IRO Journal on Sustainable Wireless Systems (ISSN: 2582-3167)
www.irojournals.com/irosws/

AI-Powered Real-Time Monitoring and Vigilance Alert System

Hariprasath R[1], Tamil M[2], Yuvaraj V[3], K. Karthick[4]

[1-3]Computer Science and Engineering, Kongunadu College of Engineering and Technology, Namakkal-Trichy Main Road, Tholurpatti Post, Thottiyam Taluk, Trichy, Tamil Nadu, India.

[4]Assistant Professor, Computer Science and Engineering, Kongunadu College of Engineering and Technology, Namakkal-Trichy Main Road, Tholurpatti Post, Thottiyam Taluk, Trichy, Tamil Nadu, India.

E-mail: [1]hariprasathrenganathan@gmail.com, [2]tamil08mctv@gmail.com, [3]yuviveera2002@gmail.com, [4]karthivel.me@gmail.com

Abstract

Real-time monitoring is increasingly complex across various fields, including security, transportation, and business operations, requiring manpower availability around the clock for monitoring and alerting. The system involves delayed alerts in the absence of manpower, potentially leading to uncertainty regarding monitoring status. The proposed system employs real-time monitoring and alerts through updates. Upon system initiation, such as with a webcam utilizing OpenCV and facial recognition libraries, algorithms analyze and process data. Detection of activity within the coverage area triggers immediate alerts to the owner through email or SMS, using the respective contact information provided during system setup. Continuous monitoring provides regular updates on the situation, aiming to enhance security and operational efficiency by promptly notifying owners of unidentified situations or human resource shortages.

Keywords: Real-time Monitoring, Alert Notification, Artificial Intelligence, Security.

1. Introduction

In the digital age, real-time monitoring has become increasingly important due to the complexity of modern life and the need to be constantly vigilant over various environments

Extended version of article received from 4th INTERNATIONAL CONFERENCE on Artificial Intelligence,5G Communication and Network Technology
DOI: https://doi.org/10.36548/jsws.2024.1.005
Received: 07.03.2024, received in revised form: 04.04.2024, accepted: 16.04.2024, published: 30-04-2024

and assets [1]. Ensuring effective monitoring has traditionally required a significant workforce, presenting challenges in terms of cost, scalability and reliability [2]. However, with advances in artificial intelligence and automation technologies, the field of monitoring and surveillance is undergoing a transformation [3]. The proposed system introduces an innovative approach to real-time monitoring and vigilance through AI-based recognition and alerting mechanisms [5]. Using sophisticated machine learning models for object detection, tracking, and facial recognition, the system is able to autonomously analyse live video streams to identify and respond to potential threats or anomalies. One of the key advantages of the system is its ability to function seamlessly in a variety of environmental conditions, including various weather conditions and lighting scenarios [6, 7]. This adaptability ensures consistent and reliable performance regardless of external factors. Additionally, the system's automated alerting mechanism enables immediate alerts to be sent to relevant stakeholders or authorities in case of detected incidents or suspicious activities. By leveraging email alerts, users can receive real-time alerts and take immediate action to address potential security breaches or security issues [8]. As a result, real-time surveillance and vigilance system powered by artificial intelligence not only improves security measures, but also offers cost-effective and efficient solutions for a wide range of sectors, including but not limited to residential security, commercial security. Public safety and infrastructure monitoring. With its potential to revolutionize traditional monitoring practices, the system is poised to become an indispensable tool for active threat monitoring and detection in the future [10].

2. Related Work

S.no	Author & Year of Publication	Journal	Title of the paper	Advantages &Limitations
1	Jianghai Lan, Xingguo Jiang, Guojun Lin, 2023 [4]	IEEE Access	Expression Recognition Based on Multi-Regional Coordinate Attention Residuals	Focus on important areas of the face while leveraging residual connections to improve feature extraction but Limited dataset scope and high computation cost

2	Muhammad Haris Kaka Khel, Kushsairy Abdul Kadir, 2023 [9]	IEEE Journal	Realtime Crowd Monitoring-Estimating Count, Speed and Direction of People Using Hybridized YOLOv4	Potentially analyze crowds in real-time, providing valuable information for crowd management and safety but the effectiveness of the system in managing extremely large crowds remains unclear.

