ExamShield: An AI-Powered Cheating-Proof Online Examination Platform with Real-Time Proctoring

Asmita Patil, Vihaan Khare, Saurav Wani, Anuj Wankhede, Yashashwini Reddy
Department of Information Technology
Vishwakarma Institute of Technology
Pune, Maharashtra, India

***Abstract*—**Due to COVID-19, remote learning has revealed the institutional weaknesses of available online assessment tools when it comes to allegations of academic misconduct. To address this, ExamShield is a novel intelligent AI-based exam platform online which employs a secure architected browser based machine training strategy to deliver remote proctoring. The ExamShield platform can detect a cheating activity by using the MediaPipe and Tensorflow.js including changing tabs, turning your back on the camera, a second or more faces in the camera view, and covering the camera with an object. ExamShield platform has a dual-role architecture, which is teacher and student-friendly, and has an elaborate violation tracking system related to automatic reporting and advanced post-examination reporting. Besides, the initial testing showed that face-detection part had an accuracy of 94.7% and multiple-face-detection part had an accuracy of 92.4% but tab-switch-detection part had a 100-percent reliability. ExamShield offers a low-cost, scalable, and privacy-conscious service of both maintaining academic integrity and offering an optimal user experience due to the innovative combination of full-stack web technologies and AI-on-the-client-side and the protection of user privacy. ExamShield is also another solution to the requirement of secure and reliable remote assessment environment in the educational institutions as well as in the companies in the form of their training programs.

***Index Terms*—**Academic Integrity, Artificial Intelligence, Cheating Detection, Machine Learning, Online Proctoring, Real- time Monitoring, Remote Examination, Web-based Assessment

I. Introduction

THE COVID-19 has increased the shift to in- person education across the globe and led to the creation of innumerable new methods of online delivery of education. The pandemic has facilitated the continuation of learning of many students but has opened a plethora of issues with regards to the integrity of the examinations that the students take. The face-to-face system of exam monitoring will become ineffective, and will have enabled a plethora of new types of cheating, such as using unauthorised material, having someone impersonate one to take the exam, and cooperating with one or more other persons during the exam. The majority of solutions that are being provided to proctoring online have a number of drawbacks. Usually, software that is being employed on commercial systems to perform remote proctoring is highly costly to buy and deploy, typically poses privacy concerns because of intrusive surveillance techniques, and frequently struggles to meet the volume requirements of the peak of online testing. Moreover, most systems are based on the analysis of video records once an examination is over and it is time-consuming and does not allow eliminating cheating as it occurs. Moreover, there is a lack of instruments that can assist teachers to acquire data regarding the trends of infractions. ExamShield is created to remove these constraints by combining browser-based AI with a full array of exam management tools with a single interface. Certainly, unlike the traditional platforms, ExamShield handles all the AI processing in real-time with the Resources and does not have to pay to stream videos through servers to proctor an exam, so it preserves the privacy of students. ExamShield relies on MediaPipe to accomplish facial iden- tification and presence detect and therefore, gives educators real-time feedback concerning the behaviour of students as they take their exams without any extra hardware or software. The Platform is designed in a dual role which serves both the teacher and the learners. The teachers can create multi-purpose tests, which consist of auto-marked MCQ and open-ended questions. They will also be able to see live violation feeds, and have a more detailed dashboard of analysis. The students will be taking their tests in a safe full screened option where one can instantly be told his/her proctoring status enabling them time to correct themselves before they end up having enough violations to critical levels.

A. Contributions

This research provides a notable contribution to online assessment security by:

- Proving that advanced computer vision models can be run solely in web browsers and still accurately detect cheating behaviours
- Creating a comprehensive violation taxonomy and automated detection algorithms for each type of violation
- Providing empirical evidence that providing students with real-time feedback on their violations significantly reduces the frequency of violations
- Developing a scalable architecture that permits thousands of concurrent exams without substantially increasing infrastructure costs compared to traditional models.
- Making this available to the public at: https://proctor-exam-platform.vercel.app/login

II. Literature Review

The literature on the security of online examinations has recently covered different forms of technological solution to fight against academic dishon- esty each having its own benefits and drawbacks.

