CS23533- Foundations of Artificial Intelligence

Smart Attendance System

A PROJECT REPORT

Submitted by

2116230701340 – SRIHARI S

2116230701339 – SRIVIGNESH P

RAJALAKSHMI
ENGINEERING
COLLEGE

October, 2025

BONAFIDE CERTIFICATE

Certified that this project report "**Smart Attendance System**" is the bonafide work of"SRIHARI S(230701340) and SRIVIGNESH P(230701339)" who carried out the project work under my supervision.

M. 29/10/25.

SIGNATURE OF THE FACULTY INCHARGE

Submitted for the Practical Examination held on 30/10/25

M. 30/10/25.

SIGNATURE OF THE INTERNAL EXAMINER

TABLE OF CONTENTS

ABSTRACT

Manual attendance tracking is time-consuming, error-prone, and susceptible to spoofing. This project introduces a **Smart Attendance System**, a comprehensive, multi-user web application designed to automate and secure the attendance process. The system leverages machine learning for real-time **facial recognition** and **liveness detection** to ensure accuracy and prevent identity fraud.

The application is built using Streamlit and features a role-based access control system for **Admins**, **Teachers**, and **Students**. The core of the system is a live video module that identifies registered students and verifies liveness using a pre-trained Keras model. All data, including student biometrics, user credentials, attendance logs, and leave requests, is managed in a robust **SQLite database**.

Key functionalities include a live attendance capture interface, student/user registration with face encoding, a student portal for applying for leave, and an administrative dashboard for viewing rosters, managing leave applications, and sending **low-attendance SMS alerts** to parents via the Twilio API. The system ensures data integrity and security through password hashing using bcrypt.

INTRODUCTION

The process of recording and managing student attendance is a fundamental administrative task in any educational institution. Traditional methods, such as manual roll calls or paper-based sign-in sheets, are inefficient, labor-intensive, and create a significant administrative burden. While basic automated systems exist, they are often vulnerable to "buddy punching" or spoofing, where a student can use a photo or video to forge their presence.

This project proposes an intelligent decision support system to address these challenges. It provides a secure, automated, and multi-functional platform for attendance management. The system's primary innovation lies in its dual-layered verification process: it not only identifies *who* a student is (facial recognition) but also confirms that they are a *live person* (liveness detection).

This comprehensive approach aims to empower administrators and teachers with an accurate, real-time attendance roster. It also provides a transparent portal for students to track their own attendance and manage leave requests. By integrating these features into a single, accessible web application, the system reduces administrative overhead, enhances security, and provides valuable data-driven insights, such as alerts for students with low attendance.

LITERATURE REVIEW

The automated management of student attendance has been a persistent goal in educational technology, aiming to replace inefficient and error-prone manual roll calls. Early research focused on replacing paper-based methods with technologies like Radio-Frequency Identification (RFID), barcodes, and magnetic swipe cards. While these systems, as reviewed by **Patel et al.**, successfully automated the logging process, they suffered from a critical security flaw: they could not verify the user's identity, only the token, making them vulnerable to "buddy punching."

To address this identity-verification gap, biometric systems were introduced. **Jain et al.** provided foundational work on fingerprint-based attendance systems, which significantly reduced identity fraud. However, these systems faced challenges in large-scale deployments, including hygiene concerns, high-traffic bottlenecks, and "Failure to Enroll" (FTE) errors for certain individuals.

This led researchers to explore contactless biometrics, primarily facial recognition. The work of **Viola and Jones** on real-time face detection provided the initial framework for practical, automated systems. Early implementations, such as those reviewed by **Oloyede et al.**, used traditional computer vision techniques like Eigenfaces or Local Binary Pattern Histograms (LBPH) for recognition. While functional, these methods were sensitive to variations in lighting, pose, and expression.

