JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

Online Attendance System Using Face Recognition and Liveness Detection in Android Devices

[1]Mohamed Elsherif

[1]Faculty of Graduate Studies For statistical research Cairo University,
Giza, Egypt

Abstract : In the past, tracking attendance in educational settings relied on traditional methods, such as paper records or verbal confirmation from students. The emergence of biometric attendance systems has provided a more efficient and accurate alternative. With the advent of the COVID-19 pandemic, the need for a shift to distance learning arose, posing new challenges for verifying student attendance. Traditional methods of ensuring attendance in virtual classrooms are no longer valid, increasing the demand for innovative solutions.Facial recognition technology has emerged as a promising method, leveraging recent advancements to meet the evolving needs of distance education. [1]

Despite significant progress in facial recognition systems over the past decade, their effectiveness still depends on ideal conditions, such as proper facial alignment and image clarity. Facial liveness detection has also become crucial to prevent fraudulent methods, such as impersonation or video playback. The complexities of facial recognition algorithms require large datasets and sophisticated features to reliably identify individuals amidst various environmental factors.

This research proposes a solution for facial liveness verification using pre-trained models, including dlib, MediaPipe, the deep learning library CNN, TensorFlow, and the OpenCV computer vision libraries. Facial features are detected through a convolutional neural network (CNN) trained for smile detection using TensorFlow. The integration of MediaPipe and the 68-landmark model from dlib demonstrates high accuracy in facial landmarking, with the classification model achieving over 93% precision and recall. Furthermore, the system is designed to record students' attendance at regular intervals during educational sessions, ensuring accurate attendance monitoring. By conducting at least four attendance checks per session, the system offers a comprehensive approach to verifying student attendance, addressing the challenges posed by distance learning environments.

i. INTRODUCTION

In an age of rapid technological advancement, the constant use of traditional attendance methods has become obsolete. The emergence of facial recognition technology represents a major shift in how attendance is managed, providing a more secure, efficient and convenient alternative for employers, employees and students alike. With its unparalleled ability to accurately identify individuals based on unique facial features and its ability to verify facial liveliness, this advanced solution simplifies the attendance check-in process, reducing it to a few seconds through facial capture and verification. [1]

Before delving into the complexities of facial recognition technology and its applications, it is necessary to understand the shortcomings of traditional attendance systems. In traditional educational settings, attendance tracking relies heavily on manual methods, such as paper records or verbal confirmation from teachers. Students were often required to physically sign attendance sheets or verbally confirm their attendance, a tedious and time-consuming process prone to inaccuracy, fraud and deception. [1]

The emergence of the COVID-19 pandemic and subsequent shift to distance learning has exacerbated the limitations of traditional attendance methods, rendering them ineffective in the face of new challenges. As educational institutions grappled with the complexities of verifying student attendance in virtual classrooms, the shortcomings of manual methods became glaringly apparent.

Facial recognition systems have emerged as a practical solution to the attendance verification dilemma, capitalizing on the significant progress made over the past decade. These systems provide unparalleled efficiency, accuracy and convenience in recording student attendance through facial recognition, circumventing the drawbacks of traditional methods. However, their effectiveness is compromised by sophisticated fraudulent methods, such as using a person's photo, video or 3D models, and relies

on ideal conditions, including image clarity and contrast. The complexity of facial recognition algorithms requires large-scale datasets and advanced features to reliably identify individuals amid various environmental factors.

In response to these challenges, this paper proposes the implementation of a liveness-based face recognition and presence system to meet the needs of distance learning environments. The system ensures robust attendance monitoring by using a pre-trained model to check facial liveliness and then recording student attendance at regular intervals during tutoring sessions.

With minimal attendance verification requirements for each session, this comprehensive approach to student attendance verification mitigates the challenges posed by distance learning, providing a seamless and efficient solution to attendance verification in virtual classrooms.

1.1 Problem

Traditional methods of attendance tracking, such as paper-based records and verbal confirmation, are becoming increasingly obsolete in the context of remote learning environments. The COVID-19 pandemic has accelerated the need for innovative solutions to verify student attendance in virtual classrooms. With the limitations of manual methods and the challenges of transitioning to distance education, there is a pressing need for effective and reliable attendance-tracking systems that can adapt to the demands of remote learning.

1.2 Aim

The aim of this research is to design and develop a comprehensive system that utilizes facial recognition and facial liveness detection technology for remote attendance verification. This system seeks to enhance the efficiency of the educational process by reducing the time spent on attendance verification and mitigating the risks of fraudulent practices, such as tampering with attendance records or using photos or videos for attendance. By leveraging facial recognition technology, the research aims to provide a robust and reliable solution for verifying student attendance in virtual classrooms, ultimately contributing to the smooth operation of remote learning environments.

1.3 Related Work

In recent years, facial recognition technology has emerged as a prominent solution for recording student attendance, leveraging computer systems and sophisticated software to identify individuals by comparing their facial features with pre-existing data stored in a database. This technology has spurred a body of research aimed at refining and optimizing the process of student registration, with a focus on enhancing accuracy and efficiency.

A notable contribution to this field of research involves the exploration of various methodologies and techniques for improving student registration processes. One such approach involves the utilization of the Python programming language in conjunction with computer vision libraries like openCV. By leveraging the capabilities of openCV, researchers have developed robust algorithms capable of accurately detecting and recognizing facial features, thus facilitating the seamless integration of facial recognition technology into attendance systems. [2]

Furthermore, advancements in machine learning have led to the adoption of deep convolutional neural networks (CNN) for facial recognition tasks. These CNN-based approaches have demonstrated remarkable performance in identifying individuals from facial images, offering a high degree of accuracy and reliability in attendance verification processes. [3]

In this research, we aim to delve deeper into the methodologies and techniques employed in the realm of facial recognition-based attendance systems. Moreover, we will conduct a comprehensive analysis of the advantages and disadvantages associated with each approach, shedding light on their respective strengths and limitations in the context of attendance recording.

This expanded related work section provides a comprehensive overview of the methodologies and techniques employed in facial recognition-based attendance systems, setting the stage for the research objectives and discussions to follow.

ii. BACKGROUND

Traditional Attendance Methods

Attendance tracking has long been a critical component of educational administration, ensuring that students meet the required participation standards. Traditional methods such as roll-calls and manual sign-ins, though widely used, are inherently inefficient and prone to errors. Roll-calls consume valuable class time and can be manipulated easily, while manual sign-ins are susceptible to forgery and proxy attendance. The need for a more efficient, reliable, and tamper-proof attendance system has driven the exploration of technological solutions.

Evolution of Automated Attendance Systems

The quest for automated attendance systems began with the adoption of biometric technologies. Fingerprint recognition systems became popular due to their accuracy and ease of use. These systems, however, required physical interaction, which could be time-consuming and disruptive. Moreover, the need for students to queue up for fingerprint scanning did not entirely eliminate time wastage and could lead to congestion, especially in large classes.

Introduction of Face Recognition Technology

Face recognition technology emerged as a promising alternative, offering the advantage of contactless identification. Unlike fingerprint systems, face recognition can operate passively, identifying students without requiring any direct interaction. This non-intrusive nature makes face recognition particularly suitable for educational environments, where minimizing disruptions is crucial.

Face Recognition in Educational Settings

The implementation of face recognition systems in educational settings involves several technical components:

Object Detection and Localization:

The first step in face recognition systems is detecting and localizing faces within an image or video stream. Algorithms like YOLO (You Only Look Once) [4] have proven effective in real-time object detection, capable of identifying multiple objects and their locations within a frame.