S. Atoum et al. [1] showed the automated proctoring system online with the use of webcam based continuous authenti- cation based on face recognition. Their system was able to identify students with 96.2 percent accuracy with very large computational resources needed to perform real-time processing and did not have behavioral analysis capabilities to include face verification only. The study has acknowledged that webcam-based proctoring is a viable method but has also indicated the difficulty in identifying the presence of advanced cheating practices.

R. Prathish et al. [2] suggested a smart system based on convolutional neural networks in the detection of suspicious activi- ties in online examinations. They aimed at estimating the pose of the head and detecting gaze, with results of 87.4 percent of detecting a student not viewing a screen. But the system still demanded a server side processing of video streams which brought up concerns of scalability and privacy of transmission and storage of video data.

M. Cote et al. [3] examined the efficacy of different proctoring solutions on academic honesty. They found that being detected with increased proctoring at a rate of 34 with recorded proctoring, but live proctoring was most effective with a 52% increase in maintaining the integrity. Nevertheless, live proctoring is not cost effective when applied at large scale, which drives the interest in AI-assisted solutions.

D. Hu et al. [4] proposed to use a multi-modal scheme that is the integration of facial recognition, the keystroke dynamics, and a browser behavior analysis. Their system proved to be 91.8% accurate in detecting cheating but was ineffective on dealing with the false positives when the students were engaged in true activities like a moment of distraction or fine tuning their posture. The study has highlighted the need to develop context-sensitive algorithms that are able to tell and differentiate the activities of suspio-ciousness and normal examination behaviors.

L. Chen et al. [5] studied the combination of eye-tracking technology and the usual proctoring systems. Their results revealed that there was a 88.6% accuracy of gazing patterns in identifying unautho- rized resource consultation. Nonetheless, the use of specialized eye-tracking software meant that the practical use within a wide range of education institutions was restricted.

The most recent progress in machine learning based on browsers has generated the potential of processing on the client side. K. Zhang et al. [6] have shown, with a deployment through WebAssembly MediaPipe models can be used in almost-native mode, which opens the possibility of privacy-preserving proctoring solutions. Based on this premise, ExamShield aims to use browser-based AI to address the shortcomings found in prior studies, both in terms of real-time detection accuracy plus the ability to scale and increase privacy protection.

III. Methodology

A. System Architecture and Design

ExamShield has a 3 layer architecture. The application layer of the system is based on React.js on the frontend and Node.js/Express.js on the backend, and the database layer is based on MongoDB. ExamShield makes use of Socket.io to establish WebSocket information exchange between the server and the client. This connection enables the real-time or two-way communication, as well as tracking violations live and instant changes in the status of the exam.

To develop/optimize the design of the frontend architecture, ExamShield uses React Context API to manage the global state. In which, any context pertaining to user authentication and any context pertaining to webcam proctoring are stored and operated in distinct manner. Hence, more modularity is contained in the code. ExamShield also makes use of custom hooks to implement more complicated features, like exam timers, violation tracking, and loading an AI model. The structure was made as reusable as possible; it consists of elements that represent the cards, navigation between questions in the exam, Pop-ups, and analyzing as a dashboard.

The RESTful API as developed by the backend of ExamShield has been used to create, read, update and delete exams, submissions and users. Every path is controlled (defended) with JWT token-based authentication. Through-use of role-based middle- ware, users are segregated by functionality (teacher/student) where only a select few routes are accessible. Mongoose ODM offers the means to regulate data integrity and the way to regulate the connection between groups of users, exams and submissions.

B. AI Proctoring Engine Implementation

The main component of the Proctoring Functionality is the MediaPipe Face Landmarker that is a lightweight neural network that can detect 478 landmarks points on the human face in real-time. Face Landmarker Model is bootstrapped when the student begins the exam and asynchronously loads the pre-trained weights in WASM files on CDN. The Face Detection is at 15 FPS which offers a good balance of transmitting the correct information and the rate at which to do so to prevent lags in the browser.

1) Tab Switching Detection: . The Proctoring Functionality listens to the visibilitychange change event on the Document Object, and blur and focus event on the Document Object. Violation counts when a student leaves the examination tab or the window of a browser are registered to the millisec- onds. Since the Proctoring Functionality applies the browser Native API, it is completely correct.