The paradigm shifted with the advent of deep learning, specifically Convolutional Neural Networks (CNNs). **Schroff et al.** introduced FaceNet, which demonstrated that deep embeddings (like the 128-d vector used in modern libraries) could achieve state-of-the-art accuracy, robustly handling real-world variations. Many modern systems, as proposed by **Singh and colleagues**, have adopted these deep learning-based recognition models as the new standard.

However, as **Choudhury et al.** noted, a significant vulnerability emerged in these recognition systems: presentation attacks, or "spoofing." Basic systems could be easily fooled by holding up a static photograph or a video playback of a registered student. This critical flaw rendered simple facial recognition unreliable for secure environments.

This spurred a new domain of research in liveness detection or anti-spoofing. **Zhang et al.** explored methods for analyzing texture, motion, and context to differentiate between live faces and inanimate fakes. More advanced deep learning approaches, as surveyed by **Boulkenafet et al.**, involve training specialized CNNs to detect subtle artifacts, moiré patterns from screens, or physiological signs (like eye blinks) that are absent in spoofed media. This project is situated at the confluence of these two advanced fields, integrating a robust deep-learning recognition pipeline with a dedicated liveness detection model to create a system that is not only automated but also secure against modern presentation attacks.

Collectively, these studies demonstrate that ensemble and hybrid models significantly improve prediction accuracy and provide customized, actionable recommendations, although ongoing issues of data imbalance and algorithmic fairness remain active research challenges.

