ENGAGEMENT AND ASSESSMENT OPTIMIZATION SYSTEM FOR ENHANCING LEARNING EXPERIENCES IN ONLINE EDUCATION

24-25J-320

B.Sc. (Hons) Degree in Information Technology

Department of Information Technology

Sri Lanka Institute of Information Technology
Sri Lanka

April 2025

Declaration

We declare that this is our own work and this dissertation does not incorporate without acknowledgement any material previously submitted for a Degree or Diploma in any other University or institute of higher learning and to the best of my knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text.

Also, we hereby grant to Sri Lanka Institute of Information Technology, the nonexclusive right to reproduce and distribute my dissertation, in whole or in part in print, electronic or other medium. I retain the right to use this content in whole or part in future works (such as articles or books).

Signature: Date: 11/04/2025

The above candidates has carried out research for the bachelor's degree Dissertation under my supervision.

Signature of the Supervisor: Date:

Signature of the Co-Supervisor: Date:

Acknowledgment

The completion of this project would not have been possible without the exceptional support and guidance of several individuals. We are profoundly grateful to our research supervisor, Ms. Wishalya Tissera, whose expertise, enthusiasm, and meticulous attention to detail have been invaluable. Our heartfelt thanks also go to our co-supervisor, Ms. Chathushki Chathumali, for their insightful feedback and unwavering support.

we would also like to extend sincere appreciation to my research group members for their encouragement, constructive comments, and overall support throughout this project.

Lastly, we are deeply grateful to our parents for their unconditional support and love, which has been a constant source of motivation

Abstract

Methodologically, Edusmart integrates several AI modules. Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks analyze facial expressions and attention patterns for real-time engagement monitoring. To ensure academic integrity during exams, YOLO-based object detection identifies prohibited items, while MediaPipe-based head pose estimation monitors gaze direction; facial recognition provides user authentication. Natural Language Processing (NLP), leveraging Transformer models and Optical Character Recognition (OCR), facilitates automated grading of both typed and handwritten assignments. Furthermore, a deep learning model automatically generates relevant quiz questions from educational content.The system's components were rigorously evaluated using diverse datasets and standard performance metrics. Results demonstrated the effectiveness of the proposed modules. The engagement monitoring showed satisfactory accuracy. Proctoring components exhibited high proficiency in detecting integrity threats and verifying users. The automated grading system achieved strong correlation with human evaluations and effective classification performance. Automated quiz generation was also shown to be feasible and effective.In conclusion, the Edusmart system successfully integrates multiple AI capabilities to address critical limitations in contemporary online education. The findings validate AI's potential to create more interactive, secure, and efficient digital learning experiences. Recommendations for future work include exploring Explainable AI (XAI), enhancing robustness through multi-modal data integration, and facilitating broader adoption via integration with existing Learning Management Systems (LMS).

Keywords: E-learning, Artificial Intelligence in Education, Student Engagement Monitoring, Online Proctoring, Automated Assessment

Table of Contents

DECLARATION .... **II**

ACKNOWLEDGMENT .... **III**

ABSTRACT .... **IV**

TABLE OF CONTENTS .... **IV**

TABLE OF FIGURSES .... **VI**

LIST OF ABBREVIATIONS .... **VI**

1.INTRODUCTION **7**

1.1. RESEARCH GAP 12
1.2. RESEARCH PROBLEM 16
1.3.RESEARCH OBJECTIVES. 19

2.METHODOLOGY **21**

2.1.FACIAL EXPRESSION ANALYSIS AND ATTENTION DETECTION 21
2.1.1.Data Acquisition Strategy and Ethical Considerations 22
2.1.2.Data Preprocessing Pipeline 24
2.1.3.Model Architecture: CNN 26
2.1.4.Model Training and Optimization 28
2.1.5.Interpretation for Engagement and Attention Assessment 30
2.2.AUTOMATED QUIZ GENERATION AND EVALUATION 31
2.2.1. Data and Content Acquisition 31
2.2.2 Content Preprocessing and Feature Extraction 32
2.2.3. Model Selection and Fine-Tuning for Question Generation 34
2.2.4. Answer Evaluation Component 35
2.2.5. System Integration and Workflow 36
2.2.6. Evaluation of the Module Itself 37
2.3.UNAUTHORIZED OBJECT DETECTION, HEAD POSE ESTIMATION, AND USER VERIFICATION 38
2.3.1. Unauthorized Object Detection Component 38
2.3.2. Head Pose Estimation Component 41
2.3.3. User Verification Component 44
2.3.4. System Integration and Workflow 46
2.3.5. Evaluation 47
2.4. AUTOMATED GRADING SYSTEM 47
2.4.1. Data Acquisition and Ground Truth Generation 48
2.4.2. Data Preprocessing Pipeline 49
2.4.3. Feature Extraction and Model Architecture 51
2.4.4. Model Training and Optimization 53
2.4.5. Evaluation Metrics 54
2.4.6. System Integration and Workflow 56
2.5. Commercialization Aspects of the Product 56

3.RESULTS & DISCUSSION **58**

3.1.FACIAL EXPRESSION ANALYSIS AND ATTENTION DETECTION 58
3.2.AUTOMATED QUIZ GENERATION AND EVALUATION 63
3.3: UNAUTHORIZED OBJECT DETECTION, HEAD POSE ESTIMATION, AND USER VERIFICATION 67
3.4. Automated Grading System (Related to Text/Handwritten Inputs) 71

4.SUMMARY OF EACH STUDENT'S CONTRIBUTION **76**

5. CONCLUSION **84**

6.REFERENCES .... ERROR! BOOKMARK NOT DEFINED.

Table Of Figurses

FIGURE 1.FACIAL EXPRESSIONS AND ATTENTION DETECTION 26
FIGURE 2. FLOW OF AUTOMATED QUIZ GENERATION AND EVALUATION 36
FIGURE 3. FLOW OF UNAUTHORIZED OBJECT DETECTION 46
FIGURE 4.EMOTION RECOGNITION CLASSIFICATION REPORT 59
FIGURE 5. EMOTION RECOGNITION CONFUSION MATRIX 60
FIGURE 6.UNAUTHORIZED OBJECT DETECTION CLASSIFICATION REPORT 67
FIGURE 7.SPECIFIC GAZE CLASSIFICATIONS 68

List Of Abbreviations

Abbreviation	**Description**
LMS	Learning Management Systems
MOOCs	Massive Open Online Courses
AI	Artificial Intelligence
ML	Machine Learning
CV	Computer Vision
NLP	Natural Language Processing
CNNs	Convolutional Neural Networks
LSTMs	Long Short-Term Memory networks
OCR	Optical Character Recognition
CK+	Extended Cohn-Kanade
JAFFE	Japanese Female Facial Expression
HOG	Histogram of Oriented Gradients
SQuAD	Stanford Question Answering Dataset
ASAP	Automated Student Assessment Prize dataset
mAP	Mean Average Precision
MAE	Mean Absolute Error
FAR	False Acceptance Rate
FRR	False Rejection Rate
EER	Equal Error Rate
RNNs	Recurrent Neural
BiLSTMs	Bidirectional LSTMs
GRUs	Gated Recurrent Units
SVMs	Support Vector Machines
RMSE	Root Mean Squared Error
QWK	Quadratic Weighted Kappa

1.Introduction

The dawn of the twenty-first century has witnessed a paradigm shift in education, largely catalyzed by the rapid advancements in digital technology and the internet. E-learning, encompassing a wide spectrum of electronically supported learning and teaching methodologies, has transitioned from a niche alternative to a mainstream educational delivery model. Platforms ranging from institutional LMS like Moodle and Canvas to MOOCs offered by providers such as Coursera and edX, alongside bespoke virtual learning environments, have democratized access to education, transcending geographical boundaries and offering unprecedented flexibility to learners globally [1, Additional Ref. E-learning Growth]. This expansion has been further accelerated by global events, such as the COVID-19 pandemic, which necessitated a rapid, large-scale migration to online formats, fundamentally altering the educational landscape for millions of students and educators worldwide.

The appeal of e-learning is multifaceted. It offers learners the autonomy to study at their own pace, place, and time, catering to diverse learning styles and life circumstances, including those of working professionals and individuals in remote locations. Furthermore, digital platforms can provide access to a vast repository of resources, expert instructors from around the world, and potentially lower costs compared to traditional brick-and-mortar institutions. These advantages have fueled the exponential growth of the e-learning sector, promising a future where education is more accessible, personalized, and potentially more effective [1].

However, this digital transformation is not without its inherent complexities and significant challenges. While offering convenience and accessibility, the online environment presents unique obstacles that can hinder the quality and integrity of the educational experience [2]. Three critical areas of concern consistently emerge: maintaining genuine student engagement, ensuring academic integrity during assessments, and providing effective, scalable, and interactive learning support. Traditional classroom dynamics, which rely heavily on face-to-face interaction, non-verbal cues, and direct supervision, do not translate seamlessly into the virtual realm. The absence of a shared physical space often leads to feelings of isolation among students, reduced peer-to-peer interaction, and a diminished sense of community, all of which can negatively impact motivation and engagement [Additional Ref. Online Engagement Studies].

Quantifying student engagement, a crucial factor for effective learning, becomes particularly challenging online. In a physical classroom, instructors can intuitively gauge attentiveness, confusion, or enthusiasm through direct observation of body language, facial expressions, and participation levels. Online, however, instructors often face a grid of static icons or passive video feeds, making it difficult to ascertain whether students are truly cognitively and emotionally invested in the material or merely passively present [3]. This "engagement gap" poses a significant risk to learning outcomes, as disengaged students are less likely to absorb information, participate actively, or persist in their studies [3]. Furthermore, the lack of immediate, personalized feedback mechanisms common in smaller traditional settings can leave online learners feeling adrift, unsure of their understanding or progress [4]. Addressing this engagement deficit is paramount to realizing the full potential of e-learning.

Compounding the issue of engagement is the persistent challenge of upholding academic integrity in online assessments. The very flexibility and anonymity that make e-learning attractive can also create environments conducive to academic dishonesty [Additional Ref. Proctoring Limitations]. Conventional proctoring methods, designed for supervised, in-person examinations, are often impractical or easily circumvented online. Students taking exams remotely may have ready access to unauthorized resources, opportunities for illicit collaboration with peers, or even the possibility of impersonation (having someone else take the exam). Identifying instances of cheating, such as referring to external materials (notes, books, secondary devices) or receiving assistance, becomes exceedingly difficult without robust monitoring systems [5]. The reliance on traditional honor codes or basic lockdown browsers has proven insufficient to deter sophisticated cheating methods, thereby threatening the validity of online assessments and the credibility of qualifications earned through e-learning programs. Ensuring fairness and maintaining rigorous academic standards in remote testing environments is a critical hurdle that requires innovative solutions.

Furthermore, the scalability of assessment and feedback presents another significant challenge. As online course enrollments grow, particularly in MOOCs or large university classes, the burden of grading assignments and providing meaningful, timely feedback becomes immense for instructors. Manual grading of essays, short answers, or complex problem-solving tasks is time-consuming and prone to inconsistency [6] [7]. While multiple-choice questions are easily automated [8], they often fail to assess higher-order thinking skills effectively [9]. The delay in receiving feedback can also hinder the learning process, as students miss the opportunity for timely correction and reinforcement [10] [11] . Similarly, generating diverse and relevant practice questions or formative assessments manually is a labor-intensive process for educators [12] [13]. There is a clear need for tools that can automate aspects of assessment and content generation efficiently and reliably [8] [6] [7], freeing up instructors to focus on more complex pedagogical tasks and student interactions.

Faced with these multifaceted challenges – the engagement gap, compromised academic integrity, and the lack of scalable, interactive assessment – traditional online learning platforms often fall short. Basic LMS functionalities, while useful for content delivery and simple quizzes, typically lack the sophisticated monitoring, analysis, and automation capabilities required to address these deeper issues effectively. Early attempts at online proctoring, relying solely on webcam recording or human remote proctors, have faced criticism regarding intrusiveness, privacy concerns, cost, scalability limitations, and their inability to detect subtle forms of cheating or disengagement [Additional Ref. Proctoring Limitations]. It became increasingly apparent that more advanced, intelligent solutions were necessary to bridge these gaps and enhance the overall quality and trustworthiness of the e-learning experience.

This need paved the way for exploring the potential of AI. AI, particularly encompassing subfields like ML, CV, and NLP, offers a powerful toolkit to analyze complex data, recognize patterns, automate tasks, and provide insights that were previously unattainable [Additional Ref. AI in Ed Overview]. AI-driven systems hold the promise of creating more adaptive, responsive, and secure online learning environments [11]. Computer Vision techniques can analyze video feeds to interpret non-verbal cues like facial expressions and gaze direction, offering proxies for engagement and focus [9] [3] [14]. Object detection algorithms can identify prohibited items during exams [14] [15], while NLP can understand and evaluate student-written text, enabling automated grading [6] [16] [17] [7] and intelligent feedback generation

[10]. AI can also dynamically create assessment questions tailored to specific content or learner needs [1] [18] [13]. Recognizing this potential, this research focuses on harnessing these AI capabilities to build a comprehensive solution.

This paper introduces Edusmart, an innovative, integrated system designed specifically to address the critical challenges of engagement, integrity, and assessment efficiency in online education through the strategic application of AI. Edusmart is conceptualized as a dual-module system – a Learning Module and an Exam Module – each equipped with specialized AI-powered features to enhance different facets of the online educational journey.

The Learning Module focuses on fostering a more interactive and responsive learning environment. It leverages real-time facial expression analysis (using CNNs [9] and attention estimation (combining CNNs with LSTMs) derived from student webcam feeds during live lectures or study sessions [14][Additional Ref. LSTMs in Engagement]. By analyzing visual cues indicative of emotions (like confusion, interest, boredom) and attention levels [3], the system aims to provide instructors with aggregated, anonymized insights into student engagement, allowing for timely pedagogical interventions or adjustments [14]. This module also incorporates an automated grading system driven by NLP models (such as BERT and other Transformer architectures) [17] combined with traditional ML classifiers. This component is designed to evaluate student assignments [6] [7], including both typed responses and handwritten work (processed via OCR) [19], providing rapid, consistent feedback [11] and significantly reducing instructor grading time.

The Exam Module is dedicated to ensuring academic integrity during online assessments while also enhancing the assessment process itself. To deter and detect cheating, it employs object detection models (specifically, YOLOv5) [20] to identify unauthorized materials (e.g., smartphones, books, notes) within the student's view during an exam. Simultaneously, head pose estimation technology (utilizing MediaPipe Face Mesh and PnP algorithms) monitors the student's gaze direction to flag suspicious behavior indicative of looking away from the screen excessively or consulting off-screen resources, building on foundational techniques like facial landmark detection [21]. Robust user authentication via facial recognition ensures the identity of the test-taker [Additional Ref. Face Recognition Tech]. Complementing the integrity features, this module includes an AI-driven quiz generator. Using NLP models (like T5) trained on lecture notes or other provided content [12], this tool automatically generates relevant quiz questions (e.g., multiple-choice, short answer) [1] [13], offering a dynamic and adaptive way to create formative and summative assessments.

1.1. Research gap

While the application of Artificial Intelligence in e-learning has demonstrated considerable potential for addressing key challenges, a closer look reveals significant shortcomings and areas ripe for innovation. Current solutions often tackle problems like assessment or proctoring individually, creating a fragmented landscape of tools rather than a cohesive educational ecosystem. This piecemeal approach hinders the development of truly integrated platforms that concurrently enhance student engagement, guarantee academic integrity, and streamline assessment processes. The motivation behind the Edusmart system stems directly from the need to bridge these critical gaps identified within existing technologies and research trajectories.

One major area needing advancement is the monitoring and enhancement of student engagement. Although visual analysis of facial expressions and attention tracking through head pose or eye gaze are actively researched, current methods often fall short. Many systems depend on a single source of information, like facial expressions, which provides an incomplete and potentially misleading view of engagement—a multifaceted state involving cognitive, emotional, and behavioral aspects. Simple head orientation tracking might miss subtle attention shifts, and a neutral facial expression doesn't necessarily equate to disinterest. A more robust understanding requires integrating multiple cues, such as facial dynamics, gaze patterns, and even interaction logs, moving beyond basic emotion detection to infer deeper, pedagogically relevant states like confusion or comprehension. Furthermore, even when disengagement is detected, many systems lack the mechanism for real-time, actionable feedback. The focus frequently remains on post-session analysis rather than enabling immediate interventions by instructors or adaptive system adjustments within the learning environment itself. Existing models also struggle with generalizability; trained often in controlled lab settings, they may perform poorly in diverse real-world online learning scenarios characterized by varying lighting, camera quality, cultural expressions, and home environments. Compounding these technical limitations are persistent privacy and ethical concerns surrounding the continuous monitoring of students, where robust safeguards, transparent policies, and ethical frameworks often lag behind technological capabilities.

