EduVision

Transforming Classroom Dynamics with AI-Driven Behavioral Analysis

Abdulrhman G. Alahmadi, Waleed A. Albishri, Ebrahim M. Sharka

Faculty of Computing and Information Technology

King Abdulaziz University

13/5/2024

EduVision

Transforming Classroom Dynamics with AI-Driven Behavioral Analysis

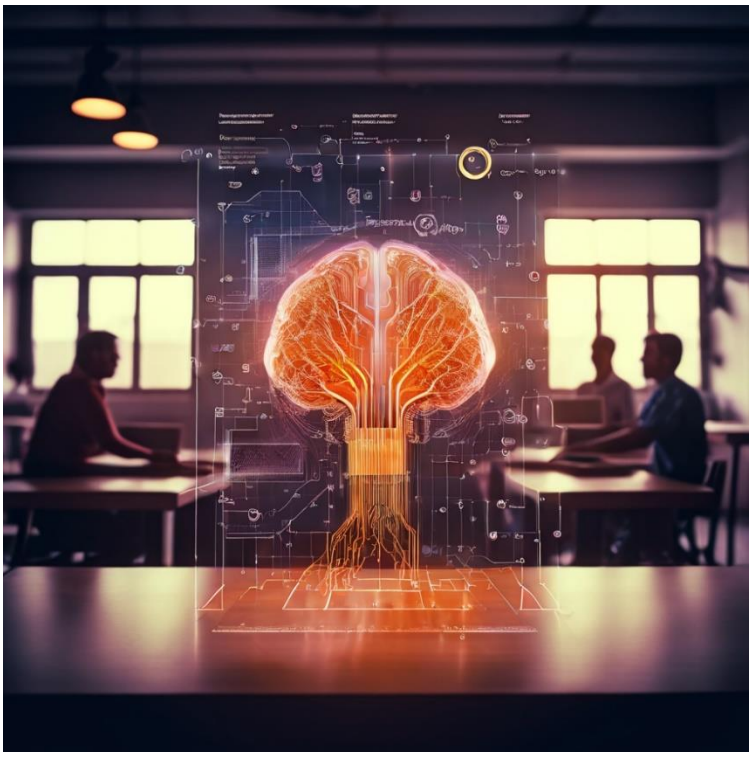
Abdulrhman G. Alahmadi, Waleed A. Albishri, Ebrahim M. Sharka

This report is submitted in partial fulfillment of the requirements for the award of Bachelor of Science in Computer Science

Supervised by: Dr.Mohammed Y. Dahab
Faculty of Computing and Information Technology

King Abdulaziz University

13/5/2024



Declaration of Originality

We hereby declare that this project report is based on our original work except for citations and quotations, which had been duly acknowledged. We also declare that it has not been previously and concurrently submitted for any other degree or award at KAU or other institutions.

Abdulrhman G. Alahmadi

Waleed A. Albishri

Ebrahim M. Sharka

Abstract (English)

EduVision emerges as a groundbreaking initiative in the realm of educational technology, with its primary focus on the creation of a sophisticated computer vision system. This system is meticulously designed to analyse student behaviour within classroom settings.

By bridging a notable gap in conventional educational methodologies, EduVision leverages the prowess of artificial intelligence to significantly enhance the learning environment. The project's core objective is to recognize and analyse various behavioural patterns exhibited by students during classroom sessions.

This innovative approach aims to provide educators with invaluable insights into student engagement and participation, thereby facilitating a more personalized and effective teaching strategy. Furthermore, EduVision underscores the importance of adapting to modern educational needs by integrating cutting-edge technology into the traditional classroom setting.

The implementation of this system promises to revolutionize the conventional classroom dynamic. By providing feedback to educators, it enables them to tailor their teaching methods to better suit the individual needs and learning styles of their students. This adaptability is crucial in fostering an interactive and engaging learning experience, thereby optimizing student performance and academic success.

In summary, EduVision sets a new standard in educational technology, offering a novel approach to student behaviour analysis through the application of computer vision and artificial intelligence. Its implementation in classrooms holds the potential to transform the educational landscape, making learning more interactive, personalized, and effective.

المستخلص

تبرز "إديوفيجن (EduVision) "كمبادرة رائدة في مجال التكنولوجيا التعليمية، حيث تركز بشكل أساسي على إنشاء نظام متطور للرؤية الحاسوبية. تم تصميم هذا النظام بدقة لتحليل سلوك الطلاب داخل الفصول الدراسية.

من خلال سد الفجوة الملحوظة في المنهجيات التعليمية التقليدية، تستفيد "إديوفيجن" من قوة الذكاء الاصطناعي لتحسين بيئة التعلم بشكل كبير. الهدف الأساسي للمشروع هو التعرف على الأنماط السلوكية المختلفة التي يظهرها الطلاب أثناء الحصص الدراسية وتحليلها.

يهدف هذا النهج المبتكر إلى تزويد المعلمين برؤى قيمة حول مشاركة الطلاب وانخراطهم، مما يسهل استراتيجية تدريس أكثر تخصيصًا وفعالية. علاوة على ذلك، تؤكد "إديوفيجن" على أهمية التكيف مع الاحتياجات التعليمية الحديثة من خلال دمج التكنولوجيا المتطورة في بيئة الفصل الدراسي التقليدية.

يعد تنفيذ هذا النظام بمثابة ثورة في ديناميكيات الفصل الدراسي التقليدية. من خلال تقديم ملاحظات للمعلمين، يمكّنهم من تكييف أساليب التدريس الخاصة بهم لتتناسب بشكل أفضل مع الاحتياجات الفردية وأنماط التعلم لطلابهم. هذه المرونة ضرورية لتعزيز تجربة تعليمية تفاعلية وجذابة، مما يؤدي إلى تحسين أداء الطلاب ونجاحهم الأكاديمي.

باختصار، تضع "إديوفيجن" معيارًا جديدًا في التكنولوجيا التعليمية، حيث تقدم نهجًا مبتكرًا لتحليل سلوك الطلاب من خلال تطبيق الرؤية الحاسوبية والذكاء الاصطناعي. إن تطبيقها في الفصول الدراسية يحمل إمكانية تحويل المشهد التعليمي، مما يجعل التعلم أكثر تفاعلية وتخصيصًا وفعالية.

Table of Contents

List of Tables

List of figures

CHAPTER 1

Introduction and background of the project

1.Introduction

EduVision represents a breakthrough in education technology, focusing on the development of a computer vision system for analyzing student behavior in classrooms.

The project addresses a significant gap in traditional educational methods by leveraging artificial intelligence to enhance the learning experience.

The system aims to recognize and analyze a wide spectrum of student behaviors, offering real-time feedback to educators. This feedback is crucial for adapting teaching methods to improve student engagement and learning outcomes.

EduVision targets educators, educational institutions, and government bodies, aiming to elevate the standard of education through technology.

The project confronts challenges such as incomplete behavior recognition, lack of systematic feedback, and insufficient data-driven insights in current educational practices.

The solution combines the latest in computer vision, machine learning, and data analysis to revolutionize classroom dynamics, making it a pioneering effort in the field of educational technology.

2.Problem Definition

In education technology, there is a need for innovative solutions that can transform traditional classrooms into engaging learning environments. Our project aims to address a non-trivial problem related to Artificial intelligence by developing a computer vision system for monitoring and analyzing student behavior in classrooms.