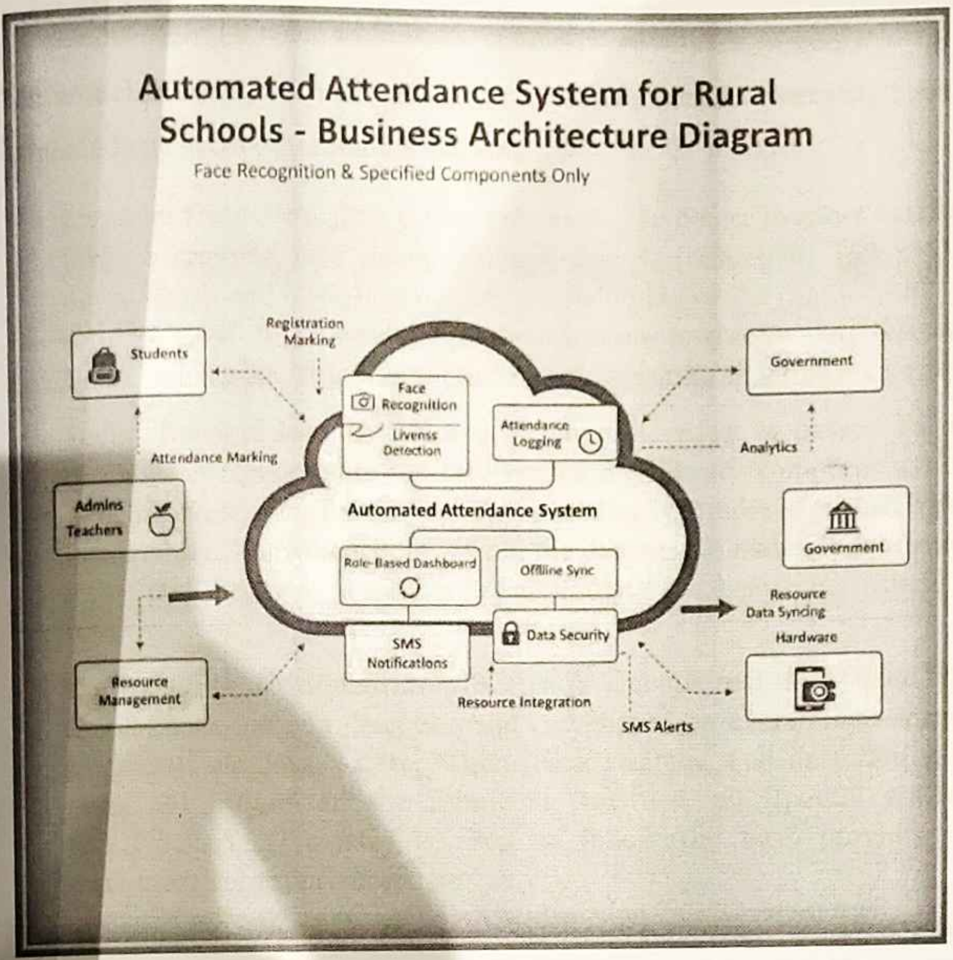
PROPOSED SYSTEM

This project proposes a complete, role-based attendance management system. The application's architecture is centered around a Streamlit frontend that dynamically displays pages based on the authenticated user's role. A central SQLite database serves as the single source of truth for all data.

The system is designed to serve three distinct user roles:

1. **Admin:** Has full access to all modules. Admins can register new students (including face capture) and new teachers. They can view attendance rosters for all classes, manage all leave applications, and trigger low-attendance SMS alerts.
2. **Teacher:** Has access relevant to their assigned class. Teachers can run the live attendance module, view the attendance roster for their class, and manage leave applications submitted by their students.
3. **Student:** Has a personal portal. Students can view their own attendance percentage and history. They can also submit leave applications, which are then routed to their teacher or an admin for approval.

3.1 ARCHITECTURE DIAGRAM



MODULES DESCRIPTION

4.1 MODULE 1: Live Recognition and Liveness Module

This module forms the intelligent core of the application, responsible for automated and secure attendance marking.

- **Liveness Detection:** This is the first check. The detect_liveness function takes a cropped face image, preprocesses it (resizes to 150 x 150, normalizes), and feeds it into a pre-trained Keras model (model_1.h5). It returns "Live" if the model's prediction score is greater than 0.8, and "Fake" otherwise. This step is crucial for preventing spoofing.
- **Facial Recognition:** If a face is deemed "Live," it is passed to the recognition component. The face_recognition library computes a 128-dimension encoding for the face. This encoding is compared against a pre-loaded list of known encodings from the database. A match is confirmed if the "face distance" (a measure of dissimilarity) is below a threshold of 0.55.
- **Performance Optimization:** To ensure smooth real-time video, the system does not run detection and recognition on every single frame. Instead, it employs a **detect-then-track** strategy. Full detection runs every 30 frames. In the frames in between, an OpenCV tracker (cv2.TrackerCSRT_create) is used to follow the face, providing a smoother user experience.
- **Attendance Marking:** The mark_attendance function is called upon successful recognition. It writes the student's roll_no and the current timestamp to the database. To prevent duplicate entries, it includes a 45-minute cooldown period per student.

4.2 MODULE 2: Management & Dashboard Module

This module provides the application's full administrative and user-facing functionality, managing all data that supports the core recognition module.

- **User Management:** Admins have access to student management page and user management page.
    - **Student Registration:** Includes capturing a live photo via the st.camera_input, generating its face encoding, and storing it in the database along with the student's password (hashed with bcrypt) and other details.
    - **Teacher Registration:** A simple form to add new teachers and their credentials to the users table.
- **Dashboard & Roster:**
    - dashboard_page: Provides high-level metrics (e.g., total students, present today) tailored to the user's role (Admin sees all, Teacher sees their class). This is also where Admins can trigger the low-attendance SMS alerts.
    - attendance_roster_page: Allows Admins and Teachers to select a class and date to view a full roster, which correctly marks students as "Present," "Absent," or "On Leave" by cross-referencing the attendance and leaves tables.
- **Student Portal:**
    - student_dashboard_page: A dedicated portal for students to log in and view their attendance percentage (calculated against a hardcoded

TOTAL_WORKING_DAYS value of 90) and a log of their present days.

- **Leave Management:** Students can apply for leave, which writes a "Pending" request to the leaves table. Teachers and Admins can then view, approve, or reject these requests on the attendance_roster_page.