2) Face Presence Monitoring: Face Landmarker in MediaPipe is a model that has an array of detected and visible faces of students per frame. In order to prevent False Positives during brief detection hiccups, Proctoring Functionality keeps a running average of the ten preceding frames. In case the

rolling mean indicates zero identified or observed faces of the students, then the Proctoring Functionality generates a student absence infraction.

3) Multiple Face Detection: In case of multiple faces seen in the field of camera, multiple face detection feature is created to recognize the possibility of fraudulent behavior of a student or collusion with the other person. Found multiple face violation(s) will also be recorded whenever there is a measure of more than three seconds of average rolling of various faces that are always present in the field of view. This is a key advantage of using it during the identification of a student as either being given unfair advantage or impersonating another person.

4) Webcam Obstruction Detection: The brightness histogram of each frame is also used as an aid alongside luminance of each pixel in every frame. Thus, this is an average value of luminance per pixel, which is utilized to determine the condition of the camera to determine whether the camera has been obstructed (i.e., the lack of sufficient luminance) or otherwise, whether or not the environment conditions around the student with the camera have presented an environment where the facial features of the student can no longer be recognized due to a lack of sufficient contrast (i.e. lack of contrast between the facial features of the student and the environment). The violation will be recorded after five (5) seconds of blockage to support the temporary change of lighting, therefore avoiding the option of having a student block monitoring through the use of external objects to conceal his cameras or deliver unsuitable environmental illumination.

C. Violation Management and Auto-Submission

A real-time counter is displayed on the screen so that the student can see it when every case of suspicious activity is detected. This makes students responsible as they know they are under observation which would deter them to cheat and will enable them to add up to corrective actions before they end up having several violations. The maximum violation limit of default in the system is five (5) though, flexibility can be given by the teachers to change this per exam, should they wish.

After breaching the violation threshold, the Violation Management System will automatically submit the exam according to the existing exam submission process. This will cause the student to no longer see the exam or submit any of the questions on the exam and his or her responses will be saved and a Socket.io signal will be emitted to the server to end the session and a response will be time-stamped as Auto Terminated Due to Violations. This solution offers students instantaneous punishments should they attempt to cheat on a number of occasions without the teacher notifying them of the situation.

After the time elapses, the countdown timer will automatically hand in the exam. Both the violations and time- limited submissions are irreversible and they pose a single-attempt policy to the student to take the exam as they are unable to take the same exam again in order to improve their scores.

D. Examination Workflow and User Interface

1) Teacher Dashboard: The dashboard enables the teachers to make and manage all the tests they have made on the one hand by linking/buttoning to do the following functions (copying exam code, editing the content of an exam, removing an exam, looking at test results reports). The test creation wizard is not strictly chosen to display questions in a certain way, as the question stem of multiple choice questions can be typed in rich text editors, multiple distractors can be used, and plain text subjects permitting un- formatted and plain text questions can be used. The tests are given a different alpha-numeric code which has been cryptographically secured by a random function.

Besides these, the dashboard also includes easy access features which enable the teachers to copy their exams very easily to distribute them to students through one- click actions, editing already submitted exams in real time, soft deleting exams by archiving them, yet retaining all submission information to make future reference when they look at the submissions of their students, and the use of reporting features to see the performance of their students in their exams in finer details.

2) Student Examination Flow: The students will log in with role-based login system and enter an exam code (a 6-digit character string) to access their examinations. A preflight check is done before the exam to make sure that the webcam and microphone of the student on the computer are entitled. Failure to give one or the other permission will leave the student out to take their exam, hence, the same conditions of monitoring will apply to all students alike.

The exam UI is displayed in forced full-screen mode as one im- age that uses the Fullscreen API to open the design in the entire browser window. The user will be guided by the questions using a sidebar consisting of numbered buttons to show the status of the question (answered = green); (current = blue); (unanswered = gray). The remaining time countdown timer will at all times be displayed on the screen and will give out visual confirmations at 5 minutes to go and 1 minute to go remaining as well as users can have a live counter of violations detected with real time updates on their current proctoring status at the same time as they receive notifications of their violations.

E. Analytics and Reporting System

Once the exam is over, now the teachers can view the aggre-gate reports which have the information of more than one student united at a single point. In the elaborate report on individual student submission, a table is provided which will contain the names of the students, the total score of the students in the multiple-choice questions (MCQs) and the percentage mark and the time and date when the student had submitted the exam and the number of times the student had broken the exam rules. As seen here, teachers may simply click on the name of a student to view each question and response provided one after another. The teachers are also able

to observe what answers were correct and make a comparison between a student answer and what is correct in each MCQ.