2.1 Project Scope

This project involves creating and deploying a reliable computer vision system that can recognize a wide range of student behaviors and offer feedback.

The main objective is to enhance classroom interactions, upgrade teaching techniques, and raise the standard of education.

2.2 Target Users

The target users for our system include:

Educators: Classroom teachers and instructors who benefit from the insights into student behavior, allowing them to adapt their teaching methods to enhance student engagement and learning outcomes.

Educational Institutions: Schools, colleges, and universities can use the system to assess and improve the overall classroom environment, thus ensuring a more conducive atmosphere for learning.

Government Bodies: Educational authorities and policymakers can utilize the data and insights generated by the system to make informed decisions about education policies and resource allocation.

2.3 Key Issues

Incomplete Behavior Recognition: Current classroom monitoring systems often focus on student mood and poses during the class ignoring whether they focus on educators or not, limiting educators' ability to gain a holistic understanding of classroom dynamics.

Lack of Feedback: Traditional classroom observation methods lack the capacity to provide accurate feedback for the whole classroom to teachers, hindering their ability to adapt their teaching methods and maintain an optimal learning environment.

Limited Data-Driven Insights: The lack of a strong computer vision system obstructs the gathering and analysis of valuable data on student behavioral patterns, impeding informed decision-making at institutional and policy levels.

Inefficiencies in Education Delivery: Ineffectual monitoring systems can lead to suboptimal teaching methodologies and student engagement, ultimately affecting the quality of education delivered.

2.4 Suggested Solution

We have come up with a practical and thorough solution to tackle the problems we have identified. Our proposed system will incorporate cutting-edge computer vision technology, machine learning algorithms, and data analytics to offer a multi-dimensional approach to the challenges faced in conventional classrooms.

Comprehensive Computer Vision System: Our team will develop and put into operation a cutting-edge computer vision system that can precisely identify and classify various student behaviors. To achieve top-notch accuracy in behavior recognition, we will utilize deep learning methods and real-time image processing.

Feedback Mechanism: The system will incorporate a feedback mechanism that delivers insights to educators. Ensuring educators can adapt their teaching strategies promptly.

Phased Development Approach: The project will follow a structured development process, including Requirements Specification, Analysis, Design, Implementation, Testing, and Deployment phases.

Each phase will be meticulously planned and executed to ensure the successful realization of the project's objectives.

By combining cutting-edge technology with a systematic development approach, our solution aims to advance education, foster greater student engagement, and ultimately lead to improved academic performance.

3.Project

3.1Project Overview

Project Name: EduVision

Project Start Date: September 2023

Project End Date: June 2024

3.2 Project Scope

The project aims to accomplish the following:

Phase 1: Design and Dataset

Phase 2: Implementation

3.2 Project Tasks

Phase 1: Design and Dataset

Task 1: Initial Presentation (100%)

Task 2: Report 1 (100%)

Task 3: Presentation 1 (100%)

Task 4: Report 2 (100%)

Task 5: Presentation 2 (100%)

Task 6: Final Report (100%)

Task 7: Poster Session (100%)

Task 8: Final Presentation (100%)

Phase 2: Implementation

Task 10: Writing The Code (100%)

3.3 Milestones

Phase 1: Design and Dataset

Milestone 1: End of Phase 1

Phase 2: Implementation

Milestone 2: End of Phase 2

3.4 Project Scheduling

Figure 1 project Scheduling.

3.5 Resource Allocation

The project tasks will be completed as a group effort, with all team members collaborating and contributing to each task.

3.6 Project Timeline

Phase 1: Design and Dataset

September 19, 2023 - Initial Presentation

October 11, 2023 - Report 1

October 12, 2023 - Presentation 1

November 5, 2023 - Report 2

November 8, 2023 - Presentation 2

December 2, 2023 - Final Report

December 4 - Poster Day

December 6 - Final Presentation

Phase 2: Implementation

June 2, 2024 - Writing the Code

CHAPTER 2
Literature Reviews

4. literature review

4.1 YOLO based Human Action Recognition and Localization[1]

The paper explores using YOLO (You Only Look Once) for human action recognition and localization in videos. YOLO is an object detection model that uses a single convolutional neural network to predict bounding boxes and class probabilities directly from images. The authors trained a YOLO model on the LIRIS human action dataset which contains 10 classes of human-object and human-human interactions.

The model can simultaneously detect, localize and classify human actions in videos with a frame rate of 15-16 FPS, enabling real-time performance.

Their approach recognizes actions using fewer frames than typical methods, often just a single frame, without needing optical flow data. Periodic frames are processed rather than the whole video.

The best model achieves 88.4% accuracy on the LIRIS test set for combined detection, localization and classification.

The advantage of using YOLO is its speed and efficiency in being able to process images in a single pass through the network.

This could enable real-time spatiotemporal action localization.

Overall, the paper shows promising results for using YOLO for human action analysis in videos and demonstrates YOLO's capability for our use case.

Table 1 yolo paper criteria

Prompt criteria	Comments
Presentation	The presentation of the paper was excellent
Relevance	The information presented is the methodology we will be using in our case
Objectivity	The author aimed to present real time action detection method and did so
Method	The paper does not describe any specific methodology for data collection as the authors used the existing LIRIS Human Activities dataset to train and test their model.
Provenance	All relevant papers were correctly referenced and provided at the end of the paper; it was published on the research gate.
Timeliness	The paper was published in 2018.

4.2 An Effective Behavior Recognition Method in the Video Session Using Convolutional Neural Network [2]

The research paper titled "An Effective Behavior Recognition Method in the Video Session Using Convolutional Neural Network" offers significant insights that can be leveraged to enhance the Eduvision model. Eduvision aims to detect student behavior in a classroom setting.

The paper's methodology and findings are particularly relevant for improving the accuracy and efficiency of behavior recognition in Endovision's video-based monitoring system.

Key Findings and Their Relevance:

Target Detection and Behavior Recognition: The paper proposes integrating a target detection phase before behavior recognition.

This approach, involving extracting the body region from videos, can be applied to Eduvision to focus on students and reduce background noise interference.

I. Fragmentation and Stochastic Sampling: By fragmenting video sessions and using stochastic sampling, the paper addresses the challenge of modeling long-duration videos. This technique could help Eduvision effectively analyze extended classroom sessions, capturing comprehensive behavioral patterns over time.

II. Improved Loss Function: The adoption of an improved loss function (focal loss function) to tackle classification challenges and sample imbalances can be beneficial for Eduvision, especially when dealing with varied and subtle student behaviors.

III. Use of Convolutional Neural Networks (CNNs): The paper emphasizes the superiority of CNNs in image processing, which could be instrumental in enhancing Endovision's image analysis capabilities.

IV. Comparison with Traditional Methods: The paper highlights the limitations of traditional computer vision and optical flow methods, suggesting that Eduvision could benefit from advanced techniques like Recurrent Neural Networks (RNNs) for more effective behavior analysis.

V. Experimental Validation: The paper's methods have been validated on benchmark datasets (UCF-101, HMDB-51, Kinetics), suggesting a high potential for successful application in real-world scenarios like classrooms.

Incorporating the methodologies and findings from this paper could significantly enhance the Eduvision model's ability to accurately recognize and analyze student behavior in a classroom environment.

The paper's advanced approach to video-based behavior recognition, particularly in dealing with complex video data and ensuring accurate behavior classification, aligns well with the objectives of Eduvision.

4 .3 Figures and Tables for Reference:

General flowchart of behavior recognition.

