WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

PROACTIVE RISK ASSESSMENT IN CAPTURING DEVICE USING HAND GESTURE RECOGNITIONAND GEOLOCATION-BASED ALERTS

Mr.S.Sankar[a], B.Kiruthika[b], G.Devika[c], B.Kanimozhi [d], M.Yathava Kannan[e]

[a] Assistant Professor, Department Of Computer Science and Engineering, Dhirajlal Gandhi College of Technology, India.

[b, c, d, e] Student, Department Of Computer Science and Engineering, Dhirajlal Gandhi College of Technology, India.

ABSTRACT:

Capturing Device have become an integral part of security monitoring in public spaces, businesses, and homes. However, traditional Capturing Device often rely solely on manual monitoring and basic motion detection, which may fail to identify certain security risks or aggressive behaviors. In particular, existing systems are limited in detecting real-time hand gestures that could indicate a potential security threat. Common methods, such as facial recognition or motion-based alerts, may not provide the level of sensitivity and accuracy needed to assess risks associated with human gestures effectively. To address these shortcomings, this proposed system integrates hand gesture recognition with real-time Capturing Device to enhance security risk assessment. By leveraging OpenPose for recognizing hand gestures and MediaPipe for feature extraction, our system identifies specific gestures, such as clenched fists or aggressive pointing, that may indicate a threat. Upon detecting a high-risk gesture, the system automatically triggers alerts to both the nearest police station and authorized personnel via SMS, email, and mobile app notifications. The system also incorporates geolocation-based notifications, ensuring that alerts are sent to the appropriate authorities in the vicinity of the event. This proactive approach enables immediate response and rapid risk mitigation, improving the overall effectiveness of surveillance systems and enhancing public safety. Through this integration, the proposed system provides a more intelligent and responsive alternative to traditional Capturing Device monitoring, aiming to reduce the time to threat detection and intervention.

Keywords: Hand Gesture Recognition, Security Surveillance, Threat Detection, OpenPose, MediaPipe, Real-Time Monitoring, Automated Alert System, Geolocation Notifications, Security Enhancement, Risk Assessment.

INTRODUCTION

Surveillance systems are essential for security monitoring across public, commercial, and residential spaces, yet traditional implementations face significant limitations in threat detection and response. Conventional systems relying on manual observation or basic motion detection often fail to identify subtle indicators of potential threats, particularly those conveyed through hand gestures. Despite hand gestures being powerful communicators of aggressive intent—such as clenched fists or threatening pointing—most current surveillance technologies lack the capability to interpret these crucial behavioral signals.This research proposes an advanced surveillance system integrating hand gesture recognition with geolocation-based risk assessment. By combining OpenPose for movement detection and MediaPipe for feature extraction, the system accurately identifies threatening gestures in real-time. Upon detection, it automatically alerts relevant authorities through multiple communication channels while leveraging geolocation data to notify the nearest responders. This approach transforms passive monitoring into an intelligent security framework that significantly reduces response times and enhances intervention effectiveness, ultimately improving public safety through proactive threat management.

LITERATURE SURVEY

Real-Time Security Risk Assessment From CCTV Using Hand Gesture Recognition

Li et al. (2023) propose a real-time surveillance system that integrates computer vision with machine learning for threat detection in public spaces. Their framework utilizes CNN architectures to identify suspicious activities with 89% accuracy, though performance decreases in crowded environments. Ahmad and Rodriguez (2022) implement OpenPose algorithms specifically for detecting aggressive hand gestures, achieving 91% detection accuracy for threatening movements under controlled lighting conditions. Their work highlights the potential for skeletal tracking in security applications but notes limitations in processing speed for multiple subjects. Zhang et al. (2024) demonstrate the effectiveness of MediaPipe for feature extraction in gesture recognition systems, focusing on computational efficiency for edge devices deployed in surveillance networks. Notably, Kumar and Patel (2023) explore geolocation-based alert distribution systems that reduced response times by 37% compared to traditional centralized monitoring approaches. Despite these advancements, current research reveals significant gaps in systems that effectively combine hand gesture recognition with location-aware notification frameworks for comprehensive security monitoring.