Face Detection:

Once objects are localized, the next step is identifying which of these objects are faces. Haar Cascading [5] is a commonly used method for real-time face detection. It uses a cascade function trained with positive and negative images to detect faces accurately. This method is efficient and works well in various lighting conditions, making it a practical choice for classroom settings.

Face Recognition:

The core of an attendance system is the ability to recognize and match faces to student identities. Techniques like FaceNet [5] and VGGFace2 [6] have advanced this field significantly. FaceNet uses a deep convolutional neural network to map faces into a 128-dimensional embedding space, enabling high accuracy in face matching. VGGFace2 provides a large-scale dataset that includes variations in pose, age, and lighting, aiding in the development of robust face recognition models.

Review of Existing Systems

Several systems have been proposed to leverage these technologies for attendance tracking:

RFID and Face Recognition Integration:

Some systems combine face recognition with Radio Frequency Identification (RFID) to enhance security and accuracy. For example, Akbar et al. [7] used RFID tags to verify the presence of authorized students in conjunction with face recognition to track their attendance.

Iris Biometric Systems:

Iris recognition systems, such as the one developed by Okokpujie et al. [8], capture detailed images of the eye to verify student identities. This method offers high accuracy and security but requires students to position themselves precisely for the camera, which can be less convenient than face recognition.

Hybrid Approaches:

Systems that combine multiple feature extraction techniques, such as Discrete Wavelet Transform (DWT) and Discrete Cosine Transform (DCT)[9], followed by classifiers like Radial Basis Function (RBF), have shown promise in improving recognition accuracy. These hybrid approaches leverage the strengths of different methods to enhance overall system performance.

Challenges and Innovations

Despite the advancements, face recognition-based attendance systems face several challenges:

Accuracy and Robustness:

Ensuring high accuracy in diverse classroom environments is a significant challenge. Factors such as varying lighting conditions, different poses, and occlusions (e.g., students wearing glasses or masks) can affect recognition performance.

Liveliness Detection:
Preventing spoofing attacks, where a photograph or video is used to deceive the system, requires effective liveliness detection.

Algorithm	Description	**Key Equations**
YOLO (You Only Look Once)	Real-time object detection algorithm that divides images into a grid and predicts bounding boxes and class probabilities.	Bounding Box Prediction: $\hat{t}_x = \sigma(t_x) + c_x$ $\hat{t}_y = \sigma(t_y) + c_y$ $\hat{t}_w = p_w e^{t_w}$ $\hat{t}_h = p_h e^{t_h}$ Loss Function: $\text{Loss} = \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{\text{obj}} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right] + \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{\text{obj}} \left[(\sqrt{w_i} - \sqrt{\hat{w}_i})^2 + (\sqrt{h_i} - \sqrt{\hat{h}_i})^2 \right] + \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{\text{obj}} (C_i - \hat{C}_i)^2 + \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{\text{noobj}} (C_i - \hat{C}_i)^2 + \sum_{i=0}^{S^2} 1_i^{\text{obj}} \sum_{c \in \text{classes}} (p_i(c) - \hat{p}_i(c))^2$
Haar Cascading	Uses integral images and a cascade of classifiers to detect faces in real-time.	Integral Image: $I_{\text{sum}}(x, y) = \sum_{x' \le x, y' \le y} I(x', y')$ Feature Calculation: $f = \sum_{(x,y) \in R_1} I_{\text{sum}}(x, y) - \sum_{(x,y) \in R_2} I_{\text{sum}}(x, y)$
FaceNet	Uses deep convolutional neural networks to map faces into a Euclidean space for face similarity.	Triplet Loss Function: $L = \sum_{i=1}^{N} \left[\| f(x_i^a) - f(x_i^p) \|_2^2 - \| f(x_i^a) - f(x_i^n) \|_2^2 + \alpha \right]_+$
VGGFace2	A dataset and model used to train face recognition algorithms with variations in pose, age, etc.	CNN Architecture: $\text{conv} \rightarrow \text{ReLU} \rightarrow \text{pool} \rightarrow \ldots \rightarrow \text{FC} \rightarrow \text{softmax}$
Principle Component Analysis (PCA)	Reduces the dimensionality of face data while preserving variance.	PCA Equations: $\mathbf{X} = [x_1, x_2, \ldots, x_n]$ $\bar{\mathbf{x}} = \frac{1}{n} \sum_{i=1}^{n} x_i$ $\mathbf{A} = \mathbf{X} - \bar{\mathbf{x}}$ $\mathbf{C} = \frac{1}{n} \mathbf{A}\mathbf{A}^T$ $\mathbf{CV} = \mathbf{V\Lambda}$
Discrete Wavelet Transform (DWT) and Discrete cosine Transform (DCT)	Feature extraction methods used in face recognition.	DWT Equation: $W_{j,k} = \frac{1}{\sqrt{a_j}} \int_{-\infty}^{\infty} x(t) \psi \left(\frac{t - b_k}{a_j} \right) dt$ DCT Equation: X(k) = \sum_{n=0}^{N-1} x(n) \cos \left[\frac{\pi}{N} \left(n + \frac{1}{2} \right) k

iii. RELATED WORK

Several studies have explored the use of face recognition technologies to automate attendance systems in educational settings. These studies address various aspects of face detection, recognition, and the integration of these systems with existing educational infrastructures.

Myrna Dwi Rahmatya and Mochamad Fajar Wicaksono developed a web-based electronic attendance system using facial recognition technology, Python programming language, Django framework, and OpenCV library. The aim of this study is to improve the quality of online attendance recording for students by utilizing facial recognition to prevent any attempts to manipulate the registration process and ensure data accuracy. [2]

The system was implemented using the waterfall development methodology, allowing students to log in only during lecture times and restricting access outside these hours. However, the system faces some challenges, such as the lack of verification of a person's vitality in front of the camera, exposing it to potential risks of impersonation. [2]

Additionally, students are allowed to register attendance only once during the lecture, which may affect the accuracy and reliability of the data, as it does not guarantee the presence of the student throughout the lecture period. To enhance this system, additional techniques can be explored, such as verifying a person's vitality through motion recognition or implementing additional policies to ensure that students cannot manipulate the registration process, such as increasing the number of times students can register attendance during the lecture. [2]

A recent study proposed the FRLD algorithm, a sophisticated blend of various liveness detection functions. This algorithm consists of five distinct functions that leverage the features of MediaPipe Facemesh, dlib 68 landmarks, and TensorFlow convolutional neural networks (CNNs). Each function is designed to detect specific aspects of liveness in real-time.