General Flowchart of Behavior Recognition: This figure illustrates the overall process of behavior recognition as proposed in the paper.

Including it in your report would help in visually explaining the step-by-step methodology of behavior recognition that could be applied to Eduvision.

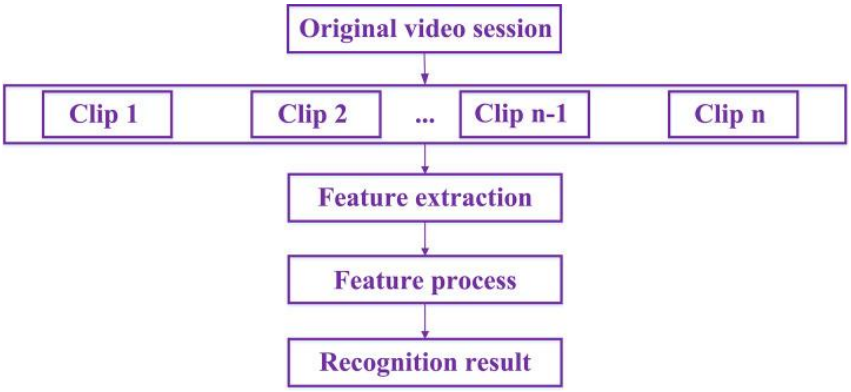


Figure 2 ***General flowchart of behavior recognition.***

Flowchart of the target detection process.

Flowchart of the Target Detection Process: This diagram would show the detailed process of how the target detection phase is integrated before the behavior recognition step.

It's useful for understanding how to focus the Eduvision system on relevant subjects (students) in a video.

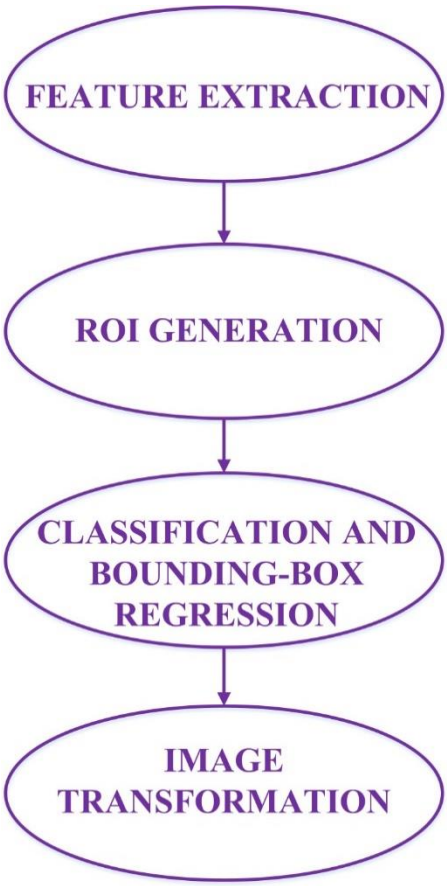


Figure 3 ***Flowchart of the target detection process.***

Image transformation process.

Image Transformation Process: This figure probably demonstrates how images are transformed during the recognition process, which might include steps like cropping, alignment, or normalization.

This can be important for Eduvision when processing classroom video feeds.

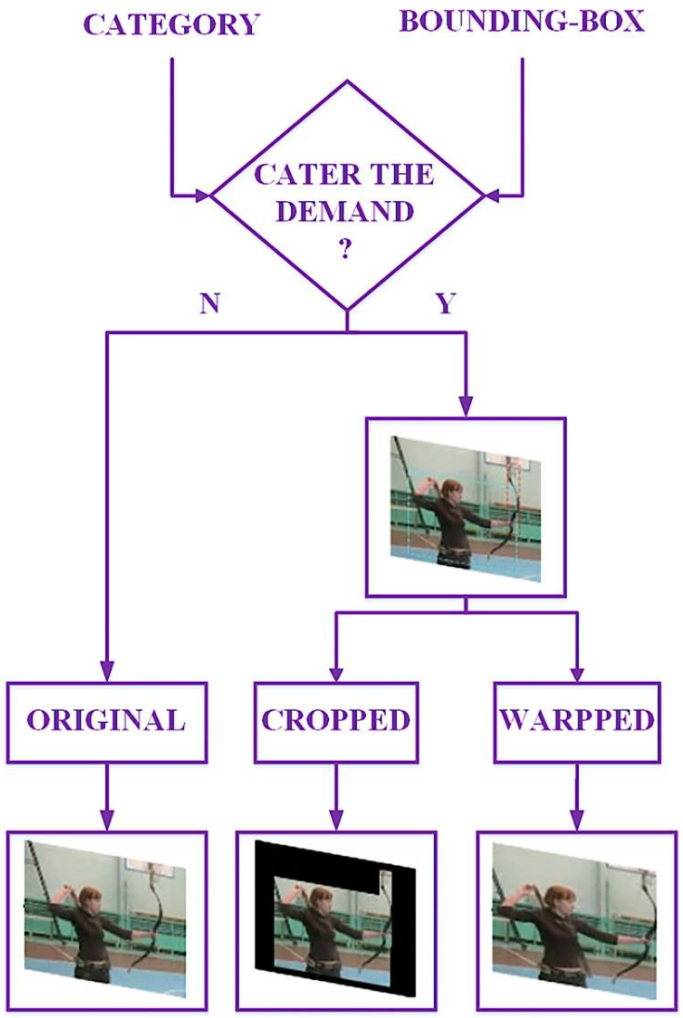


*Figure 4 i**mage transformation process.***

Table 1 & 2: Effect of different alpha values on accuracy.

The effect of different α and different partitions on hmdb-51 about accuracy.

α	Partition1	Partition2	Partition3	Average
0.10	60.6	56.5	58.7	58.6

α	Partition1	Partition2	Partition3	Average
0.25	76.6	73.6	74.9	75.0
0.50	***76.8***	73.8	***75.2***	***75.3***
0.75	76.7	***73.9***	75.1	75.2
0.90	76.7	73.8	75.1	75.2
1.00	76.7	73.8	75.1	75.2

Figure 5 ***Effect of different alpha values on accuracy.***

Table 2

The effect of different α and different partitions on ucf-101 about accuracy.

α	Partition1	Partition2	Partition3	Average
0.1	76.8	77.4	78.4	77.5
0.25	95.4	96.3	95.4	95.7
0.5	95.5	96.3	***95.9***	***95.9***
0.75	***95.7***	***96.4***	95.6	***95.9***
0.9	95.5	96.2	95.7	95.8
1	95.6	96.3	95.8	***95.9***

Figure 6 ***Effect of different alpha values on accuracy.***

Table 7: Accuracy comparison of different methods in behavior recognition.

