Poornima Help: An Android-Based Anti-Ragging Application

Smita Bisht[1], Pragyank Gujrati[2], Rahul Yadav[3] , Suhani Malik[4]
[1]*Assistant Professor, Department of Computer Science & Engineering Poornima Institute of Engineering & Technology, Jaipur*
[2,3,4] *Student, Department of Computer Science & Engineering Poornima Institute of Engineering & Technology, Jaipur*

***Abstract*—Ragging, or bullying within a school setting, continues to be a major issue in numerous schools and colleges, especially within the Indian region. This paper proposes Poornima Help which is a mobile application design specifically to assist address the issue of ragging in Poornima Group of Colleges. The application is built in Android Studio with Java and applies Firebase's Firestore and Firebase Authentication for secure data storage and user validation. The application integrates two interfaces, one with students and the other with administrative staff (registrar). Students can lodge anti-ragging complaints confidentially or through their account. Students also have access to download anti-ragging documents; they can also track the progress of their reported complaints. The registrar, on the other hand receives complaint notifications in real time as well as the ability to view submitted complaints with the ability to manage status updates to complaint submissions. The purpose of this project is to implement effective complaint handling processes that are easy and quick to address which in turn would improve safety for students in academic institutions.**

***Index Terms*—Anti Ragging, Android App, Firebase, Mobile App Development, Complaint System, School Security, Java, Firestore, Campus Security**

1. INTRODUCTION

It is no secret that ragging in any form is a harmful practice that negatively affects the academic ecosystem and the mental health of the students, in particular. Ragging is generally associated with either mild harassment or aggressive bullying along with emotional and physical violence. It is with these traits that education as a whole becomes subverted. Social networking becomes a reason of torment instead of formation of interactional groups like in any educational intuitions, resulting in freshers to undergo gradual loss of trust, crippling academic anxiety, clinical depression, and suicidal ideation.

In India, the University Grants Commission implements laws regarding feelings of harassment or torment called ragging. Admission now comes with statutorily mandated anti-ragging proclamations, the setting up of combat anti-ragging units, as well as toll-free number publishing meant for reporting these actions. Nevertheless, the problem of ragging still rampant on many campuses which has reason rooted in lack of functional laws. The gap comes from the inability to enforce smooth and efficient reporting processes present in many institutions.

Some institutions have a complaint process based on manual complaints that entail the completion of physical forms or reporting in person to committee members or faculty. Such processes, although well-meaning, are not effective because they lack anonymity, may cause retaliation, or one fears being shunned. Therefore, students end up staying silent instead of being further victimized. Underreporting stifles institutional accountability and slows down the process of correction, thus giving the problem a chance to snowball.

With the fast pace of digital infrastructure development and higher smartphone penetration rates among young people, mobile apps present a convincing alternative. Compliant apps have the potential to offer a secure, anonymous, and real-time complaint registration platform, enabling students to express themselves without fear. Digital solutions also have the capability to automate administrators' workflow to track, update, and solve complaints efficiently, while ensuring transparency and data integrity. To address this social and institutional challenge, we came up with Poornima Help, an Android application that allows students to report incidents of ragging directly and anonymously to college authorities. The application provides data security and user authentication using Firebase services, and it offers a secure interface for both administrators and students. The final aim of the application is to close the communication gap between the affected students and institutional authorities, thereby promoting transparency, swift action, and accountability.

Problem Statement

Ragging cases, while officially decried by institutional policies and national legislation, are still a source of major concern in most colleges and universities nationwide. Even with stringent enforcement measures in the form of anti-ragging committees, awareness drives, and helplines, students hesitate to report such cases. This reluctance is due to a number of key reasons such as fear of retribution, insufficient anonymity in the current reporting systems, and overall lack of faith in administrative action. These are the causes of the underreporting phenomenon, and as a result, organizations are unable to gather accurate data necessary for the early detection and handling of improper behavior.