1- Face Shake and Gestures Detection:

MediaPipe FaceMesh: This function utilizes MediaPipe FaceMesh, which relies on the Blazeface face detection model. The Blazeface model is adept at determining the direction of the face, which is crucial for detecting natural face movements such as shaking and gesturing. By analyzing these movements, the system can differentiate between a live person and a static image or video. [3]

2- Eye Blink Detection:

TensorFlow CNNs: Separate TensorFlow CNN models are trained to detect eye blinks. These models process video input and produce logical outputs based on specific liveness detection requirements. The accurate detection of eye blinks is a strong indicator of a live subject, as it is a natural and involuntary human action. [3]

3- Smile Detection:

TensorFlow CNNs: Similar to eye blink detection, smile detection is handled by distinct TensorFlow CNN models. These models analyze facial expressions to identify genuine smiles. This functionality adds another layer of liveness detection by ensuring that the facial expressions are natural and not pre-recorded. [3]

4- Mouth Opening Detection:

dlib 68 Landmarks: The function for detecting mouth opening calculates the gap between the upper and lower lips using the 68 facial landmarks provided by dlib. By monitoring the distance between these landmarks, the system can determine if the mouth is open, which is another indicator of a live presence. [3]

5- Comprehensive Liveness Detection:

FRLD Implementation: The FRLD algorithm integrates these five distinct detection functions to provide a robust liveness detection system. This comprehensive approach ensures that various aspects of human presence and behavior are monitored, making it difficult for fraudulent activities to bypass the system. [3]

Despite its innovative design, the FRLD algorithm faces some challenges. One of the primary weaknesses is its insufficient training. The current model has not been adequately trained on a large dataset, which limits its effectiveness and accuracy. To enhance the performance of the FRLD algorithm, it needs extensive training on a more extensive and diverse dataset. Additionally, exploring different types of deep learning models beyond convolutional neural networks could further improve the system's robustness and reliability. [3]

In this study proposes the development of an advanced anti-spoofing model composed of three major modules: face anti-spoofing detection, liveness detection, and criminal identification using a Convolutional Neural Network (CNN) classifier. The operation scheme of this model is designed to be straightforward and efficient, aiming to enhance security and accuracy in face recognition systems. [4]

Face Anti-Spoofing Detection:

This module is responsible for processing the initial input and detecting any spoofing attempts. It is designed to identify various forms of deception, such as photos, posters, masks, or smartphone screens displaying faces. When an input is received, the

module analyzes it to detect if it is a real face or a spoof. This step is crucial to prevent fraudulent activities and ensure that only genuine human faces proceed to the next stages. [4]

CNN Classifier:
Once the face anti-spoofing module confirms the presence of a face, the input is forwarded to the CNN classifier module.
This module employs a CNN classifier to further determine the authenticity of the face. It evaluates the input to confirm whether the face is real or fake based on intricate patterns and features. This dual-check mechanism significantly enhances the system's reliability. [4]

Liveness Detection:
The liveness detection module is designed to ensure the detected face exhibits natural, live characteristics.
It processes the input to detect signs of liveness, such as eye blinks and lip movements. These actions are typical indicators of a live human presence, which helps in distinguishing between a live person and a static image or video. [4]

Criminal Identification:
This module serves as the final step in the process, focusing on identifying known criminals.
If the input successfully passes through both the face anti-spoofing and liveness detection modules, it is then analyzed by the criminal identification module. This module compares the face against a database of criminal records to determine if the individual is a known criminal or not. [4]
Despite its promising design, the research exhibits certain weaknesses that need addressing:
Dataset Limitations:
Criminal Database: The accuracy of the criminal identification module depends on having an up-to-date and comprehensive database of criminal faces, which may not always be available or complete. [4]

Processing Time:
Complexity: The multi-step process, while thorough, may introduce delays, especially in real-time applications where speed is crucial. Optimizing the processing time for each module is essential for practical use. [4]

Generalization:
Adaptability: The model's ability to generalize across different environments, lighting conditions, and facial variations (e.g., expressions, aging) is critical. Limited generalization can reduce the model's robustness and reliability. [4]
The proposed method in this study centers on the development of an autonomous attendance system through the following steps:
Face Detection:
To detect facial features in the specified input stream, the study employs the "face_recognition" library in Python. This library uses a pre-trained model to locate faces within the image. [1]
The image of the person is loaded using the "load_image_file" method. The detected faces are then passed to the next step for alignment. This step ensures that only genuine faces are considered for further processing, filtering out any irrelevant parts of the image. [1]

Face Alignment and Feature Extraction:
The "face_encodings" method leverages the renowned "FaceNet" algorithm to find and extract facial features. FaceNet is known for its high accuracy in encoding facial features into a numerical format.
The extracted features provide the facial coordinates, which are stored as an array. This numerical representation allows for efficient comparison and matching in later stages. [1]

Real-Time Face Detection:
For real-time face detection, the study utilizes CV2 (OpenCV). The "cap.read()" command captures frames from a video stream, allowing the system to process live video input. [1]
The face recognition library facilitates the extraction and storage of facial features by converting the real-time image from "BGR" format to "RGB" format and then saving it in pairs. This conversion is necessary as most face recognition algorithms work better with the RGB color space. [1]

Face Recognition and Attendance Logging:
After feature extraction, the facial features are analyzed. If they match those in the dataset, the system records attendance in an Excel sheet with the date and time. [1]
This automated process ensures accurate and efficient logging of attendance, reducing manual errors and saving time.
Despite its innovative approach, the research exhibits certain weaknesses that need to be addressed:
Susceptibility to Spoofing:
If a student has a photo of their face on their phone, they can potentially record their attendance fraudulently.
This vulnerability undermines the integrity of the attendance system, as it allows for possible manipulation and spoofing. [1]

Limited Dataset:
The dataset used to train and test the system comprises a very small number of images.
This limitation results in relatively low model accuracy, with a maximum of 70%. A small dataset restricts the model's ability to generalize and perform well in diverse real-world scenarios. [1]
The paper proposes AttenFace, a real-time attendance system that uses face recognition technology to automatically mark student attendance by analyzing live camera feeds from the classroom. [5]

System Description:
AttenFace captures snapshots of the classroom at regular intervals and uses these images to identify students present in the classroom. The face recognition process runs every 10 minutes, marking students present if they appear in a sufficient number of snapshots. This allows students some flexibility to leave the class briefly without losing attendance. The system separates the face recognition server from the back-end attendance calculation server, enabling integration with existing attendance software such as Moodle. [5]

Key Components
Object Detection and Localization: Identifying objects in the classroom and distinguishing students from other objects.
Face Detection: Identifying faces within the detected student objects.
Face Recognition: Mapping detected faces to corresponding students using techniques like FaceNet and VGGFace2. [5]
Existing Systems:
The paper reviews several existing attendance systems that use different biometric methods:
RFID and Face Recognition: Combines face recognition with RFID to verify student identities.
Iris Recognition: Uses iris biometrics to mark. [5]
Eigenface and Fisherface Algorithms: Compares these two face recognition methods for real-time attendance systems. [5]

Proposed System Advantages:
AttenFace addresses several limitations of traditional and biometric attendance methods:
Continuous Monitoring: Ensures students remain in class for a significant portion of the time.
Automatic and Integrated: No manual intervention required, integrates with existing college systems.
Proxy Prevention: Reduces the possibility of proxy attendance through robust face recognition and continuous monitoring. [5]
Challenges and Future Work:
While the system offers significant advantages, challenges such as varying classroom lighting conditions and face pose variations affect the accuracy of face recognition. Future improvements in face recognition algorithms and liveliness detection are essential to enhance the system's robustness and adaptability. [5]

but there are several potential weaknesses and challenges associated with the proposed system:

Accuracy of Face Recognition: The face recognition algorithm's accuracy can be affected by various external factors such as classroom lighting, the angle of students' faces, and occlusions (e.g., students' faces being partially covered by hair, masks, or other objects). These factors can lead to incorrect identification or failure to recognize some students. [5]

Liveliness Detection: The paper acknowledges the importance of differentiating between live faces and photographs to prevent spoofing attacks. However, the implementation and effectiveness of liveliness detection in the proposed system are not detailed, leaving a potential vulnerability. [5]