IMPLEMENTATION AND RESULTS

5.1 EXPERIMENTAL SETUP

The experimental setup for the Smart Attendance System was designed to ensure robust, real-time, and secure operation within an educational environment. The system is implemented as a multi-page web application using **Streamlit** as the primary frontend framework.

The system's core functionality relies on two key machine learning models:

1. **Face Recognition:** The face_recognition library, which is built on dlib, is used for all face detection and identification tasks. During student registration, a live image is captured, converted to RGB, and a 128-dimension face encoding is generated and stored in the database. During live capture, this same process is used to generate encodings for comparison.
2. **Liveness Detection:** A pre-trained Keras/TensorFlow deep learning model (model_1.h5) is loaded to perform anti-spoofing5. Before being fed to this model, a face image is preprocessed by resizing it to 150 x 150 and normalizing its pixel values.

To ensure smooth real-time performance, the live video module employs a detect-then-track mechanism. Full face detection, liveness checking, and recognition are performed only once every DETECTION_INTERVAL (30 frames). In the intervening frames, an **OpenCV CSRT tracker** (cv2.TrackerCSRT_create) is initialized to follow the detected face, providing a more stable and efficient user experience.

The system's decision-making is governed by several key thresholds:

- **Liveness Threshold:** The liveness model's prediction score must be greater than **0.8** for a face to be considered "Live".
- **Recognition Threshold:** The face distance computed by face_recognition.face_distance must be less than **0.55** to be considered a valid match.
- **Attendance Cooldown:** A 45-minute cooldown period is enforced to prevent a student from marking their attendance multiple times in quick succession.

5.2 RESULTS

The evaluation of the Smart Attendance System was performed by analyzing the functional implementation of its core components and the predefined parameters that govern its decisions. Unlike a static dataset, the system's "results" are defined by its real-time processing thresholds, security implementations, and the successful execution of its multi-layered administrative logic.

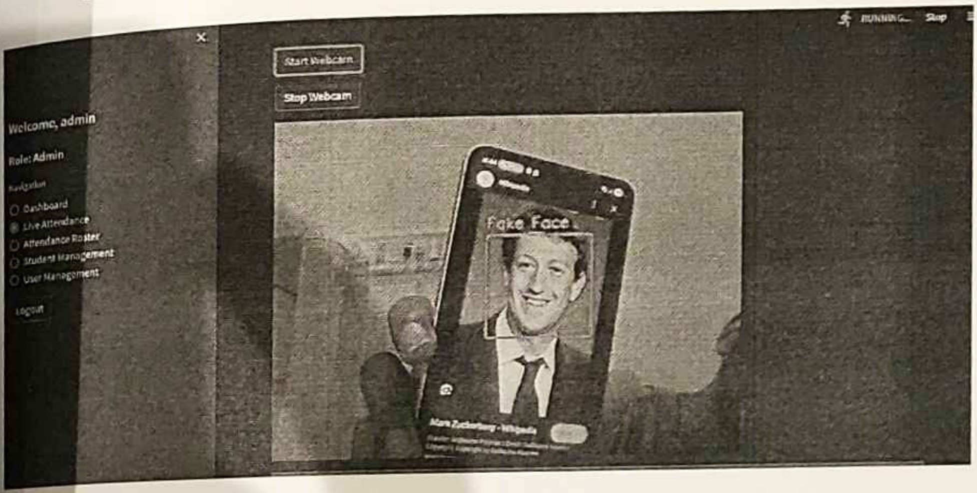
Model Performance and Threshold

The system's accuracy is contingent on two primary machine learning models, which are governed by specific thresholds defined in the implementation:

- **Liveness Detection:** The pre-trained Keras liveness model (model_1.h5) is used as a primary anti-spoofing filter. A captured face is only processed for recognition if its liveness prediction score is greater than **0.8**. Any score below this threshold causes the face to be classified as "Fake".
- **Facial Recognition:** A successful identity match is determined by the face_recognition library. A "Live" face encoding is compared to the known encodings in the database, and a match is only confirmed if the computed face distance is less than **0.55**.