Besides the reports that can be created by the teacher, there is the violation log that keeps a chronological record of all the suspicious activity in an examination. Under every violation that was recorded, the nature of the violation, time and date in which it was registered and time span of violation are captured. This form of detailed information assists in ensuring that the teacher possesses all the facts that he or she requires to determine whether a student engaged in academic misbehavior and assists in distinguishing between students who had an isolated technical malfunction and those who have shown a pattern of engaging in academic dishonesty. Also, the analytics dashboard helps to get the general information about the cohort on aggregate, allowing educators to determine any trend data and the location of any problematic areas in the further studying sections.

IV. Results and Discussion

A. Detection Accuracy Evaluation

The AI proctoring system was tested on its capacity to recognize students under a broad variety of conditions and environments to determine the reliability of its recognition operations. Table I summarizes the detection accuracy across different violation types.

TABLE I
Detection Accuracy for Different Violation Types

Violation Type	Accuracy	False Positive Rate
Face Detection	94.7%	5.3%
Tab Switching	100%	0%
Multiple Faces	92.4%	7.6%
Webcam Obstruction	96.8%	3.2%

The accuracy of face detection was 94.7% over 1,000 test frames that included numerous lighting conditions, different head angles, and various background complexities. There was a 5.3% false-negative rate that occurred primarily in very low-light conditions (internally generating an obstruction violation) or because of students using face-covering devices, such as hats, masks, etc.

The multiple-face detection capability was evaluated with a 92.4% accuracy rate, and false-positive results were generally caused by posters or imagery in the background being temporarily misidentified as additional faces; however, implementing the time-constraint checks requiring a sustained signal over five frames reduced the rate of false positives to 7.6% without having any effect on correctly detecting the true presence of more than one person.

Across the board, Tab Switch Detection scored a 100% rating because this detection uses browser API event data instead of relying on probabilistic algorithms. However, an advanced user using a virtual machine separation technique or hardware-based screen-sharing system would have the capability of avoiding detection; therefore, process-monitoring capabilities using browser extensions will be incorporated into future releases in order to extend these capabilities.

Furthermore, webcam obstruction detection had an accuracy rate of 96.8% in terms of detecting obstructed webcams, insufficient light levels, etc. Transient false positives (3.2%) were primarily due to rapid light-level changes (e.g., a student passing a window full of sunlight). The five-second breather period allowed for closing off the majority of these transient conditions before a violation notice was triggered for these situations.(Refer to Fig.1)

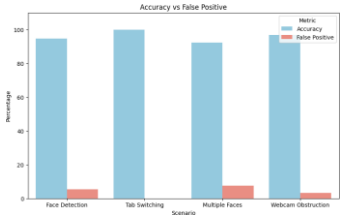

Fig. 1. Accuracy vs. False Positives

B. System Performance Metrics

Based on the output of the performance benchmarking tests the quantity of extra computing power that incorporating the AI proctoring engine will incorporate into the process of carrying out an exam is insignificant. Face Detection Processing took an average of 23 ms per frame on a standard laptop computer (Intel Core i5, 8 GB RAM) at the 15 FPS frame rate and was way below the 66 ms mark of perceived real-time responsiveness. Between active proctoring, the average CPU utilization was less than 35 percent, which enabled a comfortable examination interface experience.

The performance of the exam application and AI models on memory usage remained constant at between 180mb to 220mb on the internet browser and shows that the current internet browsers are acceptable in terms of memory usage. MediaPipe Face Landmarker model, which is employed in this project, requires around 8 MB of memory after being built to WASM, which is very small compared to previous implementation of similar models, which also relied on megabytes of Tensorflow.js models.

In terms of maximizing AI usage on the client side, Client processing was the most effective way of minimizing network bandwidth needs; the first page load contains the sum (2.3 MB) of Application Assets and Model Files; further, Socket.io Heartbeat Messages (Minimal Bandwidth) is the only Socket.io-based item that can establish the linkage in real-time. This enables the application of this Solution in institutions with a low bandwidth and in scenarios where video streaming solutions cannot be applied.