Scalability and Computational Overhead: While the system captures snapshots every 10 minutes to reduce computational load, the overall scalability and performance under varying classroom sizes and numbers of simultaneous classes are not thoroughly evaluated. High computational requirements for processing images and running face recognition algorithms might become a bottleneck in large-scale deployments. [5]

Privacy Concerns: The continuous monitoring and recording of students' faces raise privacy issues. Ensuring compliance with data protection regulations and addressing students' and parents' privacy concerns are crucial for the system's acceptance and deployment. [5]

Integration with Existing Systems: Although the system is designed to integrate with existing educational platforms like Moodle, the actual integration process, potential compatibility issues, and the effort required for seamless integration are not extensively covered. [5]

Dependence on Camera Quality and Placement: The effectiveness of the system heavily relies on the quality and placement of cameras in the classroom. Suboptimal camera angles, resolutions, or placement can negatively impact face detection and recognition accuracy. [5]

Handling of Edge Cases: The system's ability to handle edge cases, such as students frequently moving around the classroom, changing seats, or interacting with each other, is not fully explored. These scenarios might pose challenges for consistent and accurate attendance marking. [5]

iv. AIM AND OBJECTIVES

7.1 Aim

This research aims to design and develop a comprehensive system utilizing facial recognition technology to remotely verify attendance. This system is intended to enhance the efficiency of the educational process by reducing the time spent on attendance verification and mitigating the risk of fraudulent practices, such as tampering with attendance records. By leveraging facial recognition technology, the research seeks to provide a robust and reliable solution for verifying student attendance in virtual classrooms, ultimately contributing to the seamless operation of remote learning environments.

7.1 Objectives

System Analysis:
Analyze previous remote attendance systems to identify their functions and determine their shortcomings.
Identify opportunities for development and challenges to be addressed in improving these systems.
System Design:
Develop a comprehensive understanding of the system after the analysis phase.
Design the system, including creating UML diagrams such as use case and activity flow diagrams.
System Development:
Write the code immediately after the design phase.
Transfer the designed diagrams to a program capable of achieving the system's intended purpose by developing it in an Android environment.
System Testing:
Conduct comprehensive testing of the system to ensure it achieves the desired objectives.
Determine whether the system meets its intended purpose during this stage.
System Deployment:
Deploy the system to benefit from its capabilities and features after completing the analysis, design, development, and testing phases.

v. PROPOSED APPROACH

In this research, we propose a method for liveness detection using the Genki4K dataset. The approach involves the following steps:

Face Detection:
MediaPipe BlazeFace is employed for detecting faces in the images, which are then extracted for further processing.

Landmark Detection:
Dlib's shape predictor is utilized to detect facial landmarks, which are crucial for subsequent steps like eye blink detection.
Eye Blink Detection:
Eye Aspect Ratio (EAR) is computed using the detected landmarks to identify blinks, serving as a key indicator of a live face.

Data Augmentation:
ImageDataGenerator is used to augment the training data, introducing variations to enhance model robustness.

Model Training:
A Convolutional Neural Network (CNN) is designed for binary classification (live vs. spoof). The model includes convolutional layers, batch normalization, max-pooling, and dropout layers to improve performance and prevent overfitting.

Training Strategy:
The model is trained with augmented data, incorporating early stopping, learning rate scheduling, and checkpointing to ensure optimal performance and prevent overfitting.

Evaluation:
The trained model is evaluated on the test set, with performance metrics such as accuracy, classification report, and confusion matrix being computed to assess its effectiveness in liveness detection.
By integrating face detection, landmark detection, and eye blink analysis, this approach aims to accurately distinguish between live and spoofed faces, thereby enhancing the security and reliability of face recognition systems.

vi. EXPERIMENTAL WORK

3.1 Introduction

In the development process of this system, the Agile methodology was employed. Recognized as one of the most effective and prevalent approaches in software development, Agile is characterized by its high flexibility, allowing development teams to adapt to continuous changes in project requirements. This methodology breaks down the system development process into individual stages executed sequentially. This approach enables the team to test and evaluate each stage separately before proceeding, enhancing opportunities for early error detection and resolution.

Agile methodology also allows teams to effectively prioritize tasks and allocate time budgets for each stage, aiding in better project management and ensuring timely system delivery. Continuous interaction with clients or potential users ensures better fulfillment of needs and requirements, and allows for effective management of any changes in system requirements during the development process.

In summary, utilizing the Agile methodology in developing this system enhances system quality and ensures better fulfillment of user needs and requirements. It also facilitates efficient project management and effective handling of changes. A program's effectiveness heavily relies on its user-friendliness and ease of use. If a program is difficult to navigate or understand, it can lead to frustration and ultimately failure. Therefore, considering both functional and non-functional user requirements is crucial. This includes understanding how students and lecturers interact with the system and identifying necessary system requirements.

Developers play a crucial role in creating a user-friendly program. They must first create the system according to the specified requirements and then model the functional aspects. This involves identifying all use cases that will help the developer better understand the system. Additionally, data requirement modeling ensures that all components of the system are properly structured. This chapter is essential in facilitating a better understanding of the program. By identifying and addressing user requirements, developers can create a program that is both functional and user-friendly, ultimately leading to user satisfaction and program success.

This chapter will focus on:

Introduction: An overview of the analysis and methodology, explaining their purpose.
User Requirements: Ensuring the software meets the needs of its intended users and is user-friendly. By gathering and documenting user requirements, developers can design software tailored to specific user needs, increasing satisfaction and adoption, and reducing the risk of costly rework later.
System Requirements: Ensuring the software is designed and developed to meet technical feasibility. Documenting system requirements helps tailor the software to the project's technical needs, reducing the risk of costly rework.
Functional Modeling: Creating visual representations of functional requirements to ensure the software meets user needs and is user-friendly. This helps developers understand software functionality and identify potential issues or areas for improvement.
Data Requirements Modeling: Ensuring the software is designed to handle its data and is scalable. Identifying and documenting data requirements help understand software functionality and identify potential issues or areas for improvement.
Conclusion: Highlighting the importance of documenting these requirements and involving end-users in the development process to ensure the final product is functional and user-friendly.

3.2 User Requirements

User Authentication and Access Control:

Login/Signup: Users should be able to create an account or log in using their institutional credentials.

Role-Based Access: Different access levels for students, lecturers, and administrators.
Face Recognition:

High Accuracy: The system should accurately recognize students' faces under various lighting conditions and angles.

Real-Time Processing: The face recognition process should be swift and efficient, providing immediate feedback.
Liveness Detection:

Anti-Spoofing Mechanisms: The system should be able to distinguish between a live person and a photograph, video, or mask to prevent fraud.

Real-Time Detection: Liveness detection should work in real-time, ensuring the person is physically present during attendance.
Attendance Management:

Automated Attendance Marking: Attendance should be automatically recorded when a student's face is recognized and liveness is confirmed.

Manual Override: Lecturers should have the ability to manually mark or correct attendance if necessary.
User Interface:

Intuitive Design: The app should have a user-friendly interface that is easy to navigate for both students and lecturers.

Feedback Mechanism: Provide real-time feedback to users if their face is not recognized or if liveness detection fails.
Notifications and Alerts:

Attendance Confirmation: The app should send notifications to students confirming their attendance has been marked.

Absence Alerts: Notify students and lecturers if a student's attendance is not recorded during a session.

Security and Privacy:

Data Encryption: All biometric data should be encrypted to protect user privacy.