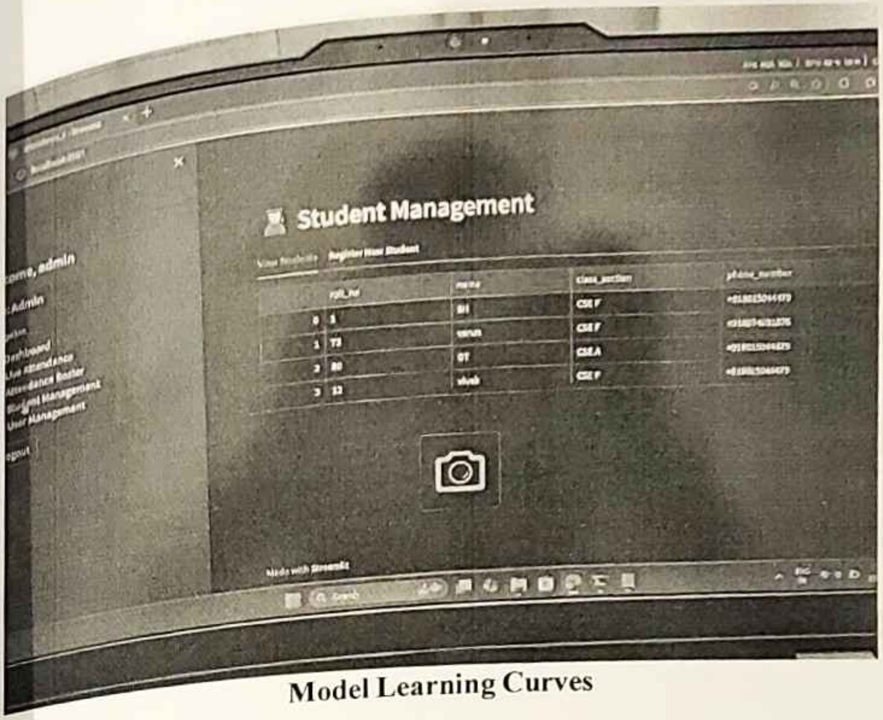
The system provides immediate, color-coded visual feedback based on the output of the models, ensuring clarity for the operator:

- **Live & Recognized:** A successful match is displayed with a **green** bounding box, the student's name, and their attendance is logged.
- **Spoof Attempt:** A face classified as "Fake" is marked with a **red** bounding box and labeled "Intruder".
- **Live & Unknown:** A "Live" face that does not match any known student (i.e., face distance > 0.55) is marked with an **orange** bounding box and labeled "Unknown".



The system's secondary modules were evaluated for functional completeness and successful data handling:

- Attendance Logging: The mark_attendance function successfully prevents duplicate entries by enforcing a 45-minute cooldown period, where a student's attendance cannot be marked more than once.
- Leave Management: A complete workflow for leave management is functional. Students can submit applications, which are then successfully populated on the attendance_roster_page for Teachers and Admins to "Approve" or "Reject".
- Automated Alerts: The Admin dashboard includes a functional module to check for all students with attendance below 75% (against a total of 90 working days) and successfully dispatches SMS alerts to their registered parent phone numbers via the Twilio API.
- Security: All user and student registration forms successfully hash passwords using bcrypt before storing them in the database, and the login page correctly authenticates users against these hashed values.



