 

HAROHALLI, KANAKAPURA ROAD – 562112

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

(DATA SCIENCE)

**MATLAB PROJECT REPORT**

ON

#### 

#### "Automatic Cheating Detection in Exam Hall"

2024-2025

BACHELOR OF TECHNOLOGY IN

COMPUTER SCIENCE & ENGINEERING (DATA SCIENCE)

**Submitted by**

Shreya Praveen – ENG23DS0037

Siri.A.K-ENG23DS0038

Thungashree.I.L-ENG23DS0041

**Under The Supervision of:**

#### Prof. Shivamma D.

#### Assistant Professor

#### Department of CSE (Data Science), DSU

DAYANANDA SAGAR UNIVERSITY

#### DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

**(DATA SCIENCE)**

#### SCHOOL OF ENGINEERING

**HAROHALLI, KANAKAPURA ROAD – 562112**



CERTIFICATE

#### It is certified that the mini project work entitled “Automatic Cheating Detection in Exam Hall” has been carried out at *Dayananda Sagar University*, Bangalore, by *Shreya Praveen – ENG23DS0037 , Siri.A.K – ENG23DS0038 , Thungashree.I.L – ENG23DS0041*, Bonafide student of fourth Semester, B.Tech in partial fulfilment for the award of degree in *Bachelor of Technology in Computer Science & Engineering (Data Science)* during academic year *2024-25*. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report deposited in departmental library.

#### The project report has been approved as it satisfies the academic requirements in respect of project work for the said degree.

**Signature of the Guide Signature of the Chairperson**

**ACKNOWLEDGEMENT**

A project's successful completion offers a sense of satisfaction, but it is never finished without expressing gratitude to everyone who contributed to its accomplishment. We would like to convey our sincere gratitude to our esteemed university, Dayananda Sagar University, for offering the first-rate facilities.

We are especially thankful to our Chairperson, Dr. Shaila S G, for providing necessary departmental facilities, moral support and encouragement. The largest measure of our acknowledgment is reserved for Prof. Shivamma D. whose guidance and support made it possible to complete the project work in a timely manner.

We would want to thank everyone who has assisted us in successfully completing this project work, both directly and indirectly. The staff has provided us with a great deal of direction and cooperation.

Shreya Praveen –ENG23DS0037

Siri.A.K – ENG23DS0038

Thungashree.I.L – ENG23DS0041

**DECLARATION**

We hereby declare that the project entitled **"Automatic Cheating Detection in Exam Hall"** submitted to Dayananda Sagar University, Bengaluru, is a bona fide record of the work carried out by me under the guidance of Prof. Shivamma D., Assistant Professor in the Dayananda Sagar University School of Engineering's Department of Computer Science and Engineering (Data Science). This work is submitted toward the partial fulfillment of the requirements for the award of a Bachelor of Technology in Computer Science and Engineering (Data Science).

Shreya Praveen-ENG23DS0037

Siri.A.K-ENG23DS0038

Thungashree.I.L-ENG23DS0041

**ABSTRACT**

The project presents the development and implementation of an automated cheating detection system designed to enhance security and integrity in examination halls. Leveraging advanced technologies such as computer vision, machine learning, and sensor- based monitoring, the system aims to detect and deter dishonest behaviors in real time. The solution integrates surveillance cameras, object recognition algorithms, and behavior analysis techniques to identify suspicious activities, such as unauthorized device usage, abnormal movement patterns, and communication between students. The system offers scalability, accuracy, and reduced dependence on manual invigilation. Experimental results and simulations demonstrate the system’s effectiveness in identifying various forms of cheating while maintaining a non-intrusive and efficient examination environment. This project contributes to the ongoing efforts to uphold fairness and trust in academic assessments through intelligent automation.

The proposed system provides a **cost-effective**, **privacy-friendly**, and **scalable** solution to maintaining examination integrity. It minimizes human error, reduces the burden on staff, and eliminates ethical concerns related to video surveillance. This approach is especially valuable in institutions with limited resources or in exams where visual monitoring is either not feasible or inappropriate.