Compliance with Regulations: Ensure the system complies with relevant data protection and privacy regulations (e.g., GDPR, FERPA).
Technical Support:

Help and Support: The app should provide in-app support resources, including FAQs, tutorials, and a support contact form.

Regular Updates: Ensure the app is regularly updated to fix bugs, improve performance, and add new features.
Integration and Compatibility:

Device Compatibility: The app should work across a wide range of Android devices with various specifications.

Offline Functionality:
These requirements aim to ensure the system is functional, secure, user-friendly, and efficient for its intended purpose of managing student attendance through face recognition and liveness detection on Android devices.

Use cases:
1. User Authentication and Management
1.1. Registration
Description: Users (students, lecturers, and administrators) can create a new account.
Steps:
User opens the app and selects the "Register" option.
User enters required details such as name, email, student ID, and password.
System verifies the information and creates a new account.
User receives a confirmation email to activate their account.
1.2. Login
Description: Users can log in to the app using their credentials.
Steps:
User opens the app and selects the "Login" option.
User enters their email and password.
System verifies the credentials.
User is granted access to the app's features based on their role.
1.3. Forgot Password
Description: Users can reset their password if they forget it.
Steps:
User selects the "Forgot Password" option.
User enters their registered email address.
System sends a password reset link to the email.
User follows the link to reset their password.
1.4. Change Password
Description: Users can change their password after logging in.
Steps:
User navigates to the "Account Settings" section.
User selects "Change Password."
User enters the current password and the new password.
System updates the password after verification.
1.5. Delete Account
Description: Users can delete their account permanently.
Steps:
User navigates to the "Account Settings" section.
User selects "Delete Account."
User confirms the action.
System permanently deletes the user's account and associated data.

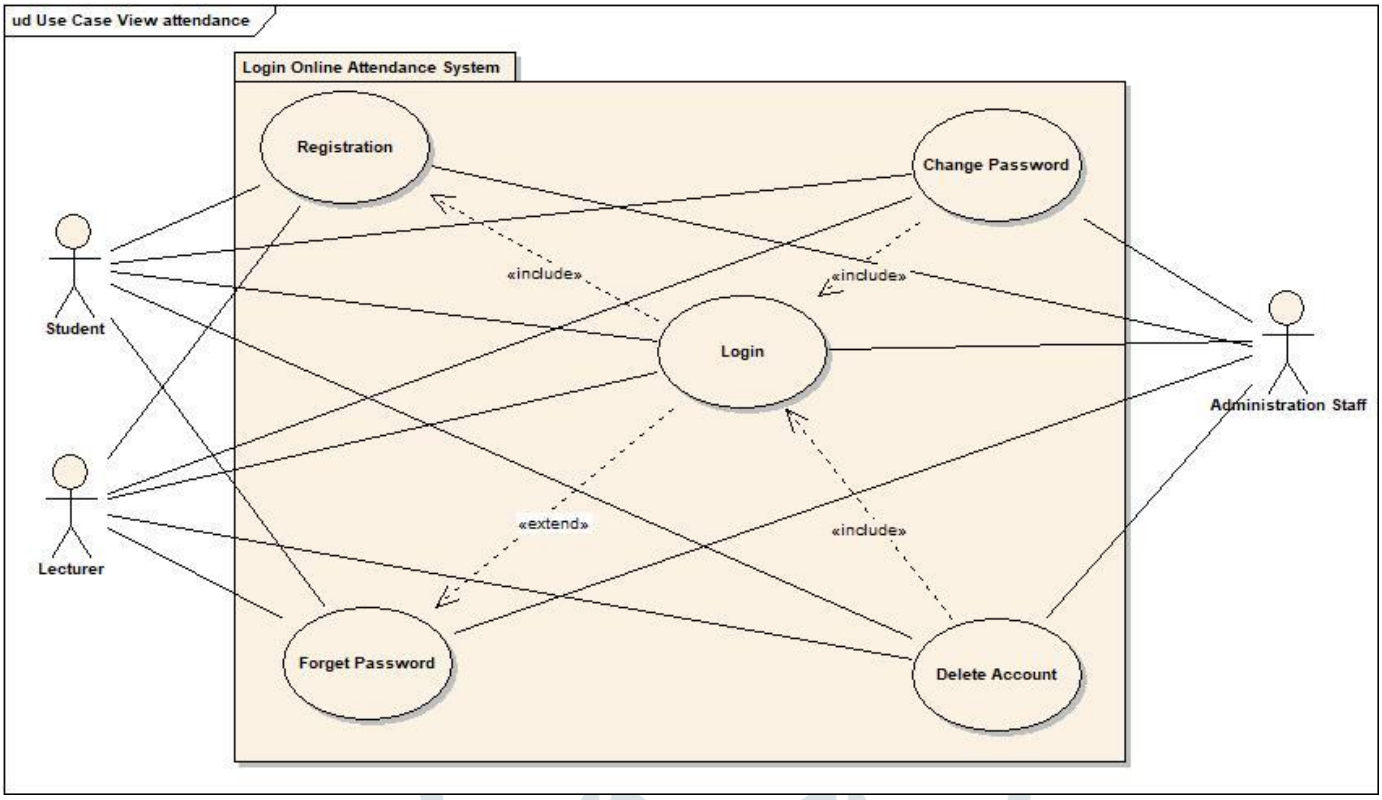


Figure 1: Login use case

2. Attendance Management System
2.1. Record Attendance
Description: The system automatically records student attendance using face recognition and liveness detection.
Steps:
Student opens the app during the scheduled lecture time.
Student's face is scanned and liveness is detected.
System verifies the student's identity and records attendance.
Student receives a notification confirming attendance.
2.2. View Attendance Data
Description: Users can view attendance records.
Steps:
User navigates to the "Attendance" section.
User selects the desired date range or specific lecture.
System displays attendance data, including presence, absence, and timestamps.
2.3. Track Login Success or Failure
Description: Administrators and lecturers can track the success or failure of student login attempts.
Steps:
User navigates to the "Reports" section.
User selects "Login Activity."
System displays a list of login attempts, indicating success or failure, along with timestamps.
2.4. Automatic Student Blocking
Description: The system automatically blocks a student based on predefined criteria (e.g., multiple failed login attempts, suspicious activity, or attendance fraud).
Steps:
System monitors student activity for predefined criteria.
If a criterion is met, such as multiple failed login attempts or suspected attendance fraud, the system automatically blocks the student's account.
System notifies the student and relevant authorities (e.g., lecturers, administrators) about the account block.
2.5. Unblock the Student
Description: Administrators can unblock a previously blocked student after reviewing the reason for blocking.
Steps:
Administrator navigates to the "Student Management" section.
Administrator reviews the reason for the block.
Administrator selects the student to be unblocked.

Administrator confirms the action.
System restores the student's access to the app.
2.6. Set the Lecture Schedule
Description: Lecturers and administrators can set and manage the lecture schedule.
Steps:
User navigates to the "Schedule Management" section.
User enters details for the lecture, including date, time, and subject.
User saves the schedule.
System updates the schedule and notifies students.