Table 7

Accuracy comparison of different methods in UCF-101 and HMDB-51.

method	**UCF-101**	**HMDB-51**
Yan [31]	82.4	48.7
Al [32]	85.8	54.9
Du [33]	*86.4*	*53.7*
Diba [34]	*89.7*	*61.1*
Liang [35]	**89.8**	**62.1**
Carreira [29]	**91.7**	**61.1**
Karpathy [14]	**94.3**	**70.9**
Ji[43]	91.1	60.5
Donahue[26]	95.6	74.8
OURSwithoutF.S.S.&F.L.	95.7	75.1

method	**UCF-101**	**HMDB-51**
OURSwithoutF.L.	95.8	75.1
OURSwithoutF.S.S.	95.8	75.2
OURS(all)	96.1	75.4

Figure 7 ***Accuracy comparison of different methods in behavior recognition.***

CHAPTER 3
Architecture Design

5. The Detailed Requirements Specification

5.1 Functional Requirements

The developed EduVision software module fulfills the following functional requirements:

1. Face Orientation Detection:
    - Looking Forward:
        - The software can detect when a student is looking forward, indicating attentiveness towards the lecture.
    - Looking Left:
        - The software can identify when a student is looking left, which may suggest engagement with colleagues or materials on the left side.
    - Looking Right:
        - The software can recognize when a student is looking right, potentially indicating interaction with colleagues or materials on the right side.
    - Looking Down:
        - The software can detect when a student is looking down, which may imply phone usage, note-taking, or disengagement from the lecture.
    - Looking Up:
        - The software can identify when a student is looking up, suggesting possible distraction or lack of focus on the lecture.
2. Phone Detection:
    - Phone Usage:
        - The software can detect when a student is using a phone during the lecture, potentially indicating distraction or disengagement.
    - Phone Obstruction:
        - The software can identify instances where a student is holding a phone in a manner that obstructs their view of the lecture.
3. Hand Raise Detection:
    - The software can detect when a student raises their hand, signifying active participation or a desire to ask a question.
4. Engagement Analysis:
    - Distraction Detection:

        - The software can identify students who are distracted based on their face orientation (looking up, down, left, or right) and phone usage.
    - Focus Detection:
        - The software can recognize students who are focused and attentive based on their face orientation (looking forward) and absence of phone usage.
5. Visualization and Reporting:
    - The software generates informative charts to visualize the analysis results, including:
        - Figure 1: Counts of classified images and detected phones for each face orientation label.
        - Figure 2: Counts of classified images and detected phones for distracted, focused, and hand raised categories.
    - The charts provide a clear overview of student engagement levels and patterns during the video lecture.

5.2 Non-functional Requirements

1. Performance
    - The model should exhibit minimal latency, providing real-time or near-real-time feedback on student behavior detection.
2. Reliability
    - The model should be highly reliable, with a low risk of crashes, errors, or data loss, ensuring consistent monitoring and reporting.
3. Security
    - Implement robust security measures to protect the data and privacy of students and faculty, including secure data transmission, storage, and user authentication.
4. Data Privacy and Compliance
    - Ensure that the model strictly adheres to data protection regulations, maintaining the privacy of student and faculty data.
5. Response Time
    - Define acceptable response times for various software functions to ensure timely alerts and notifications for detected behavior deviations.
6. Ethical Considerations
    - Develop the model with strong ethical considerations, respecting the rights of students and faculty, and promoting a supportive approach to behavior management rather than punitive measures.

5.3 Data Requirements

Our model will process a collection of videos, which can be viewed as a series of images, and its structure is as follows:

Video Input

- **Requirement**: High-resolution video lectures to enable precise detection and analysis.
- **Data Needed**: Continuous recording of classroom settings or virtual lectures.

Person Detection

- **Requirement**: Identify all students within the video frame.
- **Data Needed**: Access to pre-trained Faster R-CNN models.

Face Orientation Classification

- **Requirement**: Classify the direction of each student's face (forward, left, right, up, down).
- **Data Needed**: Integration with MediaPipe's FaceMesh for real-time facial orientation detection.

Hand Raise Detection

- **Requirement**: Detect when a student raises their hands.
- **Data Needed**: MediaPipe's Pose estimation technology to analyze body posture and hand positioning.

Phone Detection

- **Requirement**: Identify students using phones during lectures.
- **Data Needed**: Adaptation of Faster R-CNN models to specifically recognize smartphones within the video frames.

Image and Data Storage

- **Requirement**: Store processed images and categorize them based on detected features.
- **Data Needed**: Sufficient storage for saving cropped images of detected persons, face orientations, hand raises, and phone usage.

Computational Resources

- **Requirement**: Process video data in real-time or near real-time.
- **Data Needed**: Powerful processing capabilities, likely requiring GPU support for deep learning tasks.

Visualization Tools

- **Requirement**: Generate informative charts and visualizations from the processed data.
- **Data Needed**: Software capable of creating and displaying charts based on the analysis results.

5.4 Equipment or software used.

In the development of our EduVision module, we utilized the following equipment and software:

1. Computer:
    - A capable computer with a GPU (Graphics Processing Unit) to run the module smoothly and efficiently.
2. Software and Libraries:
    - Programming Language: Python
    - Key Libraries:
        - OpenCV: Used for video processing and image manipulation tasks.
        - Torch (PyTorch): Employed for loading and running the pre-trained Faster R-CNN model for person and phone detection.
        - Torchvision: Utilized for accessing the pre-trained Faster R-CNN model and its associated weights.
        - MediaPipe: Used for face orientation classification and hand raise detection.
        - NumPy: Employed for numerical computations and array manipulation.
        - Matplotlib: Used for generating visualizations and charts to present the analysis results.
3. Pre-trained Models:
    - Faster R-CNN (fasterrcnn_resnet50_fpn): A pre-trained object detection model used for person and phone detection.
    - FaceMesh (MediaPipe): A pre-trained model used for detecting facial landmarks and classifying face orientations.
    - Pose (MediaPipe): A pre-trained model used for estimating body pose and detecting raised hands.
4. Video Input:
    - The module takes a video lecture as input for analysis. The video file path is specified in the code.

6. Methods We Used to Find Information

We used a wide range of research approaches in our hunt for accurate and thorough information, drawing on the abundance of material found in numerous sources.

Through the implementation of this comprehensive strategy, we made sure that our study goals were thoroughly and comprehensively examined.

This table provides a comparison of the pros and cons of each method:

Table 2 Comparison of the pros and cons of information collection

Method	Pros	Cons
Internet Research	Vast information, convenience, timeliness	Accuracy, overload, bias
Consulting Experts in AI	Specialized knowledge, in-depth understanding, networking	Availability, subjectivity, potential cost
Seeking Guidance from a Supervisor	Experience, mentorship, accountability	Subjectivity, time constraints, dependency

7. Skills the Team Obtained

Throughout the development of the EduVision module, the team acquired and strengthened the following skills:

1. Proficiency in Python Programming:
    - The team gained expertise in using Python as the primary programming language for implementing the EduVision module.
    - They developed skills in utilizing Python libraries such as OpenCV, Torch (PyTorch), Torchvision, MediaPipe, NumPy, and Matplotlib.
2. Knowledge of Computer Vision Techniques:
    - The team acquired knowledge of various computer vision techniques, including object detection, face orientation classification, and pose estimation.
    - They learned how to apply these techniques using pre-trained models like Faster R-CNN and MediaPipe's FaceMesh and Pose models.
3. Experience with Pre-trained Models:
    - The team gained experience in working with pre-trained models, specifically the Faster R-CNN model for person and phone detection and MediaPipe's FaceMesh and Pose models for face orientation classification and hand raise detection.
    - They learned how to load and utilize these models effectively within the EduVision module.