3. Proposed Methodology

The proposed method leverages cutting-edge technologies such as facial recognition and object detection to boost the accuracy and efficiency of the monitoring process and trigger alerts for the user. It integrates artificial intelligence algorithms for real-time analysis of monitored data, enabling the identification of patterns, anomalies, and potential security threats. Additionally, it entails the development of a user-friendly interface aimed at enhancing the overall user experience, ensuring accessibility and ease of navigation for users of varying backgrounds. This system is designed to effectively operate across diverse areas, facilitating security measures and enabling prompt actions based on detected presences**.**

3.1 Initiation Module

Allows the user to start the monitoring system. Waits for user input such as email id and message to initiate the monitoring process. Initializes the webcam capture module and proceeds to monitoring and recognizing.

3.2 Real-Time Monitoring Module

The object detection and face detection enable to monitor from the live video feed from the webcam. Analyzes the video frames for any detected objects or faces. Continuously captures video frames from the webcam. Passes the frames to analysis.

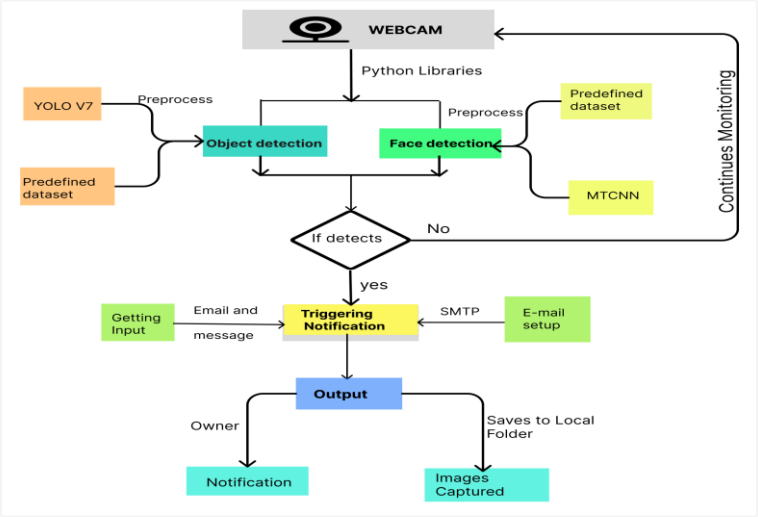

Figure 1. Proposed Method Diagram

3.3 Face Detection Module

The module receives a video frame from the real-time streaming module. The frame may undergo pre-processing steps such as resizing, normalization, or color conversion if required by the MTCNN algorithm. The MTCNN algorithm is applied to the pre-processed frame to detect faces. MTCNN performs multi-task face detection by simultaneously predicting bounding boxes and facial landmarks. MTCNN detects the presence of faces in the frame by identifying bounding boxes around them and localizes the positions of the detected faces within the frame, thus it displays the detection.

```
def face_detection(self, frame):
    rgb_frame = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
    small_frame = cv2.resize(rgb_frame, (0, 0), fx=0.5, fy=0.5)
    faces = self.detector.detect_faces(small_frame)
    faces_detected = []
    if faces:
        for result in faces:
            x, y, w, h = result['box']
            x *= 2
            y *= 2
            w *= 2
            h *= 2
            cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 255, 0), 2)
            faces_detected.append(frame[y:y+h, x:x+w])
    return frame, faces  detected
```

Figure 2. Coding for Face Detection

The code implementation initiates the way to draw the frame around the faces if detected while processing and the detected faces triggers the alert generation if detected. Let it been further in alert generation module.

3.3.1 Techniques/Algorithm

a. MTCNN

MTCNN (Multi-task Cascaded Convolutional Neural Networks) is a deep learning model used for face detection and facial landmark localization. MTCNN is responsible for detecting candidate face regions in an input image. It uses a shallow CNN to generate face bounding boxes along with a probability score indicating the presence that performs bounding box regression to accurately adjust the size and location of the candidate face bounding boxes. It uses a deeper CNN to refine the coordinates of the bounding boxes and improve the localization accuracy. MTCNN is used for face detection, specifically in the face detection method of the EmailInputWindow class. This method takes an input frame, applies MTCNN for face detection, and returns the frame with bounding boxes drawn around detected faces. The detected faces can then be further processed or analyzed based on the requirements of the application.