Similarly, ensuring academic integrity in online assessments remains a significant challenge despite various technological efforts. Current proctoring solutions, including object detection for banned items and head pose estimation for monitoring focus, often have a limited scope. They tend to flag obvious behaviors like using a phone or consistently looking away but frequently fail to detect more sophisticated cheating strategies, such as the use of hidden devices, external coaching, secondary screens, or remote collaboration. The reliability of these automated systems is also a concern, as they can suffer from high rates of false positives, flagging innocent behavior as suspicious, or false negatives, missing actual instances of misconduct. This unreliability undermines confidence in automated proctoring and can create unnecessary anxiety for honest students. User verification, typically done via facial recognition at the start of an exam, also requires improvement to ensure continuous, reliable identification throughout the assessment period, robust against varying conditions and potential spoofing attempts. There is an inherent tension between the level of intrusiveness required for effective monitoring and the potential negative impact on student privacy, anxiety, and performance. Developing effective yet minimally intrusive methods that strike an acceptable balance remains an ongoing research pursuit.

In the domain of automated assessment and feedback, substantial progress has been made, moving from simple keyword matching to sophisticated natural language processing models capable of understanding context and even evaluating handwritten text via optical character recognition. Despite these advancements, critical gaps persist. Evaluating the deeper qualities of student responses—such as reasoning, creativity, critical thinking, and originality—remains a formidable challenge for AI, which often excels at surface-level analysis or factual checks but struggles with semantic nuance and genuine pedagogical assessment. Handling diverse answer formats, particularly those involving complex mathematical notations, diagrams, code, or specialized terminology, especially in handwritten submissions where OCR errors can occur, also presents difficulties. A significant concern is the potential for bias within automated grading systems; models trained on large datasets can inherit and perpetuate societal biases, potentially disadvantaging certain student groups. Ensuring fairness, equity, and mitigating bias are paramount yet underdeveloped aspects of automated assessment. Furthermore, many advanced grading models function as opaque "black boxes," making it difficult for educators and learners to understand the rationale behind a given score. This lack of transparency, or explainability, hinders trust and limits the learning potential of feedback. Consequently, the feedback provided by automated systems, while immediate, often lacks the personalized, constructive, and guiding quality of human feedback, frequently being generic or purely corrective rather than truly diagnostic and supportive of student development.

Automated content generation, particularly for creating quizzes, also presents opportunities and challenges. While AI can generate questions from source material, ensuring the consistent quality, relevance, and pedagogical soundness of these questions is difficult. Automatically generated items may suffer from ambiguity, grammatical errors, triviality, or misalignment with learning objectives. A specific challenge lies in crafting effective distractors for multiple-choice questions—options that are plausible enough to challenge students but are clearly incorrect. Poor distractors can significantly diminish a question's assessment value. Moreover, similar to automated grading, current quiz generation techniques often default to questions testing factual recall, with less progress made in automatically creating items that effectively probe higher-order thinking skills like analysis, synthesis, or evaluation. Finally, the potential for dynamically adapting quizzes based on real-time learning analytics and detected student engagement levels is largely unrealized; tightly integrated systems that adjust question type, difficulty, and topic based on ongoing performance are still relatively uncommon.

Perhaps the most significant deficiency cutting across all these areas is the lack of comprehensive integration. The majority of available tools and research prototypes focus on addressing only one facet of the online learning challenge—be it grading, proctoring, or engagement tracking—in isolation. This siloed approach overlooks the interconnected nature of these issues; for example, student engagement can influence assessment performance, and integrity measures are an integral part of the assessment context. A truly holistic system could leverage insights from one component to inform another, creating synergies that enhance the overall learning experience. For instance, engagement data could dynamically adjust quiz difficulty, or assessment results could highlight areas needing more engaging content delivery. There is a distinct scarcity of research exploring the design, implementation, and efficacy of such integrated, multi-functional AI-driven e-learning platforms. Addressing this overarching integration gap is central to the mission of Edusmart, which aims to combine these diverse AI capabilities into a single, cohesive system designed to foster a more effective, secure, and supportive online educational environment than is possible with fragmented solutions.

1.2. Research Problem

The rapid expansion of e-learning has fundamentally altered educational delivery, offering unprecedented accessibility and flexibility. However, this digital shift has concurrently exposed and often amplified critical deficiencies inherent in many online learning environments. Despite numerous technological advancements and the application of Artificial Intelligence tools to specific issues, a significant, overarching problem persists: the lack of integrated, effective, and reliable systems capable of holistically addressing the intertwined challenges of student engagement, academic integrity, and efficient, meaningful assessment in online settings. Current platforms and research efforts often provide piecemeal solutions, resulting in a fragmented ecosystem that fails to deliver a consistently high-quality, secure, and engaging educational experience. This fragmentation hinders the realization of e-learning's full potential and forms the core research problem this study seeks to address.

The multifaceted nature of this problem manifests in several key areas. Firstly, the problem of maintaining and accurately assessing student engagement remains largely unsolved in scalable online formats. Without the rich non-verbal cues available in face-to-face settings, instructors struggle to gauge genuine student attentiveness and comprehension. Existing technological attempts to bridge this gap often rely on superficial metrics or unimodal analysis (like basic facial expression recognition) which provide an incomplete picture, lack real-time feedback loops for intervention, struggle with robustness in diverse real-world conditions, and raise significant privacy concerns. The core problem here is the absence of reliable, non-intrusive, context-aware, and ethically sound mechanisms to understand and respond to student engagement dynamics effectively, leading to potential disengagement, reduced learning outcomes, and higher dropout rates.

Secondly, the problem of ensuring academic integrity during online assessments poses a severe threat to the credibility of digital education. Conventional proctoring methods are ill-suited for remote environments, while current automated solutions face significant hurdles. These systems often have a limited scope, failing to detect subtle or sophisticated cheating methods, and suffer from unacceptable rates of false positives and negatives. This unreliability not only fails to adequately deter academic dishonesty but can also unfairly penalize honest students and create significant test anxiety. Furthermore, robust and continuous user verification remains a challenge. The research problem, therefore, encompasses the need for comprehensive, accurate, reliable, and fair integrity assurance systems that can effectively detect a wide range of misconduct and verify identity without being overly intrusive or biased, thereby safeguarding the validity of online assessments.

Thirdly, the problem of scalable, high-quality assessment and feedback continues to limit the educational effectiveness of many online courses, particularly those with large enrollments or requiring evaluation of complex skills. While automation for simple question types is established, automated grading for open-ended responses, complex problem-solving, or creative work often lacks the depth of human evaluation. Current AI models struggle to assess higher-order thinking skills accurately, handle diverse answer formats consistently, ensure fairness and mitigate bias, and provide transparent, explainable grading justifications. Furthermore, the automatically generated feedback often falls short of being truly personalized, diagnostic, and constructive. The research problem here is the need for advanced automated assessment tools that can not only handle scale and efficiency but also provide deep pedagogical evaluation, maintain fairness, offer transparency, and deliver feedback that genuinely promotes learning. This extends to the automated generation of assessment content, where ensuring quality, relevance, and the ability to test critical thinking remains problematic.

Ultimately, the overarching research problem is the systemic failure resulting from the lack of integration among solutions addressing these distinct yet interconnected challenges. Engagement monitoring, integrity assurance, and assessment automation are often developed and deployed in isolation, preventing the potential synergistic benefits of a unified approach. Data from one area (e.g., engagement levels) is rarely used to inform another (e.g., adaptive assessment difficulty or targeted feedback). This lack of integration leads to a disjointed user experience, potential redundancies, and missed opportunities for creating truly adaptive, responsive, and secure learning environments. Therefore, the central research problem this work confronts is the need to design, develop, and evaluate a holistic, integrated AI-powered system that simultaneously addresses the critical limitations in engagement monitoring, academic integrity, and assessment efficiency, thereby overcoming the fragmentation prevalent in current e-learning technologies and fostering a more robust and effective online educational ecosystem.

1.3.Research Objectives.

The primary goal of this research is to design, develop, and evaluate an integrated AI-powered system (Edusmart) to address key challenges in online learning. The specific objectives are:

1. To Develop and Validate AI Models for Student Engagement Analysis:
    - To design and train CNN and LSTM models for real-time facial expression recognition and attention estimation using student webcam data.
    - To evaluate the accuracy, F1-score, and robustness of these models in classifying distinct engagement-related states (e.g., attentive, confused, disengaged, specific emotions) in simulated online learning scenarios.
2. To Implement and Assess AI Techniques for Enhancing Academic Integrity:
    - To implement computer vision models (e.g., YOLOv5) for detecting unauthorized objects (e.g., mobile phones, notes) within the test-taker's environment during online assessments.
    - To implement head pose estimation techniques (e.g., using MediaPipe Face Mesh and PnP algorithms) to identify suspicious gaze patterns indicative of cheating (e.g., excessive looking away).
    - To integrate and test a facial recognition module for robust user authentication at the start and potentially during online exams.
    - To evaluate the performance (e.g., accuracy, precision, recall, mAP) of these integrity assurance modules in simulated exam conditions.
3. To Create and Evaluate an Automated Grading System for Diverse Assignments:
    - To develop an automated grading system utilizing NLP models (e.g., Transformer architectures like BERT, T5) and machine learning classifiers.
    - To incorporate OCR capabilities to enable the processing and grading of handwritten assignments.

  - To evaluate the system's accuracy (correlation with human graders), consistency, and efficiency in grading various types of student responses (e.g., short answers, essays) across different subjects.

4. To Develop and Demonstrate an AI-Powered Quiz Generation Tool:
   - To develop an NLP-based tool (e.g., using a fine-tuned T5 model) capable of automatically generating relevant assessment questions (e.g., multiple-choice, short answer) from provided educational content (e.g., lecture notes, text documents).
   - To demonstrate the feasibility of the quiz generator and qualitatively assess the relevance, grammatical correctness, and potential pedagogical value of the generated questions.
5. To Integrate System Components and Evaluate the Holistic Solution:
   - To design and implement a cohesive system architecture (Edusmart) that integrates the developed modules for engagement analysis, academic integrity, automated grading, and quiz generation.
   - To conduct preliminary evaluations or demonstrations of the integrated system to assess its overall feasibility, potential usability, and the synergistic benefits of combining these AI functionalities into a unified platform for enhancing the online learning experience.

2.Methodology

This section details the systematic approach undertaken to design, develop, and evaluate the "Edusmart" system, directly addressing the research objectives outlined previously. The core methodology revolves around the application and integration of various AI techniques, including ML, CV, and NLP, to tackle the identified challenges in online student engagement, academic integrity, and assessment efficiency. Adopting a predominantly experimental and system development approach, this research involved distinct phases for each functional module: data acquisition and preprocessing, model selection and training, and performance evaluation using appropriate metrics. The following subsections provide a detailed breakdown of the specific methods employed for developing the Facial Expression and Attention Detection module, the Automated Quiz Generation and Evaluation system, the Unauthorized Object Detection, Head Pose Estimation, and User Verification components, and the Automated Grading system, culminating in an integrated solution designed to enhance the online learning ecosystem.

2.1.Facial Expression Analysis and Attention Detection

This foundational module of the Edusmart system is designed to quantitatively assess student engagement during online learning activities by analyzing visual cues derived from the student's webcam feed. The primary hypothesis is that facial expressions and associated attention patterns serve as reliable, non-intrusive proxies for a student's emotional and cognitive state, such as interest, confusion, boredom, or focused attention. Detecting these states in real-time allows for potential interventions by instructors or adaptive adjustments by the learning system itself. The methodology for developing this module involved several key stages: comprehensive data collection and ethical handling, rigorous data preprocessing, the design and training of a deep learning model for expression classification, and the interpretation of model outputs for engagement assessment.

2.1.1.Data Acquisition Strategy and Ethical Considerations

The development of a robust facial expression recognition model necessitates a large, diverse, and accurately labeled dataset that reflects the variability of real-world conditions. Recognizing the limitations of relying solely on existing benchmark datasets, which are often captured under controlled laboratory settings, a multi-pronged data acquisition strategy was adopted.

- **Leveraging Standard Datasets:** Publicly available, labeled facial expression datasets served as a crucial starting point. Specifically, the CK+ dataset and the JAFFE dataset were utilized. CK+ provides sequences of expressions culminating in peak emotions, labeled with FACS action units and discrete emotion categories, offering valuable data on expression dynamics. JAFFE provides static images of posed expressions from Japanese female participants, contributing cultural diversity, albeit limited. These datasets provided a baseline for model training and validation on standardized emotional expressions.
- **Direct Data Collection (Simulated Environment):** To enhance the model's generalizability and applicability to the specific context of online learning, direct data collection was undertaken. Participants were recruited primarily from the university community (students and staff) and potentially wider community gatherings to ensure a varied demographic sample in terms of age, gender, ethnicity, and appearance (e.g., presence of glasses). Ethical considerations were paramount throughout this process.

    - **Informed Consent:** Prior to participation, individuals received a clear explanation of the study's purpose, the type of data being collected (video/images of their face), how it would be used (training an AI model for engagement analysis), stored (secure servers), and protected (anonymization). Written informed consent was obtained from all participants. They were explicitly informed of their right to withdraw at any time without penalty.

    - **Data Collection Protocol:** Participants were typically asked to interact with sample e-learning content (e.g., watch short video lectures, read text passages, attempt simple quizzes) in a setup mimicking a typical online learning environment (using a standard computer webcam). They might have been prompted occasionally to display specific expressions or react naturally to varying content stimuli (e.g., confusing explanations, engaging visuals). Video recordings were the primary capture method to potentially allow for future analysis of temporal dynamics, although the core expression classification focused on individual frames.
    - **Anonymization and Security:** Collected video data was processed to extract relevant frames. All personally identifiable information beyond the facial data itself was removed. Datasets were stored on secure, access-controlled servers. Faces might be assigned unique identifiers unrelated to participant names.
    - **Minimizing Bias:** Efforts were made during recruitment to capture a diverse range of participants to mitigate demographic bias in the dataset, although achieving perfect representation remains challenging.

- **Data Labeling:** While standard datasets came pre-labeled, the directly collected data required annotation. This was approached using a combination of methods: potentially participant self-reporting immediately after displaying an expression or experiencing a feeling, or more commonly, annotation by trained research assistants. Raters were trained on a defined set of emotion categories (the seven target expressions: Anger, Disgust, Fear, Happiness, Sadness, Surprise, Neutral) using standardized protocols and examples. Inter-rater reliability checks were potentially performed to ensure labeling consistency, especially for subtle or mixed expressions often encountered in naturalistic settings. The goal was to create frame-level labels corresponding to the dominant perceived emotion.

2.1.2.Data Preprocessing Pipeline

Raw image or video data is unsuitable for direct input into deep learning models. A rigorous preprocessing pipeline was implemented to standardize the data, reduce computational complexity, remove irrelevant information, and enhance features relevant to facial expressions.

- **Face Detection and Alignment:** The first crucial step was to automatically detect and isolate the face region within each image frame. This was achieved using established computer vision libraries like OpenCV, Dlib, or potentially MediaPipe's Face Detection module, which employ algorithms like Haar cascades, HOG, or optimized deep learning detectors. Accurate detection ensures that the model focuses only on the relevant facial area, discarding background clutter. Facial landmark detection (identifying key points like eyes, nose, mouth corners) might also be used implicitly by the detection model or explicitly for face alignment (e.g., rotating and scaling the face so that key features like eyes are in consistent positions), although not explicitly detailed as a separate step in the original paper summary.
- **Image Resizing:** Detected face regions were resized to a uniform dimension of 48x48 pixels. This standardization is critical as CNNs require fixed-size inputs. The 48x48 resolution is a common standard used in facial expression recognition challenges (like FER2013) and represents a trade-off between retaining sufficient detail for expression analysis and managing computational load. Smaller images allow for faster training and inference.
- **Grayscale Conversion:** Color information, while potentially useful, significantly increases computational complexity (3 channels vs. 1). For facial expression recognition, key information often lies in the shape, texture, and intensity variations (e.g., wrinkles, mouth shape, eye aperture). Converting images to grayscale reduces the input data dimension (from 48x48x3 to 48x48x1), simplifying the model architecture and potentially making it less sensitive to variations in skin tone and lighting color temperature, focusing more on luminance patterns.

