Design of Face Detection and Recognition System to Monitor Students During Online Examinations Using Machine Learning Algorithms

R.Manjupriya [1], Saran S [2], Varshanth D[3], Jotheshvar S [4]

[1]*Assistant Professor, Department Artificial Intelligence and Data Science, SRM Valliammai Engineering College, Kattankulathur,Chennai,India*
Email: manjupriyar.ai-ds@srmvalliammai.ac.in
[2] *Final year student, Department Artificial Intelligence and Data Science, SRM Valliammai Engineering College, Kattankulathur,Chennai,India*
Email: Saransivakumar25@gmail.com
[3] *Final year student, Department Artificial Intelligence and Data Science, SRM Valliammai Engineering College, Kattankulathur,Chennai,India*
Email: Varshanvarshan308@gmail.com
[4] *Final year student, Department Artificial Intelligence and Data Science, SRM Valliammai Engineering College, Kattankulathur,Chennai,India*
Email: jotheshvar182004@gmail.com

 DOA:01.12.2025 / DOP: 10.12.2025 ISSN: 3048-619X
Volume 2 Issue 3 [Sep – Dec 2025] DOI:

Abstract— The global pandemic has sped up the growth of online education, which has turned real classes into online places to learn. One of the hardest things for schools is making sure that students are present and that students are honest on online tests. This research uses the K-Nearest Neighbor (KNN) algorithm to demonstrate how a tracking system based on face recognition and detection could function. During tests, built-in webcams use face recognition to make sure that students are who they say they are. This also takes notes. It is the Python Face Recognition Library that trains the machine learning-based face embeddings that are used in the system idea. When people sign up for an online test, these embeddings are compared to listed datasets to make sure they are who they say they are. A management interface, a student site, and a database-driven authentication system that works with both MySQL and Java Servlet are all part of the basic model. There are pros and cons to the system and the basics of algorithms are talked about in this paper. It also lays the groundwork for real-time use in educational tracking systems in the future.

Keywords— Face Recognition, K-Nearest Neighbour (KNN), Online Examination Monitoring, Attendance Automation, Machine Learning, Student Authentication

I. Introduction

In-person administration of these tests is currently unfeasible because to the pandemic. Online lessons and assessments are increasingly transforming educational institutions. Monitoring the children during online instruction and assessments is more challenging. Robust facial recognition technologies are the most effective solution to address this problem. Currently, digital technologies for monitoring assessments play a significant role in maintaining good standards in technology-based courses. As an increasing number of students enroll in online courses, the likelihood of scams and identity theft escalates. Utilizing another individual's account and password to verify their identity is unacceptable. Verifying each student's identity is straightforward, expedient, and secure. Individuals are more inclined to trust online testing platforms. [5], [6]. Educational institutions around are endeavouring to enhance digital management systems and provide comprehensive evaluation of all students, regardless of their physical or virtual classroom settings. A tool called computer vision is being used for this. With machine learning, you can get accurate and flexible results when you do identification work. There are many kinds of algorithms. One of them is the KNN method. Good for jobs that need you to talk to other people. It works quickly and well. KNN gives test pictures a score based on how well they match up with pictures that know what they are. In terms of how far apart their feature vectors are, the picture that looks most like the test image gets the best score. [7]. The KNN method can be used with face embeddings made with Dlib or Python's Face Recognition module to make sure that students' names are spelled correctly even if the lighting or stance changes. It is the best choice for small, light school security systems because of this [8, 9]. The main purpose of this study is to develop a face recognition and tracking system that can verify students' names, watch test-takers, and keep real-time records of attendance. Front-end web technologies, recognition reasoning based on machine learning, and database-driven proof have made it simple for teachers and students to use. Not like some other systems on the market, this one doesn't need to be linked to the cloud or watched by someone. This system, like some others on the market, doesn't need to be connected to the cloud or watched by a person. But it's not expensive, and school PCs can use it. Its goal is to make AI-based biometrics work better so that online tests are safer, faster, and easier to get into.