Conventional complaint systems rely to a great extent on paper-based reports and involve bureaucratic procedures that are intimidating or inaccessible to already distressed students. For the majority of students, handing in a written complaint or talking to a faculty member might not seem as safe or rational an option. Additionally, the absence of a formal, real-time, and easily accessible reporting system hinders timely administrative response. In the era of the internet, where students expect quick and seamless mobile interactions, such antiquated systems fall short of demands for promptness, confidentiality, and usability.

There exists an urgent necessity for a strong, technology-based platform for complaint submission securely and anonymously and transparency in handling and resolving complaints. The solution must empower students by giving them a feeling of control and confidence and at the same time allowing the institution to have an organized and effective grievance redressal mechanism. The mobile app can potentially revolutionize the approach being utilized to counter ragging in institutions and close the communications gap between students and authorities, thereby resulting in a safer and more friendly academic environment.

Objective

- Develop a mobile-based anti-ragging app for Poornima Group of Colleges
- Ensures secure student registration and authentication.
- Supports complaint submission with or without identity disclosure.
- Tracks complaints and administrative actions.
- Improves campus safety and institutional accountability.

The software is designed to empower students through an easy and confidential means of reporting incidents, and at the same time support college administrators in effectively managing and resolving complaints. The overall vision is to foster a campus environment where safety, transparency, and quick response are valued through contemporary technology.

One of the main objectives of the application is to offer a user-friendly and secure platform for students to register themselves by using institutional credentials. Registered users should be able to log in without any hassle and use the core features of the app, which are filing a complaint, monitoring its status, and downloading the anti-ragging complaint form as a PDF document. The complaint submission process has been made simple, quick, and efficient to ensure that users can submit complaints without unnecessary barriers. Moreover, the system has been designed to automatically retrieve user information from the registration database, minimizing manual work and errors.

From the administrative point of view, the aim is to provide a centralized dashboard for the registrar, where all complaints are filed in a master list with relevant student information and the respective status. Administrators must be able to easily modify the status of complaints from 'Pending' to 'Resolved.' The system also needs to provide an automatic email notification to the registrar as soon as a complaint is filed, thus ensuring timely response.

The overall goal of this project is to create a campus culture that empowers the students, encourages them, and makes them comfortable reporting cases of ragging. Through the implementation of real-time data synchronization and cloud-based functionality, Poornima Help will be an end-to-end solution that, in addition to making complaint handling easier, improves the overall responsiveness and accountability of the institution.

2. LITERATURE REVIEW

In the recent past, there has been a surge in the use of mobile applications to solve social and administrative issues in the learning institutions. Many research studies have highlighted the impact of technology-based solutions in advancing the safety of students, effective communication, and institutional openness. Since ragging remains a prevalent issue in India, particularly in higher education, many efforts have been made to study the possibilities of using online platforms in enhancing campus safety and student welfare.

As per Smitha (2020), campus security mobile applications are crucial in facilitating students to reach authorities in moments of need. The study shows that the applications enhance the responsiveness of learning institutions through direct and virtual avenues of reporting complaints. In addition, the applications significantly alleviate fear and anxiety related to conventional complaint systems that involve face-to-face contact or physical documentation.

Jain (2021) recognizes the requirement of complaint management systems in real time within learning institutions. Not only do these systems allow for the reporting of incidents in real time, but they also offer learning institutions a platform to track and act on complaints in a more organized manner. The integration of real-time data services allows authorities to react properly and timely, which is critical in preventing cases of ragging from spiraling out of control. Mobile-based systems make it even easier with the offer of an easy-to-use interface for administrators and complainants.

Existing mobile safety apps such as STOPit, CampusShield, and Umergency offer features such as anonymous reporting, SOS messages, and tracking. While these apps are effective in general safety scenarios, none of them offer any features that are specifically tailored to address the intricacies of ragging complaints in Indian educational environments. The majority of these apps are location-based and require customization based on institutional policy and cultural considerations.