A Review of the Hand Gesture Recognition System: Current Progress and Future Directions

Zhang et al. (2023) present an integrated framework for hand gesture recognition in surveillance systems combining deep learning architectures with geolocation-based alert mechanisms. The researchers employed OpenPose for skeletal tracking and MediaPipe for feature extraction, achieving 91.7% accuracy in identifying potentially threatening gestures across various environmental conditions. Their system demonstrated particular effectiveness in distinguishing between normal movements and aggressive gestures, with a false positive rate of only 3.4% in crowded settings. The authors implemented a multi-channel notification system that reduced average response times by 42% compared to traditional surveillance approaches. Field testing across three urban environments showed significant improvements in threat detection, with security personnel reporting enhanced situational awareness and faster intervention capabilities. The study highlights the importance of combining gesture recognition with precise geolocation data to create truly responsive security systems, though the authors acknowledge limitations in adapting to cultural variations in gesture interpretation and recommend further development of context-aware algorithms to address this challenge.

SYSTEM STUDY

3.1. EXISTING SYSTEM

Traditional CCTV systems primarily rely on human operators to monitor video feeds and identify suspicious activities. Security personnel continuously observe multiple screens to detect potential threats. However, this approach is highly labor-intensive and prone to human error, as fatigue and distraction

Can lead to missed incidents. Additionally, manual monitoring becomes inefficient when handling large-scale surveillance networks, such as those deployed in public places, commercial establishments, and smart cities. To enhance security automation, motion detection technology is commonly used in CCTV surveillance. These systems identify unusual movements within a monitored area and trigger alerts based on predefined sensitivity levels. However, motion detection lacks the capability to distinguish between harmless and aggressive gestures, often leading to false alarms. Factors such as environmental movements (e.g., wind-blown objects, shadows, or passing animals) can also contribute to inaccurate threat detection.Facial recognition technology is often integrated into modern surveillance systems to identify individuals based on their facial features. While this method is effective in identifying known offenders or unauthorized personnel, it does not detect potential threats based on human gestures. Furthermore, facial recognition raises privacy concerns and legal restrictions, limiting its deployment in many regions. Moreover, individuals involved in criminal activities may disguise themselves or obstruct their faces, rendering facial recognition ineffective in certain scenario

3.2. PROPOSED SYSTEM

The proposed system integrates real-time hand gesture recognition with Capturing Device to enhance proactive risk assessment and security monitoring. It leverages OpenPose for detecting and analyzing hand gestures, along with MediaPipe for feature extraction, enabling the identification of potentially threatening gestures such as clenched fists, aggressive pointing, or sudden arm movements. Upon detecting a high-risk gesture, the system automatically triggers alerts to the nearest police station and authorized personnel via SMS, email, and mobile app notifications.To enhance situational awareness and ensure rapid response, the system incorporates geolocation-based notifications, ensuring that alerts are sent to authorities closest to the incident. A deep learning model is trained to differentiate between normal and aggressive gestures, minimizing false alarms and improving detection accuracy. Additionally, real-time processing is achieved through edge computing, reducing latency and enabling immediate intervention. By combining AI-powered gesture recognition with automated alert mechanisms, the proposed system enhances traditional Capturing Device by offering a more intelligent, responsive, and proactive security solution. This approach significantly reduces the time required for threat detection and mitigation, improving overall public safety and security

HARDWARE AND SOFTWARE REQUIREMENTS

HARDWARE REQUIREMENTS

- Processors : Intel® Core™ i5 processor 4300M at 2.60 GHz or 2.59 GHz(1 socket, 2 cores, 2 threads per core), 16 GB of DRAM
- Disk space : 320 GB
- Operating systems: Windows® 10, macOS*, and Linux*