4. Video Processing Skills:
    - The team developed skills in video processing using OpenCV, including reading video frames, applying object detection and classification techniques, and saving processed images.
    - They gained knowledge of video manipulation techniques such as frame skipping and image cropping.
5. Data Analysis and Visualization:
    - The team acquired skills in analyzing the processed data and generating meaningful insights from the results.
    - They learned how to use NumPy for numerical computations and data manipulation.
    - They gained experience in creating informative visualizations and charts using Matplotlib to present the analysis results effectively.
6. Problem-Solving and Adaptation:
    - The team developed problem-solving skills by adapting existing techniques and models to suit the specific requirements of the EduVision module.
    - They learned how to modify and customize the code to handle various scenarios and edge cases encountered during the development process.
7. Collaboration and Project Management:
    - The team strengthened their collaboration skills by working together to develop and integrate different components of the EduVision module.
    - They gained experience in project management, including task allocation, timeline management, and effective communication within the team.

8. Diagrams

8.1 Use Case Diagram

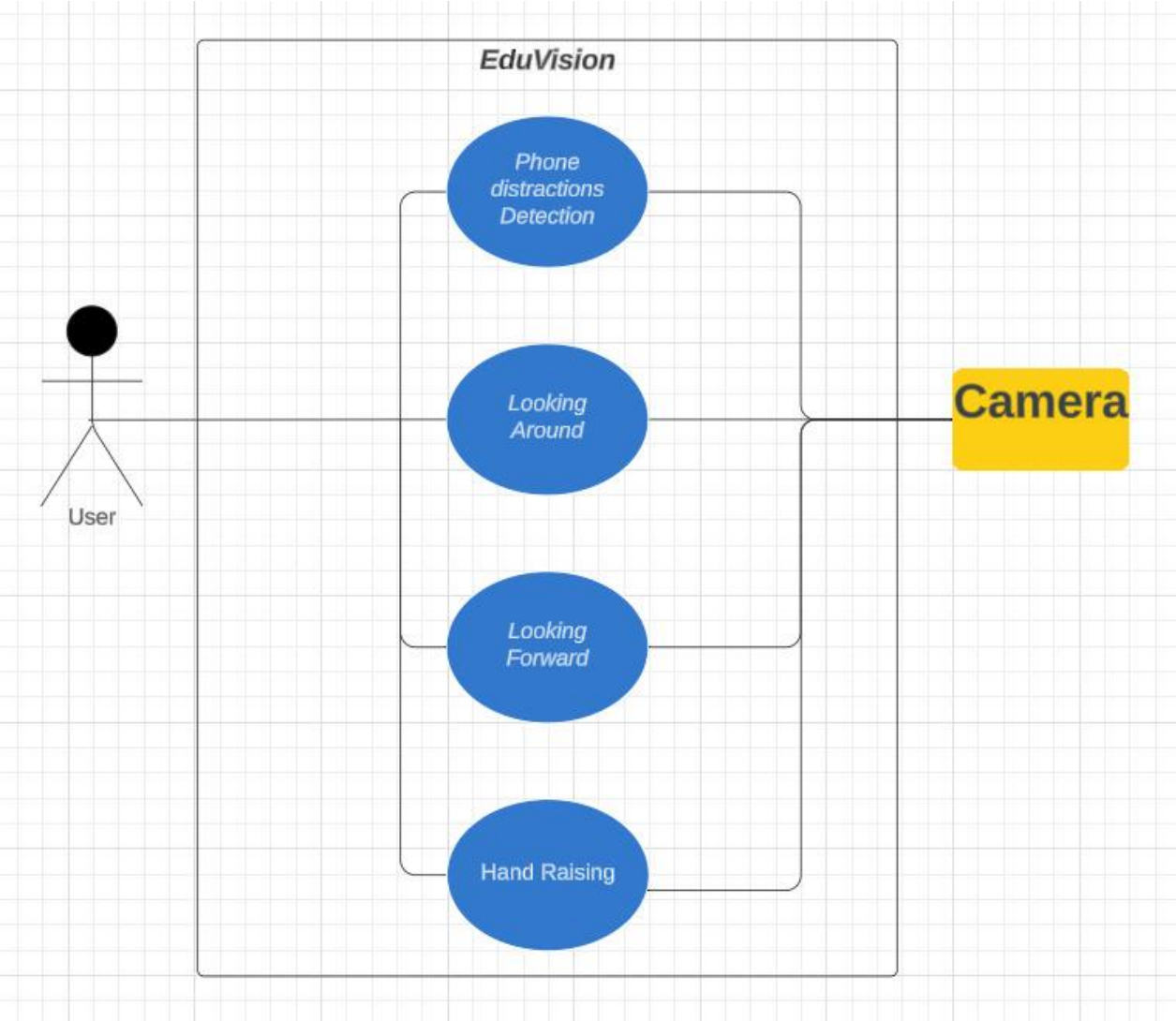


Figure 8 use case diagram.

The above use case diagram has two systems: EduVision and the Camera system.
Within the EduVision system, there are four distinct use cases: Phone Distractions Detection, Looking Around Detection, Looking Forward Detection, and Hand Raising Detection.