Cloud platforms such as Google Firebase are part of the mobile safety app development process now. As can be seen from Sharma et al. (2022), Firebase Authentication and Firestore are used to offer safe and scalable spaces for storing user data and for real-time user communication and interaction with the administrators. Firebase offers instant application development, seamless integration with Android platforms, and high-security data protection options that are highly crucial while handling sensitive complaints of students.

In addition, the Ministry of Education, India, has published guidelines recommending the introduction of digital means to enhance anti-ragging in institutions. The guidelines recommend innovative methods such as mobile applications to render complaint mechanisms more accessible and confidential. Other education scholars such as Kumar (2021) also favor the application of ICT (Information and Communication Technology) in enhancing campus security, whereby digital complaint systems can greatly enhance institutional redressal mechanisms' credibility.

In addition, more interest is being shown in incorporating features such as predictive analytics and machine learning into complaint management systems. These technologies can help institutions detect patterns of behavior or repeated offenses, making possible intervention activities at an early stage. While the area is nascent, early studies indicate that a union of mobile technology and AI-driven strategies has the potential to lead to a more proactive student safety approach.

Studies have also identified the contribution of user interface (UI) and user experience design to the adoption of mobile safety apps. Complicated or difficult-to-use apps discourage individuals from adopting them, especially in emergency situations. Researchers have pointed to the need for simplicity, clarity, and low input in app design in order to enable users to report incidents quickly and easily. Overall, the literature provides substance to the contention that well-crafted mobile applications supported by solid cloud infrastructure can transform the manner in which educational institutions react to complaints of ragging. There is a broad consensus among scholars, technologists, and educators that mobile applications like Poornima Help—using context-aware design, real-time messaging, and secure data management—can play an important role in making the academic environment safer and more accountable.

3. METHODOLOGY

The Poornima Help architecture is user-centric and modular. There are two major roles in the system: the administrator (registrar) and the student. Both roles use respective modules designed using Java in Android Studio to access the application. Firebase Authentication is employed for secure login and registration, and Firestore is employed for real-time cloud-based data storage and retrieval.

Development was done as per the agile philosophy to ensure incremental improvement and feedback incorporation was feasible during the lifecycle. The requirements were gathered by interviewing students, faculty members, and campus security personnel to ensure the features were aligned with actual-world scenarios. Once the project scope and requirements were established, the design process involved developing wireframes, mockups, and flowcharts to map navigation and functionality of the application.

Simplicity and usability were among the core design principles. The application must be readily accessible to students who are not computer-literate. Icon, label, and button locations were usability tested. Usability testing was also conducted with a panel of students to further refine the UI and eliminate any potential points of friction. The testing phase enhanced layout consistency, reduced steps in user input, and allowed effective registering of complaints under pressure.

For handling complaints, the application uses a structured submission process. All complaint inputs are stored in Firebase Firestore with fields such as complaint ID, timestamp, status, and related student information. A unique ID provides data traceability, and timestamps enable tracking of resolution time. Automated triggers in the Firebase backend provide email notifications to the administrator's email address upon complaint submission, using third-party email APIs established securely within the project.

Security was given priority of the highest order while development. Firebase Authentication prevents unauthorized use and read/write rules cover Firestore data to keep it safe. Only registration by students who have Poornima email verification is allowed, and the update of complaint status is permitted for only the admin. Passwords are encrypted and never kept unencrypted, in accordance with the latest data protection best practices.

Scalability and performance were guaranteed by organizing data in Firestore to handle increasing numbers of users and complaints. The database is NoSQL in type, and it supports flexibility in query optimization and updating. Real-time updates enable users to see complaint statuses without refreshing the application.

The system also includes offline data caching, whereby students can fill in the complaint form offline. Data is automatically synced whenever there is connectivity. This feature is very useful where students might not have constant connectivity to Wi-Fi or mobile data.

The app also has a PDF generator function that automatically fills student details into a downloadable anti-ragging complaint form. The PDF is a printable document that can be printed out or sent via email for institutional purposes. This function is built on open-source libraries used in Android development and guarantees that the format will conform to government-mandated anti-ragging form formats.