**TABLE OF CONTENTS**

|  |  |
| --- | --- |
| **CHAPTER** | **PG. NO.** |
| 1. INTRODUCTION | 7-7 |
| 1. OBJECTIVES AND SCOPE OF WORK | 8-8 |
| 1. DESCRIPTION OF WORK | 9-10 |
| 1. METHODOLOGY | 11-12 |
| 1. SOURCE CODE | 13-20 |
| 1. RESULT | 21-23 |
| 1. CONCLUSION | 24-24 |
| 1. REFERENCES | 25-25 |

**INTRODUCTION**

Ensuring academic integrity during examinations is a critical challenge faced by educational institutions. Traditional methods of invigilation, which rely heavily on human supervision, are often limited in their ability to detect subtle or well-concealed forms of cheating. With the increasing use of technology in education, there is a growing need for intelligent, automated systems that can monitor examination environments more effectively and fairly.

The project focuses on the design and implementation of an automated cheating detection system using MATLAB . The system utilizes classical image processing and computer vision techniques available within the MATLAB environment. The approach emphasizes motion detection, face tracking, object recognition using feature-based methods and behavior analysis to identify suspicious activities during exams.

This introduces an alternative approach to automate cheating detection in examination halls **without relying on visual monitoring**. Instead, it focuses on non-visual technologies that provide a balance between efficiency, privacy, and affordability. By integrating hardware components such as **RF signal detectors**, **motion sensors**, **noise sensors**, and **RFID-based identity verification systems**, the proposed solution aims to detect common cheating behaviors like unauthorized device usage, physical movement patterns indicative of communication, and impersonation.

### 

### 

### OBJECTIVE AND SCOPE OF WORK

### The objective of this project is to develop an automated system for detecting cheating behaviors in an examination hall using MATLAB-based image processing and classical computer vision techniques. The primary goal is to identify and flag suspicious activities such as frequent head movements, handling of unauthorized objects, and possible communication between examinees. By automating the monitoring process, the system aims to minimize human error and reduce the dependence on continuous manual invigilation, thereby enhancing the overall fairness and integrity of the examination process. The solution is designed to be non-intrusive, ensuring that it does not interfere with the examinees’ experience, while also being scalable for deployment across different types of examination environments.

### The scope of the project includes the implementation of the system entirely in MATLAB, leveraging its robust image processing and classical computer vision capabilities. The system focuses on detecting cheating-related activities based on predefined visual patterns and behavioral cues, such as unusual movements, object passing between candidates, and the presence of multiple faces within a single video frame. Suspicious events detected by the system will be logged or flagged for review by human invigilators, enabling a more efficient and accurate evaluation process. Furthermore, the system is designed with adaptability in mind, allowing it to be customized for use in various educational institutions, including schools, colleges, and universities, regardless of the scale or layout of the exam hall.

### DESCRIPTION OF WORK

### This project involves the development of an automated system to detect potential cheating behavior during examinations using MATLAB-based video analysis and classical image processing techniques. The core functionality is implemented in the fodsmalpractice\_main function, which guides the user through selecting an exam surveillance video file and initiates a multi-step analysis pipeline. The system begins by reading frames from the selected video using MATLAB’s VideoReader and detects faces using the built-in vision.CascadeObjectDetector. It tracks the horizontal movement (X position) of the detected face to identify suspicious behaviors, such as large head movements or prolonged absence of the face, which may indicate cheating attempts like communication with others or leaving the frame to access unauthorized materials.

### The heart of the system lies in the analyze\_video function. This function uses MATLAB’s VideoReader to extract frames from the video and applies the vision.CascadeObjectDetector to detect faces within each frame. The system monitors the horizontal position (X coordinate) of the detected face across frames. If a sudden shift in position exceeding the defined movement threshold is detected, or if the face is missing from the frame for a specified duration, the system considers these as possible indicators of malpractice. Each detected event is logged with a timestamp, frame number, and reason ("Large Movement" or "Face Lost"). Simultaneously, the function maintains a detailed log of each frame's data, including the frame number, timestamp, face presence, and face X-position. This log serves as a comprehensive dataset for further analysis.