8.2 Activity Diagram

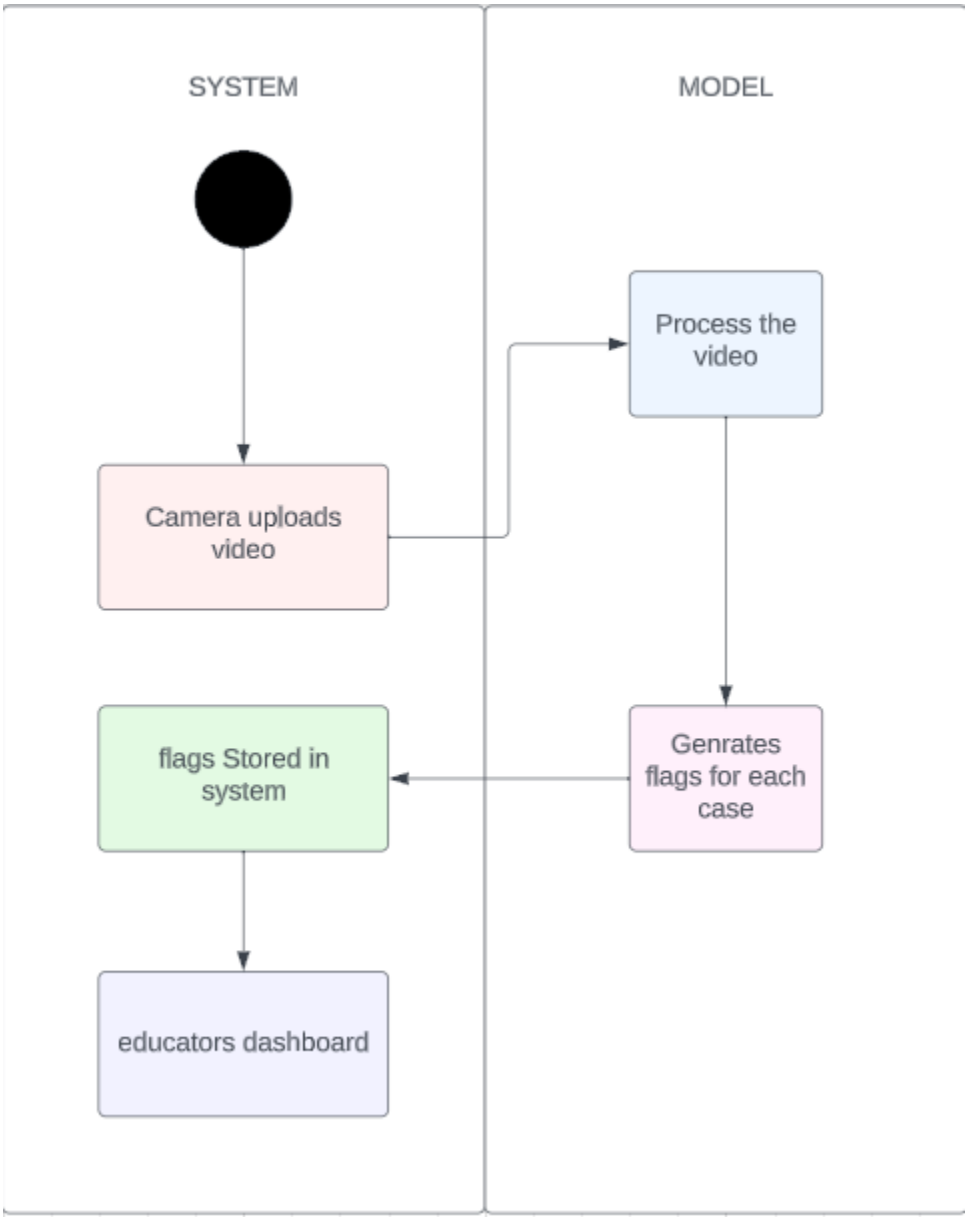


Figure 9 Activity Diagram

CHAPTER 4
Data Gathering

9. Dataset

Dataset:
In the development of the EduVision module, we utilized a dataset generated from the provided video lecture. The video was processed to extract relevant frames and analyze student behaviors and engagement levels. The following specific behaviors and activities were captured and analyzed:

1. Face Orientation:
    - Looking Forward: Indicates that the student is attentive and focused on the lecture.
    - Looking Left: Suggests that the student may be interacting with colleagues or materials on their left side.
    - Looking Right: Indicates potential engagement with colleagues or materials on the student's right side.
    - Looking Up: May imply distraction or lack of focus on the lecture.
    - Looking Down: Could indicate phone usage, note-taking, or disengagement from the lecture.
2. Phone Detection:
    - Phone Usage: Identifies instances where a student is using their phone during the lecture, potentially signifying distraction or disengagement.
3. Hand Raise Detection:
    - Detects when a student raises their hand, indicating active participation or a desire to ask a question.

Data Generation Methodology:

1. Person Detection:
    - The video lecture was processed frame by frame using the Faster R-CNN model to detect persons in each frame.
    - Cropped images of the detected persons were saved for further analysis.
2. Face Orientation Classification:
    - The cropped person images were processed using MediaPipe's FaceMesh model to detect facial landmarks.
    - Based on the detected landmarks, the face orientation of each student was classified into one of the five categories: looking forward, left, right, up, or down.
    - The classified images were saved with the corresponding face orientation labels.
3. Phone Detection:
    - The classified person images were further analyzed using the Faster R-CNN model to detect the presence of phones.
    - Images with detected phones were saved separately, indicating potential phone usage during the lecture.
4. Hand Raise Detection:
    - The cropped person images were processed using MediaPipe's Pose model to estimate body pose and detect raised hands.
    - Images with detected raised hands were saved separately, signifying active participation or inquiry.