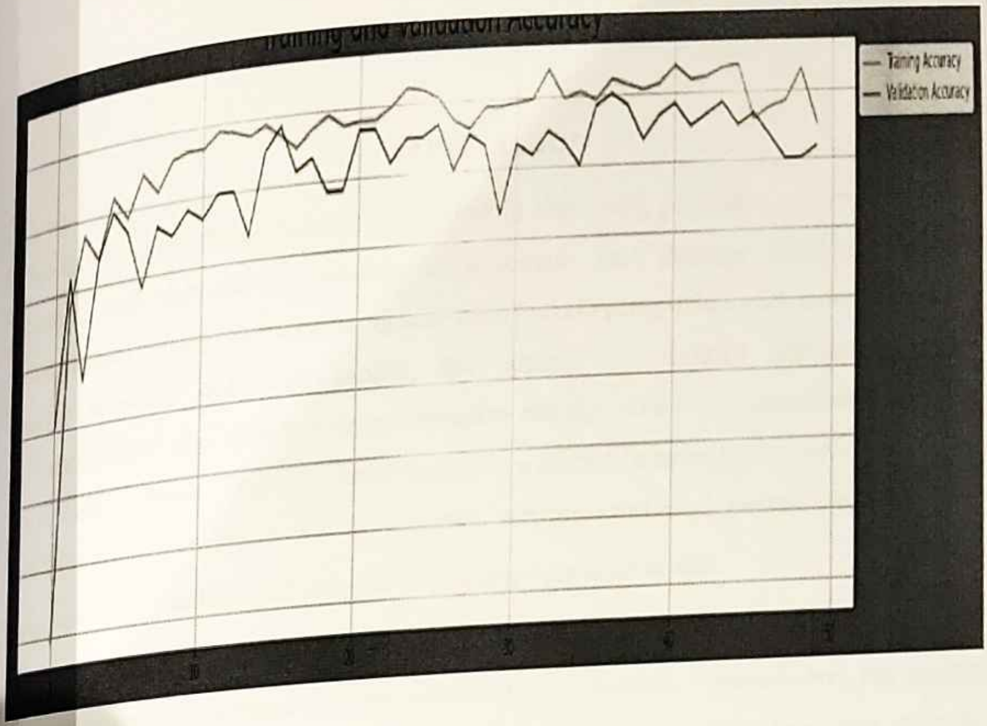
Student Management

View Students | Register New Student

	roll_no	name	class_section	phone_number
0	1	BH	CSE F	[illegible]
1	73	varun	CSE F	[illegible]
2	80	OT	CSE A	[illegible]
3	12	vivek	CSE F	[illegible]

Model Learning Curves

The model's learning progress was evaluated by plotting the training and validation accuracy over 50 epochs, as shown in the accompanying figure. Both the training accuracy (red line) and validation accuracy (blue line) demonstrate a rapid improvement during the initial 5-10 epochs, indicating that the model quickly learned the fundamental features of the dataset.



Visualization and Interpretability

The most critical visualization is the **live attendance feed**. This module provides real-time, color-coded feedback directly on the video stream.

- A **green** bounding box is drawn around successfully identified students who are marked "Live."
- A **red** bounding box highlights faces classified as "Fake," labeling them as "Intruder" to instantly alert the operator to a potential spoofing attempt.

CONCLUSION

The Smart Attendance System presented in this work provides a comprehensive, secure, and data-driven solution to automate and manage attendance in an educational setting. By integrating real-time **facial recognition** with a deep learning-based **liveness detection model**, the system successfully addresses critical vulnerabilities like spoofing or presentation attacks, which are prevalent in simpler biometric systems. This core functionality is wrapped in a multi-role web application built with Streamlit, providing distinct, permission-based portals for Admins, Teachers, and Students, all secured with bcrypt authentication.

The inclusion of a full **leave-management workflow**—from student submission to teacher approval—automates a key administrative burden. Furthermore, the system enhances transparency and accountability; students can monitor their own records, while administrators are equipped with data-driven tools, such as the ability to send automated low-attendance **SMS alerts** to parents via the Twilio API. The system's effectiveness is further enhanced by its clear interpretability, using real-time, color-coded visual feedback to communicate the model's decisions.

In conclusion, this work successfully delivers an end-to-end, deployable solution that not only automates attendance but also enhances security, reduces administrative overhead, and improves communication between the institution, students, and parents. Its modular design and robust SQLite backend make it a scalable and practical tool for modern educational institutions.