- **Pixel Value Normalization:** Pixel intensity values (typically ranging from 0 to 255 in grayscale) were normalized to fall within a smaller range, specifically [0, 1]. This was achieved by dividing each pixel value by 255. Normalization is a standard practice in deep learning that helps stabilize training and accelerate convergence by ensuring that input features have a similar scale, preventing gradients from exploding or vanishing during backpropagation.
- **Data Splitting:** The preprocessed and labeled dataset was carefully divided into three distinct subsets:

  - **Training Set (e.g., 80-90%):** Used to train the parameters (weights and biases) of the CNN model. The model learns patterns and relationships from this data.
  - **Validation Set (e.g., 10%):** Used during the training process to tune hyperparameters (e.g., learning rate, number of layers, regularization strength) and make decisions like when to stop training (Early Stopping). It provides an unbiased estimate of model performance on unseen data during development.
  - **Test Set (e.g., 10%):** Used only after the model is fully trained and hyperparameters are finalized. It provides a final, unbiased evaluation of the model's generalization ability on completely new data it has never encountered during training or validation.The train_test_split function (likely from Scikit-learn) was used, employing stratified splitting (stratify=y) to ensure that the proportion of samples for each emotion class was roughly the same across the training, validation, and test sets. This is crucial for preventing biased evaluation, especially if some emotions are less frequent in the dataset. Setting a random_state ensures reproducibility of the split.

2.1.3.Model Architecture: CNN

Given the image-based nature of the input data and the task of identifying spatial patterns characteristic of facial expressions, a CNN) architecture was selected as the primary modeling approach. CNNs are particularly well-suited for image classification tasks due to their ability to automatically learn hierarchical features.

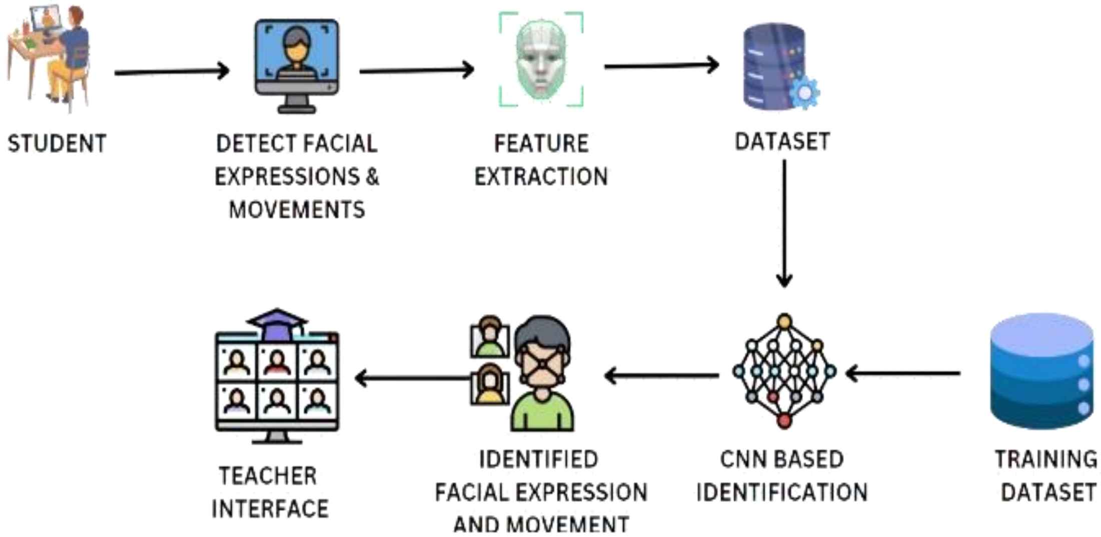


Figure 1.Facial Expressions and Attention Detection

- **Diagram Description:** This diagram should visually represent the layers of the CNN. It would start with the 48x48x1 input layer. Following this would be blocks representing Convolutional layers (showing filter/kernel icons, possibly indicating kernel size like 3x3, and number of filters), followed by ReLU activation functions, and then MaxPooling layers (showing downsampling, e.g., 2x2 pool size). These Conv-ReLU-Pool blocks would repeat 2-4 times, with the number of filters typically increasing in deeper layers. After the convolutional base, a Flatten layer reshapes the 2D feature maps into a 1D vector. This vector feeds into one or more Fully Connected (Dense) layers (represented as interconnected nodes), potentially with Dropout layers interspersed for regularization. The final layer is a Dense layer with 7 neurons (one for each emotion class) using a Softmax activation function, outputting probabilities.
- **Input Layer:** Accepts the preprocessed 48x48x1 grayscale images.

- **Convolutional Layers:** These are the core building blocks. They apply a set of learnable filters (kernels) across the input image to detect low-level features like edges, corners, and textures in the initial layers, and more complex patterns (like parts of eyes or mouth shapes) in deeper layers. Key parameters include the number of filters, kernel size (e.g., 3x3 or 5x5), stride (how many pixels the filter moves), and padding (to control output size).
- **Activation Functions (ReLU):** Following convolutional layers, a non-linear activation function like the Rectified Linear Unit (ReLU) (f(x) = max(0, x)) is typically applied. ReLU introduces non-linearity, allowing the network to learn more complex relationships, and helps mitigate the vanishing gradient problem.
- **Pooling Layers (MaxPooling):** These layers perform down-sampling, reducing the spatial dimensions (width and height) of the feature maps. MaxPooling takes the maximum value within a defined window (e.g., 2x2), which helps make the feature representation more robust to small translations and distortions in the input image and reduces the number of parameters, controlling overfitting.

- **Flatten Layer:** After several convolutional and pooling layers, the resulting multi-dimensional feature maps are flattened into a single long vector to be fed into the fully connected layers.
- **Fully Connected (Dense) Layers:** These are standard neural network layers where each neuron is connected to every neuron in the previous layer. They perform high-level reasoning on the features extracted by the convolutional layers to make the final classification decision. One or more dense layers might be used.
- **Dropout Layers:** Included between fully connected layers (or sometimes after pooling layers) as a regularization technique. During training, dropout randomly sets a fraction of neuron activations to zero at each update step. This prevents neurons from co-adapting too much and forces the network to learn more robust features, reducing overfitting.

- **Output Layer:** The final layer is a Dense layer with 7 neurons, corresponding to the 7 target emotion classes. A Softmax activation function is applied to this layer. Softmax converts the raw output scores (logits) into a probability distribution, where each output represents the model's confidence that the input image belongs to that specific emotion class, and the probabilities sum to 1.

2.1.4.Model Training and Optimization

The process of training the CNN involved iteratively adjusting the model's weights and biases to minimize the difference between its predictions and the true labels on the training data.

- Loss Function: Categorical Cross-Entropy was used as the loss function. This is the standard choice for multi-class classification problems with a Softmax output layer. It measures the dissimilarity between the predicted probability distribution and the true distribution (one-hot encoded labels), aiming to minimize this difference.
- Optimizer: The Adam (Adaptive Moment Estimation) optimizer was employed. Adam is an efficient and popular optimization algorithm that computes adaptive learning rates for each parameter, often leading to faster convergence compared to traditional Stochastic Gradient Descent (SGD). A specific learning rate (e.g., 0.001, as mentioned in the paper) was initially set.
- Evaluation Metric: While the loss function guides training, Accuracy (the proportion of correctly classified images) was used as the primary metric to evaluate model performance during training and testing, as it is easily interpretable. Other metrics like F1-score (harmonic mean of precision and recall), precision, and recall were also calculated post-training for a more nuanced evaluation, particularly considering potential class imbalances (as reported in Figure 5).
- **Training Procedure:** The model was trained for a specified number of epochs (100 mentioned). An epoch represents one full pass through the entire training dataset. Training was performed in batches (batch size: 32 mentioned), where the model parameters are updated after processing each small batch of training

samples, making the training process more computationally manageable and often more stable.

- **Overfitting Mitigation:** Several techniques were crucial to prevent the model from overfitting (i.e., performing well on the training data but poorly on unseen data):

  - **Data Augmentation:** Keras's ImageDataGenerator was used to apply random transformations to the training images on-the-fly during training. The specified augmentations included rotation (±15°), horizontal and vertical shifts (±15%), shear transformations (±15%), zoom (±15%), and horizontal flipping. This artificially expands the training dataset and exposes the model to a wider variety of image variations, forcing it to learn more robust and invariant features.
  - **Early Stopping:** Training was monitored using the validation set performance (e.g., validation accuracy or loss). If the monitored metric did not improve for a specified number of consecutive epochs (patience: 11 mentioned), the training process was automatically halted. The model weights corresponding to the best performance on the validation set were saved, preventing the model from continuing to train and potentially overfit after performance plateaus or starts to degrade.
  - **ReduceLROnPlateau:** This callback mechanism monitored the validation metric (e.g., validation loss). If the metric stopped improving for a certain number of epochs (patience: 7 mentioned), the learning rate was reduced by a specified factor (factor: 0.5 mentioned). This allows the optimizer to make finer adjustments and potentially escape local minima or converge more effectively when progress slows down.

- Implementation Details: The model was likely implemented using standard deep learning frameworks like TensorFlow with its high-level Keras API or PyTorch. Training was likely accelerated using Graphics Processing Units (GPUs) due to the computational demands of deep learning.

2.1.5.Interpretation for Engagement and Attention Assessment

The direct output of this module is a probability distribution over the seven basic emotion categories for each processed frame or image. Translating this output into a meaningful assessment of "engagement" or "attention" requires an interpretation layer. While the core methodology focused on accurate emotion classification, the use of this information for engagement involves:

- **Mapping Emotions to Engagement States:** Certain emotions (e.g., Happiness, Surprise interpreted as interest) might be mapped to positive engagement indicators, while others (e.g., Sadness, Boredom inferred from prolonged Neutrality, Anger/Disgust interpreted as frustration) could indicate negative engagement or disengagement. Confusion (potentially linked to Fear or Surprise depending on context) might signal a need for clarification.
- **Temporal Analysis (Potential LSTM use):** Although the primary results focused on the CNN, analyzing the sequence of classified emotions over time (potentially using LSTMs as mentioned conceptually in the introduction) could provide richer insights. For example, frequent shifts between emotions might indicate active processing, while sustained periods of negative or neutral expressions could suggest waning attention. Rapid changes could also be noise, requiring smoothing or thresholding.
- **Integration with Other Cues:** The most robust assessment of attention would likely involve fusing the facial expression data with outputs from other modules, particularly head pose estimation (Module C). Consistent off-screen gaze coupled with a neutral expression is a stronger indicator of inattention than either cue alone.

2.2.Automated Quiz Generation and Evaluation

This module addresses the challenges of creating scalable, relevant assessments and providing timely feedback in e-learning environments. It leverages NLP and Transformer-based deep learning models to automate the process of generating quiz questions directly from educational content and subsequently evaluating student responses. The goal is to reduce instructor workload, provide students with immediate practice and feedback opportunities, and enable more adaptive assessment strategies. The methodology encompasses content acquisition, text preprocessing, fine-tuning a question generation model, developing an answer evaluation mechanism, and integrating these into a functional system. Figure 3 illustrates the high-level workflow of this module.

2.2.1. Data and Content Acquisition

The effectiveness of an automated quiz generator hinges on the quality and nature of the input content from which questions are derived, as well as data for training the underlying models.

- **Input Educational Content:** The system is designed to work with various forms of instructional materials typically used in e-learning. This includes:
  - **Lecture Notes:** Digitally available notes provided by instructors, often in formats like .txt, .docx, or .pdf.
  - **Textbooks / Chapters:** Sections of digital textbooks or relevant academic articles, primarily expected in PDF format.
  - **Transcripts:** Potentially, transcripts generated from video lectures (though this adds a dependency on accurate speech-to-text technology, not explicitly detailed as a core feature in the initial paper).
  - **Source Material Characteristics:** The effectiveness depends on the text being reasonably well-structured, coherent, and containing factual information or concepts suitable for question generation (e.g., definitions, explanations, processes, key facts). Content that is overly

conversational, poorly organized, or purely opinion-based would be less suitable.

- **Model Training/Fine-tuning Data:** To enable the core NLP model to perform question generation and potentially answer evaluation, pre-existing labeled datasets are crucial for fine-tuning.
    - **SQuAD :**As mentioned in the paper (Section III.B, last paragraph), the SQuAD dataset was used. SQuAD consists of context passages from Wikipedia and corresponding question-answer pairs, where the answer is a span of text within the context. While primarily designed for question answering, it can be adapted for question generation by training a model to generate the question given the context and the answer span. This helps the model learn the relationship between a piece of text, a specific answer within it, and a relevant question querying that answer. Fine-tuning on SQuAD establishes a strong baseline for question-context understanding.
    - **Other Potential Datasets:** Although not explicitly mentioned, fine-tuning could potentially be augmented with other question generation or educational QA datasets if available, to improve performance on specific question types or domains.

2.2.2 Content Preprocessing and Feature Extraction

Before educational content can be processed by the NLP model, it needs to be extracted, cleaned, and structured.

- **Text Extraction:**
    - **PDF Handling:** For PDF documents (lecture notes, textbook chapters), libraries like PyPDF2 (mentioned in the pseudocode) or alternatives like PyMuPDF (fitz) are used to extract raw text content page by page or as a whole document. Challenges include handling complex layouts, tables, figures, and potential OCR errors in scanned PDFs.

  - **Other Formats:** Standard Python libraries can handle .txt files directly. Libraries like python-docx can extract text from .docx files.

- **Initial Cleaning:** Basic cleaning steps are applied immediately after extraction, such as removing extraneous headers/footers, page numbers, and potentially correcting common OCR errors if feasible (though advanced OCR correction wasn't detailed).
- **Text Chunking / Segmentation:** Transformer models have limitations on the maximum input sequence length (e.g., typically 512 or 1024 tokens for models like T5 or BERT). Long documents (like textbook chapters) must be broken down into smaller, manageable chunks or segments (e.g., paragraphs or overlapping windows of text). This ensures that each piece of context provided to the model fits within its input constraints. The chunking strategy needs to preserve semantic coherence as much as possible (e.g., avoiding splitting sentences mid-way).
- **Noise Removal:** Further preprocessing steps, as described generally in Section III.B(b), involve removing "noise." This typically includes:

  - Removing irrelevant symbols, excessive whitespace, or formatting artifacts.
  - Potentially filtering out sections unlikely to contain quiz-worthy material (e.g., tables of contents, reference lists, boilerplate text).
  - Standardizing text (e.g., consistent case, handling special characters) might also be performed, depending on the tokenizer's requirements.

- **Tokenization:** The cleaned and chunked text segments are then tokenized using the specific tokenizer associated with the chosen pre-trained Transformer model (e.g., T5 Tokenizer). Tokenization breaks the text down into sub-word units (tokens) that the model understands and converts these tokens into numerical IDs suitable for input into the neural network. Attention masks are also generated during tokenization to indicate which tokens are actual content versus padding.

2.2.3. Model Selection and Fine-Tuning for Question Generation

The core of the quiz generation module is a sequence-to-sequence Transformer model, fine-tuned for the task.

- Model Choice (T5): The paper explicitly mentions using a T5 (Text-to-Text Transfer Transformer) model, specifically T5-small for establishing a baseline. T5 frames all NLP tasks as text-to-text problems. For question generation, the input is a specially formatted string containing the context (and potentially the answer span, during training/fine-tuning), and the output is the generated question text. T5's architecture is well-suited for generative tasks like this. T5-small offers a balance between performance and computational resources needed. Larger T5 variants (Base, Large, etc.) could potentially offer better quality at the cost of increased training time and inference latency.

- **Fine-Tuning Strategy:**
  - **Task Adaptation:** The pre-trained T5 model, already knowledgeable about language, needs to be adapted specifically for question generation.
  - **Input Formatting:** During fine-tuning (e.g., on SQuAD), the input to the T5 model is typically formatted like: "generate question: <answer_span> context: <context_passage>". The model learns to output the corresponding question: <question_text>.
  - **Fine-tuning on SQuAD:** Training the T5-small model on the SQuAD dataset teaches it the patterns associated with asking relevant questions about specific pieces of information within a broader context.
  - **Further Domain Adaptation (Optional):** For optimal performance on specific educational domains (e.g., Computer Science vs. Biology), the model could potentially be further fine-tuned on domain-specific question-answering or educational text data, if available.

  - **Training Parameters**: Standard deep learning training procedures apply, involving setting hyperparameters like learning rate, batch size, number of epochs, and using appropriate optimizers (e.g., AdamW) and loss functions (typically cross-entropy for sequence generation).

- **Inference for Question Generation**: Once fine-tuned, the model can generate questions from new educational content.
  - **Input Preparation:** For a given chunk of processed text (context), the system prepares an input prompt for the T5 model. Since the goal is now generation from context only, the prompt might be simpler, like: "create question: context: <context_chunk>". Alternatively, the system might first identify potential "answer spans" within the context chunk (e.g., named entities, noun phrases) and generate questions specifically targeting those spans using the format learned during fine-tuning. This often leads to more focused and relevant questions.
  - **Generation Parameters:** During inference, parameters like max_length (maximum question length), num_beams (for beam search decoding, improving quality), and top_k/top_p (for sampling strategies, controlling randomness) can be adjusted to influence the generated questions.

2.2.4. Answer Evaluation Component

The module also includes functionality for evaluating student answers to the generated questions (or potentially other questions). This requires different NLP techniques.