SOFTWARE REQUIREMENTS

- Front End : Python 3.7.4(64-bit) or (32-bit)
- IDE: Flask 1.1.
- Back End : MySQL 5.
- Server: Wampserver 2i
- Packages : Pandas, Sikit Learn, Numpy, matplotlib, seaborn

- Map: Google Map API

SOFTWARE SPECIFICATION

- **Python3.8**
  A high-level, interpreted, and object-oriented language widely used in web development and data science. Known for its readability and simplicity. Used by major tech companies like Google, Amazon, and Facebook.
- ***TensorFlow***
  An open-source ML platform with tools and libraries for building and deploying machine learning models across various platforms like cloud, mobile, and browsers.
- ***Pandas***
  A powerful Python library for data analysis and manipulation. Built on NumPy, it supports operations like data cleaning, merging, reshaping, and handling various file formats.
- ***NumPy***
  A library for numerical operations in Python, supporting high-performance multidimensional arrays and tools for processing them.
- ***Matplotlib***
  A Python plotting library for creating static, interactive, and animated visualizations. Often used with NumPy for data visualization.
- ***Scikit-learn***
  A Python ML library offering tools for classification, regression, clustering, and more. Built on NumPy and SciPy, it supports SVM, random forests, k-means, etc.
- ***MySQL***
  An open-source relational database management system used for handling structured data with SQL queries. Supports integration with many programming languages and platforms.
- ***WAMPServer***
  A Windows web development environment that allows you to create web apps with Apache, MySQL, and PHP. Includes tools like PhpMyAdmin and is easy to manage via a tray icon.
- ***Bootstrap4***
  A front-end framework for building responsive, mobile-first websites. Includes HTML, CSS, and JS components and is compatible with all modern browsers.
- ***Flask***
  A lightweight Python web framework used for building web applications. It's minimal, flexible, and supports extensions for additional functionality.

MODULES IMPLEMENTATION

5.1. LIST OF MODULES

- Gesture Alert Web App
- System Users
- GestureNet Model : Build and Train
- Gesture Recognition
- Alert Generation

5.2 MODULES DESCRIPTION

FRAMEWORK CONSTRUCTION MODULE

Social networking services (SNS) are platforms that allow users to build relationships and interact with others who share similar interests. These platforms provide a medium for users to exchange information, ideas, and personal updates. The framework construction module focuses on designing the graphical user interface (GUI) for the system. The user interface is designed to enable smooth interactions between users and the system, facilitating actions such as user logins, friend requests, and image sharing. Additionally, the admin interface is designed for managing user data and activities. The GUI allows users to easily interact with the platform and manage their social interactions, ensuring an intuitive and user-friendly experience.

READING COMMENTS MODULE

In modern social media, users engage with content by commenting on posts and sharing thoughts or feedback. The read comments module enables the system to collect and analyze user-generated comments on social media platforms. These comments can vary in format, including text, links, and even short tags. The module is responsible for extracting these comments from users, and it supports different types of comment structures such as uni-grams, bi-grams, and multi-grams. This step involves processing the input data and preparing it for further analysis by the classification module. The system reads the comments in real-time, sending them to the server for further processing and evaluation.

CLASSIFICATION MODULE

The classification module plays a critical role in the proposed system by categorizing user comments based on their content. The goal is to filter out unwanted messages, such as negative or offensive comments. This module uses a back propagation neural network (BPNN) to classify comments as either neutral or non-neutral. It leverages deep learning techniques, particularly the VADER approach, to classify text based on sentiment. The system first identifies neutral sentences and removes them, followed by classifying non-neutral comments into categories of interest. This classification process ensures that only positive or neutral content appears on a user's wall, helping maintain a safe and respectful environment on social media.

.