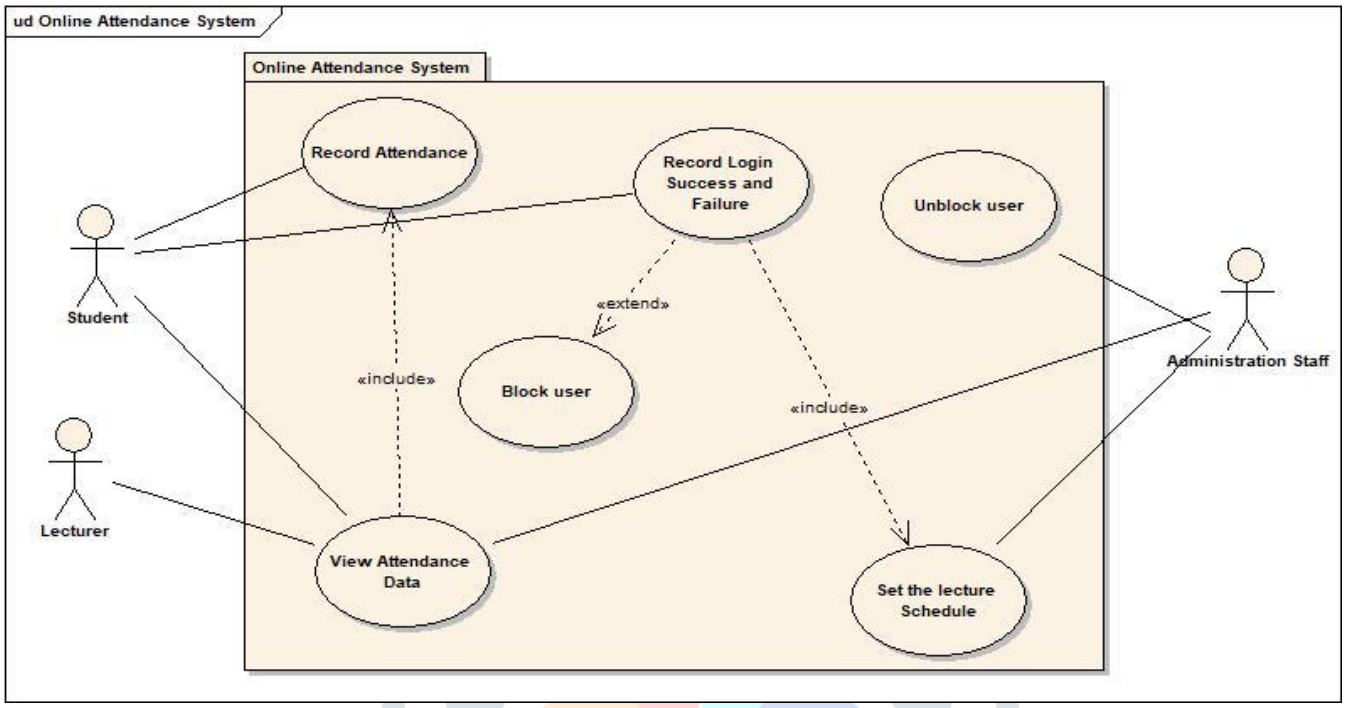


Figure 2: Online attendance use case

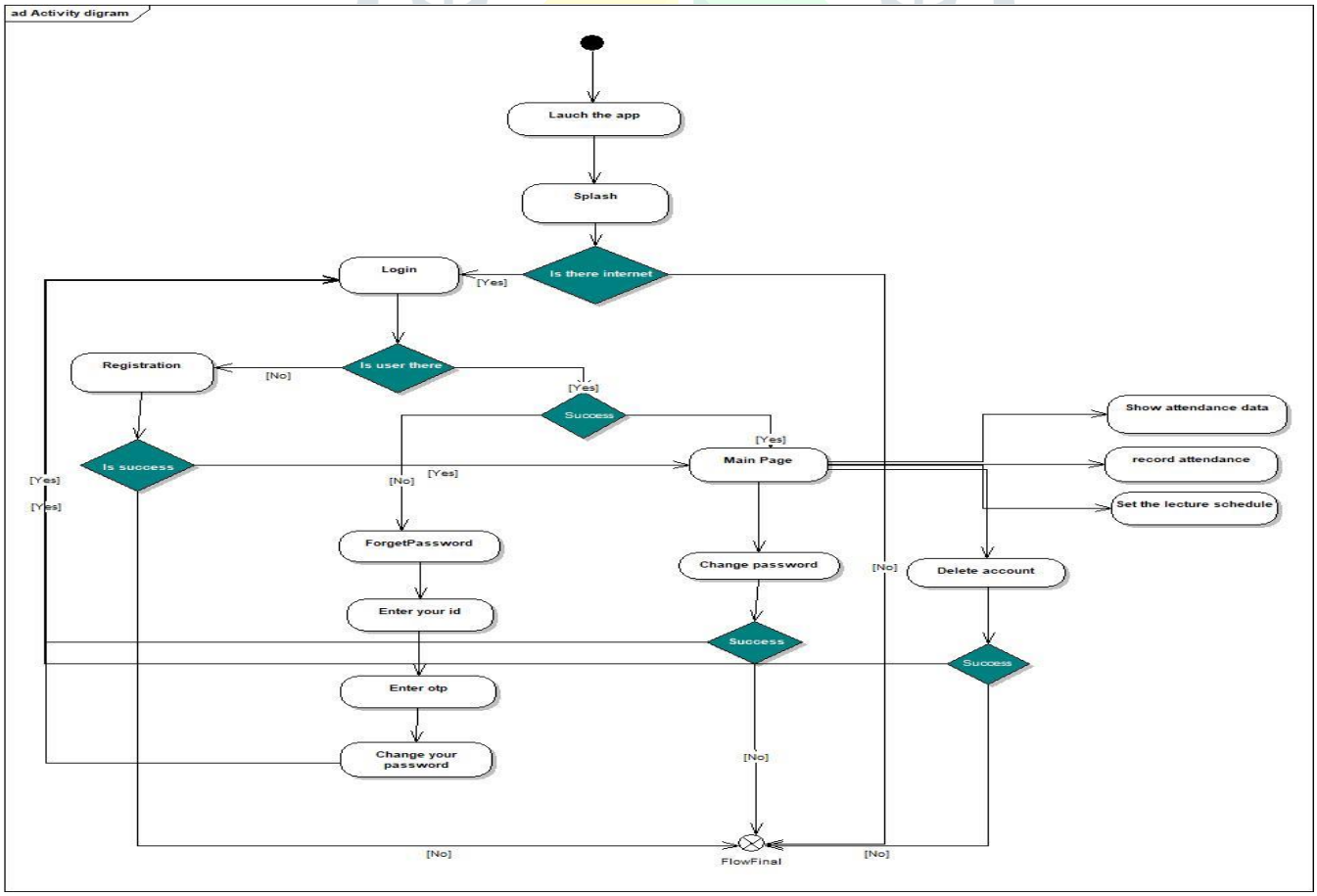
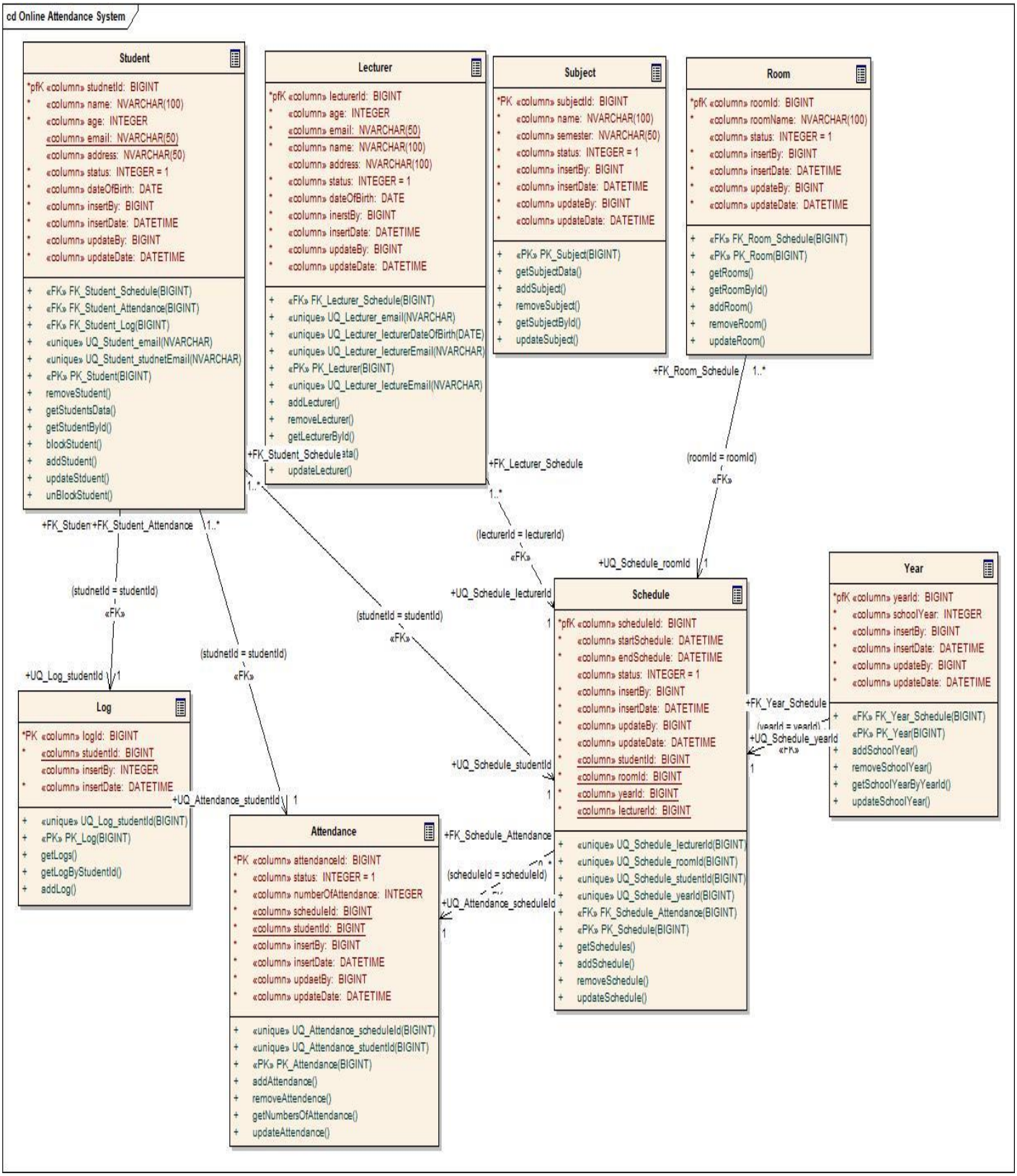