### During processing, each frame is analyzed to determine whether a face is present and to measure changes in position. If a face is not detected for a specified number of consecutive frames (lostFrameLimit), or if a sudden large movement is detected exceeding a threshold (moveThresh), the system flags this as a possible malpractice event. All events and frame-wise data are logged in a structured format for further analysis. The data is then cleaned using the clean\_log function to remove invalid or incomplete entries.

### To support post-exam evaluation, the system includes a rich set of visualization tools. The visualize\_data function provides exploratory data analysis (EDA) plots, such as face detection over time, position tracking, movement magnitudes, and histograms to help identify abnormal behaviors visually. Additionally, the pca\_analysis function performs a manual principal component analysis (PCA) on the movement and face detection data to reveal dominant behavioral patterns and summarize the variance in the data. Finally, all cleaned data and detected malpractice events are exported to an Excel report for human review. This approach provides a non-intrusive, efficient, and scalable solution to assist invigilators by automating the detection of suspicious behavior during exams using only video input and classical computer vision techniques.

### 

### 

### METHODOLOGY

### The methodology of this project is centered around using classical computer vision techniques in MATLAB to detect suspicious activities during an examination through video analysis. The process begins with video input selection, where the user is prompted to choose a recorded video file of the exam session. Once the video is loaded, the system processes it frame by frame using MATLAB’s VideoReader function. Each frame is analyzed with the help of a pre-trained face detection algorithm—vision.CascadeObjectDetector—to identify and track the presence and position of the candidate’s face throughout the duration of the exam.

### The primary behavior indicators used to detect potential malpractice are sudden or large movements of the face in the horizontal direction and the prolonged absence of a detectable face. These are considered as signs of potential cheating, such as turning to communicate with someone, attempting to access unauthorized materials, or leaving the camera view intentionally. The system continuously compares the current position of the face (X coordinate) with the previous frame and calculates the difference in position. If this difference exceeds a predefined threshold (moveThresh), or if the face is not detected for a specified number of consecutive frames (lostFrameLimit), the event is flagged as suspicious.

### All relevant data, including frame number, timestamp, face detection status, and X-position, is logged in a structured table. The log is then cleaned to remove missing or invalid entries using the clean\_log function. Next, the visualize\_data function generates various plots to provide a visual overview of face presence, movement trends, and anomalies over time. This allows for an intuitive understanding of the candidate’s behavior during the exam.

### To add a statistical dimension to the analysis, the system performs Principal

### Component Analysis (PCA) on the movement and face detection data. The PCA is implemented manually through normalization, covariance matrix calculation, and eigen decomposition. The reduced data is visualized in a 2D scatter plot, helping to identify unusual patterns or deviations from normal behaviour. Finally, all the cleaned log data and detected malpractice events are exported to an Excel report, allowing examiners or invigilators to conduct further reviews.

### Through this structured and non-intrusive methodology, the system provides an efficient and scalable solution to automate cheating detection, reduce manual workload, and uphold academic integrity using only video input and classical image processing techniques.