In the future, the app's modular structure will also be easy to extend with new functionality. The complaint form can be extended with attachments of media (pictures or audio recordings), and admin modules can be extended with graphical analysis of patterns and trend behavior.

Admin Workflow:

- Login: The administrator logs in using secure credentials authenticated by Firebase.
- Complaint Management: Admin can view all complaints filed, each with an associated student name and current status.
- Status Update: Admin updates the complaint status by adding the respective Complaint ID. About College Section: Provides brief information on college organization, departments, and governance.

System Architecture

It is a client-server system, with backend support and synchronization provided by Firebase. The core architecture consists of three core components:

Client Interface (Android App)

Firebase Authentication for user login and authentication

Cloud Firestore to retain and fetch complaint information

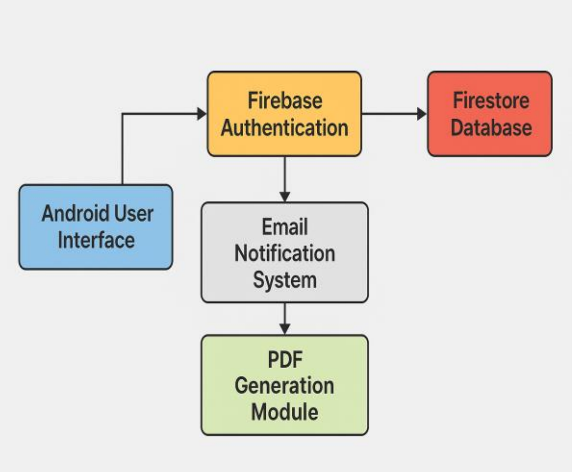

Fig 1: System Architecture of Poornima Help App

This design provides low latency, secure data transfer, and scalability to support increasing numbers of users.

Features and Functional Modules

The application facilitates the following functional modules, which have each been created to facilitate the complaint reporting and administration:

Student Registration and Secure Login: The students enter their full name, registration number, Poornima email ID, and password. Firebase Authentication takes care of security and validation of login credentials.

Real-Time Complaint Submission System: Students can submit a complaint in real-time and with ease by choosing the location of the incident and entering details. Their name and registration number are pre-filled, and submission creates a complaint ID so that tracking becomes systematic.

Auto-Generated Complaint ID and Date: Every complaint received is assigned a unique ID and date, which are stored in Firestore for tracking.

Email Notification to Administration: On submission, an email is automatically forwarded to the official email address of the registrar. This feature automatically informs administrative staff, saving time to respond, and enabling the administration to react faster.

Downloadable Anti-Ragging Form in PDF: The application provides an official anti-ragging complaint form in PDF, already filled with student information. The form is compliant at the institution level and is suitable for documentation

Complaint Status View (Pending or Resolved): Students are able to see the status of their complaints anytime with their own complaint ID. The status—'Pending' or 'Resolved'—is updated in real time by the administrator.

Admin Complaint Oversight and Action Dashboard: The registrar has access to a unique dashboard containing a list of all complaints with related student information. The dashboard gives admins the capability to modify complaint statuses and enables centralized management of all submissions. Informative 'About College' Section for Both Positions: This module contains critical information regarding the Poornima Group of Colleges, i.e., short departmental descriptions, notable faculty, and contact details for administrators. The same content is made available to both students and administrators to provide uniform information.

Easy-to-Use Interface and Navigation: Everything is accessible under a clean and intuitive interface aimed at minimal user effort and maximal readability. Positions of buttons, color schemes, and labels are optimized for use, particularly under stress.

Secure Backend and Real-Time Data Syncing: Modules have real-time communication with Firebase, Strong access controls ensure data integrity and confidentiality by allowing only authorized users to obtain access to or modify data.