3.4 Object Detection Module

The module starts by receiving a video frame from the real-time streaming module, which captures the live video feed from the webcam and undergo pre-processing steps to enhance quality or reduce noise. Pre-processing can include resizing, normalization, or noise reduction techniques The pre-processed frame is passed through the YOLOv7 object detection algorithm. YOLOv7 analyzes the frame and detects various objects present within it and YOLOv7 identifies the detected objects based on predefined classes. The module displays the detection results.

3.4.1 Techniques/Algorithm

a. YOLOv7

YOLOv7 is a state-of-the-art object detection model that belongs to the YOLO (You Only Look Once) family of models. Its's an improvement over previous version like YOLOv3, offering better accuracy and efficiency. YOLOv7 is a state-of-the-art object detection model that belongs to the YOLO (You Only Look Once) family of models. YOLOv7 takes an input image or a batch of images for preprocesses.This preprocessing step usually involves resizing the images to a fixed size and normalizing pixel values. Finally, the processed predictions are often visualized on the original input images, with bounding boxes drawn around detected objects along with their class labels and confidence scores. The module receives detection results from the object detection or face detection modules. If the module identifies a specific event, it proceeds to generate an alert message. Triggers an alert when individuals are detected. Formats the alert message and sends it to the notification module for delivery.

3.5 Alert Generation and Notification Module

The Notification module receives an alert message from the Alert Generation Module indicating a specific event or condition that requires notification. It composes an email notification message based on the received alert, including relevant information such as images, messages. As the user can view it and take the action towards the event if anything is found misleading based on owner's favour.

```
msg = MIMEMultipart()
msg['From'] = sender_email
msg['To'] = recipient_email
msg['Subject'] = "Someone is there "
with open(image_path, 'rb') as img_file:
    image = MIMEImage(img_file.read())
    msg.attach(image)
with smtplib.SMTP_SSL(smtp_server, 465) as server:
    server.login(username, password)
    server.sendmail(sender_email, recipient_email, msg.as_string())
```

Figure 3. Coding for Alert Generation and Notification Module

3.5.1 Technique/Algorithm

a. SMTP Server

The SMTP (Simple Mail Transfer Protocol) server is used to send emails. The system uses the SMTP server provided by Gmail (smtp.gmail.com) with port 587.

This server requires TLS (Transport Layer Security) encryption for secure communication. The system collects this information from the framework and prepares to send an email. The system constructs an email message using the MIME Multipart class from the email.mime.multipart module. It sets the sender, recipient, subject, and body of the email using the information provided by the user. The system establishes a connection to the SMTP server using the smtplib.SMTP class. Once authenticated, the system sends the email message using the sendmail method of the SMTP server object. The email is sent to the recipient's email address specified by the user. Overall, the SMTP server facilitates the transmission of emails from the user's email to the recipient's email address the with the specified message and optional attachments. It ensures that the email is securely delivered over the internet.

4. Experimental Results

When the suggested system is put into practice, it performs real-time surveillance using Python libraries to initiate monitoring through webcam. The webcam functions are utilized for monitoring purposes. During the monitoring process, if any individuals are detected in the frame, the system processes face detection as a primary step to identify faces. Subsequently, the alert generation module is activated to send alerts to the user using the code implementation. These alerts include images of the individuals detected within the webcam's frame. Through these alerts, users can be informed about the arrival of strangers or known persons. If an unknown person is detected, the user can take appropriate actions. Thus, the practice involves automated real-time surveillance for security purposes in various locations to monitor the arrival of individuals.

4.1 Getting Input and Start Monitoring

When the prototype is activated, it will present a framework for users to input their email address, password, and message. Users can then initiate the monitoring process by

activating the start function, which utilizes the OpenCV library to access the webcam through Python code implementation. Thus, the prototype will begin functioning.