Figure 10 Student Looking Down

Figure 11 Student Looking Forward

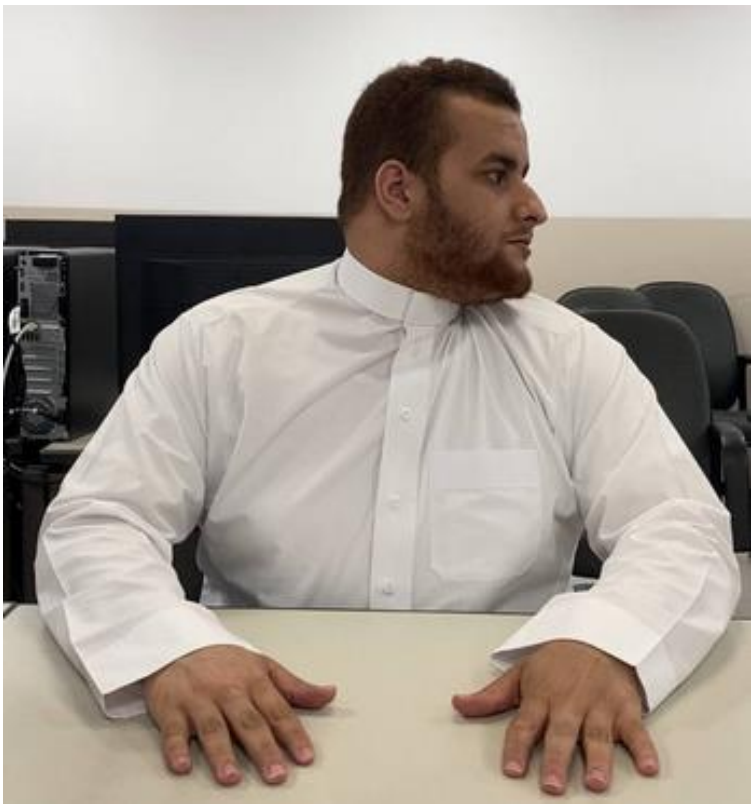


Figure 12 Student Looking Right

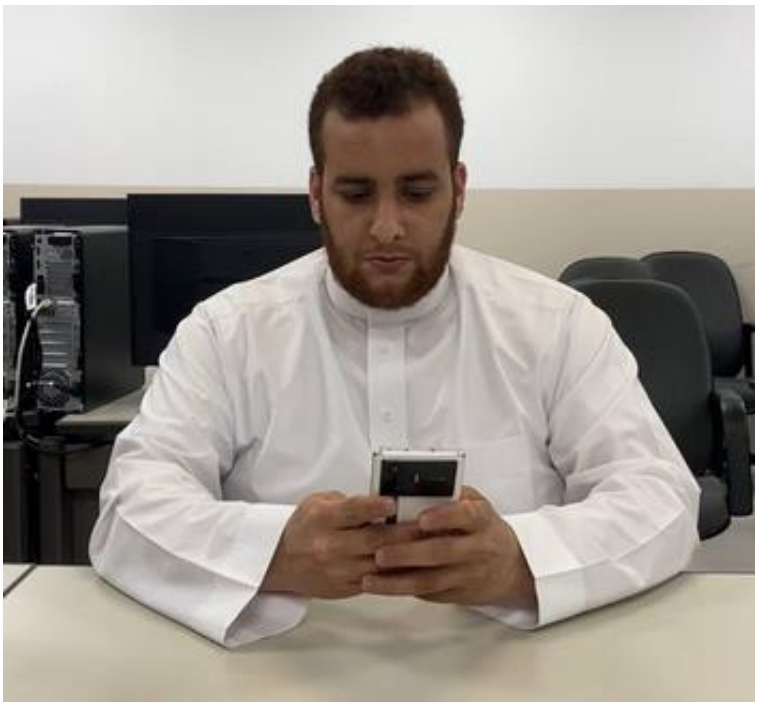


Figure 13 Student Holding the Phone

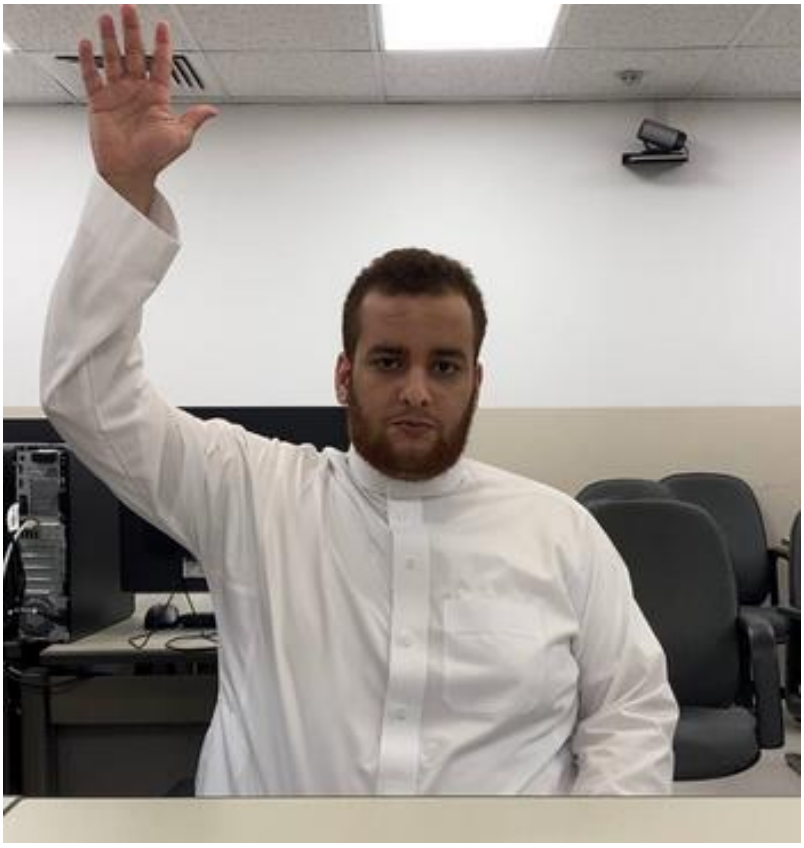


Figure 14 Student Hand Raising

CHAPTER 5
Implementation

10. The main functions

10.1 Detect Persons Function

The detect_persons function is a critical component of our vision-based processing system, designed to identify and segment human figures in video frames. This function leverages cutting-edge object detection algorithms to ensure high accuracy and efficiency in real-time applications.

From YOLOv8 to Faster R-CNN

Initially, our implementation utilized the YOLOv8 model, trained on the COCO dataset, which is renowned for its ability to detect objects in images swiftly. However, it sometimes sacrifices accuracy for speed, particularly in complex scenes or with smaller objects.

To enhance our system's capability and accuracy, we transitioned to using the Faster R-CNN framework with a ResNet-50 backbone. Faster R-CNN stands out for its precision in detecting objects. Unlike YOLOv8, Faster R-CNN employs a two-stage process: the first stage generates region proposals where objects might be located, and the second stage classifies the content of these proposals and refines their boundaries. The integration of the ResNet-50 backbone, known for its deep residual learning framework, further improves the model's ability to learn robust features without compromising the training efficiency due to vanishing gradients.

Operational Mechanism

The primary objective of employing the detect_persons function is to detect as many individuals as possible in each frame. This capability is pivotal for applications requiring accurate crowd analysis or enhanced surveillance measures. Once a person is detected, the function segments the individual from the frame, crops the image around the detected region, and saves it into an 'Output_Segmentation' directory. This systematic saving of segmented images facilitates subsequent processing or review stages.

The input to the detect_persons function is a video stream. As the video progresses, each frame is analyzed independently to ensure that all persons are detected and processed accordingly. This method guarantees that our system is not only effective in identifying individuals but also scalable to handle inputs from multiple sources simultaneously, thereby enhancing its applicability in diverse scenarios.

10.2 classify face orientations Function

The classify_face_orientations function is an advanced component within our image processing suite that specializes in facial keypoint detection. This function utilizes the powerful capabilities of the MediaPipe library, renowned for its robust and efficient face mesh models.

Integration with MediaPipe Face Mesh

Leveraging MediaPipe's face mesh model, this function provides high-fidelity detection of facial landmarks across a wide range of poses and lighting conditions. MediaPipe's face mesh is a holistic, machine learning-based solution that maps 468 3D facial landmarks to deliver precise facial feature detection, which is pivotal for our analysis.

Operational Workflow

Upon invocation, classify_face_orientations retrieves images from the Output_Segmentation directory—these images are outputs from the detect_persons function, which has already isolated frames with individuals. For each image, the function applies the MediaPipe face mesh to detect and map the facial keypoints. This detailed facial landmark detection is crucial for the subsequent classification phase.

Classification Based on Head Orientation

Following the keypoint detection, the function analyzes the spatial arrangement of these keypoints to classify the orientation of the head. This classification may include categories such as 'Looking_Left', 'Looking_Right', 'Looking_up', 'Looking_down', and 'Looking_Forward'. All those classes will be saved in Classified_Images directory. The ability to classify head orientation is essential for applications such as engagement tracking, attention analysis, and interactive systems where user orientation is a key parameter.

10.3 detect hand raised Function

The detect_hand_raised function is a sophisticated module within our image analysis suite that focuses on identifying and counting instances of hand-raised gestures. This function extends beyond simple detection, incorporating a counting mechanism to track the frequency of the hand-raised pose across frames, which is essential for applications requiring gesture recognition and frequency analysis.

Utilizing MediaPipe Pose for Skeletal Tracking

To achieve precise gesture detection, the function utilizes the MediaPipe Pose library. This library is adept at mapping the human body’s skeletal structure in real time with high accuracy. By applying this model, we can extract the coordinates of key body joints—specifically the shoulders, elbows, and wrists—necessary for determining the posture of the hand.

Gesture Detection through Angular Measurement

The core of the gesture detection lies in the calculation of the angle formed at the elbow, using the coordinates of the shoulder, elbow, and wrist. This angular measurement is pivotal; a hand-raised gesture is typically characterized by an angle less than 30 degrees. This specific threshold helps in distinguishing a raised hand from other arm positions effectively.

Operational Workflow

Upon execution, detect_hand_raised accesses the pre-segmented images from the Output_Segmentation directory, where each frame potentially contains an individual. For each frame, the function calculates the aforementioned angle. If the angle is below 30 degrees, indicating a raised hand, the function not only stores this particular frame in the Classified_Images directory under the class Hand_Raised but also increments a counter tracking the number of times a hand-raised gesture occurs.

Enhanced Functionality with Counting Mechanism

The inclusion of a counting mechanism is particularly beneficial in educational or meeting settings, where it's useful to quantify participant engagement by tracking how often individuals raise their hands. This data can be used for further analysis or feedback to enhance interaction and responsiveness in such environments.

10.4 detect phones Function

The detect_phones function serves as a specialized component of our image processing framework, aimed at identifying mobile phones within images. This function extends the capabilities of our object detection system by specifically targeting phone detection in various environments and scenarios.

Utilizing Faster R-CNN with ResNet-50

Similar to the detect_persons function, detect_phones employs the Faster R-CNN model equipped with a ResNet-50 backbone. This combination offers a powerful detection mechanism, leveraging deep learning to identify objects with high precision and reliability. The ResNet-50 backbone is particularly effective due to its deep residual networks that facilitate learning from a vast number of layers without degradation in performance, making it ideal for the complexities of phone detection in diverse settings.

Integration with Classified Images

The function operates on images stored in the Classified_Images directory, which are the outputs from the classify_face_orientations function. This ensures that the input images have

already been processed for facial orientations, which might provide contextual clues enhancing the accuracy of phone detection—such as detecting phones in close proximity to identified faces.