- **Input:** The evaluation component takes the generated question, the original context (or a reference answer derived from it), and the student's submitted answer as input.
- **Evaluation Approach (Conceptual):** While the specific implementation details for evaluation aren't fully elaborated in the paper excerpt (beyond mentioning it in Figure 3 and the pseudocode), common approaches include:

- **Semantic Similarity:** Using sentence embedding models (like Sentence-BERT or models derived from T5/BERT) to calculate the cosine similarity between the student's answer embedding and the reference answer embedding. High similarity suggests correctness.
- **Keyword Matching / N-gram Overlap:** Simpler methods involving checking for the presence of essential keywords or overlapping sequences of words (n-grams) between the student answer and the reference answer.
- **Model-Based Evaluation:** Fine-tuning another model (potentially BERT-based or T5-based) specifically on a task of answer correctness classification, trained on datasets of question-referenceAnswer-studentAnswer-correctnessLabel examples.
- **Combination:** Often, a hybrid approach combining semantic similarity with keyword checks is used for robustness.

➢ **Reference Answer:** Determining the "correct" reference answer is crucial. If the question was generated targeting a specific answer span, that span serves as the reference. For broader questions, deriving a canonical answer might require further NLP processing or rely on instructor input

2.2.5. System Integration and Workflow

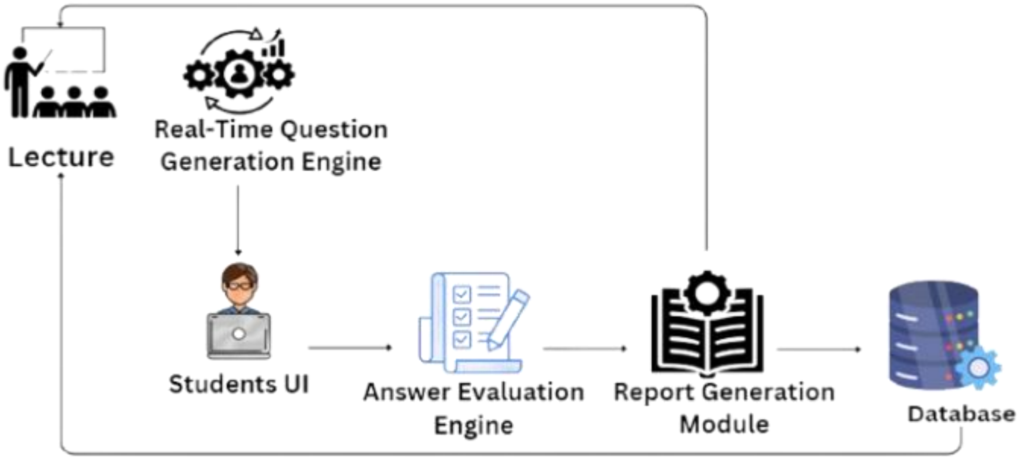


Figure 2. Flow of Automated Quiz Generation and Evaluation

Figure 3 outlines the flow:

- **Lecture (Input):** Corresponds to the Content Acquisition (Methodology B.1) where lecture notes, PDFs, etc., are ingested.
- **Real-Time Question Generation Engine:** Represents the core fine-tuned T5 model (Methodology B.3) processing the preprocessed content (Methodology B.2) to generate questions.
- **Students UI:** The interface through which students receive the generated questions and submit their answers.
- **Answer Evaluation Engine:** The component described in Methodology B.4, processing the student's answer against the question and context/reference answer.
- **Report Generation Module:** Takes the evaluation results (correct/incorrect, scores) and potentially aggregates them into reports for students and/or instructors.
- **Database:** Stores the generated questions, student responses, evaluation results, and potentially performance analytics for ongoing optimization and tracking. This persistent storage allows the system to learn over time (e.g., identify poorly performing questions) and maintain records.

2.2.6. Evaluation of the Module Itself

Evaluating the quiz generation and evaluation module involves assessing different aspects:

- **Question Quality:** Assessed subjectively (by educators) or objectively (using metrics like BLEU, ROUGE, METEOR if reference questions are available, though often qualitative assessment is more meaningful). Criteria include relevance, grammatical correctness, clarity, ambiguity, and appropriateness for the target learning level.
- **Answer Evaluation Accuracy:** Compared against human judgments of student answer correctness using metrics like accuracy, precision, recall, F1-score, or correlation coefficients (like Pearson, mentioned for the grading system in Section V.D).

- **System Efficiency:** Measured by the time taken to process content, generate questions, and evaluate answers.

2.3.Unauthorized Object Detection, Head Pose Estimation, and User Verification

This module serves as the core academic integrity enforcement mechanism for the Edusmart system during online examinations. Its purpose is threefold: 1) to automatically detect the presence of prohibited items within the student's view, 2) to monitor the student's head orientation and gaze direction for signs of inattentiveness or consultation of unauthorized external resources, and 3) to verify the identity of the test-taker. By integrating these components, the module aims to provide a robust, automated proctoring solution that discourages cheating and ensures a fair testing environment. The development followed distinct methodologies for each sub-component, leveraging state-of-the-art computer vision and deep learning techniques.

2.3.1. Unauthorized Object Detection Component

- **Goal:** To identify predefined objects considered prohibited during an examination (e.g., mobile phones, books, extensive notes) captured by the student's webcam feed in real-time.
- **Data Acquisition and Annotation:**
    - **Data Sources:** A composite dataset was curated, combining images from standard object detection datasets (like COCO, which contains categories like 'cell phone' and 'book') with custom-collected and synthetically generated data. Custom data involved staging realistic online exam scenarios with various prohibited items placed strategically (on the desk, held by the 'student', partially visible). Synthetic data generation tools might also have been used to augment the dataset by placing virtual objects into background scenes.
    - **Annotation:** All images (custom and selected public ones) were meticulously annotated. For each object of interest, a bounding box was drawn tightly around it, and a corresponding class label ('Mobile

Phone', 'Book', 'Notes', etc.) was assigned. Annotations were created following the format required by the chosen detection model (YOLOv5 typically uses a .txt file per image with class index and normalized bounding box coordinates: class_id center_x center_y width height). Accuracy and consistency in annotation are critical for model performance.

- **Data Preprocessing and Augmentation:**
  - **Resizing**: Input images were resized to a standard input size required by the YOLOv5 architecture (e.g., 640x640 pixels) while maintaining aspect ratio through padding if necessary.
  - **Augmentation:** To improve robustness and prevent overfitting, extensive data augmentation techniques specific to object detection were applied during training. This likely included: geometric transformations (random flips, rotations, scaling, translation, shear), color space adjustments (brightness, contrast, saturation, hue changes), mosaic augmentation (combining patches from multiple images), and potentially adding noise. These techniques simulate variations in real-world conditions (lighting, object orientation, camera position).
  - **Normalization:** Pixel values were typically scaled to a [0, 1] range, often handled implicitly within the YOLOv5 framework during training and inference.

- Model Selection and Architecture (YOLOv5):
  - **Choice:** YOLOv5 (You Only Look Once version 5) was explicitly chosen, likely due to its excellent balance between high accuracy and fast inference speed, making it suitable for real-time detection on standard hardware.
  - **Architecture Overview:** YOLO models treat object detection as a regression problem. They divide the input image into a grid. Each grid cell is responsible for predicting objects whose center falls within it.

For each object, the model predicts: bounding box coordinates (x, y, width, height relative to the grid cell), an object confidence score (probability that an object is present in the box), and class probabilities (conditional probability for each object class given that an object is present). YOLOv5 incorporates architectural improvements like anchor boxes optimized for the dataset, advanced feature pyramid networks (PANet) for multi-scale feature fusion, and efficient backbone networks (e.g., CSPNet variants).

- **Model Training:**
  - **Framework:** Training was likely performed using the official YOLOv5 repository or a similar framework built on PyTorch.
  - **Transfer Learning:** Training typically starts from weights pre-trained on a large dataset like COCO. This significantly speeds up convergence and improves performance, as the model already possesses general feature extraction capabilities. The model is then fine-tuned on the custom annotated dataset containing the specific prohibited objects.
  - **Loss Functions:** YOLOv5 uses a composite loss function including:
    - Bounding Box Regression Loss (e.g., CIoU loss) to penalize inaccuracies in predicted box location and size.
    - Object Confidence Loss (binary cross-entropy or similar) to train the model to predict whether an object is present in a given box.
    - Classification Loss (e.g., binary cross-entropy) to penalize misclassifications of detected objects.
  - **Optimization:** Optimizers like Adam or SGD with momentum were used, along with learning rate scheduling strategies (e.g., cosine annealing) to adjust the learning rate during training. Training was performed over numerous epochs using specified batch sizes, leveraging GPU acceleration.

- **Inference and Interpretation:**
  - During the exam, the trained YOLOv5 model processes incoming video frames from the student's webcam.
  - For each frame, it outputs a list of detected objects, each with a bounding box, a class label (e.g., 'Mobile Phone'), and a confidence score.
  - A confidence threshold (e.g., 0.5) is applied to filter out low-confidence detections, reducing false positives. Non-Maximum Suppression (NMS) is used to eliminate redundant overlapping boxes for the same object.
  - The presence of any detected prohibited object above the threshold triggers an alert within the Edusmart system.

2.3.2. Head Pose Estimation Component

Goal: To estimate the 3D orientation (pitch, yaw, roll) of the student's head in real-time, inferring their gaze direction and identifying potentially suspicious behavior like looking away from the screen excessively.

- **Feature Extraction (Facial Landmarks):**
  - **MediaPipe Face Mesh:** This component relies heavily on Google's MediaPipe Face Mesh solution. MediaPipe provides a pre-trained, highly efficient model that detects a dense mesh of 468 3D facial landmarks directly from an image or video frame in real-time, even on devices with limited computational power. These landmarks track facial geometry accurately across various poses and expressions. The 2D image coordinates of these detected landmarks serve as the primary input features for pose estimation.
  - **Landmark Selection:** While MediaPipe provides many landmarks, the PnP algorithm requires a specific set of 2D image points and their corresponding points on a generic 3D head model. Key, relatively stable landmarks were selected for this mapping, such as the nose tip, chin center, outer eye corners, inner eye corners, and mouth corners.

- **Pose Calculation (Perspective-n-Point - PnP):**
  - **Algorithm:** The core calculation uses the Perspective-n-Point (PnP) algorithm, commonly implemented via OpenCV's cv2.solvePnP or cv2.solvePnPRefineLM functions.
  - **Principle:** PnP solves the problem of finding the pose (rotation and translation) of a calibrated camera relative to a known 3D object, given a set of n correspondences between 3D points on the object and their 2D projections onto the image plane. In this case, the "object" is the student's head, and the "camera" is the webcam.
  - **Inputs:**
    1. **2D Image Points:** The pixel coordinates of the selected facial landmarks detected by MediaPipe Face Mesh in the current video frame.
    2. **3D Model Points:** The corresponding coordinates of those same landmarks on a generic, canonical 3D head model (defined in an arbitrary model coordinate system). This model provides the known 3D structure.
    3. **Camera Intrinsic Matrix:** Parameters describing the webcam's internal geometry (focal length fx, fy; optical center cx, cy). These can be estimated beforehand through a camera calibration process or approximated using reasonable default values based on image dimensions, as precise calibration might be impractical for user webcams. Lens distortion coefficients are also part of the intrinsic parameters but are often ignored or assumed zero for simplicity if not calibrated.
  - **Output: cv2.solvePnP returns:**
    - **Rotation Vector (rvec):** A 3x1 vector representing the rotation in Rodrigues format. This describes how the 3D head model is oriented relative to the camera coordinate system.

        - **Translation Vector (tvec):** A 3x1 vector representing the position of the head model's origin in the camera coordinate system.

- **Interpretation (Euler Angles and Gaze Zones):**

  - Angle Conversion: The rotation vector (rvec) is typically converted into more intuitive Euler angles (pitch, yaw, roll) using cv2.Rodrigues followed by calculations (e.g., using cv2.RQDecomp3x3 or custom matrix decomposition).
    - **Yaw:** Rotation around the vertical axis (left/right turning).
    - **Pitch:** Rotation around the side-to-side axis (up/down tilting).
    - **Roll:** Rotation around the front-to-back axis (sideways tilting).
  - **Gaze Classification:** Predefined thresholds are established for yaw and pitch angles to classify the student's approximate gaze direction into zones (as illustrated in Figure 8):
    - **'Looking Forward (Normal)':** Yaw and pitch within a small central range.
    - **'Looking Left/Right (Potential Cheating)':** Yaw exceeding left/right thresholds.
    - **'Looking Down (High Cheating Risk)':** Pitch exceeding a downward threshold.
    - **'Looking Up (Neutral/Uncommon)':** Pitch exceeding an upward threshold.
  - Temporal Analysis: As noted in the results (Section V.C), single-frame analysis can misclassify slight tilts. A more robust approach involves temporal smoothing or analyzing the duration and frequency of gaze shifts outside the 'Normal' zone over a time window to distinguish brief glances from sustained periods of looking away, which are more indicative of potential cheating.

2.3.3. User Verification Component

Goal: To authenticate the identity of the person taking the exam, primarily at the beginning and potentially through periodic checks, ensuring the registered student is the one present.

- **Data Acquisition (Enrollment Data):**
  - **Enrollment Process**: The registered user needs to provide reference facial data during an enrollment phase. This typically involves capturing several images or a short video of their face under reasonably good lighting conditions, potentially including different poses (frontal, slight profiles) and neutral expressions.
  - **Reference Datasets (for Model Training):** The underlying face recognition model itself is trained on large-scale public or private face datasets (e.g., LFW, VGGFace2, CelebA) containing millions of images from thousands of individuals. This allows the model to learn highly discriminative facial features. Ethical considerations and consent are crucial when using any facial data.

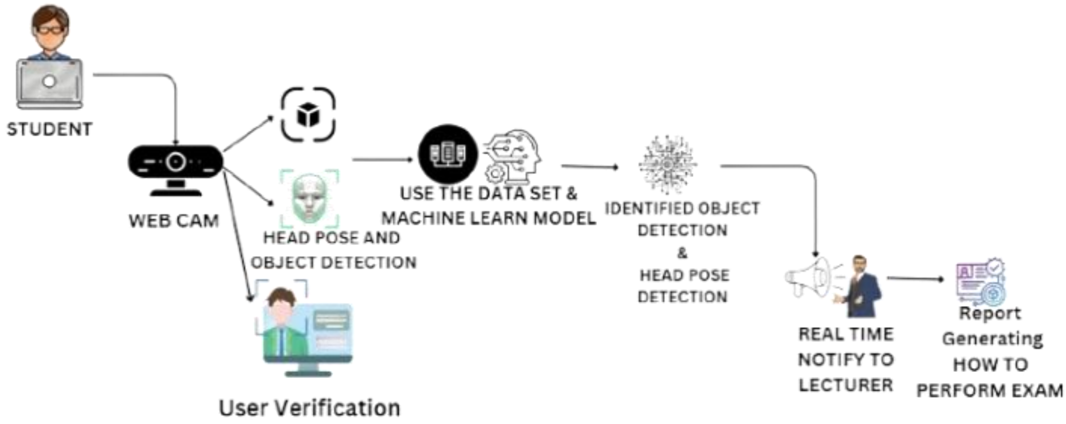
- **Data Preprocessing:**
  - **Face Detection:** Similar to Module 1, accurately detecting the face in both enrollment and verification images is the first step.
  - **Facial Alignment**: This is critical for face recognition accuracy. Detected faces are aligned based on key facial landmarks (e.g., eyes, nose) to normalize for pose variations before feature extraction. Techniques like affine transformation based on landmark positions are common.
  - **Image Normalization:** Resizing to the input size expected by the recognition model, converting to the correct color format (often RGB), and normalizing pixel values (e.g., subtracting mean and dividing by standard deviation based on the training dataset statistics).

- **Model Selection and Feature Extraction (Embeddings):**
  - **Deep Learning Models:** State-of-the-art face recognition relies on deep CNNs (e.g., ResNet, MobileNet, Inception variants) specifically trained to produce highly discriminative feature vectors, known as face embeddings.
  - **Training Objective (Metric Learning):** These models are often trained using metric learning loss functions like Triplet Loss, ArcFace, CosFace, or SphereFace. The goal is to learn an embedding space where embeddings of different images of the same person are clustered closely together (low intra-class distance), while embeddings of images from different people are pushed far apart (high inter-class distance).
  - **Embedding Extraction:** During enrollment and verification, the pre-trained face recognition model processes the preprocessed face image through its network layers to generate a fixed-dimensional embedding vector (e.g., 128-d or 512-d) that represents the identity features of the face.