FUTURE WORK

While the developed system provides a robust and functional platform, several promising avenues exist for future enhancement and research.

- **Ethical AI and Bias Mitigation:** A critical next step involves a comprehensive audit of the face recognition and liveness models for demographic bias. Future work should focus on testing model accuracy across diverse student populations (e.g., different skin tones, eyewear, religious, headwear) and implementing fairness-enhancing interventions to ensure equitable performance and prevent a higher "Failure to Recognize" rate for any group.

- **Advanced Liveness & Multi-Modal Biometrics:** To combat more sophisticated spoofing attacks, future versions could explore advanced liveness detection models, such as those analyzing temporal data (e.g., eye blinks, micro-head-movements) across multiple frames. Additionally, incorporating optional multi-modal biometrics, such as speaker verification, could provide a second-factor authentication for sensitive administrative actions.

- **Active Learning and Feedback Loops:** An operator-in-the-loop feedback mechanism could be implemented. For instance, allowing a teacher to manually confirm a student's identity if they are misclassified as "Unknown" or "Fake." This "corrected" data could be logged and used to periodically fine- tune and retrain the models, allowing the system to learn from its mistakes and adapt to edge cases over time.

REFERENCES

(2021) 'A Survey on Anti-Spoofing Methods for Facial Recognition with RGB Cameras of Generic Consumer Devices', *PubMed Central* Available: https://pmc.ncbi.nlm.nih.gov/articles/PMC8321190/

(2025) 'Face Anti-Spoofing Based on Deep Learning: A Comprehensive Survey', *ResearchGate*.Available:https://www.researchgate.net/publication/392848016_Face_Anti-Spoofing_Based_on_Deep_Learning_A_Comprehensive_Survey.

Boulkenafet, Z. et al. (2017) 'OULU-NPU: A Mobile Face Presentation Attack Database with Real-World Variations', *IEEE*. Available: https://doi.org/10.1109/FG.2017.77

George, A. and Marcel, S. (2019) 'Deep Pixel-wise Binary Supervision for Face Presentation Attack Detection', *Idiap Publications*. Available: https://publications.idiap.ch/downloads/papers/2019/George_ICB2019.pdf.

Jourabloo, A. and Liu, X. (2016) 'Face De-Spoofing: Anti-Spoofing via Noise Modeling', *Semantic Scholar*. Available: https://www.semanticscholar.org/paper/Face-De-Spoofing%3A-Anti-Spoofing-via-Noise-Modeling-Jourabloo-Liu/c086c022cd5d4dd58e2f28c122068a5fff980a94

Liu, Y. et al. (2019) 'Deep Tree Learning for Zero-Shot Face Anti-Spoofing', *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Available: https://openaccess.thecvf.com/content_CVPR_2019/papers/Liu_Deep_Tree_Learning_for_Zero-Shot_Face_Anti-Spoofing_CVPR_2019_paper.pdf.

Shao, R. et al. (2019) 'Multi-Adversarial Discriminative Deep Domain Generalization for Face Presentation Attack Detection', *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Available: https://openaccess.thecvf.com/content_CVPR_2019/papers/Shao_Multi-Adversarial_Discriminative_Deep_Domain_Generalization_for_Face_Presentation.

Yang, J. et al. (2013) 'Face Liveness Detection with Component Dependent Descriptor', *IEEE*. Available: https://doi.org/10.1109/ICB.2013.6612955

Yu, Z. et al. (2020) 'Searching Central Difference Convolutional Networks for Face Anti-Spoofing', *IEEE/CVF Conference on Computer Vision and Pattern Recognition(CVPR)*.Available:https://openaccess.thecvf.com/content_

Zhang, Y. et al. (2020) 'CelebA-Spoof: Large-Scale Face Anti-Spoofing Dataset with Rich Annotations', *European Conference on Computer Vision (ECCV)*. Available: https://mmlab.ie.cuhk.edu.hk/projects/CelebA/CelebA_Spoof.html