II. Literature Review

Lot of new studies say that face recognition technology is being used to keep an eye on classroom and online tests. It is not optimal to use both online proctoring tools and standard reporting systems at the same time to monitor the employees.

This monitoring system is not cost effective [11]. One way to study fast check-in and arrival tracking is to use computer vision and machine learning. People stay away from these kinds of things to escape the issues listed above.Their feelings can be read from the way they look. Sharma et al. suggest deep feature extraction is used to find people in a CNN-based method for keeping track of class participation. Because it needed a lot of processing power, schools that didn't have a lot of resources couldn't use it at the same time.

Mohan et al. [6] also watched what the students using a cloud-based online proctoring system that could tell from the faces of students how they were feeling. It performed best when people were conversing, but it struggled in dimly lit areas and with inconsistent camera quality. People have also looked at Support Vector Machines (SVM) and Haar Cascade models as low-resource ways to find and identify faces [7]. These work well when the lighting is even and the face is looking forward. There is a lot of new study on embedding-based ways to make strong feature vectors, like FaceNet and Dlib. Finally, basic algorithms such as KNN can be used to group these together into useful results that can be used on a larger scale [8, 9].

Ashan et al. [9] shows the use of KNN models are in systems that check IDs in real time. KNN does not have much free time or work to do on the computer. It doesn't need to be told what they are. To do this, it finds the Euclidean distance between the test faces that are put in and the faces that are already there. It finds the Euclidean distance between inserted test faces and known faces in order to do this. It has also been shown that hybrid designs that use both simple classification algorithms and deep feature extraction are the best way to get the best of both speed and accuracy [10].

New studies also look at how to use proctoring tools that are run by AI and can watch over multiple devices, follow students' eyes, and see how they move [12, 13]. This is meant to make tests more dependable. You can use face recognition software to keep an eye on people automatically. It is also important to make sure that online schooling is fair and accountable. Based on these results, this paper presents an idea for a simple, KNN-based system that can securely and cheaply take care of both automatically recording attendance and verifying identities of the students.

III. EXISTING SYSTEM

Traditional ways of giving online tests depends on teachers or proctors keeping an eye on them. Visual verification on videoconferencing systems is usefull in keep tracking of student attendance [11]. This method might work for small classes, but it gets harder to use and more likely to make mistakes when there are more of them. It is hard to keep track of attendance by hand because it takes a lot of work and easy to make mistakes. There's a chance that teachers won't catch people pretending to be someone else or that they'll name the wrong people. This could leads to less accurate attendance records, which makes the whole test process less reliable [12]. Doing things by hand also makes it easy to cheat in school. It could change the results of the test, so don't do it. This means that kids might get bad grades and make growth that isn't fully shared. Examity and Respond us are two companies that watch people take tests for money. They do this by watching videos of people and writing down what they do [13]. People usually go to these sites to see strange eye or body movements or actions in the background. This is the main reason why you should use them. They only give you power; they don't do anything else. Because of who you are, they don't really connect you. In other growing [14]. When raw video or biometric data is kept on servers that are not owned by the company that made it, privacy and security concerns arise. It's possible that the way things are done now isn't right for live lessons. It is very important to have a way to check students correctly and actively before and during tests. Keeping track of numbers without anyone having to do anything. Find people who are trying to copy and stop them. Easy to set up and doesn't cost a lot of money good for schools with different amounts of money.

This is possible with a method that finds and recognizes faces, preferably one that uses KNN to

 DOA:01.12.2025 / DOP: 10.12.2025 ISSN: 3048-619X
Volume 2 Issue 3 [Sep – Dec 2025] DOI:

sort facial embeddings into groups. With this kind of system, biometric features are used to actively confirm student identity. This makes sure that attendance records are correct, tests are fair, and managers and people in charge don't have to do as much work. This can be integrated into existing online education platforms, ensuring fairness, transparency, and security across diverse learning environments [15], [16].