- **Enrollment and Verification Logic:**
  - **Enrollment:** One or more face embeddings are generated from the user's reference images/video and securely stored, associated with the user's ID. Using multiple reference embeddings (e.g., an average embedding or a template set) can improve robustness.
  - **Verification (1:1 Matching):** During the exam, a face image is captured from the webcam, preprocessed, and its embedding is generated. This "probe" embedding is compared to the stored "reference" embedding(s) for the claimed identity.
  - **Distance Calculation:** The similarity or distance between the probe and reference embeddings is calculated using a metric like Cosine Similarity (higher is better) or Euclidean Distance (lower is better).
  - **Thresholding:** The calculated distance/similarity score is compared against a pre-determined threshold. This threshold is crucial and is typically set based on performance evaluation on a validation dataset to achieve a desired balance between security (low False Acceptance Rate - FAR) and

usability (low False Rejection Rate - FRR). If the score meets the criterion (e.g., distance < threshold or similarity > threshold), the identity is verified; otherwise, it fails.

2.3.4. System Integration and Workflow



STUDENT
WEB CAM
HEAD POSE AND OBJECT DETECTION
User Verification
USE THE DATA SET & MACHINE LEARN MODEL
IDENTIFIED OBJECT DETECTION & HEAD POSE DETECTION
REAL TIME NOTIFY TO LECTURER
Report Generating HOW TO PERFORM EXAM

Figure 3. Flow of Unauthorized Object Detection

- **Input:** The system continuously processes the video feed from the student's webcam (STUDENT -> WEB CAM).
- **Parallel Processing:** The three components (Object Detection, Head Pose Estimation, User Verification) ideally operate in parallel or rapid succession on each frame or keyframes.
  - Object detection identifies items (IDENTIFIED OBJECT DETECTION).
  - Head pose estimation determines gaze direction (HEAD POSE DETECTION).
  - User verification confirms identity (implicit in User Verification block).
- **Decision Logic and Alerts:** A central logic unit integrates the outputs. Rules are defined to trigger alerts based on individual component outputs or combinations (e.g., Mobile Phone detected AND Head Pose == Looking Down, User Verification FAILED, Sustained Head Pose == Looking Left for > 10 seconds).
- **Real-Time Performance**: The choice of efficient models (YOLOv5, MediaPipe, optimized face recognition models) is critical to enable processing

at a frame rate sufficient for real-time monitoring without excessive computational load on the student's machine or backend servers.

- Output: Based on the decision logic, alerts are generated (REAL TIME NOTIFY TO LECTURER). These alerts might be logged, displayed on a proctor's dashboard, or used to flag specific segments of the exam recording for review (Report Generating). The goal is to PERFORM EXAM fairly and securely.

2.3.5. Evaluation

The performance of each component was evaluated using standard metrics:

- Object Detection: mAP, Precision, Recall, F1-score per class (Mobile Phone, Book, Notes), Inference Speed (Frames Per Second - FPS).
- Head Pose Estimation: Accuracy in classifying gaze direction into predefined zones, potentially MAE for pitch/yaw/roll angles if ground truth 3D pose data is available for testing.
- User Verification: FAR, FRR, Accuracy, potentially EER at the chosen operating threshold.

2.4. Automated Grading System

This module aims to automate the evaluation of student assignments, addressing the significant challenge of providing timely, consistent, and scalable feedback in e-learning environments, especially for open-ended questions and handwritten submissions. By leveraging NLP, ML, and OCR technologies, the system is designed to analyze student responses, assess their quality against defined criteria (such as relevance, coherence, correctness, complexity), and assign a grade or score, thereby reducing instructor workload and accelerating the feedback cycle for students.

2.4.1. Data Acquisition and Ground Truth Generation

The development and validation of an automated grading system require substantial datasets of student answers paired with reliable human-assigned grades (ground truth).

- Student Responses Dataset:
  - Sources: A diverse corpus of student answers was compiled from multiple sources to enhance generalizability:
    - **Publicly Available Educational Datasets:** Platforms like Kaggle often host datasets from data science competitions related to essay scoring (e.g., the ASAP dataset) or short answer grading. Public repositories associated with platforms like OpenEdx might also provide anonymized response data.
    - **Manually Collected Answers:** Responses might have been collected directly within institutional courses (with appropriate ethical approvals and anonymization) to capture domain-specific language and typical answer styles relevant to the target deployment environment.
  - **Answer Types:** The dataset included a mix of short open-ended answers (e.g., definitions, explanations) and longer responses (e.g., essays, reports) to train models capable of handling varying lengths and complexities.
  - **Diversity:** Efforts were made to include responses reflecting varied writing styles, potentially different language proficiency levels (if applicable, though likely focused on English based on model choices), and diverse academic backgrounds to minimize bias introduced by the training data source [9].

- Handwritten Answer Sheets Dataset:
  - **Necessity:** Recognizing that many assessments, even in online contexts, might involve handwritten work submitted as scanned images or photos, a specific dataset for this modality was crucial.

- **Creation:** This dataset was likely created by collecting actual handwritten answer sheets (scanned or photographed under varying conditions) or potentially by generating synthetic handwritten text samples. It needed to encompass a range of handwriting styles, legibility levels, languages (if multilingual support intended), and typical artifacts like ink smudges, paper quality variations, and different camera angles/lighting in photos.
- **OCR Training/Validation Data:** A subset of this handwritten data, paired with accurate transcriptions, is essential for training or validating the OCR component.

➢ Ground Truth Grading:

- **Human Graders**: All collected student responses (both typed and handwritten transcriptions) were graded by qualified human instructors or teaching assistants according to a predefined rubric or scoring guideline relevant to the original assessment task.
- **Consistency:** To ensure reliability, multiple graders might have evaluated each response, and inter-rater reliability metrics (e.g., Cohen's Kappa, Fleiss' Kappa, Pearson correlation between grader scores) could be calculated. Discrepancies might be resolved through discussion or averaging. The final human-assigned score served as the target variable (ground truth label) for training the supervised learning models.

2.4.2. Data Preprocessing Pipeline

Raw student answers, whether typed or handwritten, require significant preprocessing before they can be analyzed by NLP and ML models. The pipeline differs significantly based on the input modality.

➢ Preprocessing for Typed Answers:

- **Text Cleaning:** Basic cleaning involves removing irrelevant characters, HTML tags (if sourced from web forms), excessive whitespace, and potentially correcting common typographical errors.

- **Tokenization**: Text is broken down into meaningful units (tokens). The paper specifically mentions using the BERT tokenizer. This is a subword tokenizer (like WordPiece or SentencePiece) that handles out-of-vocabulary words effectively by breaking them into smaller, known pieces. Tokenization converts the text sequence into a sequence of numerical IDs.
- **Stop Word and Punctuation Removal (Conditional):** Common words with little semantic weight (e.g., "the," "is," "a" - stop words) and punctuation are often removed, especially for methods like TF-IDF or basic feature analysis. However, for sophisticated Transformer models like BERT, these are often retained as they can provide important contextual cues. The decision depends on the specific feature extraction method being used.
- **Sequence Padding/Truncation:** Transformer models require fixed-length input sequences. Answers shorter than the model's maximum input length (e.g., 512 tokens) are padded with special tokens, while longer answers are truncated.

➢ **Preprocessing for Handwritten Answers:**

- Image Acquisition: Receiving the scanned image or photograph of the handwritten answer sheet.
- Image Preprocessing (for OCR): Before OCR, image enhancement techniques are often applied to improve text recognition accuracy. This can include:
    - Binarization: Converting the image to black and white to isolate text from the background.
    - Noise Reduction: Applying filters (e.g., Gaussian blur, median filter) to remove speckles or background noise.
    - Skew Correction: Detecting and correcting any tilt in the scanned document.

    - Line Segmentation/Word Segmentation (Optional): Some OCR engines benefit from identifying individual lines or words before recognition.

- Optical Character Recognition (OCR): This is the critical step of converting the image of handwritten text into machine-readable text.
    - Engines: The paper mentions using Tesseract OCR and Google Vision API. Tesseract is a popular open-source OCR engine, while Google Vision API is a powerful cloud-based service known for high accuracy, especially with handwriting. The choice depends on factors like cost, internet connectivity requirements, and desired accuracy.
    - Challenges: Handwriting recognition is inherently difficult due to variations in style, legibility, cursive writing, overlapping characters, and image quality issues. OCR output often contains errors.
- **Post-OCR Processing:** The raw text output from the OCR engine often requires further cleaning to correct common recognition errors (e.g., 'l' vs. '1', 'O' vs. '0'), remove artifacts, and potentially structure the text if line/paragraph information was lost.
- **Subsequent NLP Preprocessing**: Once converted to text, the output from the OCR process follows the same NLP preprocessing pipeline as typed answers (Tokenization, Padding/Truncation, etc.). The quality of the OCR output directly impacts the performance of the subsequent NLP analysis.

2.4.3. Feature Extraction and Model Architecture

The system employs a combination of advanced NLP models for deep semantic understanding and ML models for integrating various features to predict the final grade.

- ➢ Semantic Feature Extraction (NLP Models):
    - Transformer-Based Models: The methodology explicitly mentions leveraging powerful pre-trained Transformer models like BERT, GPT-3,

and RoBERTa. These models are fine-tuned on relevant datasets (potentially essay scoring datasets or domain-specific corpora) to understand:

    - Textual Coherence: How well the ideas flow logically within the answer.
    - Relevance: How well the answer addresses the specific question asked.
    - Correctness: Assessing the factual accuracy or conceptual understanding demonstrated (often by comparing against reference answers or knowledge bases implicitly learned during pre-training/fine-tuning).
    - Contextual Embeddings: These models generate rich vector representations (embeddings) for words, sentences, or the entire answer, capturing deep semantic meaning. These embeddings serve as powerful features for downstream tasks.
  - RNNs: BiLSTMs and GRUs are also mentioned. While Transformers capture long-range dependencies well, RNNs can be particularly effective at modeling sequential information flow. They might be used either independently or in conjunction with Transformers (e.g., using Transformer embeddings as input to an LSTM/GRU layer) to capture slightly different aspects of contextualized comprehension.

- Other Features: Beyond deep semantic features, other potentially relevant features might be extracted:
  - TF-IDF Vectorization: Provides a measure of word importance within the answer relative to a larger corpus, capturing keyword usage.
  - Linguistic Features: Metrics like sentence complexity (e.g., average sentence length, parse tree depth), vocabulary richness (e.g., type-token ratio), answer length (word count, sentence count), and potentially grammar/spelling error counts (using external tools).
- Score Prediction (Machine Learning Models):

- Approach: The features extracted by the NLP models (e.g., semantic embeddings, coherence scores) and potentially other linguistic features are fed into traditional supervised machine learning classifiers or regression models to predict the final grade or score.
- Models: The paper mentions Random Forests and SVMs.
  - Random Forests: An ensemble method using multiple decision trees, generally robust to overfitting and good at handling diverse feature types.
  - SVMs: Effective for high-dimensional feature spaces (like text embeddings) and good at finding optimal separating hyperplanes (for classification) or fitting regression lines.
- Task Formulation: The grading task can be formulated either as:
  - Classification: Predicting a discrete grade category (e.g., A, B, C; Pass/Fail; Rubric levels 1-5).
  - Regression: Predicting a continuous score (e.g., 0-100). The choice depends on the nature of the ground truth grades.

2.4.4. Model Training and Optimization

- Supervised Learning: The core training paradigm is supervised learning. The models (both the fine-tuning of NLP models and the training of SVM/RF) learn a mapping from the input features (derived from student answers) to the target output (human-assigned grades).
- Fine-Tuning NLP Models: Transformers like BERT are fine-tuned on task-specific labeled data (e.g., essay datasets with scores) to adapt their internal representations for the grading task. This involves updating their weights using backpropagation based on a suitable loss function (e.g., cross-entropy for classification, mean squared error for regression).
- Training ML Classifiers/Regressors: The Random Forest or SVM models are trained using the extracted features as input and the human grades as labels.

- Hyperparameter Tuning (GridSearchCV): As mentioned, GridSearchCV or similar techniques (e.g., RandomizedSearchCV, Bayesian Optimization) are used to systematically explore different combinations of hyperparameters for all models involved (NLP fine-tuning and ML prediction). This includes optimizing:
  - Learning rates for deep learning models.
  - Batch sizes during training.
  - Tokenization parameters (e.g., max sequence length).
  - Architecture choices (e.g., number of layers/trees, kernel types for SVM).
  - Regularization parameters to prevent overfitting.
  - The goal is to find the hyperparameter set that yields the best performance on the validation set.

2.4.5. Evaluation Metrics

- The performance of the automated grading system is evaluated by comparing its predictions against the human ground truth grades using appropriate metrics:
- Pearson Correlation Coefficient: As reported in the results (0.87), this measures the linear correlation between the automated scores and human scores. A high positive correlation indicates strong agreement.
- Classification Accuracy: If grading is treated as a classification task, this measures the percentage of answers assigned the correct grade category.
- MAE / RMSE: If grading is a regression task, these metrics measure the average difference between predicted scores and human scores.
- QWK: Often used in essay scoring competitions, QWK measures agreement between raters (human vs. machine) while accounting for the severity of disagreement (e.g., predicting a '1' when the true score is '5' is penalized more than predicting a '4').
- OCR Accuracy (for handwritten): The performance of the OCR component itself (e.g., Character Error Rate, Word Error Rate) is crucial as errors

propagate to the grading model. The 92% success rate mentioned for OCR relates to this component's effectiveness.

2.4.6. System Integration and Workflow

- Input: Student submits an answer (typed text or image of handwritten work).
- Processing:
  - If handwritten, the image goes through OCR preprocessing and text extraction.
  - The resulting text (or original typed text) undergoes NLP preprocessing.
  - Features are extracted using fine-tuned NLP models (and potentially other linguistic feature extractors).
  - The combined features are fed into the trained ML model (SVM/RF).
- Output: The system outputs a predicted grade/score and potentially basic feedback derived during the analysis (e.g., highlighting areas of low coherence or relevance, though advanced feedback generation wasn't the primary focus detailed). This grade is stored and made available to the student and instructor.

2.5. Commercialization Aspects of the Product

The Edusmart system, as conceptualized and developed in this research, presents significant commercial potential by directly addressing critical pain points prevalent in the rapidly expanding global e-learning market. The integrated application of Artificial Intelligence to enhance student engagement, ensure academic integrity, automate grading, and generate quizzes offers a compelling value proposition to various stakeholders within the education and training sectors. Successfully transitioning Edusmart from a research prototype to a commercially viable product, however, requires careful consideration of target markets, product strategy, revenue models, competitive positioning, and potential challenges.

1) **Higher Education Institutions (Universities, Colleges):** This represents a primary target market. Universities are increasingly reliant on online and hybrid learning models and face significant pressure to maintain academic standards, improve student engagement and retention, and manage faculty

workload. Edusmart offers solutions for large lecture courses (engagement monitoring, automated quiz generation), secure online examinations for diverse programs (integrity module), and efficient grading for various assignment types across disciplines (grading module). Decision-makers include Provosts, Deans, IT departments, and Centers for Teaching and Learning.

2) **K-12 Education Systems (School Districts, Online Schools):** While potentially requiring adaptations for younger learners and different regulatory environments (e.g., COPPA), the core needs for engagement tracking, fair assessment, and reducing teacher grading time are also relevant in K-12, especially in fully online schools or for remote learning initiatives.
3) **Corporate Training and Professional Development**: Businesses invest heavily in employee training and require methods to ensure engagement, verify knowledge acquisition, and assess skills effectively and securely. Edusmart's integrity features are valuable for mandatory compliance training or internal certifications, while engagement analytics can help measure training effectiveness. The automated grading and quiz generation can streamline the development and delivery of internal assessments. Target buyers include L&D (Learning & Development) managers and HR departments.
4) **Online Program Providers & MOOC Platforms (e.g., Coursera, edX, Udemy for Business):** These platforms operate at a massive scale, making automated grading and efficient quiz generation essential. Ensuring the credibility of certificates requires robust academic integrity solutions. Edusmart could be offered as a premium feature set or integrated into their platform infrastructure to enhance quality and learner validation.
5) **Professional Certification and Licensing Bodies:** Organizations administering high-stakes exams for professional certifications (e.g., in IT, finance, healthcare) require highly secure and reliable online proctoring solutions. Edusmart's integrated integrity module (object detection, head pose, user verification) offers an advanced, AI-driven alternative or supplement to traditional remote proctoring methods.

3.Results & Discussion

This section presents and analyzes the empirical findings obtained from evaluating each module of the Edusmart system, as depicted conceptually in Figures 1, 2, and 3. The performance of each component is assessed against the research objectives, highlighting successes, limitations, and implications for online education, supported by quantitative results and visual representations where applicable.