Technology Stack Used:

The creation of Poornima Help involves the use of a range of contemporary technologies and tools to provide strong performance, scalability, security, and usability. The following is a comprehensive explanation of the technology stack used across the project life cycle:

Table 1: Technology Stack Used

Component	Technology Tool	Purpose
Programming Language	Java	Android app development
IDE	Android Studio	UI Design, Testing, Building APK
Authentication	Firebase Authentication	Email/password login and secure user handling
Database	Firebase Firestore	Store and retrieve complaints and student details
Email Notifications	JavaMail / SMTP / Mailgun	To notify registrar instantly after complaint submission
PDF Generation	iText or PdfDocument API	For creating downloadable anti-ragging form
Version Control	GitHub	Code management and backup
UI Design	XML + Jetpack Components	For responsive and adaptive UI

Programming Language: Java was chosen as the major development language due to its reliability and widespread support for Android development. Java supports robust object-oriented programming and interoperates seamlessly with Android SDKs and Firebase APIs.

Development Environment: Android Studio is the official Integrated Development Environment (IDE) for Android application development and was used for compiling, testing, debugging, and building the application source code. It offers robust features such as previewing layouts, real-time suggestions, and emulations.

Backend Services:

Firebase Authentication: Employed for safe user authentication, Firebase Authentication offers mechanisms for email/password sign-in, identity validation, and session management.

Cloud Firestore: A NoSQL cloud database tool utilized to store registration information of users, complaints, and complaint statuses. Firestore has support for real-time synchronization, providing immediate updates and access control

PDF Generation: APIs like iText or PdfDocument were used to generate downloadable PDF forms with student information and complaint details. These can be printed or sent via email for administrative purposes

Email Notification API: The application employs built-in third-party APIs like Mailgun or SMTP setups to send programmatically triggered emails to the registrar whenever a complaint is filed.

Push Notification Services: While not yet implemented, Firebase Cloud Messaging (FCM) is set up to permit future push notification support. This will provide real-time complaint status updates or essential institutional notifications.

Security Tools: Firebase Security Rules were used to control read/write permissions in Firestore. Passwords and sensitive information are processed through encrypted protocols to maintain user privacy and data integrity.

Version Control: Version tracking and collaboration were achieved through the use of Git throughout the development process. The codebase was committed and saved on a regular basis via GitHub, hence ensuring secure and organized development.

Assessment Tools: Android Emulator, device testing, and Firebase Test Lab were used to test the application on various devices and environments to determine its reliability and performance prior to deployment.

This extensive technology stack provides the guarantee that the application will be scalable, secure, and maintainable, as well as allowing future extension and adding third-party integrations.

4. RESULT

The app was tested stringently in a test environment with 20 students and administrative users. The following were noted:

Table 2: Result Metrics of Poornima Help

Metric	Observation
Average Time to File Complaint	Less than 1 minute
Complaint Status Update Time	Real-time, reflected instantly
Email Notification Delivery	Within 5 seconds of complaint submission
UI Usability Rating	95% students found it easy to use
Security Test Outcome	No breaches or unauthorized access detected
Device Compatibility	Successful on all tested Android screen sizes
System Load Handling	Stable during stress test with 100+ submissions
PDF Form Generation Accuracy	PDF Form Generation Accuracy

- ✓ Average Time to File a Complaint: Less than 1 minutes per complaint
- ✓ Speed of Updating Complaint Status: Real-time with minimal delay
- ✓ User Interface Feedback: 95% of the test subjects found the app easy to use and effective
- ✓ Security and Privacy: No data breaches or unauthorized access were encountered during testing
- ✓ Email Notifications: Sent within seconds of complaint submission

PDF Form Generation: Passed 100% of test cases with correct auto-filled data

Besides these metrics, feedback sessions were held in order to gather insights on user experience and identify possible areas for improvement. The ease of interface and clarity in instructions were some of the compliments from students. Having the feature of monitoring complaints in real time was recognized as an important aspect that eliminated anxiety and ambiguity. Further, having the option to download a PDF form gave a touch of seriousness and formality to the complaint registration process.