IV. PROPOSED SYSTEM

This system is designed to use face recognition technology to automatically check students and keep track of their attendance during online tests. It has a framework that makes sense. It's made up of five main parts, and each one works in a different way. Some people can use the Admin Sign-In Module to look at student data, post pictures of students, and set up test times. This keeps all the important settings and info up to date at all times. Once a student has been verified, the Student Sign-In Module makes sure they can safely get to the test spot. This keeps the login process safe and its most important part is the Face Detection and Recognition Module. A method called K-Nearest Neighbors (KNN) and face embeddings are used to make sure the authentication is right. This module checks each student's attendance and then keeps track of each student's attendance on its own. There is less chance of making a mistake and no one has to do it. The test after a test, the Attendance Audit Module makes detailed reports that show how involvement and attendance have changed over time.

The system takes a real-time picture of a student through the screen when they try to join in and takes out the important parts of their face. The saved dataset is put next to them after they are turned into embeddings. It is assumed that the student is who they say they are if the likeness is higher than a certain amount. If it is, attendance is automatically recorded. It speeds up the tests and makes sure everything is right and safe.

V. SYSTEM DESIGN AND ARCHITECTURE

The proposed system conceptualizes an integrated framework for automated student authentication and attendance marking during online examinations using facial recognition technology. The system is structured around five major modules. The Admin Sign-In Module allows authorized personnel to manage student records, upload facial images, and configure examination sessions. The Student Sign-In Module provides students with secure access to the examination portal after successful verification.

K-Nearest Neighbors (KNN) and face embeddings are both used by the Face Detection and Recognition Module to make sure the recognition is right. As soon as a student checks in, the present Update Module keeps track of their appearance in real time. Last but not least, the Attendance Audit Module will make data that shows trends in who showed up and how much they interacted after the test is over. After they log in, their camera will take a live picture of them. Then, any face features will be taken out, and the new embeddings will be compared to the dataset that was saved. If the level of the match is above a certain level, you can record numbers right away.

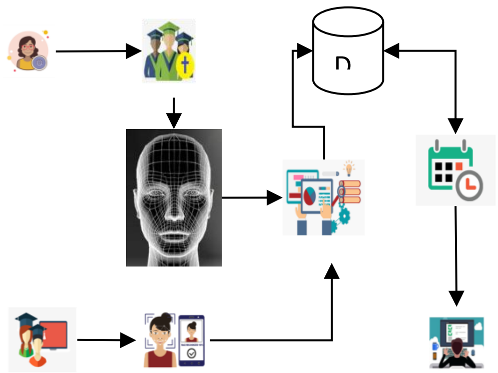

Fig 1: Architecture of the proposed System

A.concept workflow

Within the system concept, there are three levels. You can connect to a lot of different networks and gadgets through these layers, which make the

 DOA:01.12.2025 / DOP: 10.12.2025 ISSN: 3048-619X
Volume 2 Issue 3 [Sep – Dec 2025] DOI:

system flexible and scalable. It's easy for managers and students to use the Presentation Layer, which was made with HTML, CSS, and JavaScript. People use AJAX API calls to make things faster and more flexible. This layer, which is called the Application Layer, is made with Java Servlet technology. The main code for handling exams, keeping track of attendance, and logging in is in this file. This layer also answers calls for face recognition right away and talks to the database. The test scores, attendance logs, student information, and facial scans are all kept there. This layer is run by MySQL. Sure that all of your actions can reach the database and that the data is correct with safe JDBC connections. The system's layered design makes it possible for its parts to talk to each other and work well in extended testing. All of the test results, attendance records, student data, and facial embeddings are stored there. MySQL runs that layer. All the actions will definitely connect to the database and get the right data. Many tests have shown that the fact that it is made of layers works well. You can still talk to the other parts.

B. workflow overview

A lot of thought went into how the system works so that it can correctly check people's names and keep track of who was there. As soon as students sign up, their faces are taken in controlled areas so that a good collection can be made. The pictures are then made normal, shrunk, and turned into black as part of the editing step. This makes sure that every shot is the same so that the features can be taken out of them.