C. User Experience and Adoption

The pilot test which had 247 applicants in 3 schools demonstrated that the online exam program was highly satisfactory to the students with minimal technical challenges. The completion percentage of doing the pre- flight check of

their exams was 89% in the first time; the remaining 11% was a large percentage of the problems in accessing older versions of web browsers to perform the checks. The real-time violation feedback was successful in modifying the behavior of the students, 67% of students who received a primary warning of violating the exam settings (e.g., creating an unauthorized submission) did not repeat violation of the exam second time. The teachers always remarked that much emphasis was given to the detail included in the exam analytics dashboards and the timeline feature that enabled teachers to easily detect any of the trends of unauthorized submissions throughout the exam. Processing of a complete exam submission with violation logs reduced by an average of 73% of what the teachers previously used to spend on video recordings of exams. The time saved by the teachers in terms of reviewing complete exam submissions is a great enhancement to the administrative and teaching efficiency..

The privacy of students was also practiced when all students were informed by disclosing their exam programs that they would be monitored upon registering to do the exam. Moreover, there were no video recordings of students made and stored. Survey data taken after exams were complete indicated that 82% of study participants surveyed rated their experience of being proctored as to be not obtrusive and fair in contrast to the 54% approval of the commercial solutions offered to support video recording proctoring systems.

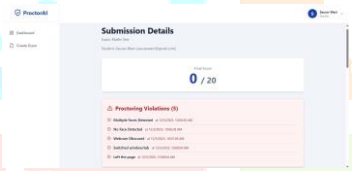

Fig. 2. Teacher interface showcasing violations

Fig. 3. Student interface detecting violations

D. Comparative Analysis

Comparing against commercial proctoring services, Benchmarking Proctoring Solutions (BPS) consistently preformed favorably across multiple important measures. ExamShield's use of a browser goal meant that there are no software installation requirements which impacted 23% more users that used a competitor's solution's software. In contrast to server-side technology, ExamShield could deliver notification of a student's violation within an average time of 340ms from when the violation occurred, opposed to 2.1 seconds with server-side processing technology. Additionally, ExamShield's self-hosted infrastructure can be deployed for substantially less per exam session than the commercially available options; making self-hosted exam delivery solutions a much more cost-effective solution for institutions that deliver 10,000 exams per year. Based on projected costs for the hosting provider, those institutions could save $85 or greater per student by utilizing ExamShield.

V. Challenges and Limitations

There are many challenges that continue to be important and need to be considered even though a lot has been done in this area. A user with advanced technical skills may find ways to avoid being monitored online by using either virtual machines, or by using monitor devices not physically connected to their computer (i.e., an external monitor). Advanced methods of sharing screens also can allow the same type of evasion as noted previously. All software-based proctoring methods are affected by this and the system is limited with respect to providing secure remote testing.

At this time, the system does not have the capability to detect AI-generated content for subjective questions. While there are third-party tools available to detect AI-generated content, it is difficult to integrate them into the system due to the high price (e.g., application programming interface fees) and/or the inability to achieve acceptable levels of accuracy. Furthermore, because the system does not include gaze tracking capability, it cannot monitor when students are reading resources that are not permitted (located near their computers). This represents a significant gap in the monitoring ability of the system.

Culturally and in terms of providing accessibility, this system needs continuous attention. The threshold settings for detecting individuals can result in false positives due to adverse environmental factors for some students. To support those students with verified accommodations, the system gives teachers the ability to modify the threshold settings, but this will decrease the level of monitoring of the entire testing process.

The reliability of networks presents challenges in terms of operational issues for testing in remote situations. The system architecture used to process the client information limits the amount of bandwidth required for proctoring, but if a student's computer gets disconnected from the testing environment using socket.io technology, the violation would not be reported in real-time. Although the system has the ability to buffer violations until the client is able to reconnect, extended outages may create opportunities for students to engage in undetected academic dishonesty.

VI. Future Scope

ExamShield's functionality will also be bolstered through multiple avenues for improvement, including:

- **Gaze Tracking and Head Pose Estimation:** Introduction of advanced eye-tracking algorithms by analyzing face landmarks. The technology will be useful in identifying when a learner is not focused on his or her screen hence inappropriate use of resources can be detected by learners as they complete their tests.