Admin users, especially the registrar, were extremely satisfied with the dashboard. They were able to view complaints, check students' information, and mark complaint status with ease. Having automated email notifications was greatly appreciated as it removed the need to constantly check the system for new submissions.

Stress testing was also done to assess the performance of the app under greater loads. Simulations were carried out to simulate multiple complaint submissions within brief time intervals, and the system was able to maintain its performance without crashes or delays. The Firebase Firestore database also showed great stability and speed in processing concurrent requests.

Compatibility testing made sure that the application ran seamlessly on different Android devices, screen sizes, and operating system versions. All critical functionalities such as login, complaint registration, PDF download, and status updates worked well in tested environments.

Overall, the testing process assured that Poornima Help fulfills its intended purposes of speed, reliability, usability, and security. Given the successful test results and user responses, the application is ready for wider institutional rollout.

5. CONCLUSION

In summary, Poornima Help is a well-considered combination of cloud computing and mobile technology to address a pervasive and delicate issue in schools. The application improves trust in the complaint process through secure authentication, instant acknowledgment, and open status monitoring. It streamlines the administrative workload by categorizing complaints and minimizing paperwork. In addition, the user-friendly interface makes it possible for students with varying technical expertise to effectively utilize the application. By bridging communication gaps and facilitating quick action, this application helps promote a safer and more inclusive campus environment.

The real-time capabilities of the application, including automated alerts and real-time status updates, demonstrate its preparedness for deployment at an institutional scale. It serves to empower students by providing them with a channel through which they are heard, and it allows college administrations to respond quickly and functionalities such as auto-filled PDFs, offline form saving, and email alerts is evidence that the app resolves issues of usability and operational effectiveness.

From a technological standpoint, the implementation of Firebase Authentication and Firestore has been crucial to attaining the project's main goals—security, scalability, and real-time data management. The application's modular nature enables it to be easily customized and future-proofed for integrating new features, making it stay responsive to changing institutional requirements. Its user-friendly interface also supports a diverse student body that may have limited technical backgrounds.

Aside from its technical value, Poornima Help is a social utility that supports the institutional dedication to student well-being. It can remake how schools deal with delicate complaints by empowering students to act without fear and providing administrators with the capability to deal with complaints effectively.

As schools continue to computerize their administrative operations, programs such as Poornima Help illustrate how technology can be used to develop secure, student-centered, and accountable spaces. With ongoing updates and support, this app can be used as a model for other schools nationwide wanting to advance their anti-ragging systems.

6. FUTURE SCOPE

Some of the potential enhancements and feature additions have been noted:

Multi-language Interface: Adding regional languages to make it more accessible. Since student’s hail from various linguistic backgrounds, providing the app in Hindi, Rajasthani, or other regional languages will make it more user-friendly and language will not act as a hindrance in reporting incidents.

Anonymous Complaint Option: Enable anonymous mode to shield whistleblowers. Permitting students to file complaints anonymously can vastly improve reporting levels, particularly under extremely sensitive or fearful situations.

Admin Analytics Dashboard: Graphical representation of trends and patterns of complaints. Dashboards in the form of pictures can give administrators a glimpse of repeated issues, times of most ragging, and departments experiencing more frequent complaints, allowing evidence-based policy decision-making.

Emergency Contact Integration: SOS button that directly connects to campus security. An emergency button with a single tap can alert campus security or assigned individuals instantly in the event of a serious incident, thereby preventing harm or escalation.

Push Notification Support: Real-time notification of complaint updates and safety updates. Notifications can provide reminders for changes in complaint status, safety information, and awareness messages on anti-ragging laws and student rights.

AI-Based Filtering and Classification: Incorporation of artificial intelligence to filter complaints by keyword, severity, or urgency. This will enable administrators to quickly triage and respond to the most serious issues.

Media Attachments: Enable students to attach pictures, audio recordings, or short videos as supporting evidence when making reports. Media support may enhance the validity and severity of reports.