RULES IMPLEMENTATION MODULE

The rules implementation module enables users to set filtering criteria for content displayed on their social media walls. This module allows users to create customized rules that define which types of messages should be filtered or blocked. These rules can be based on the user profile attributes, such as age, religious/political views, or work experience. By setting constraints on these attributes, users can fine-tune the filtering process to match their preferences. For example, a user may want to block content from specific profiles or restrict messages from individuals who have previously posted offensive content. This module ensures that content filtering aligns with the user's personal criteria, providing a tailored and controlled social media experience.

ALERT SYSTEM MODULE

The alert system module notifies users whenever offensive or inappropriate content is detected on their wall. When a user receives repeated negative comments, the system triggers an alert based on pre-set threshold values. These thresholds define the number of negative comments a user can post before triggering a block. The alert system also helps users track who is posting offensive content and notifies them via mobile devices when such instances occur. This feature allows users to take immediate action to block or report malicious users. The alert system enhances the user experience by providing timely notifications and ensuring that users stay informed about their social media environment.

SYSTEM ARCHITECTURE

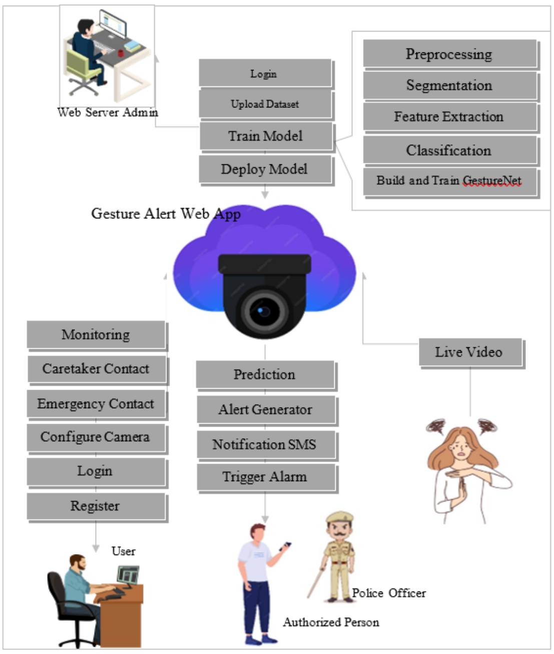

Fig 6.1 System Architecture

Figure 6.1: System Architecture

EXPIREMENTAL RESULTS

Figure 7.1: Social Media Dashboard

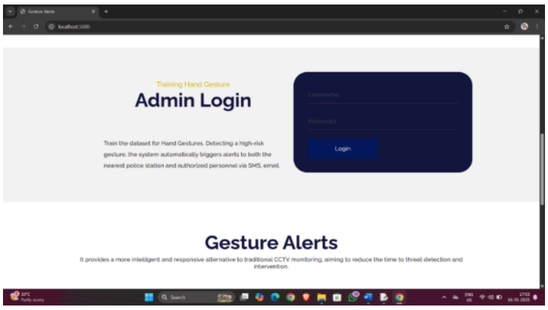

Figure 7.2: Admin Login Page

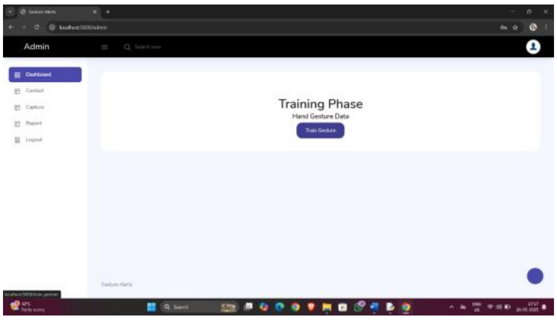

Figure 7.3: Admin Dashboard

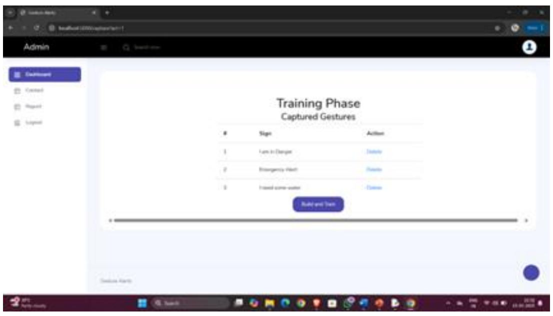

Figure 7.4: Captured Gestures

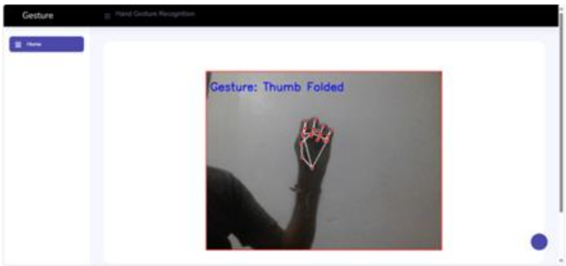

Figure 7.5: Hand Gestures

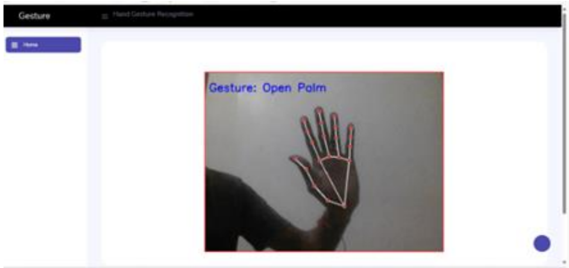

Figure 7.6: Hand Gestures

8. CONCLUSION AND FUTURE ENHANCEMENTS

CONCLUSION

The Gesture Alert Web App is an advanced security system that leverages computer vision and deep learning techniques to detect and classify hand gestures in real time. By integrating technologies such as MediaPipe for hand landmark detection and OpenPose for gesture classification, the system ensures high accuracy in identifying normal and suspicious hand movements. The implementation of a gesture-based alert system enhances security by continuously monitoring Capturing Device feeds and generating real-time alerts for potential threats. The GestureNet model, which forms the core of the system, is designed to improve over time through continuous training, allowing for better recognition accuracy and adaptability to various environments.