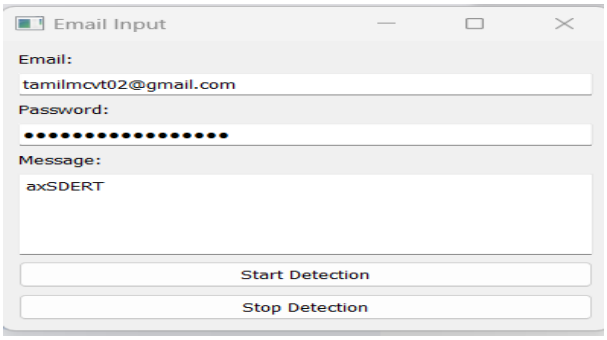

Figure 4. Getting Input and Start Monitoring

4.2 Face Detection and Objection Detection Processing

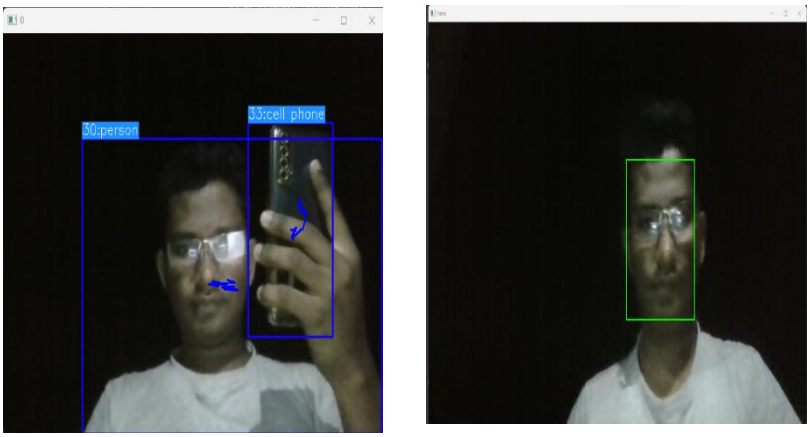

Figure 5. Face Detection and Object Detection

"When the monitoring process is initiated through the start function, the webcam engages in monitoring and employs the face detection and object detection modules to detect activity within the frame it monitors. Upon detecting an individual crossing or approaching the webcam, it triggers the alert generation to notify the user. This notification is generated by analyzing the frame data pre-processed with the help of algorithms and Python libraries.

4.3 Notification and Saving to Local Storage

Thus, the monitoring process involves both face detection and object detection, which triggers alert generation through email along with captured images of individuals detected during monitoring. The alert is initiated using the SMTP server. Not only are the images sent

via email, but they are also saved in local storage, allowing users to access them both through email and locally. In the email notification, the message entered during the framework initiation is displayed.

The monitoring continues streaming until the user initiates the stop function, which is provided in the framework where user inputs are entered. It composes an email notification message based on the received alert, including relevant information such as images and messages. Users can then view the notification and take appropriate action if any discrepancies are found, based on the owner's preferences.

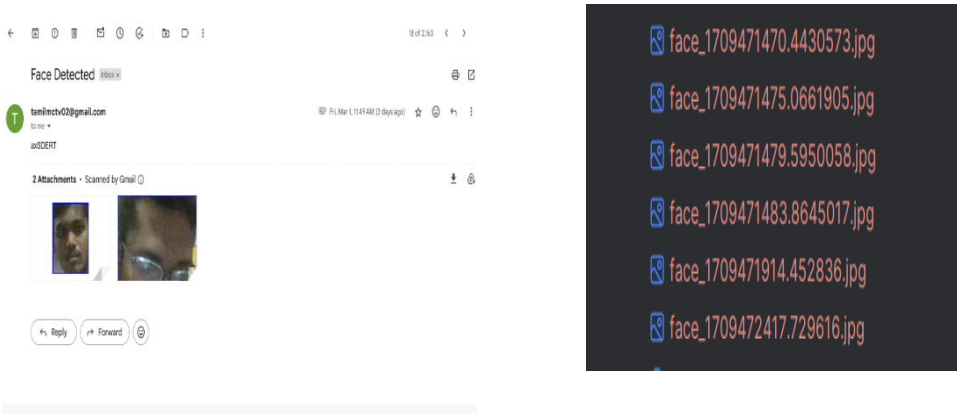

Figure 6. Notification and Saving to Local Storage

- **Operating system** : Windows 10/11
- **Tools** : PyCharm
- **Programming Language** **:** Python
- **Framework** : PyQt5