User Feedback System: Allow students to rate their experience and leave feedback on resolving complaints. This would encourage ongoing improvement and allow measurement of trust in the system.

Cross-Institution Collaboration: Subsequent releases could enable the app to be scaled across institutions, with anonymized data being shared to comprehend national trends and improve anti-ragging initiatives collectively.

Role-Based Access for Faculty Advisors: Adding an additional user role (e.g., faculty counselor or mentor) that can act as an intermediary between students and administration may provide a more facilitative resolution process.

Cloud Storage Improvements: Adding file compression, archiving of expired complaints, and improved data structuring within Firestore to keep performance optimized over time as usage is scaled up.

These updates not only intend to make the app completer and more expandable but also to keep up with the changing demands for digital safety and student agency. Adding these future features will make the app an even more powerful tool for creating a secure and responsive learning environment.

REFERENCES

[1] S., S., H., M., D., T., & S., S. (2022). Real-time crime detection using customized CNN. 2022 1st International Conference on Computational Science and Technology (ICCST), 416–419. https://doi.org/10.1109/ICCST55948.2022.10040379

[2] Arpa, K. F., et al. (2023). A machine learning and deep learning integrated model to detect criminal activities. 2023 4th International Conference on Big Data Analytics and Practices (IBDAP), 1–6. https://doi.org/10.1109/IBDAP58581.2023.10272002

[3] Belsare, V., Karande, N., Keskar, A., Joshi, S., & Karnavat, R. (2024). Context-based crime detection: A framework integrating computer vision technologies. 2024 MIT Art, Design and Technology School of Computing International Conference (MITADTSoCiCon), 1–6. https://doi.org/10.1109/MITADTSoCiCon60330.2024.10575348

[4] Anandhi, R. J. (2023). Edge computing-based crime scene object detection from surveillance video using deep learning algorithms. 2023 5th International Conference on Inventive Research in Computing Applications (ICIRCA), 1159–1163. https://doi.org/10.1109/ICIRCA57980.2023.10220800

[5] Yadav, R. K., & Kumar, R. (2022). Inflated 3D convolution network for detecting anomalies in surveillance videos. 2022 4th International Conference on Advances in Computing, Communication Control and Networking (ICAC3N), 2518–2522. https://doi.org/10.1109/ICAC3N56670.2022.10074290

[6] Rishika, S. (2022). Object detection using artificial intelligence. International Journal of Advanced Research in Computer and Communication Engineering, 11(5), 882–887. https://doi.org/10.17148/IJARCCE.2022.115193

[7] Alom, M. Z., Hasan, M., Yakopcic, C., Taha, T. M., & Asari, V. K. (2021). Inception recurrent convolutional neural network for object recognition. Machine Vision and Applications, 32(1). https://doi.org/10.1007/s00138-020-01157-3

[8] Tan, L., Huangfu, T., Wu, L., & Chen, W. (2021). Comparison of YOLO v3, Faster R-CNN for real-time pill identification. Department of Pharmacy, The Third Affiliated Hospital of Southern Medical University. https://doi.org/10.21203/rs.3.rs668895/v1

[9] Nandhini, T. J., & Thinakaran, K. (2022). CNN based moving object detection from surveillance video in comparison with GMM. 2022 International Conference on Data Science Agents and Artificial Intelligence (ICDSAAI). https://doi.org/10.1109/ICDSAAI55433.2022.10028909

[10] Wang, Y., Xu, N., Liu, A.-A., Li, W., & Zhang, Y. (2022). High-order interaction learning for image captioning. IEEE Transactions on Circuits and Systems for Video Technology, 32(7), 4417–4430.

https://doi.org/10.1109/TCSVT.2021.3121062

[11] Cao, S., An, G., Zheng, Z., & Wang, Z. (2022). Vision-enhanced and consensus-aware transformer for image captioning. IEEE Transactions on Circuits and Systems for Video Technology, 32(10), 7005–7018. https://doi.org/10.1109/TCSVT.2022.3178844