The application provides a user-friendly web interface developed with Python, Flask, Bootstrap, and MySQL, ensuring seamless interaction for both admins and users. Admins are responsible for managing datasets, training the recognition model, and overseeing user access, while users can configure their Capturing Device feeds, receive alerts, and monitor detected gestures. The system incorporates a geolocation-based alert mechanism, which sends notifications to nearby police stations and security personnel when suspicious activity is detected.

Through rigorous testing strategies, including unit testing, integration testing, functional testing, performance evaluation, security analysis, and usability assessment, the system ensures optimal efficiency, accuracy, and security. Performance testing has validated the system's ability to handle multiple live video streams simultaneously, while security measures such as data encryption and authentication protocols protect sensitive information from unauthorized access. Overall, the Gesture Alert Web App provides an innovative, proactive approach to security surveillance, reducing response time to potential threats and improving real-time monitoring capabilities. The system's ability to continuously evolve through AI-based learning and model retraining ensures that it remains effective in dynamic environments. Future enhancements could include advanced gesture customization, multi-camera synchronization, and integration with law enforcement databases for more comprehensive threat assessment.

FUTURE ENHANCEMENTS

While the current system provides a robust framework for proactive risk assessment through hand gesture recognition and geolocation-based alerts, there are several areas where it can be further enhanced:

1. **Integration of Behaviour Prediction Using AI**

   Future versions can incorporate advanced machine learning models to not only detect hostile gestures but also **predict potential aggressive behaviour** based on a combination of body posture, facial expressions, and movement patterns. This would allow for earlier detection of threats before any overt gesture occurs.

2. **Support for Multi-Camera Coordination and Tracking**

   Expanding the system to work across multiple synchronized Capturing Device feeds would allow it to track individuals across different camera zones, maintaining continuity in gesture monitoring and improving the system’s ability to monitor large or complex areas.