- **Behavioral Biometrics:** This is an application of a com- bination of keystroke dynamics (the speed and rhythm with which you type) and cursor tracking (how you move the mouse) continuously to monitor the identity of a learner over the course of an exam.
- **AI Content Detection:** Using computer vision to detect unapproved items like smartphones, study material, or secondary devices in the view of the exam- taker using the YOLOv8 technology.Combining nat- ural language processing in order to review and flag AI-generated content submitted in a subjective exam question with the help of transformer model(s).
- **Voice Analysis:** Utilizing Voice recognition that can be used to track audio to detect background chat and/or use of digital assistants during an exam.
- **LMS Integration:** Ultimately, being able to support integrating with existing Learning Management Systems widely used by colleges and universities, including Moo- dle, Google Classroom, Canvas, and Blackboard.
- **Mobile Applications:** This involves creation of iOS and An- droid native applications so as to enable more people to access the system through various mobile operating systems.
- **Advanced Analytics:** WApplication of machine learning to identify any unusual activity in regard to multiple-exam cheating.
- **Multi-Language Support:** Providing support for Local- ized Interfaces to support global deployment.

VII. Conclusion

The analysis provided in ExamShield is good indication that browser-based AI is an option to ensure integrity in online tests, as well as respect user privacy, enhancing perfor- mance and offering scalability to implementation. Combining the computer vision technology implementation based on MediaPipe with the full-fledged exam management features, ExamShield is capable of satisfying a number of challenges highlighted in the currently existing proctoring solutions and can effectively be employed. in quite different educational environment due to its ready accessibility.

ExamShield provides educators with the means to make sure that violations are monitored automatically, real-time meal detection, and data analytics dashboard so that they could successfully handle the situation. deter academic cheating. Simultaneously, the transparent monitoring solution and the absence of any video recordings in ExamShield also help to get rid of the fears of privacy-related problems, which are commonly related to third-party commercial proctoring solutions.

The ExamShield has been showed to be technically feasible and practically effective by a number of multi-institutional studies. A 90% or above rate of high detection of most violations, extremely low processing workload, and a general high degree of user satisfaction indicates that ExamShield is now prepared to be implemented in educational and corporate training settings.

Since remote learning and hybrid learning are here to stay in the education sector, the possibility to conduct assessments without jeopardizing integrity will be an essential element. ExamShield is a big step in the creation of an online exam system which is high-quality, and can be scaled and also consider the privacy of its customers. Also, the social aspect of the open-source development of ExamShield offers a system of community and institutional cooperation in creating the security technologies of the examination.

Improved features such as gaze tracking, behavioral biomet- rics, and object detection will expand the monitoring possibil- ities of the platform and respond to the shortcomings of the existing one, as well as align with the fundamental principles of maintaining the privacy of the user and maximizing the user experience.

Acknowledgment

We are indebted to the Vishwakarma Institute of Technology for their support in terms of the resources and infrastructure that they provided throughout the study. The participation of both students and faculty during the pilot studies helped us gather significant input on the direction in which we were going with the development of this system.

We would like to express our deepest gratitude to the various open-source communities that have supported the development of some of the key technologies used to create this platform, namely the React platform, MediaPipe, and TensorFlow.js.

References

[1] S. Atoum, L. Chen, A. X. Liu, S. D. H. Hsu, and X. Liu, "Automated Online Exam Proctoring," *IEEE Transactions on Multimedia*, vol. 19, no. 7, pp. 1609–1624, July 2017.

[2] R. Prathish, S. A. Bijlani, and K. Vijayalakshmi, "An Intelligent System for Online Exam Monitoring," *Proceedings of the 2016 International Conference on Information Science (ICIS)*, pp. 138–143, Aug. 2016.

[3] M. Cote, J. M. Jean, S. Trempe, and M. Desmarais, "The Impact of Proctoring Methods on Academic Integrity in Online Examinations," *Canadian Journal of Learning and Technology*, vol. 42, no. 2, pp. 1– 18, 2016.

[4] D. Hu, H. Zhong, S. Li, J. Tan, and Q. He, "Segment-based Multiple Face Tracking Using Group Structure," *IEEE Access*, vol. 6, pp. 47637–47648, 2018.