### 

### SOURCE CODE

### function fodsmalpractice\_main()

### % === MAIN SCRIPT ===

### [vidFile, vidPath] = uigetfile({'\*.mp4;\*.avi;\*.mov'}, 'Select Exam Video File');

### if isequal(vidFile,0)

### disp('No video selected. Exiting.');

### return;

### end

### videoFullPath = fullfile(vidPath, vidFile);

### moveThresh = 50;

### lostFrameLimit = 10;

### [malpracticeEvents, detectionLog] = analyze\_video(videoFullPath, moveThresh, lostFrameLimit);

### cleanedLog = clean\_log(detectionLog);

### visualize\_data(cleanedLog);

### pca\_analysis(cleanedLog);

### reportFile = fullfile(pwd, 'fodsmalpractice\_report.xlsx');

### writetable(cleanedLog, reportFile);

### disp(['📄 Report saved to: ' reportFile]);

### if ~isempty(malpracticeEvents)

### disp('⚠ Malpractice suspected! Events detected:');

### for i = 1:length(malpracticeEvents)

### fprintf(' - Frame %d at %.2f s: %s\n', ...

### malpracticeEvents(i).FrameNum,malpracticeEvents(i).TimeSec,

### malpracticeEvents(i).Reason);

### end

### else

### disp('✅ No malpractice detected.');

### end

### end

### %% === LOCAL FUNCTIONS ===

### function [events, log] = analyze\_video(videoFile, moveThresh, lostFrameLimit)

### vid = VideoReader(videoFile);

### faceDetector = vision.CascadeObjectDetector();

### faceLostFrames = 0;

### previousX = [];

### events = struct('FrameNum', {}, 'TimeSec', {}, 'Reason', {});

### log = table([], [], [], [], 'VariableNames', {'FrameNum','TimeSec','X','FaceFound'});

### fig = figure('Name','Exam Monitoring','NumberTitle','off');

### frameNum = 0;

### cheatingTriggered = false;

### cheatingDisplayFrames = 0;

### cheatingDisplayLimit = 50;

### while hasFrame(vid)

### frame = readFrame(vid);

### frameNum = frameNum + 1;

### bboxes = step(faceDetector, frame);

### imshow(frame); hold on;

### if ~isempty(bboxes)

### currentX = bboxes(1,1);

### faceLostFrames = 0;

### if ~isempty(previousX)

### dx = abs(currentX - previousX);

### if dx > moveThresh && ~cheatingTriggered

### cheatingTriggered = true;

### events(end+1) = struct('FrameNum', frameNum, 'TimeSec', vid.CurrentTime, 'Reason', 'Large Movement');

### cheatingDisplayFrames = cheatingDisplayLimit;

### end

### end

### previousX = currentX;

### rectangle('Position', bboxes(1,:), 'EdgeColor', 'g', 'LineWidth', 2);

### title('Face Detected');

### faceFound = true;

### else

### faceLostFrames = faceLostFrames + 1;

### faceFound = false;

### if faceLostFrames >= lostFrameLimit && ~cheatingTriggered

### cheatingTriggered = true;

### events(end+1) = struct('FrameNum', frameNum, 'TimeSec', vid.CurrentTime, 'Reason', 'Face Lost');

### cheatingDisplayFrames = cheatingDisplayLimit;

### end

### title(sprintf('No Face Detected: %d frames', faceLostFrames));

### end

### if cheatingTriggered && cheatingDisplayFrames > 0

### text(10, 30, 'CHEATING DETECTED', ...

### 'Color', 'red', 'FontSize', 20, 'FontWeight', 'bold', ...

### 'BackgroundColor', 'yellow');

### cheatingDisplayFrames = cheatingDisplayFrames - 1;

### end

### hold off;

### log = [log; {frameNum, vid.CurrentTime, double(previousX), faceFound}];

### pause(0.01);

### if ~isvalid(fig)

### break;

### end

### end

### close(fig);

### end

### function cleanedLog = clean\_log(log)

### maskValidX = ~isnan(log.X) & ~cellfun(@isempty, num2cell(log.X));

### cleanedLog = log(maskValidX, :);

### cleanedLog.X = double(cleanedLog.X(:));

### cleanedLog.TimeSec = double(cleanedLog.TimeSec(:));

### cleanedLog.FrameNum = double(cleanedLog.FrameNum(:));

### cleanedLog.FaceFound = logical(cleanedLog.FaceFound);

### end

### function visualize\_data(T)

### T.X = double(T.X(:));

### T.TimeSec = double(T.TimeSec(:));

### T.FaceFound = logical(T.FaceFound);

### if height(T) < 2

### dx = zeros(height(T),1);

### else

### dx = [0; abs(diff(T.X))];

### end

### figure('Name','EDA','NumberTitle','off','Position',[100 100 1200 600]);

### subplot(2,2,1);

### plot(T.TimeSec, T.FaceFound, '-b', 'LineWidth', 1.5);

### xlabel('Time (s)'); ylabel('Face Found'); title('Face Detection Over Time'); grid on;

### subplot(2,2,2);

### plot(T.TimeSec, T.X, '-r'); xlabel('Time (s)'); ylabel('X Position'); title('X Position Over Time'); grid on;

### subplot(2,2,3);

### plot(T.TimeSec, dx, '-m'); xlabel('Time (s)'); ylabel('|ΔX|'); title('Face Movement Magnitude'); grid on;

### subplot(2,2,4);

### histogram(dx, 30); xlabel('|ΔX|'); ylabel('Frequency'); title('Movement Histogram'); grid on;

### end