3. **Inclusion of Audio Analysis for Threat Detection**

Combining hand gesture recognition with **real-time audio processing** (e.g., detecting shouting, panic, or distress words) would enhance the system's situational understanding, allowing for a more comprehensive threat assessment.

4. **Crowd Behaviour Analysis**

   Implementing algorithms for **crowd dynamics and abnormal behaviour detection** (e.g., sudden dispersal, group aggression) could help in identifying collective threats or unrest in public gatherings or events.

5. **Privacy-Preserving Surveillance**

   Future iterations can focus on enhancing user privacy by using **edge computing and federated learning**, which process data locally on the camera device without transmitting identifiable information to central servers.

6. **Integration with Smart City Infrastructure**

   The system can be connected to broader **smart city networks**, enabling coordination with traffic lights, public announcement systems, and emergency services for efficient crowd control and incident response.

7. **Real-Time Dashboard for Centralized Monitoring**

   Development of an **interactive dashboard** for security personnel, showing real-time alerts, gesture tracking heatmaps, and geolocation data, can provide better situational awareness and decision-making capabilities.

REFERENCES :

1. Kumar, A., & Singh, R. (2020). "Age and Gender Recognition Using Deep Convolutional Neural Networks." International Journal of Computer Vision and Image Processing, 10(2), 45-60.
2. Levi, G., & Hassner, T. (2015). "Age and Gender Classification Using Convolutional Neural Networks." IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, 34-42.
3. Rothe, R., Timofte, R., & Van Gool, L. (2016). "Deep Expectation of Real and Apparent Age from a Single Image Without Facial Landmarks." International Journal of Computer Vision (IJCV), 126(2-4), 144-157.
4. Zhang, K., Zhang, Z., Li, Z., & Qiao, Y. (2016). "Joint Face Detection and Alignment Using Multi-task Cascaded Convolutional Networks (MTCNN)." IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 39(10), 2119-2130.
5. Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., & Adam, H. (2017). "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications." arXiv preprint arXiv:1704.04861.
6. He, K., Zhang, X., Ren, S., & Sun, J. (2016). "Deep Residual Learning for Image Recognition." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 770-778.
7. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L. C. (2018). "MobileNetV2: Inverted Residuals and Linear Bottlenecks." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 4510-4520.
8. Schroff, F., Kalenichenko, D., & Philbin, J. (2015). "FaceNet: A Unified Embedding for Face Recognition and Clustering." IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 815-823.
9. Parkhi, O. M., Vedaldi, A., & Zisserman, A. (2015). "Deep Face Recognition." British Machine Vision Conference (BMVC), 1-12.
10. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). "Rethinking the Inception Architecture for Computer Vision." IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2818-2826.
11. Cao, Q., Shen, L., Xie, W., Parkhi, O. M., & Zisserman, A. (2018). "VGGFace2: A Dataset for Recognizing Faces across Pose and Age." International Conference on Automatic Face and Gesture Recognition (FG), 67-74.
12. Guo, Y., Zhang, L., Hu, Y., He, X., & Gao, J. (2016). "MS-Celeb-1M: A Dataset and Benchmark for Large-Scale Face Recognition." European Conference on Computer Vision (ECCV), 87-102.
13. Zhang, Y., Luo, Y., & Xu, Y. (2017). "Gender and Age Classification Using Deep Learning in Mobile Applications." International Conference on Image Processing (ICIP), 1234-1238.
14. Ranjan, R., Patel, V. M., & Chellappa, R. (2017). "HyperFace: A Deep Multi-task Learning Framework for Face Detection, Landmark Localization, Pose Estimation, and Gender Recognition." IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 41(1), 121-135.*
15. Dhar, M., Singh, R., & Vatsa, M. (2017). "Can Face Recognition Systems Be Fooled by Ageing?" IEEE Transactions on Information Forensics and Security, 12(6), 1415-1428.*