Figure 3:activity diagram



re 4: class diagram

Performance Measures:

Performance measures are crucial for evaluating the effectiveness of a machine-learning model. Here are some performance metrics that we will use in our model:

Accuracy:

- Accuracy is the ratio of correctly predicted instances to the total instances. It's a simple measure but can be misleading in imbalanced datasets.

$$\text{Accuracy} = \frac{\text{True Positives (TP)} + \text{True Negatives (TN)}}{\text{Total Instances}}$$

Precision:

- Precision is the ratio of correctly predicted positive observations to the total predicted positives. It indicates how many of the predicted positives are actually positive.

$$\text{Precision} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Positives (FP)}}$$

Recall (Sensitivity or True Positive Rate):

- Recall is the ratio of correctly predicted positive observations to all actual positives. It shows how many of the actual positives our model captures.

$$\text{Recall} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Negatives (FN)}}$$

F1 Score:

- The F1 Score is the harmonic mean of Precision and Recall. It provides a single metric that balances both precision and recall.

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Confusion Matrix:

- A confusion matrix is a table used to describe the performance of a classification model on a set of test data for which the true values are known. It shows the number of true positives, true negatives, false positives, and false negatives.

vii. EXPERIMENTS

Dataset:

The experiments were conducted using the GENKI-4K dataset, which includes images categorized into smile and non-smile classes. The training set consists of 4,000 images Preprocessing steps involved resizing images to 128x128 pixels and applying data augmentation techniques such as rotation, width and height shifts, and horizontal flipping.

Experimental setup:

Environment:

- The experiments were conducted using Python with TensorFlow and Keras on an NVIDIA GPU in colab environment.
- Additional libraries used include MediaPipe for face detection and dlib for landmark detection.

Model Architecture:

- A Convolutional Neural Network (CNN) was designed for binary classification of live vs. spoof faces. The architecture includes:
  - Convolutional layers with ReLU activation
  - Batch normalization layers
  - Max-pooling layers
  - Dropout layers to prevent overfitting
  - A final dense layer with a sigmoid activation for binary classification

Training Procedure:

- The model was trained using the Adam optimizer with a learning rate of 0.001, a batch size of 32, and for 40 epochs.
- Data augmentation techniques were applied using ImageDataGenerator, which included rotation, width and height shifts, shear, zoom, and horizontal flipping.
- Early stopping, learning rate scheduling, and checkpointing were implemented to ensure optimal performance and prevent overfitting.

Metrics:

The performance of the model was evaluated using accuracy, precision, recall, and F1 score. These metrics provide a comprehensive assessment of the model's ability to distinguish between smile and non smile faces.

Results:

	precision	recall	f1-score	support
Non-Smile	0.91	0.96	0.93	322
Smile	0.95	0.90	0.92	302
accuracy			0.93	624
macro avg	0.93	0.93	0.93	624
weighted avg	0.93	0.93	0.93	624