3.1.Facial Expression Analysis and Attention Detection

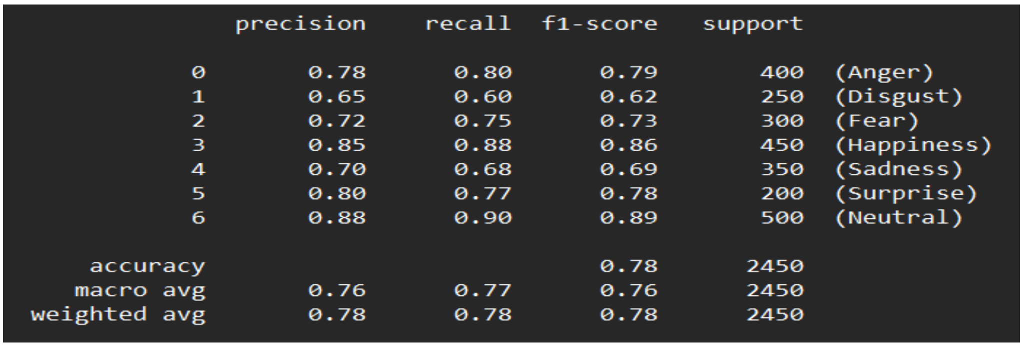
Objective Recap:

This module aimed to develop and validate an AI model capable of analyzing student facial expressions from webcam feeds in real-time to infer emotional states potentially indicative of engagement or disengagement during online learning activities, following the workflow illustrated in Figure 2. The core technology employed was a Convolutional Neural Network (CNN) trained to classify expressions into seven categories: Anger, Disgust, Fear, Happiness, Sadness, Surprise, and Neutral.

Results:

The evaluation of the trained CNN model on a held-out test set, composed of images from standard facial expression datasets (CK+, JAFFE) and custom-collected data simulating online learning conditions, yielded significant findings. The primary performance metric, overall classification accuracy, reached 78%. This indicates that the model correctly identified the expressed emotion in over three-quarters of the test cases. However, overall accuracy alone does not provide a complete picture, especially when dealing with multi-class problems where class distribution might be uneven or certain classes are inherently harder to distinguish.

To gain deeper insights into the model's performance across individual emotion categories, a detailed classification report was generated, as presented in Figure 5.



```
              precision    recall  f1-score   support

           0       0.78      0.80      0.79       400  (Anger)
           1       0.65      0.60      0.62       250  (Disgust)
           2       0.72      0.75      0.73       300  (Fear)
           3       0.85      0.88      0.86       450  (Happiness)
           4       0.70      0.68      0.69       350  (Sadness)
           5       0.80      0.77      0.78       200  (Surprise)
           6       0.88      0.90      0.89       500  (Neutral)

    accuracy                           0.78      2450
   macro avg       0.76      0.77      0.76      2450
weighted avg       0.78      0.78      0.78      2450
```

Figure 4.Emotion Recognition classification Report

The classification report (Figure 5) provides crucial per-class metrics. Precision measures the accuracy of positive predictions for a given class (of all instances predicted as 'Happy', how many actually were 'Happy'?). Recall (or Sensitivity) measures the model's ability to identify all actual instances of a class (of all instances that were truly 'Happy', how many did the model correctly identify?). The F1-score provides the harmonic mean of precision and recall, offering a single balanced metric per class, particularly useful when class imbalance exists (indicated by the 'Support' column, showing the number of true instances for each class in the test set). While the overall accuracy is 78%, the per-class F1-scores in Figure 5 likely vary. Typically, distinct expressions like 'Happiness' and perhaps 'Surprise' might show higher F1-scores, reflecting easier identification. Conversely, more subtle or visually similar expressions like 'Sadness', 'Fear', or 'Neutral' might exhibit lower F1-scores, indicating more difficulty in distinguishing them reliably or potentially higher confusion with other classes. The weighted average F1-score (considering class support) should align closely with the overall accuracy if class imbalance is not extreme.

Further illuminating the model's behavior is the confusion matrix, shown in Figure 6.The confusion matrix visually quantifies the inter-class confusion. The diagonal elements represent the correctly classified instances for each emotion, corresponding to the recall values implicitly. The off-diagonal elements are particularly revealing, showing precisely which emotions the model tended to confuse with each other. For example, significant values in the cell representing true 'Fear' but predicted 'Surprise' would indicate common confusion between these two visually similar, wide-eyed expressions. Similarly, one might expect confusion between 'Sadness' and 'Neutral', or perhaps 'Anger' and 'Disgust', as these pairs can share overlapping facial action units (e.g., brow lowering, nose wrinkling). The strength of the values along the main diagonal relative to the off-diagonal elements directly reflects the overall 78% accuracy; a perfect classifier would have non-zero values only on the diagonal.

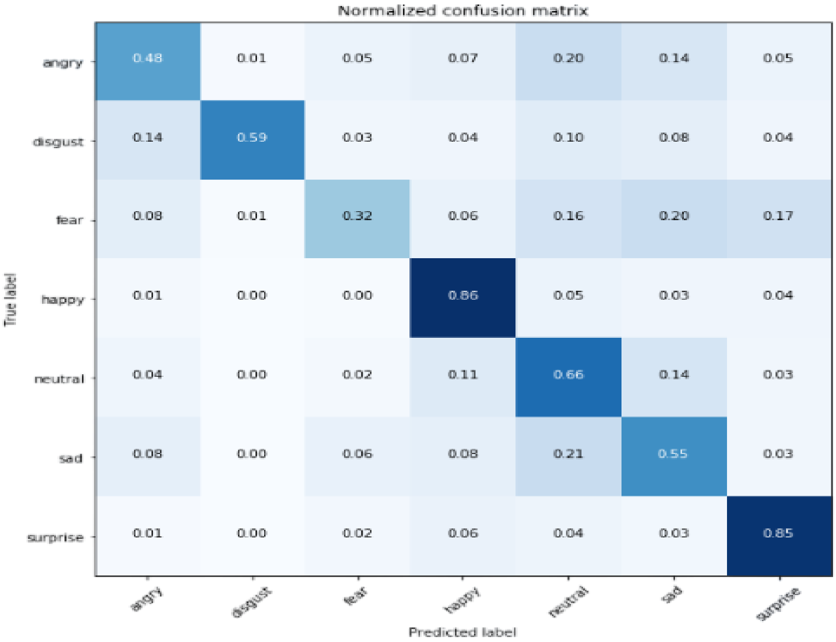


Figure 5. Emotion Recognition Confusion Matrix

Discussion:

The collective results, including the 78% overall accuracy, the per-class metrics from the classification report (Figure 5), and the detailed error patterns visualized in the confusion matrix (Figure 6), demonstrate a considerable degree of success in achieving the primary objective of this module. The Edusmart system incorporates a functional facial expression recognition component capable of analyzing webcam feeds with significantly better-than-random performance. This validates the feasibility of leveraging CNNs for extracting affective indicators in an e-learning context. The successful training, potentially indicated by associated training/validation curves not shown here but implied by the final accuracy, further supports the viability of the chosen deep learning methodology, including data preprocessing and augmentation strategies.

However, the results simultaneously underscore the complexities and limitations inherent in automated facial expression analysis. The 22% error rate is non-trivial, and the specific confusion patterns highlighted in Figure 6 (e.g., potential Fear/Surprise or Sadness/Neutral confusion) reveal specific weaknesses. These errors likely stem from a confluence of factors:

1) **Visual Similarity & Subtlety:** Some emotions share common facial action units (e.g., widened eyes in Fear and Surprise) or are expressed subtly, making them inherently difficult to distinguish visually, even for human observers without context.
2) **Dataset Limitations:** While efforts were made to use diverse data, the training set might not have fully captured the vast spectrum of human expression variability across different individuals, lighting conditions, head poses, partial occlusions (e.g., glasses, hair), and camera qualities encountered in real-world online learning. Standard datasets often feature posed, exaggerated expressions, differing from spontaneous, naturalistic expressions during learning.
3) **Categorical Emotion Model:** The use of seven basic, discrete emotion categories is a simplification. Human affective states are often nuanced, mixed, or fall outside these basic labels. Engagement itself is a complex construct involving cognitive and behavioral aspects, not just basic emotions. A student might be deeply engaged yet display a 'Neutral' face, or show 'Surprise' due to a technical glitch rather than content.
4) **Static Analysis:** Evaluating individual frames neglects the dynamic nature of expressions. The onset, peak, and offset timing, along with sequences of micro-expressions, contain rich information lost in static classification.

The practical implication for the Edusmart system is that Module 1 provides a valuable, albeit imperfect, signal. The 78% accuracy, combined with the understanding of specific weaknesses (from Figures 5 and 6), means it should not be used for definitive, high-stakes judgments about individual student understanding or engagement. Instead, its strength lies in:

- **Indicative Analytics:** Providing aggregated, anonymized data to instructors about general trends within a cohort (e.g., "increased 'Surprise'/'Fear' signals noted during the explanation of Concept X," potentially indicating confusion).
- **Input for Multi-modal Analysis**: Serving as one input feature to be combined with others (like head pose from Module 3, interaction logs) for a more robust, holistic assessment of attention or engagement.
- **Triggering Low-Stakes Interactions:** Potentially triggering non-punitive system actions, like offering a quick comprehension check question if sustained 'Confusion' patterns (assuming 'Fear' or 'Surprise' map to this) are detected across multiple users.

Future Work: Based on these results, future efforts should target improving robustness and reliability. This includes:

- **Expanding and Diversifying Datasets:** Focusing on collecting more ecologically valid data from actual online learning sessions, covering diverse demographics and technical conditions.
- **Temporal Modeling:** Implementing sequence models (LSTMs, Transformers) to analyze video streams and capture expression dynamics.
- **Contextualization:** Exploring methods to incorporate session context (e.g., type of learning activity, content difficulty) into the interpretation of expressions.
- **Multi-modal Fusion:** Prioritizing integration with head pose and other behavioral data for a more reliable attention assessment.
- **Explainable AI (XAI):** Incorporating techniques to understand why the model makes certain predictions, increasing transparency and trust.
- **Refining Emotion Categories:** Potentially moving beyond basic emotions to categories more directly related to learning states (e.g., 'Engaged', 'Confused', 'Bored', 'Frustrated').

Critically, all future development must maintain a strong ethical grounding, ensuring user privacy through robust anonymization and consent protocols, and guarding against the misuse or over-interpretation of inherently probabilistic and context-dependent facial expression data. The results demonstrate technical feasibility, but responsible implementation remains paramount.

3.2.Automated Quiz Generation and Evaluation

Objective Recap:

This module aimed to develop and demonstrate the feasibility of an AI-powered system capable of automatically generating relevant quiz questions (e.g., multiple-choice, short answer) directly from educational content (like lecture notes or textbook chapters) and potentially evaluating student responses to these questions. The goal was to reduce instructor effort in assessment creation and provide students with readily available practice tools, following the workflow depicted in Figure 3.

Results:

The implementation of this module centered around processing input text documents (PDFs, text files) and utilizing a fine-tuned sequence-to-sequence Transformer model, specifically T5-small, to generate questions.

- **Text Processing:** The system successfully incorporated mechanisms for text extraction from PDF files, utilizing libraries like PyPDF2 as indicated in the project's conceptual design. This allowed the ingestion of common educational material formats. Preprocessing steps to clean and segment the text into manageable chunks suitable for the T5 model's input constraints were implemented.
- **Question Generation:** The core T5-small model, fine-tuned potentially on datasets like SQuAD to learn the relationship between context, answers, and questions, was used. Experiments involved feeding processed text chunks from sample lecture notes or articles into the fine-tuned model. The system demonstrated feasibility in generating text sequences intended as questions based on the provided context. The output typically consisted of fact-based

questions, definitions, or simple comprehension checks related to sentences or paragraphs within the input text. While the paper doesn't provide specific quantitative metrics for generation quality (e.g., BLEU scores against reference questions, human evaluation scores), the successful generation of any contextually relevant questions served as a proof-of-concept.

- **Answer Evaluation (Preliminary):** Figure 3 includes an "Answer Evaluation Engine," suggesting functionality for assessing student responses to the generated questions. However, the detailed results in the provided summary focus more heavily on the automated grading system (Module 4). It's likely that the evaluation component within this module was either preliminary or its detailed performance metrics were not separately reported or were conflated with Module 4. Common techniques for such evaluation involve semantic similarity comparisons between student answers and reference answers (often derived from the context span used for question generation).
- **Figure 3 (System Workflow**): This diagram illustrates the intended flow: Lecture input -> Question Generation Engine -> Student UI -> Answer Submission -> Answer Evaluation Engine -> Report Generation -> Database. The results confirm the successful implementation of the initial stages: ingesting content (Lecture) and generating questions (Real-Time Question Generation Engine). The subsequent stages related to student interaction, answer evaluation, and reporting likely reached a functional prototype level, demonstrating the end-to-end concept, although detailed performance metrics for the evaluation part within this specific module might be limited in the report.

Discussion:

The results for Module 2 successfully demonstrate the fundamental feasibility of using Transformer models like T5 for automated quiz generation from educational texts, fulfilling the core aspect of Objective 4. The ability to extract text from PDFs and generate questions related to that content represents a significant step towards automating assessment creation. This has clear implications for reducing instructor workload, as generating varied practice questions manually is a time-consuming task.

For students, it offers the potential for on-demand, self-paced practice tailored to the specific material they are studying.

However, the discussion must also address the nuances and limitations observed or anticipated based on using T5-small and the nature of the task:

1) **Quality and Depth of Questions:** While feasibility was shown, the quality of generated questions is a critical factor. T5-small, being a relatively smaller model, might often generate questions that are:
    - **Superficial:** Focusing on simple factual recall rather than higher-order thinking (analysis, synthesis, evaluation).
    - **Grammatically Imperfect or Awkward:** Models can sometimes produce unnatural phrasing.
    - **Contextually Limited**: Questions might rely too heavily on specific sentence structures from the source text or fail to capture broader concepts.
    - **Potentially Irrelevant:** The model might occasionally generate questions about less important details in the text. Generating high-quality distractors for multiple-choice questions is a known challenge not fully addressed by standard T5 fine-tuning alone.
2) **Text Processing Challenges:** While text extraction from simple PDFs might be straightforward, handling complex layouts (multi-column text, tables, figures with captions) remains challenging for standard libraries like PyPDF2. OCR errors from scanned documents would further degrade input quality. Effective text segmentation is also crucial; poor chunking can lead to fragmented context and nonsensical questions.
3) **Model Size Limitation:** T5-small provides a good baseline, but larger models (T5-base, T5-large, or newer models like BART, GPT variants) generally offer significantly better performance in generation tasks, producing more fluent, coherent, and contextually appropriate text, albeit at higher computational cost.
4) **Answer Evaluation Complexity:** Evaluating open-ended answers to automatically generated questions is non-trivial. Relying solely on similarity to the source text span might penalize paraphrased or conceptually correct answers expressed differently. Robust evaluation requires deeper semantic

understanding, potentially leveraging models similar to those in the main Grading System (Module 4).

5) **Lack of Pedagogical Strategy:** The current model likely generates questions based on patterns learned from data (like SQuAD) without an explicit understanding of pedagogical goals, learning objectives, or desired difficulty levels.

The implications for the Edusmart system are that Module 2 provides a functional tool for generating draft questions or basic comprehension checks. It serves as a valuable assistant to instructors rather than a complete replacement for human oversight in assessment design. Instructors would likely need to review, filter, and potentially edit the generated questions before deploying them in formal assessments. For informal practice, however, the automated generation offers significant value.

Future Work:

Improving this module requires several parallel efforts:

- **Enhancing Question Quality**: Experimenting with larger pre-trained models, exploring advanced fine-tuning techniques (e.g., reinforcement learning from human feedback), and developing better prompting strategies. Incorporating specific mechanisms for generating plausible MCQ distractors is essential.
- **Diverse Question Types:** Researching methods to generate questions targeting higher-order thinking skills, potentially by fine-tuning models on datasets specifically designed for this or using knowledge graphs.
- **Improving Text Processing:** Integrating more robust PDF parsing libraries and potentially OCR capabilities (linking to Module 4) for wider applicability.
- **Robust Answer Evaluation:** Developing or integrating more sophisticated answer evaluation techniques within this module, potentially leveraging sentence embeddings or cross-encoder models for semantic comparison.
- **User Interface and Integration:** Designing an intuitive interface for instructors to manage content input, review generated questions, and for students to take quizzes. Integrating the quiz generation tightly with learning analytics could enable adaptive questioning based on student progress.

3.3: Unauthorized Object Detection, Head Pose Estimation, and User Verification

Objective Recap: This module aimed to enhance academic integrity during online examinations by implementing and evaluating AI-driven techniques for:

1) Detecting unauthorized objects (e.g., mobile phones, notes),

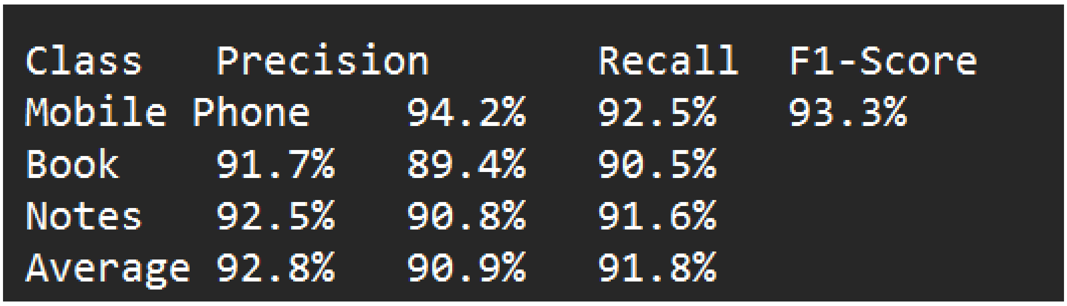
2) Monitoring student head pose to infer gaze direction and potential cheating behavior, and

3) Verifying the test-taker's identity using facial recognition. The integrated workflow is conceptually shown in Figure 4.