Operational Workflow

Upon execution, detect_phones scans each image within the Classified_Images directory. Using the Faster R-CNN model, it detects the presence of mobile phones by recognizing their typical size, shape, and expected appearance in various contexts. Each detected phone is outlined with a bounding box, and the image can be tagged or flagged for further review or processing, depending on the application requirements.

Output Directory and Naming Convention

Detected images are stored in a newly designated directory named Classified_Images_Detected. Within this directory, the images are further organized into subdirectories. Each subdirectory is named according to the class of the original image from Classified_Images plus the descriptor of the detected item, in this case, "phone". This naming convention and directory structure facilitate easy navigation and retrieval of detected images, making it highly efficient for users to locate specific detections.

Applications and Significance

The ability to detect mobile phones automatically is crucial for security applications, such as in secured facilities where phone usage is restricted. It is also valuable in behavioral studies, where phone usage patterns are analyzed, or in retail and public settings to monitor compliance with policies regarding phone usage.

CHAPTER 6
Evaluations

11. Evaluations

11.1 Evaluation of EduVision System

The evaluation of the EduVision system focuses on its capability to accurately classify and detect student behaviors in a classroom setting, specifically examining the system's recognition accuracy and operational efficacy. The results from the dataset are illustrated in two figures, providing a quantitative overview of behavior classification and phone usage.

Figure 15: Detection of Student Behaviors and Phone Usage

Figure 15 below presents the distribution of various student behaviors alongside detected instances of phone usage. Key observations include:

- **Looking Forward**: Dominantly observed with 1,329 instances, indicating a high level of student engagement directed towards the front.
- **Looking Left, Right, and Up**: Varied counts of these behaviors suggest occasional attention shifts.
- **Phone Detection**: Shows the system's capability to identify phone usage, a critical factor for assessing distractions.

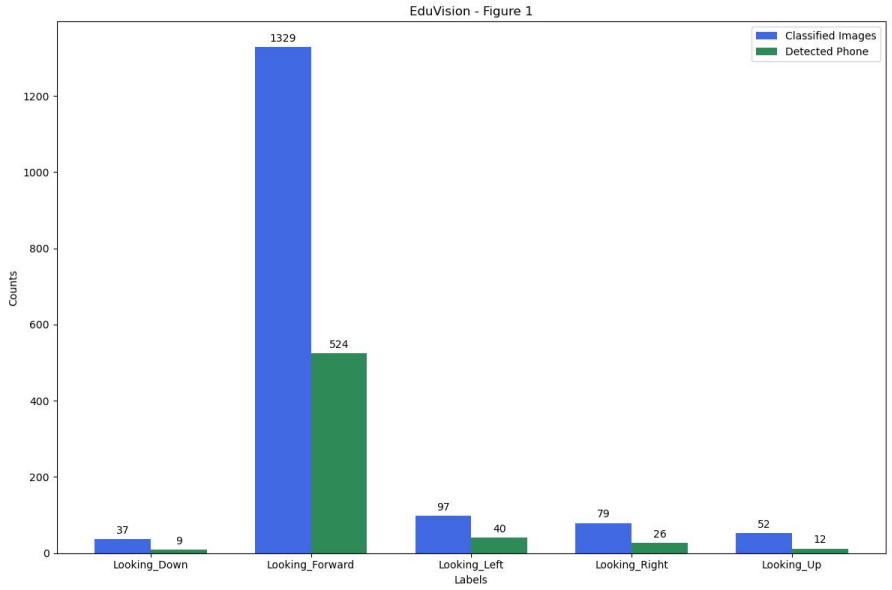


Figure 15 Overview of Classified Behaviors

Figure 16 categorizes behaviors into broader groups:

- **Distracted and Distracted with Phone**: These categories are essential for understanding the frequency and context of student distractions, particularly those involving phone usage.
- **Focused**: Represents the high frequency of students attentively looking forward, underscoring the prevalent engagement within the classroom.
- **Hand Raised**: This category notably reflects the count of unique students who raised their hands, rather than the total number of hand-raising instances, explaining the lower count observed.

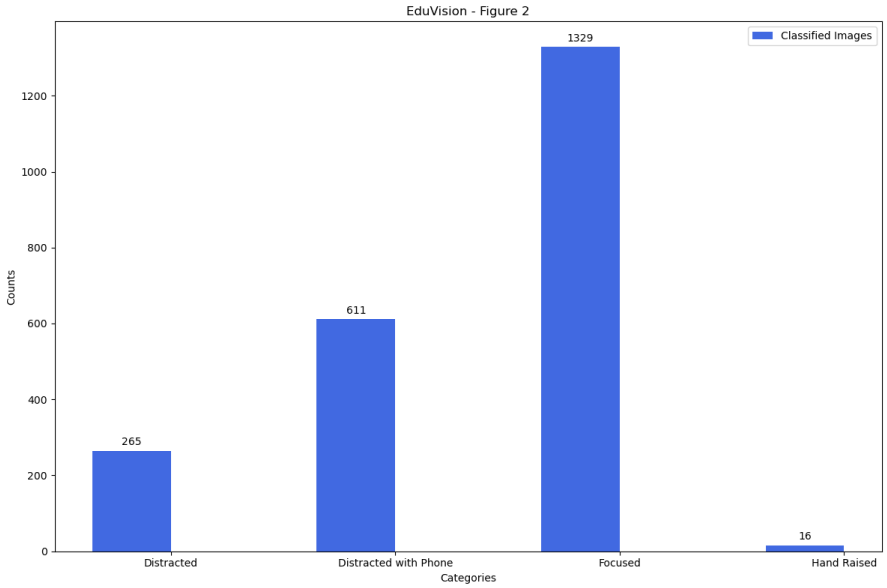


Figure 16 categorizes behaviors into broader groups

Analysis

The significant number of 'Focused' behaviors compared to 'Distracted' categories indicates a generally high level of engagement. The low frequency of 'Hand Raised' behaviors is due to the counting method, which considers only unique occurrences per student rather than all instances. This method provides a conservative estimate but may underrepresent the actual level of interactive behaviors.

CHAPTER 7
Conclusion & Future Work

12. What work do you intend to carry out in the near future

- Create an application that includes the ai model
  - It is full stack react app
  - Includes a database that can be accessed through the app
  - The user interface is easy to use and functional as well
- Monetize the app by selling or providing a SaaS to educational institutions or government bodies.

  This will require a full analysis of the market and then choosing the right business model and approach

13. Conclusion

EduVision marks a transformative step in education by integrating AI and computer vision to enhance classroom learning. This innovative approach provides educators with real-time insights into student engagement, enabling more personalized and effective teaching strategies. The project signifies a shift towards embracing technology in education, preparing students and educators for a tech-integrated future. As EduVision continues to evolve, its potential to influence broader educational policies and adapt to various learning environments remains vast. Ultimately, EduVision is setting a new standard in educational technology, fostering more dynamic, interactive, and tailored learning experiences.

14. References

1. Kadivar, N., Peteri, R., Salah, A. A. (2018). YOLO based Human Action Recognition and Localization. ResearchGate. https://www.researchgate.net/publication/326535574_YOLO_based_Human_Action_Recognition_and_Localization
2. Meng, Y., & Zhang, J. (2022). An effective behavior recognition method in the video session using convolutional neural network. PLOS ONE, 17(8), e0266734. https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0266734