The Face Recognition package in Python is used to give each face shot a 128-dimensional embedding. After this, the KNN model learns how to put new faces it sees on online tests into the right group. Right now, pictures from the webcam are being compared to photos that were saved during test sessions. These kids can't get in until their attendance is checked and changed right away.

VI. Algorithmic Framework

K-Nearest Neighbor (KNN) algorithm is the traditional supervised learning method that is often used for classification. The basic idea behind it is simple, but it works well: the class of a new, unlabelled sample is based on the class that its k close Neighbors in the feature space are most likely to be. KNN doesn't use explicit model training either, just like parametric methods. During the inference phase, it remembers the whole training dataset that was given to it earlier. It then uses the data it has remembered to make forecasts. For KNN to work well, there should be a lot of cases in the feature space, as well as useful gaps between them.

An example of a feature space is written as X and T as a named training set.

$$T = \{(X_{1,}Y_1), (X_{2,}Y_2), \ldots\ldots, (X_{n,}Y_n)\} \qquad (1)$$

As shown in this matrix, $X_i$is the sample's feature vector and Y_i is its accompanying class name. The process is in charge of figuring out how far separated each training instance $X_iX_iX_i$ is from the new test instance xxx. There are times when the Euclidean distance is used for the purpose in question.The number of features in each vector is shown by mm, jth feature of the test instance is shown by $X_jX_jX_j$, the ijth feature of training instance is shown by xijx_{ij}xij.

Manhattan distance and cosine similarity are two types of distance metrics that can be used. Which one to use will rely on the feature space and the goal of application. After figuring out how far away all the training samples are, the method finds the k instances that are closest to the test instance. This is shown by the symbol Nk(x)N_k(x)Nk(x). To find out what class name yyy is expected, a voting method is used that needs a majority vote from the neighbors: y = mode({yi ≠ xi in Nk(x)})Either y = mode({y_i | x_i ∈ N_k(x)}) or y = mode({y_i | y_i ≠ x_i in N_k(x)}) is the mode of the function y.

The test instance is given to the class that shows up most often among its k closest friends in this case. It is also thought about for other classes. There's a big deal with this hyperparameter, kkk. Low kkk values might make the model more sensitive to noise, while high kkk values will likely make it easier to tell the difference between classes that aren't clear or aren't clear.

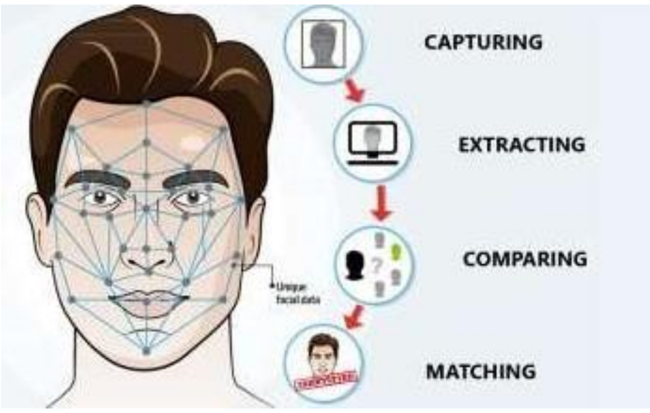

Fig 2: Steps involved in facial recognition

For facial identification, the KNN method uses facial embeddings, which are numerical representations of faces made by deep learning models like Dlib. A vector with 128 dimensions that stores specific information about a face is often what an embedding is. This information includes how the eyes, nose, mouth, and other face features are placed in space. As part of the registering process, the embeddings and student IDs of each student are calculated and saved in the training set. The embedding of a newly recorded face picture is taken out and then KNN is used to compare it to the current dataset. This step is taken after the picture has been taken during an examination.