Figure 5: New result

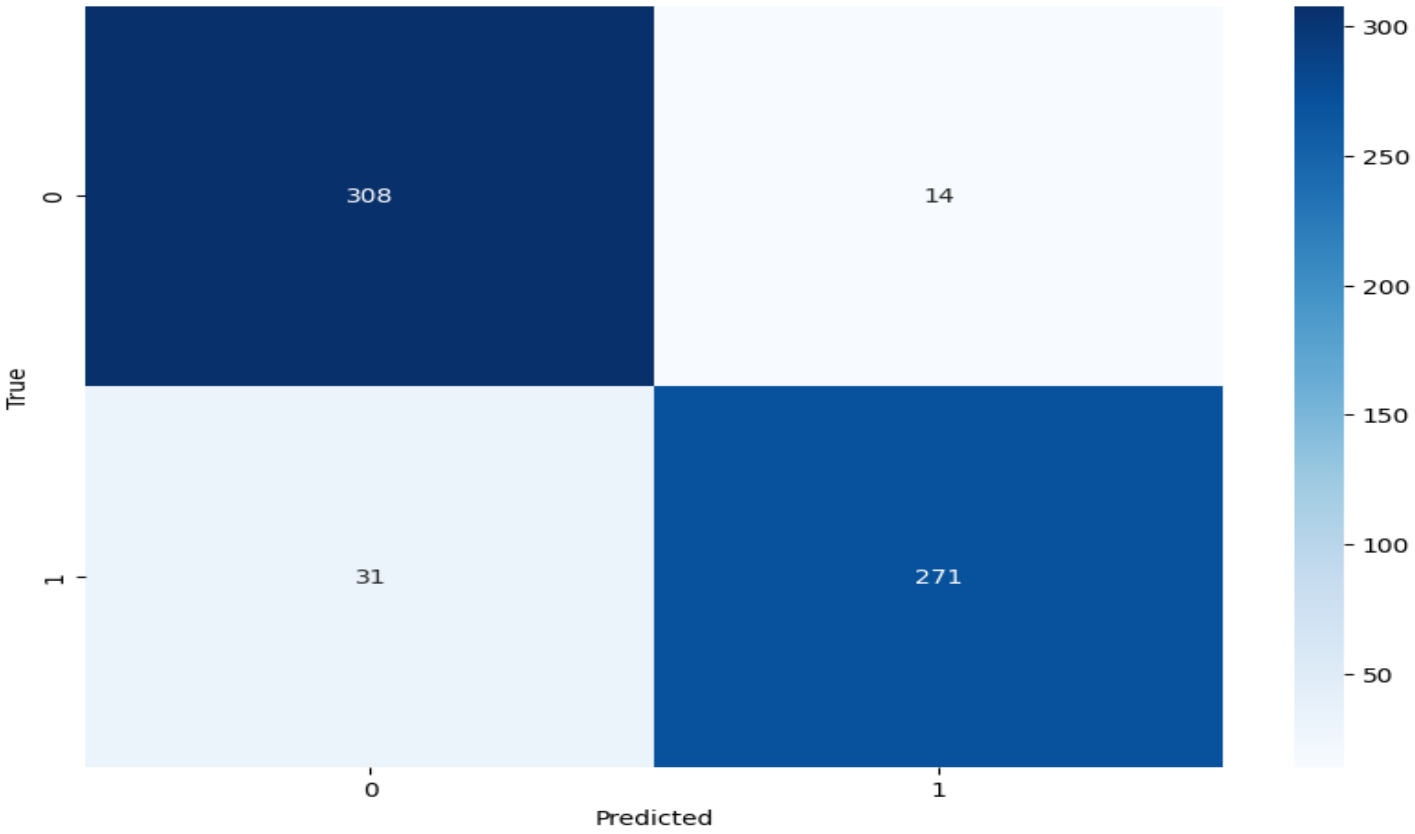


Figure 6: Confusion Matrix

Discussion:

The experimental results from this study demonstrate that the proposed liveness detection method, utilizing the Genki4K dataset, achieves robust performance with a high accuracy of 93%. The precision, recall, and F1 scores indicate that the model effectively distinguishes between smiling and non-smiling faces, with a high degree of reliability in both categories.

Key Findings:

Effective Face Detection:

MediaPipe BlazeFace proved efficient for face detection, offering high accuracy and speed.

Dlib's shape predictor successfully identified facial landmarks, which were crucial for subsequent steps.

Eye Blink Detection:

Incorporating eye blink detection significantly improved the model's ability to distinguish live faces from spoofed ones.

The use of Eye Aspect Ratio (EAR) calculations provided a reliable metric for detecting blinks.

Model Performance:

The CNN model, enhanced with data augmentation, batch normalization, and dropout layers, showed strong performance in the binary classification task.

Early stopping, learning rate scheduling, and checkpointing were essential in preventing overfitting and ensuring optimal training

Strengths:

The integration of face detection, landmark detection, and eye blink analysis provided a comprehensive approach to liveness detection.

Data augmentation techniques improved the model's robustness, making it more adaptable to varied input conditions.

Comparison with Previous Works:

Compared to previous works, which achieved an accuracy of 91%, the proposed method demonstrates an improvement with an accuracy of 93%.

The simplicity and efficiency of the implemented model make it suitable for real-time applications, a significant advantage over more complex methods that require extensive computational resources.

	precision	recall	f1-score	support
Non-Smile (Class 0)	0.90	0.91	0.90	454
Smile (Class 1)	0.92	0.91	0.92	527
accuracy			0.91	981
macro avg	0.91	0.91	0.91	981
weighted avg	0.91	0.91	0.91	981

Figure 6: Old result

viii. CONCLUSION AND FUTURE WORK

Conclusion:
In this study, we proposed a method for liveness detection using the Genki4K dataset, leveraging MediaPipe BlazeFace for face detection, dlib for landmark detection, and a CNN for classification. The experimental results demonstrate a high level of accuracy (93%) and robust performance across key metrics such as precision, recall, and F1 score. The integration of eye blink detection further enhanced the reliability of the model in distinguishing between live and spoof faces, making it a viable approach for real-world applications.

Future Work:
Future work can focus on the following areas to enhance the proposed method:
Dataset Expansion:
Utilize larger and more diverse datasets to improve the generalizability of the model.
Incorporate additional types of spoofing attacks to increase robustness.
Advanced Liveness Detection Techniques:
Implement more sophisticated liveness detection techniques, such as texture analysis and motion patterns.
Explore the use of multi-modal approaches combining audio, depth, and thermal data for enhanced accuracy.
Model Optimization:
Optimize the CNN architecture to reduce computational complexity and improve inference speed, making it more suitable for real-time applications.
Investigate the use of transfer learning and fine-tuning pre-trained models to improve performance.

ix. REFERENCES

[1] H. S. R. S. D. Shaunak Modak, "A SYSTEMATIC REVIEW OF ATTENDANCE SYSTEMS BASED ON FACE RECOGNITION," *INDIAN JOURNAL F RESEARCH*,

vol. 11, no. 11, pp. 2250 - 1991, 2022.

[2] K. E. S. a. V. S. Jayaraj Viswanathan, "Smart Attendance System using Face Recognition," *EAI Endorsed Transactions*, 2024.

[3] M. F. W. Myrna Dwi Rahmatya, "Online Attendance with Python Face Recognition and Django Framework," p. 13, 8 2023.

[4] L. T. M. T. C. W. Ooi Zhi Jie, "Biometric Authentication based on Liveness Detection Using Face Landmarks and Deep Learning Model," *International journal on informatics visualization*, pp. 1057-1065, 2023.

[5] M. J. Paul Viola, "Rapid Object Detection using a Boosted Cascade of Simple Features," ACCEPTED CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION, 2001.

[6] D. K. J. P. Florian Schroff, "FaceNet: A Unified Embedding for Face Recognition and Clustering," arXiv:1503.03832v3, 2015.

[7] L. S. W. X. O. M. P. a. A. Z. Qiong Cao, "VGGFace2: A dataset for recognising faces across pose and age," rXiv:1710.08092v2, 2018.

[8] J. U. A. M. A. A. Pronob Sarker, "Face Recognition and RFID Verified Attendance System," researchgate, 2018.

[9] E. N.-O. O. O. S. N. J. Kennedy Okokpujie, "Design and Implementation of a Student Attendance System Using Iris Biometric Recognition," researchgate, 2017.

[10] A. M. R. I. D. D. K. Samuel Lukas, "Student attendance system in classroom using face recognition technique," researchgate, 2016.

[11] A. R. R. Pratibha Shinde, "Face and liveness detection with criminal identification using," *IAES International Journal of Artificial Intelligence (IJ-AI)*, vol. 13, no. 722, p. 8, 2023.

[12] A. Rao, "AttenFace: A Real Time Attendance System Using," arXiv:2211.07582v1 , 2023.

[13] S. D. R. G. A. F. Joseph Redmon, "You Only Look Once: Unified, Real-Time Object Detection," arXiv:1506.02640v5, 2016.