5. Conclusion

In conclusion, AI-Powered Real-time Monitoring and Vigilance Alert System has emerged as a creative and promising means of security solution, particularly in situations where manpower is limited. By offering a real-time streaming GUI, individuals can monitor webcam activity from the system's initiated location. Detection processes are activated upon initiation, enabling the system to detect individuals approaching the webcam and trigger alerts through SMTP server initiation. The system processes frames through pre-processing, capturing individuals and providing image notifications through email to users. Detection and alert

generation are automated, enhancing efficiency and responsiveness. The AI-Powered Real-time Monitoring and Vigilance Alert System represents a significant advancement in security technology, offering users a proactive and intelligent solution to protect their property, assets and personnel. With its real-time monitoring capabilities, automated detection processes and seamless alert generation, this system gives users peace of mind and confidence in their security infrastructure, making it an indispensable tool in today's fast-paced and dynamic world.

References

[1] Barathi Subramanian, Jeonghong Kim, Mohammed Maray. (2022) 'Digital Twin Model: A Real-Time Emotion Recognition System for Personalized Healthcare' in IEEE access, vol.10, pp. 81156 - 81165.

[2] Duy-Linh Nguyen, Muhamad Dwisnanto Putro. (2022) 'Facemask Wearing Alert System Based on Simple Architecture With Low-Computing Devices' in IEEE access, vol. 10, pp. 29872 - 29981.

[3] Hui Qi, Chenxu Wu1, Ying Shi, Xiaobo Qi. (2023) 'A Real-Time Face Detection Method Based on Blink Detection' in IEEE access, vol. 11, pp. 28180 - 28189.

[4] Jianghai Lan, Xingguo Jiang, Guojun Lin. (2023) 'Expression Recognition Based on Multi-Regional Coordinate Attention Residuals' in IEEE access, vol. 11, pp. 63863 - 63873.

[5] Jie Huang, Anmin Huang, And Liming Wang. (2020) 'Intelligent Video Surveillance of Tourist Attractions Based on Virtual Reality Technology' in IEEE access, vol. 8, pp.159220 - 159233.

[6] Juan Jesus Ojeda-Castelo, Maria De Las Mercedes Capobianco-Uriarte. (2022) 'A Survey on Intelligent Gesture Recognition Techniques' in IEEE access, vol.10, pp. 87135 - 87156.

[7] Jumabek Alikhanov, Hakil Kim. (2023) 'Online Action Detection in Surveillance Scenarios: A Comprehensive Review and Comparative Study of State-of-the-Art Multi-Object Tracking Methods' in IEEE access, vol. 11, pp. 68079 - 68092.

[8] Kazutoshi Akita and Norimichi Ukita. (2023) 'Context-Aware Region-Dependent Scale Proposals for Scale-Optimized Object Detection Using Super-Resolution' in IEEE access, vol. 11, pp. 122141 - 122153.

[9] Muhammad Haris Kaka Khel1, Kushsairy Abdul Kadir. (2023) ‘Realtime Crowd Monitoring—Estimating Count, Speed and Direction of People Using Hybridized YOLOv4’ in IEEE Journal, vol. 11, pp. 56368 - 56379.

[10] Natalia Kowalczyk, Milena Sobotka. (2023) ‘Mask Detection and Classification in Thermal Face Images’ in IEEE access, vol.11, pp. 43349 - 43359.

Author’s Biography

Mr.M.Tamil currently pursuing the bachelor degree in B.E - Computer Science and Engineering. He has experience in initiating the real-time projects like OpenCV processing and automations that was gained during his internship periods. He was more interested in areas of AI and ML. He was passionate on emerging trends that help to gain knowledge and make the way to practices in professional background as well looking for the opportunity.

Mr. K. Karthick Assistant Professor/ CSE, Kongunadu College of Engineering and Technology, Trichy, Tamil Nadu, India. He received his bachelor degree in B.Tech. Information Technology from Odaiyappa College of Engineering and Technology under Anna University- Trichy, M.E. in Computer Science and Engineering from Kongunadu College of Engineering and Technology under Anna University-Chennai. He has 10 years of teaching experience and pursuing Ph.D., as a part-time research scholar in Anna University, Chennai. His area of interest lies in Networking and Network Security and published 10 papers in international journals and presented 15 papers in national and international conferences in that area.