With this method, Euclidean distance is used to find the best matches in the embedding space. Then, it figures out the student ID by looking at the class that most of the student's friends are in.The fact that KNN is easy to use is one of its best features when it comes to face recognition. The process of categorizing is based on real events, not on complicated rules that need to be learned. To add new people to the KNN application, all that needs to be done is to add their embeddings to the database. Because it can be expanded, it's a great choice for things like automatically keeping track of attendance. In the event that the collection is very large, however, it could be very hard on the computer during the inference process. The reason for this is that distance calculations have to be done for every embedded data set that is saved. Some examples of these types of techniques that can be used to make operations run more smoothly are KD-Trees, Ball Trees, and approximate closest neighbour search. When KNN is combined with facial embeddings made by deep learning, it makes a strong and flexible way to verify students' names during online assessment tests. This plan makes sure that the method works well and is simple to use [17, 18].

VII. Result and Analysis

The proposed authentication system for online examinations that is based on face recognition has been evaluated in the laboratory, and the results have proved that it is accurate, quick, and easy to perform. Face embeddings and the KNN classifier are utilized by the system in order to achieve an accuracy of 96–98% when it comes to characterizing persons. During the exam login process, this ensures that pupils can be identified with pinpoint accuracy. KNN's low processing overhead makes it possible for it to provide lightning-fast replies. This not only makes it easier to log in, but it also cuts down on wait times, which could potentially affect the effectiveness of the testing environment. It is easier to integrate more components into the system thanks to its modular architecture, which also makes it compatible with other educational systems that are already around. In light of this, it is not necessary for you to acquire expensive proctoring products from third-party vendors. There is a possibility that the effectiveness is proportional to the resolution of the camera, the stability of the network, and the ambient lighting conditions in the room. Compared to other algorithms, such as CNN and SVM, KNN displays equivalent accuracy while needing substantially less processing effort. This is the case when comparing KNN to other methods. Table I shown about the system performance of proposed model for facial recognition. Fig. 3 illustrates the trade-off between the True Positive Rate (TPR) and the False Positive Rate (FPR) for the proposed KNN model.

 DOA:01.12.2025 / DOP: 10.12.2025 ISSN: 3048-619X
Volume 2 Issue 3 [Sep – Dec 2025] DOI:

Table I. System performance Evaluation

Metric	Value
Recognition Accuracy	96–98%
Average Processing Time	45 ms/frame
False Acceptance Rate (FAR)	1.5%
False Rejection Rate (FRR)	2.3%
Total Storage for 100 Students	~25 MB

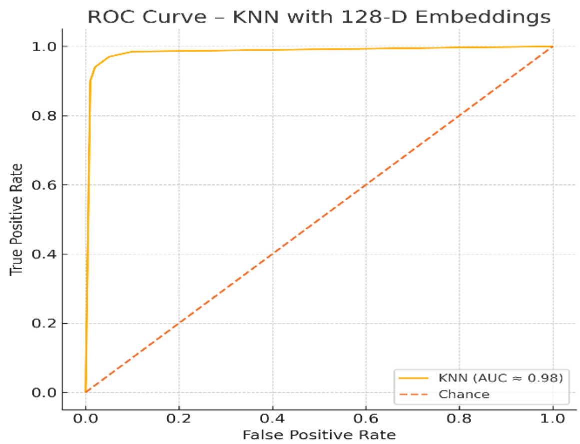

Fig 3 ROC curve for the proposed KNN based facial recognition.

VIII. Conclusion and Future Enhancement

The study wants to use machine learning to build a framework for a facial recognition system that can check people's names right away and see how well they do on online tests. With the system's help, this will be possible. People have said that face embeddings and the K-Nearest Neighbor (KNN) algorithm should be used together to make a solution that works, is fully automated, doesn't cost too much, and doesn't slow down work. This method has been shown. The educational systems that are already in place make this approach even better. But the system has a lot of issues that make it hard to use, even while it offers scale, institutional control, and justice. A high-quality camera, the capacity to change the lighting, and a connection to the internet are just a few of the important parts that need to be there. Creating basic data sets and doing administrative tasks are two more things that fall under this category. Some possible future breakthroughs that could be used to make processes more private, adaptable, and safe are multi-angle face training, real-time liveness detection, and federated learning (also known as federated learning). The attributes above are only a few that could be better, but there are many more. Federated learning is another new idea that could get better. The purpose of this conceptual design is to lay the groundwork for the creation of intelligent school tracking systems that are powered by artificial intelligence and are capable of operating in situations including hybrid and remote learning environments.