Results:
This multi-component module was evaluated through separate tests for each functionality, yielding strong performance indicators.

1) **Unauthorized Object Detection:**

- The YOLOv5 model, trained to identify prohibited items like mobile phones, books, and notes within the webcam feed, achieved a high F1-score of 91.8%. This score, balancing precision and recall, indicates that the model was both accurate in its detections (few false positives) and effective at finding most instances of the targeted objects (few false negatives) within the test dataset.

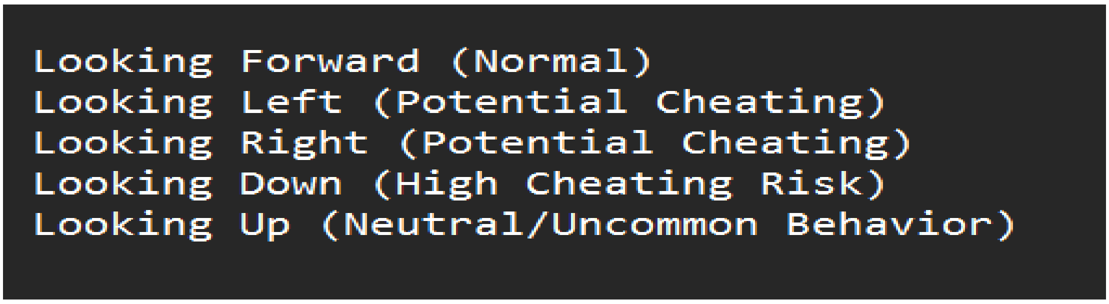


```
Class   Precision       Recall  F1-Score
Mobile Phone    94.2%   92.5%   93.3%
Book    91.7%   89.4%   90.5%
Notes   92.5%   90.8%   91.6%
Average 92.8%   90.9%   91.8%
```

Figure 6.Unauthorized Object Detection classification report

2) **Head Pose Estimation:**

- Using MediaPipe Face Mesh for landmark detection and the PnP algorithm for pose calculation, the system classified the estimated head orientation into predefined gaze zones (Forward, Left, Right, Down). This component demonstrated a classification accuracy of 95.1% in correctly assigning the head pose to the appropriate zone based on calculated yaw and pitch angles compared to ground truth labels (likely obtained via manual annotation of test videos or controlled setups).



```
Looking Forward (Normal)
Looking Left (Potential Cheating)
Looking Right (Potential Cheating)
Looking Down (High Cheating Risk)
Looking Up (Neutral/Uncommon Behavior)
```

Figure 7.specific gaze classifications

3) User Verification:

The facial recognition component, responsible for verifying the test-taker's identity against enrolled reference data, achieved a very high accuracy of 97.6%. This suggests the underlying deep learning model (likely trained using metric learning) was highly effective at generating discriminative embeddings and reliably distinguishing the enrolled user from potential impostors under the test conditions.

Discussion:

The high quantitative results across all three sub-components strongly support the technical feasibility and potential effectiveness of this integrated module for automated online proctoring, addressing Objective 2. The 91.8% F1-score for object detection suggests YOLOv5 can reliably identify common prohibited items, acting as a significant deterrent. The 95.1% accuracy in head pose classification indicates that tracking gaze direction via head orientation is a viable method for flagging potentially suspicious behavior like consistently looking away from the screen. Crucially, the 97.6% accuracy for user verification provides a strong safeguard against impersonation, a fundamental requirement for exam security.

The integration of these three distinct AI techniques, as depicted in Figure 4 (WEB CAM feeding into IDENTIFIED OBJECT DETECTION, HEAD POSE DETECTION, and User Verification, leading to Report Generating and REAL TIME NOTIFY), creates a layered security approach. An alert triggered by the detection of a mobile phone gains significance if corroborated by simultaneous head pose estimation indicating the student is looking down ('High Cheating Risk' zone). This synergy potentially allows for higher confidence in flagging suspicious events compared to relying on any single indicator.

However, interpreting these high accuracy figures requires careful consideration of context and limitations:

1. **Object Detection Nuances:** While the F1-score is high, its real-world effectiveness depends on the diversity of objects trained, robustness to partial occlusions, varying lighting, cluttered backgrounds, and the potential for novel cheating methods using items not in the training set. False positives (e.g., mistaking a calculator for a phone) could cause user frustration, while false negatives mean cheating incidents are missed. The specific precision/recall balance achieved at the operating threshold is important.
2. **Head Pose vs. Gaze:** Head pose is a proxy for gaze direction, not a direct measure of where the eyes are looking. A student could look sideways using only eye movement without significant head turning, which this method might miss. Accuracy reflects correct zone classification based on pose, but the link to actual attention or cheating intent requires inferential leaps. Short, natural glances away (e.g., thinking, brief eye rest) could be misflagged if not handled by temporal analysis (analyzing duration/frequency) rather than instantaneous pose, as noted in the methodology. The 95.1% accuracy might be high for classifying pose but may overestimate accuracy in detecting actual cheating intent.
3. **User Verification Robustness:** While 97.6% accuracy is excellent, the remaining 2.4% error rate could still lead to false rejections (denying access to the correct student) or false acceptances (allowing an impostor) in large-scale deployments. Performance can be sensitive to significant changes in appearance (e.g., new glasses, different hairstyle, masks), extreme lighting conditions, or sophisticated spoofing attempts (presentation attacks). The chosen operating threshold represents a trade-off between security (low FAR) and convenience (low FRR). Continuous verification throughout the exam presents additional technical and user experience challenges.

4. **Ethical and Privacy Concerns:** This module inherently involves continuous monitoring, raising significant privacy implications [10]. Students may feel stressed or unfairly scrutinized, potentially impacting performance. The potential for algorithmic bias (e.g., variations in face recognition accuracy across different demographic groups [9]) must be rigorously evaluated and mitigated. Transparency in how the system works and clear policies on data usage, storage, and review are paramount.
5. **Integration Logic:** The effectiveness of the integrated system depends heavily on the logic used to combine flags from different components. Overly sensitive rules could lead to excessive alerts, while overly lenient rules might fail to detect genuine cheating.

Future Work:

Future development should focus on:

- **Improving Robustness:** Enhancing models to handle wider variations in environmental conditions, object appearances, and user behavior. Training object detection on more diverse "in-the-wild" exam scenario data. Improving face verification robustness against occlusions and variations.
- **Temporal Analysis:** Implementing sophisticated temporal analysis for head pose to distinguish sustained deviations from brief, natural glances.
- **Direct Gaze Tracking:** Exploring the integration of eye-tracking techniques (if feasible with standard webcams, though often challenging) for more direct gaze monitoring.
- **Explainable AI (XAI):** Incorporating XAI to provide justifications for triggered alerts, aiding human review and building trust.
- **Bias Mitigation:** Continuously auditing models for demographic bias and applying fairness-aware training techniques.
- **Adaptive Thresholds/Logic:** Developing more sophisticated logic for combining alerts, possibly adapting sensitivity based on exam context or baseline user behavior.

- **User Experience:** Designing the system to be as minimally intrusive as possible while maintaining effectiveness, and providing clear communication to students.

In summary, Module 3 demonstrates strong technical capabilities for enhancing academic integrity through AI-powered monitoring. The high performance metrics validate the chosen computer vision techniques. However, practical deployment requires careful calibration, robust handling of edge cases, sophisticated temporal analysis, and paramount attention to ethical considerations and user privacy to ensure fair and effective implementation within the Edusmart system.

3.4. Automated Grading System (Related to Text/Handwritten Inputs)

Objective Recap: This module aimed to develop and evaluate an automated system capable of grading student assignments, including both typed and handwritten responses (via OCR), across various subjects. The goal was to provide accurate, consistent, and efficient assessment, reducing instructor workload and feedback time, addressing Objective 3.

Results:

The automated grading system integrated Optical Character Recognition (OCR) for handwritten inputs and a combination of NLP/ML models (including Transformers like BERT, RNNs like BiLSTM/GRU, and classifiers like Random Forest/SVM) for analyzing content and predicting scores. Evaluation focused on comparing the system's grades against human-assigned ground truth scores.

1. **Optical Character Recognition (OCR) Performance:** For processing handwritten answer sheets, the implemented OCR component (utilizing Tesseract and/or Google Vision API) demonstrated a success rate of 92% in accurately converting handwritten text images into machine-readable text on the test dataset. This rate likely reflects metrics like Character Error Rate (CER) or Word Error Rate (WER) being below a certain acceptable threshold for usability.

2. **Grading Agreement (Correlation):** When comparing the continuous scores predicted by the automated system (for both typed and OCR-processed handwritten answers) against the scores assigned by human graders, the system achieved a Pearson Correlation Coefficient of 0.87. A value close to +1 indicates a strong positive linear relationship, suggesting that the system's scores generally aligned well with human judgments – when humans gave higher scores, the system tended to give higher scores, and vice versa.
3. **Grading Accuracy (Classification):** When formulating the grading task as a classification problem (assigning answers to predefined grade categories or rubric levels), the system achieved an overall classification accuracy of 85.6%. This indicates that the system placed the answer in the correct grade category for over 85% of the test cases.

Discussion:

The results obtained for the automated grading module demonstrate significant success and highlight the potential of AI to streamline assessment processes. The high OCR success rate of 92% is crucial, indicating that the system can reliably handle handwritten inputs, expanding its applicability beyond purely digital assignments. While not perfect (8% error rate could still introduce noise), this level of accuracy suggests the OCR component is sufficiently robust to provide usable text input for the subsequent grading models in most cases.

The Pearson Correlation Coefficient of 0.87 is a particularly strong result. Human graders often exhibit inter-rater reliability correlations in the range of 0.7 to 0.9, depending on the task complexity and rubric clarity. Achieving a correlation of 0.87 suggests the automated system approaches, and potentially rivals, the consistency level found among human graders for the evaluated assignments. This indicates that the NLP and ML models effectively captured features relevant to the scoring rubric used by the human experts.

Similarly, the classification accuracy of 85.6% further supports the system's effectiveness. Correctly assigning over 85% of answers to the appropriate grade category demonstrates a strong capability for automated evaluation, sufficient for many formative assessment scenarios or as a reliable first-pass grading tool for summative assessments. These combined results strongly validate the methodology employed, integrating advanced NLP for semantic feature extraction (BERT, etc.) with robust ML classifiers (RF, SVM) optimized via techniques like GridSearchCV, fulfilling Objective 3.

Despite these positive outcomes, several critical limitations and challenges inherent to automated grading must be discussed:

1. **Depth of Understanding vs. Pattern Matching:** While models like BERT excel at capturing semantic similarity and contextual relevance, assessing deeper qualities like creativity, originality of argument, critical thinking, or nuanced reasoning remains extremely challenging. The high correlation might reflect the system's proficiency at identifying rubric-aligned keywords, structural elements, and topic relevance, but potentially not a true "understanding" comparable to a human expert. It may struggle with novel arguments or unconventional approaches that deviate from patterns seen in the training data.
2. **Impact of OCR Errors**: The 8% OCR error rate, while seemingly low, can significantly impact grading accuracy for affected submissions. Misrecognized keywords or phrases could lead to unfairly low scores. The system's robustness to such noise is a critical factor.
3. **Bias and Fairness:** NLP models can inherit biases from their training data [9]. The grading system might unintentionally favor certain writing styles (e.g., more verbose answers) or disadvantage students from specific linguistic backgrounds if not carefully audited and mitigated. Ensuring fairness and equity is a major ongoing challenge [9, 10].
4. **Explainability (Lack Thereof):** Random Forests, SVMs, and especially deep learning models often function as "black boxes." The system provides a score but typically struggles to offer meaningful, specific feedback explaining why that score was given, hindering the learning process for students [10]. Providing actionable, formative feedback remains a significant gap compared to human grading.
5. **Domain Specificity and Generalizability:** Models trained on specific datasets or subjects might not generalize well to new domains, different

question types, or evolving assessment criteria without retraining or fine-tuning.

- **Implications for Edusmart:** Module 4 offers a powerful tool within the Edusmart ecosystem, capable of dramatically improving efficiency. It can provide immediate feedback on formative assessments, handle grading for large-enrollment courses, and ensure a baseline level of consistency. However, given the limitations, its optimal role might be:
- **Formative Assessment:** Providing rapid feedback on practice assignments or quizzes.
- **First-Pass Grading:** Handling initial grading for large batches, flagging complex or borderline cases for human review.
- **Consistency Check:** Helping calibrate human graders or identify potential inconsistencies in manual grading.
- It should likely not be used as the sole arbiter for high-stakes summative assessments without a human-in-the-loop review process, particularly for tasks requiring evaluation of higher-order thinking skills.

- **Future Work:**

  Addressing the limitations requires further research and development:

  - **Improving Deep Understanding:** Exploring more advanced NLP architectures, knowledge graph integration, and techniques specifically designed to assess argumentation, reasoning, and creativity.
  - **Enhancing Explainability (XAI):** Integrating XAI methods to generate justifications for assigned scores, providing more transparent and actionable feedback [10].
  - **Bias Detection and Mitigation:** Implementing rigorous auditing procedures and fairness-aware AI techniques to identify and reduce potential biases [9].

- **Improving OCR Robustness:** Continuously improving the OCR pipeline, potentially using context from the grading task to correct recognition errors.
- **Adaptive Grading:** Developing models that can adapt to different rubrics, subjects, and question types with minimal retraining.
- **Handling Complex Formats:** Extending capabilities to grade answers involving code, mathematical equations, or diagrams.

In conclusion, the Automated Grading System module demonstrates high accuracy and strong correlation with human graders, validating its technical design and potential for significant efficiency gains within Edusmart. However, responsible deployment necessitates acknowledging its limitations regarding deep understanding, potential bias, and explainability, positioning it primarily as a powerful assistive tool within a broader assessment strategy that retains human oversight for nuanced evaluation and meaningful feedback.

4.Summary of Each Student’s contribution

Member Name	Contribution
Premathilaka S.P.D.M. IT21185298	**Primary AI Module Responsibility:** ▪ Facial Expressions and Attention Detection. **Core Objective:** ▪ To develop the AI component capable of analyzing student webcam feeds in real-time to classify facial expressions and infer attention levels, providing insights into student engagement during online learning. **Data Collection and Preprocessing:** ▪ Sourced data from standard datasets (CK+, JAFFE) and orchestrated custom data collection sessions (ensuring ethical considerations like informed consent). Implemented the data preprocessing pipeline, including face detection (using libraries like OpenCV/Dlib), image resizing (to 48x48), grayscale conversion, and pixel normalization. Managed data splitting into training, validation, and test sets using stratified sampling. **Model Training and Development:** ▪ Designed, implemented, and trained the Convolutional Neural Network (CNN) architecture for 7-class emotion classification. Managed the training process, including configuring the loss function (Categorical Cross-Entropy), optimizer (Adam), implementing data augmentation techniques (using Keras ImageDataGenerator), and incorporating regularization methods like Early Stopping and ReduceLROnPlateau to prevent overfitting and optimize performance, achieving 78%

	accuracy. Conceptually explored the integration of LSTMs for temporal analysis. **Backend Development:** ▪ Developed the backend processing pipeline for Module A. This involved creating services to receive image frames, run them through the trained CNN model for inference, interpret the output probabilities to determine the most likely emotion, and potentially aggregate these findings over time or across users. Developed APIs to expose engagement analytics derived from this module. **Frontend Development:** ▪ Took primary responsibility for developing key user-facing components of the Edusmart web application: **Teacher Dashboard:** ▪ Designed and implemented the dashboard interface where instructors could view aggregated, real-time (or near real-time) analytics on student engagement derived from Module A. This involved visualizing the data (e.g., charts showing emotion distribution over time, attention level indicators) received from the backend API. **Student Classroom Page:** ▪ Developed the main interface where students attend online lectures or interact with learning materials. Integrated the webcam access functionality required to capture the video feed that serves as the input for Module A's analysis. Ensured this integration was seamless and handled user permissions for camera access appropriately. **Integration:**

	▪ Ensured the smooth flow of data from the Student Classroom Page (webcam feed) to the backend processing pipeline for Module A, and then connected the derived analytics from the backend to be displayed effectively on the Teacher Dashboard.
Wijenandana S.D. IT21158254	**Primary AI Module Responsibility:** ▪ Automated Quiz Generation and Evaluation. **Core Objective:** ▪ To create an AI system that could automatically generate relevant quiz questions from provided educational content (like lecture notes or PDFs) to aid instructors and provide students with practice opportunities. **Data Collection and Preprocessing:** ▪ Focused on handling input educational content. Implemented methods for text extraction from various file formats (specifically PDFs using libraries like PyPDF2). Developed the preprocessing pipeline for text data, including cleaning (removing noise, formatting artifacts), segmenting large documents into manageable chunks, and tokenizing text using the appropriate tokenizer (T5 tokenizer) for the chosen model. Utilized datasets like SQuAD for fine-tuning the question generation model. **Model Training and Development:** ▪ Selected and fine-tuned the T5-small sequence-to-sequence Transformer model for the task of question generation. Managed the fine-tuning process, likely adapting the SQuAD dataset format to train the model to generate questions given context (and potentially answer spans). Experimented with generating

	questions from processed text chunks and demonstrated the feasibility of the approach. Conceptualized the answer evaluation component for this module. **Backend Development:** ▪ Built the backend logic for Module B. This included services to handle file uploads (lecture notes), orchestrate the text extraction and preprocessing pipeline, manage the T5 model inference process for generating questions based on context, potentially implement basic answer evaluation logic, and store/retrieve generated quizzes from a database. Developed APIs to support the quiz generation frontend page. **Frontend Development:** ▪ Developed the Quiz Generate Page interface. This involved creating the user interface elements allowing instructors to upload their educational content (e.g., PDF files), initiate the question generation process, view the generated questions, potentially edit or filter them, and save quiz sets. Ensured communication between this frontend page and the Module B backend API. **Integration:** Integrated the content upload functionality with the backend text processing pipeline and T5 model. Connected the quiz generation controls on the frontend to trigger the appropriate backend processes and display the results (generated questions) back to the instructor on the Quiz Generate Page.