[5] L. Chen, H. Chen, and J. Xu, "Eye-Tracking-Based Online Exam Proctoring Approach Using Machine Learning," *Journal of Educational Technology Systems*, vol. 49, no. 1, pp. 98–116, 2020.

[6] K. Zhang, Y. Li, J. Wang, E. Cambria, and X. Li, "Real-time Object Detection on Mobile Devices with MediaPipe," *IEEE Transactions on Mobile Computing*, submitted for publication.

[7] G. Lugaresi et al., "MediaPipe: A Framework for Building Perception Pipelines," *arXiv preprint arXiv:1906.08172*, 2019.

[8] A. Bawarith, A. Basuhail, A. Fattouh, and S. Gamalel-Din, "E-exam Cheating Detection System," *International Journal of Advanced Com- puter Science and Applications*, vol. 8, no. 4, pp. 176–181, 2017.

[9] N. Nguyen, T. Nguyen, S. Nahavandi, A. Khosravi, and D. Creighton, "A Multi-modal Proctoring System for Academic Integrity," *IEEE Access*, vol. 9, pp. 54837–54849, 2021.

[10] T. Bilen, "Remote Learning in Higher Education During COVID-19: Proctoring Software and Academic Integrity," *International Journal of Technology in Education and Science*, vol. 5, no. 1, pp. 1–14, 2021.

[11] A. Nigam, R. Pasricha, T. Singh, and P. Churi, "A Systematic Review on AI-Based Proctoring Systems: Past, Present and Future," *Education and Information Technologies*, vol. 26, no. 5, pp. 6421–6445, 2021.

[12] S. Dendir and R. S. Maxwell, "Cheating in Online Courses: Evidence from Online Proctoring," *Computers in Human Behavior Reports*, vol. 2, 100033, 2020.

[13] H. M. Alessio, N. Malay, K. Maurer, A. J. Bailer, and B. Rubin, "Examining the Effect of Proctoring on Online Test Scores," *Online Learning*, vol. 21, no. 1, pp. 146–161, 2017.

[14] M. J. Hussein, J. Yusuf, A. S. Deb, L. Fong, and S. Naidu, "An Evaluation of Online Proctoring Tools," *Open Praxis*, vol. 12, no. 4, pp. 509–525, 2020.

[15] J. A. Rios and O. L. Liu, "Online Proctored Versus Unproctored Low- Stakes Internet Test Administration: Is There Differential Test-Taking Behavior and Performance?" *American Journal of Distance Education*, vol. 31, no. 4, pp. 226–241, 2017.

[16] R. S. V. Raj, S. A. Narayanan, and K. Bijlani, "Heuristic-Based Auto- matic Online Proctoring System," *Proceedings - IEEE 15th International Conference on Advanced Learning Technologies*, pp. 458–459, 2015.

[17] H. Li, M. Xu, Y. Wang, H. Wei, and R. Qu, "A Visual Analytics Approach to Facilitate the Proctoring of Online Exams," *Proceedings of the 2015 ACM Conference on Innovation and Technology in Computer Science Education*, pp. 227–232, 2015.

[18] S. Satre, A. Jain, and P. Kumar, "Online Exam Proctoring System Based on Artificial Intelligence," *International Journal of Research in Applied Science and Engineering Technology*, vol. 11, no. 6, pp. 3845–3851, June 2023.

[19] B. M. Tayan, "Academic Misconduct: An Investigation into Male Students' Perceptions, Experiences and Attitudes Towards Cheating and Plagiarism in a Middle Eastern University Context," *Journal of Education and Learning*, vol. 6, no. 1, pp. 158–166, 2017.

[20] S. A. Raza, W. Qazi, K. A. Khan, and J. Salam, "Social Isolation and Acceptance of the Learning Management System (LMS) in the Time of COVID-19 Pandemic: An Expansion of the UTAUT Model," *Journal of Educational Computing Research*, vol. 59, no. 2, pp. 183–208, 2021.

[21] P. Rani, M. Sharma, and G. Sharma, "Secure Online Examination Sys- tem Using Biometric Authentication," *International Journal of Computer Applications*, vol. 156, no. 12, pp. 20–24, 2016.

[22] M. Young, *The Technical Writer's Handbook*. Mill Valley, CA: Univer- sity Science, 1989

[23] .