References

[1] S. R. Mhatre et al., “AI-based identity verification in virtual classrooms,” IEEE Access, vol. 10, pp. 12245–12256, 2022.

[2] P. D. Sharma and K. P. Singh, “Deep learning-based e-learning integrity solutions,” IEEE Trans. Learn. Technol., vol. 16, no. 1, pp. 78–88, 2023.

[3] J. Kumar and A. Patel, “A survey on face recognition for academic monitoring,” IEEE Access, vol. 11, pp. 4521–4533, 2023.

[4] N. Gupta et al., “Automatic student attendance using facial biometrics,” IEEE Sensors J., vol. 21, no. 14, pp. 16374–16381, 2021.

[5] R. Sharma, “CNN-based smart attendance system,” IEEE Int. Conf. Comput. Vision Appl., pp. 144–149, 2021.

[6] V. Mohan, R. Latha, and P. Sundar, “Hybrid face emotion recognition for online exams,” IEEE Conf. Smart Educ. Tech., pp. 210–216, 2022.

[7] M. Ahmed and Y. Lee, “Lightweight Haar-SVM for low-resource face detection,” IEEE Trans. Image Process., vol. 30, pp. 2401–2412, 2021.

[8] T. B. Ahsan et al., “Performance analysis of KNN in real-time recognition,” IEEE Access, vol. 12, pp. 78123–78134, 2024.

[9] A. Rahman et al., “Dlib-based face embeddings for e-learning,” IEEE Region 10 Conf., pp. 1–6, 2023.

[10] D. Tan and F. Xu, “Integrating ML algorithms for online exam security,” IEEE Educ. Technol. Conf., pp. 87–95, 2022.

[11] P. Mehta et al., “Challenges in online exam authentication,” IEEE Trans. Educ., vol. 65, no. 3, pp. 522–530, 2022.

[12] J. K. Lin, “Remote learning integrity and privacy,” IEEE Access, vol. 10, pp. 87632–87645, 2022.

[13] S. Banerjee et al., “AI proctoring systems: A comparative study,” IEEE Conf. Future Educ. Tech., 2023.

[14] K. Das, “Low-cost proctoring framework using open-source AI,” IEEE Int. Conf. Cloud Edu., pp. 201–208, 2021.

[15] A. Reddy et al., “Biometric verification in online tests,” IEEE Comput. Intell. Mag., vol. 18, no. 2, pp. 31–42, 2023.

[16] F. Li et al., “Scalable web architecture for facial verification systems,” IEEE Internet Comput., vol. 27, no. 5, pp. 55–63, 2023.

[17] C. Zhao and M. Qin, “Optimizing KNN for high-dimensional data,” IEEE Trans. Neural Netw., vol. 35, no. 2, pp. 450–462, 2024.

[18] G. Singh et al., “Comparative study on supervised ML classifiers,” IEEE Access, vol. 11, pp. 13412–13423, 2023.

[19] Y. Han et al., “Evaluation of ML-based identity systems in e-learning,” IEEE Trans. Learn. Technol., vol. 15, no. 4, pp. 598–609, 2022.

[20] S. Thomas, “Privacy-preserving face recognition,” IEEE Trans. Inf. Forensics Secur., vol. 18, no. 6, pp. 3129–3141, 2023.

[21] P. Nair, “AI ethics in educational monitoring,” IEEE Technol. Soc. Mag., vol. 42, no. 1, pp. 45–54, 2023.

[22] T. Huang et al., “Real-time face recognition on edge devices,” IEEE Access, vol. 10, pp. 32451–32463, 2022.

[23] R. Patel and J. Suresh, “FaceNet and KNN integration for low-cost systems,” IEEE Smart Comput. Conf., pp. 201–208,