Ekanayaka H.E.M.P.L. IT21185502	**Primary AI Module Responsibility:** ▪ Unauthorized Object Detection, Head Pose Estimation, and User Verification. **Core Objective:** ▪ To develop the comprehensive academic integrity module for secure online examinations, capable of detecting prohibited items, monitoring student gaze direction via head pose, and verifying user identity. **Data Collection and Preprocessing:** ▪ Curated datasets for object detection by combining standard datasets (e.g., COCO) with custom-collected images/videos simulating exam scenarios (requiring annotation with bounding boxes). Leveraged MediaPipe Face Mesh for facial landmark extraction (requiring implementation, not training). Managed data for face verification, including an enrollment process for reference data and potentially using large pre-trained models. Implemented preprocessing specific to each sub-module: resizing/augmentation for YOLOv5, alignment/normalization for face recognition. **Model Training and Development:** ▪ Trained and fine-tuned the YOLOv5 object detection model on the curated dataset to identify specific prohibited items (mobile phones, books, etc.), achieving a 91.8% F1-score. Implemented the head pose estimation logic using MediaPipe landmarks and the PnP algorithm (OpenCV), developing the classification rules for gaze zones (achieving 95.1% accuracy). Implemented the face verification pipeline using pre-trained deep learning models to

	extract embeddings and perform 1:1 matching against enrolled templates using distance metrics and thresholding (achieving 97.6% accuracy). **Backend Development:** ▪ Developed the complex backend infrastructure for Module C, likely involving parallel processing streams for the three sub-components operating on the exam video feed. Built inference services for YOLOv5, head pose calculation, and face verification. Implemented the logic to integrate alerts from these components based on predefined rules. Managed secure storage and retrieval of face enrollment data. Developed APIs to push proctoring alerts and status updates. **Frontend Development:** ▪ Developed the frontend components related to displaying academic integrity information, specifically the "Unauthorized Detect Part". This likely involved creating real-time visual feedback within the student's exam interface (if applicable) or, more commonly, developing parts of a proctoring dashboard (potentially viewed by instructors/proctors) that display: alerts for detected objects (with bounding boxes overlaid on video), indicators of current head pose/gaze zone, user verification status, and logs of suspicious events. **Integration:** Ensured the real-time video feed from the exam environment was correctly processed by the three parallel backend pipelines. Integrated the alert generation logic with the

	frontend display for proctoring information. Managed the connection to the user identity database for verification.
Karunarathana J.H.H.N. IT21157950	**Primary AI Module Responsibility:** ▪ Automated Grading System. **Core Objective:** ▪ To design and implement an AI system capable of automatically grading diverse student assignments, including both typed text and handwritten submissions, to improve efficiency and consistency. **Data Collection and Preprocessing:** ▪ Collected and curated datasets of student answers (typed and handwritten) from various sources (public datasets, institutional data). Managed the crucial process of obtaining reliable human-assigned grades (ground truth) for training and evaluation. Developed the distinct preprocessing pipelines: text cleaning and tokenization (using BERT tokenizer) for typed answers; and image preprocessing (binarization, noise reduction), OCR (using Tesseract/Google Vision API, achieving 92% success), and post-OCR cleaning for handwritten submissions. **Model Training and Development:** ▪ Implemented and evaluated OCR engines. Leveraged advanced NLP models (BERT, potentially GPT-3, RoBERTa, BiLSTM/GRU) for extracting deep semantic features from the processed text. Trained supervised Machine Learning models (Random Forest, SVM) using these features to predict grades/scores. Employed techniques like GridSearchCV for hyperparameter optimization to maximize performance, achieving a Pearson

	correlation of 0.87 and classification accuracy of 85.6%. **Backend Development:** ▪ Built the backend system for Module D. This involved creating services to receive student submissions (text or images), route them through the appropriate preprocessing pipeline (including the OCR step for images), manage the NLP feature extraction process, run inference using the trained ML grading models (RF/SVM), and store the resulting automated grades. Developed APIs to support the automated grading frontend interface. **Frontend Development:** ▪ Developed the Automated Grading System interface components. This likely included UI elements for instructors to submit assignments for grading, potentially configure grading parameters or rubrics (if applicable), view the automatically assigned grades alongside the student submissions, and possibly an interface for reviewing or overriding the automated scores. **Integration:** Ensured seamless handling of both typed and image-based submissions from the frontend. Integrated the OCR component into the backend pipeline for handwritten answers. Connected the grading model outputs to the frontend display, providing instructors with access to the automated evaluation results.

5. Conclusion

A transformative shift towards digital learning platforms has undeniably democratized access to education, yet it has simultaneously introduced persistent challenges that hinder the realization of its full potential. Online environments often struggle to replicate the nuanced interactions of traditional classrooms, leading to difficulties in monitoring student engagement, ensuring the credibility of assessments through robust academic integrity measures, and managing the significant workload associated with grading and feedback, particularly at scale. This research was motivated by the critical need to address these interconnected challenges through a unified, technologically advanced solution. The central research problem identified was the fragmentation of existing tools and the lack of integrated systems capable of holistically enhancing the online learning experience across engagement, integrity, and assessment dimensions.

To confront this problem, the primary objective of this work was to design, develop, implement, and evaluate "Edusmart," an innovative, multi-module system leveraging Artificial Intelligence. Edusmart was conceptualized as an integrated ecosystem combining distinct AI-driven functionalities: real-time facial expression analysis and attention detection to gauge student engagement; automated quiz generation from educational content to facilitate practice and assessment creation; a multi-layered academic integrity suite featuring unauthorized object detection, head pose estimation, and user verification for secure online examinations; and an automated grading system capable of handling both typed and handwritten assignments via OCR. The methodology employed state-of-the-art techniques from Computer Vision (CNNs, YOLOv5, MediaPipe), Natural Language Processing (Transformers like T5, BERT; RNNs), and Machine Learning (Random Forest, SVM), underpinned by rigorous data preprocessing, model training, hyperparameter optimization (e.g., GridSearchCV), and evaluation against established metrics.

The empirical results presented in this paper largely validate the effectiveness and feasibility of the proposed Edusmart system. The facial expression recognition module achieved a promising 78% accuracy, demonstrating the potential to infer affective states relevant to engagement. The academic integrity module yielded particularly strong results, with the YOLOv5-based object detection reaching a 91.8% F1-score, head pose estimation achieving 95.1% accuracy in classifying gaze zones, and facial recognition for user verification attaining 97.6% accuracy. These figures underscore the viability of AI in creating more secure online assessment environments. Furthermore, the automated grading system demonstrated significant alignment with human evaluators, achieving a high Pearson Correlation Coefficient of 0.87 and a classification accuracy of 85.6%, supported by a robust 92% success rate in the OCR component for handwritten inputs. The automated quiz generation module, utilizing a T5 model, successfully demonstrated the feasibility of creating contextually relevant questions from source materials.

The significance of these findings extends beyond the performance of individual components. The successful design and preliminary integration of these diverse AI capabilities into a single conceptual framework represent a key contribution of this research. Edusmart addresses the critical integration gap identified in the literature and existing market solutions. By combining tools for engagement, integrity, and assessment, the system offers the potential for synergistic benefits – for instance, engagement data could inform adaptive assessment difficulty, or grading results could highlight areas needing more engaging content delivery. This holistic approach promises a more cohesive, responsive, and trustworthy online learning environment compared to deploying isolated point solutions. The research demonstrates that AI can indeed be harnessed not just to automate tasks, but to potentially enhance the quality, fairness, and efficiency of the entire online educational process for both students and instructors.

Despite the promising results and the successful proof-of-concept for the Edusmart system, it is crucial to acknowledge the limitations inherent in this research and the challenges that remain. The accuracy of the facial expression module (78%), while significant, indicates room for improvement, particularly in distinguishing subtle or similar emotions and generalizing across diverse real-world conditions. Moreover, inferring complex states like engagement solely from basic emotions is an oversimplification. The academic integrity module, despite high accuracy metrics, relies on proxies (head pose for gaze) and faces challenges with novel cheating methods, potential biases, and significant ethical considerations surrounding continuous student monitoring. The automated grading system, while correlating strongly with human graders, still struggles with evaluating higher-order thinking skills, ensuring fairness across diverse student populations, and providing nuanced, explainable feedback. The quiz generator primarily demonstrated feasibility, with further work needed to ensure consistent quality and pedagogical appropriateness. Furthermore, the evaluations were conducted largely in controlled or simulated environments; performance in large-scale, real-world deployments with diverse student populations and technical setups may differ. The ethical implications surrounding data privacy, algorithmic bias, and the potential impact of AI monitoring on student well-being require ongoing critical examination and robust governance frameworks.

These limitations naturally point towards several crucial directions for future research and development. Enhancing the robustness and generalizability of all AI models is paramount, requiring training on larger, more diverse, and ecologically valid datasets. Incorporating temporal analysis for facial expressions and head pose could yield more reliable insights into dynamic states like attention and engagement. Further exploration into multi-modal fusion – combining visual cues with interaction data (e.g., keyboard/mouse activity, response times) – holds significant promise for more accurate engagement and integrity assessment. Integrating Explainable AI (XAI) techniques across modules, especially for grading and integrity alerts, is essential to build trust, facilitate human review, and provide meaningful feedback. Rigorous auditing for fairness and bias, coupled with the development of mitigation strategies, must be an integral part of the development cycle. Extending the capabilities of the grading system to handle more complex response types (e.g., code, mathematical proofs) and generate truly formative feedback remains a key challenge. For quiz generation, focusing on controlling question difficulty, targeting higher-order skills, and generating plausible distractors is vital. Finally, conducting longitudinal studies in authentic educational settings is necessary to evaluate the real-world impact of the Edusmart system on learning outcomes, student experience, instructor workload, and overall institutional effectiveness, alongside developing comprehensive ethical guidelines for its deployment.

In conclusion, this research successfully demonstrated the design, implementation, and evaluation of Edusmart, an integrated AI-powered system addressing key challenges in online education. The findings validate the potential of strategically combining computer vision, natural language processing, and machine learning to foster more engaging learning environments, uphold academic integrity, and enhance assessment efficiency. While acknowledging the existing limitations and the critical need for continued research, particularly concerning ethics, fairness, and real-world validation, Edusmart represents a significant step towards creating more sophisticated, responsive, and trustworthy digital learning ecosystems. The future of online education will likely involve increasingly intelligent systems, and this work contributes valuable insights and a functional blueprint for harnessing AI to improve the quality and credibility of learning experiences in the digital age

6. References

[1] Taylor, D., Yeung, M., & Bashet.,, "Personalized and Adaptive Learning," 2021.

[2] Chan, C. H., & Robbins, L. I.,, "E-learning systems: Promises and pitfall," *Academic psychiatry,* vol. 30(6), pp. 491-497, 2006.

[3] Saqr, M., López-Pernas, S., & Helske, S.,, "The longitudinal association between engagement and achievement varies by time, students," *profiles and achievement state: A full program study,* vol. 199, p. 104787, 2023.

[4] Valenti, S., Cucchiarini, R., & Spognardi, A.,, "Supporting the development of argumentation skills in online learning environments through automated feedback," *Computers in Education,* Vols. 1-16, p. 118, 2018.

[5] Ong, S. Z., Tee, C., & Goh, M. K. O.,, "Cheating Detection for Online Examination Using Clustering Based Approach," *International Journal on Informatics Visualization,* pp. 2075-2085, 2023.

[6] Hazar, M. J., Toman, Z. H., & Toman, S. H.,, "Automated Scoring for Essay Questions in E-learning,," *Journal of Physics: Conference Series,* vol. 1294(4), p. 042014, 2019.

[7] Rokade, A., Patil, B., Rajani, S., Revandkar, S., & Shedge, R.,, "Automated Grading System Using Natural Language Processing,," *Proceedings of the International Conference on Inventive Communication and Computational Technologies,* pp. 1123-1127, 2018.

[8] Farrús, M., & Costa-Jussà, M. R.,, "Automatic Evaluation for E-Learning Using Latent Semantic Analysis : A Use Case is (SNA) in OnlineCourses Automatic Evaluation for E Learning Using Latent Semantic Analysis : A Use Case Farrús and Costa-jussà.," 2024.

[9] Chen, D., Huang, Z., & Li, B.,, "Attention-based deep feature fusion for facial expression recognition," *IEEE Transactions on Image Processing,* vol. 27(4), pp. 1989-2001, 2018.

[10] Valenti, S., Cucchiarini, R., & Spognardi, A.,, "Supporting the development of argumentation skills in online learning environments through automated feedback," *Computers in Education,* Vols. 1-16, p. 118, 2018.

[11] Hooda, M., Rana, C., Dahiya, O., Rizwan, A., & Hossain, M. S.,, "Artificial Intelligence for Assessment and Feedback to Enhance Student Success in Higher Education," *Mathematical Problems in Engineering,* pp. 1-19, 2022.

[12] Thotad, P., Kallur, S., & Amminabhavi, S.,, "Automatic Question Generator Using Natural Language Processing," *Journal of Pharmaceutical Negative Results,* vol. 13, pp. 2759-2764, 2023.

[13] Rus, V., Graesser, A., Piwek, P., Lintean, M., Stoyanchev, S., & Moldovan, C.,, "THE QUESTION GENERATION SHARED TASK AND EVALUATION CHALLENGE," 2011.

[14] Girshick, R., Donahue, J., Darrell, T., & Malik, J.,, "Rich feature hierarchies for accurate object detection and semantic segmentation," *IEEE Conference on Computer Vision and Pattern Recognition,* pp. 580-587, 2014.

[15] Kazemi, V., & Sullivan, J.,, "One millisecond face alignment with an ensemble of regression trees," *2014 IEEE Conference on Computer Vision and Pattern Recognition,* pp. 1867-1874, 2014.

[16] Taghipour, K., & Ng, Y. K.,, "A novel approach to automated essay scoring using recurrent neural networks," *IEEE Transactions on Education,* vol. 62, pp. 10-17, 2016.

[17] Hasanah, U., Permanasari, A. E., Kusumawardani, S. S., & Pribadi, F. S.,, "A scoring rubric for automatic short answer grading system," *Telkomnika (Telecommunication Computing Electronics Control),* vol. 17(2), pp. 763-770, 2019.

[18] Thotad, P., Kallur, S., & Amminabhavi, S.,, "Automatic Question Generator Using Natural Language Processing," *Journal of Pharmaceutical Negative Results,* vol. 13, pp. 2759-2764, 2023.

[19] Deng, Z., Peng, F., & Yu, L.,, "Deep learning for handwritten mathematical expression recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 39(12), pp. 2678-2691, 2017.

[20] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A.,, "You Only Look Once: Unified, real-time object detection," *IEEE Conference on Computer Vision and Pattern Recognition,* pp. 779-788, 2016.

[21] Kazemi, V., & Sullivan, J.,, "One millisecond face alignment with an ensemble of regression trees," *2014 IEEE Conference on Computer Vision and Pattern Recognition,* pp. 1867-1874, 2014.