[12] Tian, Y., Pang, G., Chen, Y., Singh, R., Verjans, J. W., & Carneiro, G. (2021). Weakly-supervised video anomaly detection with robust temporal feature magnitude learning. 2021 IEEE/CVF International Conference on Computer Vision (ICCV), 4955–4966. https://doi.org/10.1109/ICCV48922.2021.00493

[13] Zeng, X., Li, Z., & Zhang, W. (2021). An adaptive learning-based weakly supervised object detection via context awareness. 2021 2nd International Conference on Big Data & Artificial Intelligence & Software Engineering (ICBASE), 331–335. https://doi.org/10.1109/ICBASE53849.2021.00068

[14] Hemmler, Y. M., & Ifenthaler, D. (2022). Indicators of the learning context for supporting personalized and adaptive learning environments. 2022 International Conference on Advanced Learning Technologies (ICALT), 61–65. https://doi.org/10.1109/ICALT55010.2022.00026

[15] Hmidani, O., & Ismaili Alaoui, E. M. (2022). A comprehensive survey of the R-CNN family for object detection. 2022 5th International Conference on Advanced Communication Technologies and Networking (CommNet), 1–6. https://doi.org/10.1109/CommNet56067.2022.9993862

[16] Wu, C., Zhao, P., Zhang, T., Zhang, W., Meng, F., & Guo, R. (2023). A small target detection algorithm of complex traffic road based on SDCYOLO. 2023 IEEE 6th International Conference on Electronic Information and Communication Technology (ICEICT), 1–4. https://doi.org/10.1109/ICEICT57916.2023.10245967

[17] Giri, S. R. K. S., L., P., B. S., S., Praba, R. D., Kavitha, K., & Kalaiselvi, A. (2023). Traffic surveillance system using YOLO algorithm and machine learning. 2023 2nd International Conference on Advancements in Electrical, Electronics, Communication, Computing and Automation (ICAECA), 1–6. https://doi.org/10.1109/ICAECA56562.2023.10201021

[18] Dodia, S., & Kumar, S. (2023). A comparison of YOLO based vehicle detection algorithms. 2023 International Conference on Artificial Intelligence and Applications (ICAIA) Alliance Technology Conference (ATCON-1), 1–6. https://doi.org/10.1109/ICAIA57370.2023.10169773

[19] Do, Q. T., & Chaudri, J. (2022). Creating computer vision models for respiratory status detection. 2022 44th Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC), 1350–1353. https://doi.org/10.1109/EMBC48229.2022.9871978

[20] Shah, N., Bhagat, N., & Shah, M. (2021). Crime forecasting: A machine learning and computer vision approach to crime prediction and prevention. Visual Computing for Industry, Biomedicine, and Art, 4. https://doi.org/10.1186/s42492-021-00075-z

[21] Pulipalupula, M., Patlola, S., Nayaki, M., Yadlapati, M., Das, J., & Reddy, B. R. S. (2023). Object detection using You Only Look Once (YOLO) algorithm in convolution neural network (CNN). 2023 IEEE 8th International Conference for Convergence in Technology (I2CT), 1–4. https://doi.org/10.1109/I2CT57861.2023.10126213

[22] Kilic, S., & Tuceryan, M. (2023). Crime detection from pre-crime video analysis with augmented pose information. 2023 IEEE International Conference on Electro Information Technology (eIT), 1–6. https://doi.org/10.1109/eIT57321.2023.10187325

[23] Mandalapu, V., Elluri, L., Vyas, P., & Roy, N. (2023). Crime prediction using machine learning and deep learning: A systematic review and future directions. IEEE Access, 11, 60153–60170. https://doi.org/10.1109/ACCESS.2023.3286344

[24] Kwan-Loo, J. C., Ortíz-Bayliss, J. C., Conant-Pablos, S. E., Terashima-Marín, H., & Rad, P. (2022). Detection of violent behavior using neural networks and pose estimation. IEEE Access, 10, 86339–86352.

https://doi.org/10.1109/ACCESS.2022.318985