### function pca\_analysis(T)

### T.X = double(T.X(:));

### if height(T) < 2

### dx = zeros(height(T),1);

### else

### dx = [0; abs(diff(T.X))];

### end

### Xnorm = (dx - mean(dx)) ./ std(dx);

### Ynorm = (double(T.FaceFound) - mean(T.FaceFound)) ./ std(double(T.FaceFound));

### M = [Xnorm, Ynorm];

### C = cov(M);

### [V, D] = eig(C);

### [eVals, idx] = sort(diag(D), 'descend');

### eVecs = V(:, idx);

### score = M \* eVecs;

### explained = 100 \* eVals / sum(eVals);

### fprintf('\nPCA Explained Variance (Manual):\n');

### disp(explained');

### figure('Name','Manual PCA','NumberTitle','off');

### scatter(score(:,1), score(:,2), 20, T.TimeSec, 'filled');

### colorbar;

### xlabel('PC1'); ylabel('PC2'); title('Manual PCA Projection'); grid on;

### end

### RESULT

### The developed system was successfully tested using recorded exam hall videos to evaluate its effectiveness in detecting potential cheating behavior. Upon execution, the system allowed users to select a video file, processed it frame by frame, and monitored the examinee's facial presence and horizontal head movement. The system accurately detected and logged frames where the examinee exhibited large head movements—exceeding the defined threshold—or instances where the face was missing from the video for a prolonged number of frames. These events were flagged as suspicious, and a summary of each detected malpractice incident, including the frame number, time stamp, and reason (e.g., "Large Movement" or "Face Lost"), was displayed in the MATLAB command window for immediate attention.

### 

### It analyzes a selected video for suspicious activities like large head movements or a lost face, displaying "CHEATING DETECTED" .

### 

### These four graphs explain the exam monitoring process: "Face Detection Over Time" shows when a face was visible; "X Position Over Time" tracks the horizontal movement of the detected face; "Face Movement Magnitude" highlights the intensity of these movements; and the "Movement Histogram" provides a distribution of all recorded movements. Together, they visually indicate periods of attention, significant shifts, and potential instances of malpractice identified.

### 

### This "Manual PCA Projection" graph, generated by the pca\_analysis function, visualizes the relationships between face movement and face detection status over time. Each point represents a moment in the video, with its position determined by how much the face moved and whether it was detected, and its color indicating the time it occurred. This helps to identify distinct behavioral patterns or clusters of activity throughout the exam.

### 

### 

### 

### CONCLUSION

### The system effectively demonstrates a foundational approach to automated exam proctoring. The code's core functionality, as seen in the above result successfully identifies faces and overlays a "CHEATING DETECTED" warning when suspicious activities like large movements or a lost face occur. This real-time feedback is crucial for immediate intervention. The "EDA" graph further validates the system's ability to track and quantify these behaviors, showing clear patterns of face detection, X-position changes, and movement magnitudes over time. Finally, the "Manual PCA" graph illustrates the system's analytical capability, transforming the raw behavioral data into a condensed, multi-dimensional representation that can potentially reveal subtle patterns and anomalies indicative of malpractice.

### In conclusion, the system provides a robust framework for monitoring exams, combining visual detection, detailed logging, and statistical analysis to identify and report potential instances of academic dishonesty. While this is a basic implementation, it lays the groundwork for more sophisticated proctoring solutions by demonstrating the feasibility of using computer vision and data analytics for behavioral anomaly detection in educational assessments.

### REFERENCES

[1] Bilen, E., & Matros, A. (2021). “Online Cheating Amid COVID-19”. Journal of Economic Behavior & Organization, 182, 196–211.

[2]Sharma, R., & Bhattacharya, A. (2021). Deep learning approaches for cheating detection in remote examinations. International Journal of Computer Applications, 183(5), 15–21.

[3]He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 770–778.

[4]Vaswani, A., Shazeer, N., Parmar, N., et al. (2017). Attention is all you need. Advances in Neural Information Processing Systems (NeurIPS), 5998–6008.