Project Proposal: University Student Attendance Monitoring App

1. Project Title:
University Student Attendance Monitoring App

2. Introduction:
The University Student Attendance Monitoring App is designed to scan photos taken in classrooms, detect students, and count the number of attendees. This application leverages AI-based image processing techniques to automate attendance tracking, reducing manual effort and improving accuracy.

3. Objectives:

- Develop a mobile application using Flutter for both Android and iOS.
- Implement image capture and gallery selection functionality.
- Integrate AI-based image processing for detecting and counting students.
- Provide an intuitive and user-friendly interface.
- Ensure efficient performance with on-device or cloud-based AI models.
- Store attendance records securely for future reference.

4. Technologies Used:

- **Frontend Development:** Flutter (Dart)
- **AI & Image Processing:** Google ML Kit, TensorFlow Lite, or OpenCV
- **Backend (Optional):** Firebase for image storage and attendance records
- **Cloud Services (Optional):** Google Vision API or AWS Rekognition

5. Key Features:

- **Photo Capture & Upload:** Users (professors or administrators) can take a photo using the camera or select an existing image from the gallery.
- **Student Detection & Counting:** AI-based detection to identify the number of students present in the classroom.
- **Attendance Record Storage:** Save attendance data for future reference.
- **User-Friendly Interface:** Clean UI with simple navigation.
- **Results Display:** Shows the detected student count along with the processed image.
- **Optional Cloud Storage:** Ability to save images and detection results for later verification.

6. Target Audience:

- University administration for automated attendance tracking.
- Professors and lecturers for monitoring classroom presence.
- Educational institutions looking to improve attendance management.

7. Expected Outcome:

- A fully functional mobile application capable of detecting and counting students in classroom images.
- An accurate AI-powered solution with minimal processing time.
- A scalable platform with potential for future enhancements like real-time video processing.

8. Timeline:

Phase	Task	Duration
Phase 1	Research & Planning	1 Week
Phase 2	UI/UX Design & Development	2 Weeks
Phase 3	AI Model Integration	2 Weeks
Phase 4	Testing & Optimization	1 Week
Phase 5	Deployment & Launch	1 Week

9. Conclusion:
The University Student Attendance Monitoring App is an innovative and practical solution that combines AI with mobile technology to automate attendance tracking. This project aims to provide a reliable tool for universities to manage attendance records efficiently, minimizing manual work and improving accuracy. Future enhancements could include real-time video processing and cloud-based analytics for more advanced functionalities.

Detecting student heads in classroom images can be done using AI-based image processing techniques. Below are some effective approaches:

1. Methods for Detecting Student Heads

A. Using Deep Learning Models

- **Google ML Kit** (On-device processing, suitable for mobile apps)
- **TensorFlow Lite** (Lightweight deep learning models for mobile)
- **OpenCV with Haar Cascade or DNN** (Classical computer vision approach)

For **head detection**, common deep-learning models include:

- **YOLO (You Only Look Once)** – Fast and accurate object detection.
- **SSD (Single Shot MultiBox Detector)** – Suitable for real-time applications.
- **MTCNN (Multi-task Cascaded Convolutional Networks)** – Good for face detection.

B. Steps to Detect Student Heads

1. **Preprocess Image:** Convert to grayscale and apply noise reduction.
2. **Use a Pre-trained Model:** Detect heads using a deep learning model.
3. **Count the Detected Heads:** Apply non-maximum suppression (NMS) to eliminate duplicate detections.
4. **Store the Results:** Save the detected count and marked image.

---

2. Preventing Poor Image Quality (Blurry or Low-Quality Images)

To prevent capturing unclear images, you can apply **pre-capture and post-capture validation techniques**:

A. Pre-Capture (Before Taking the Picture)

- **Auto-focus & Stabilization:** Enable camera stabilization to prevent motion blur.
- **Lighting Detection:** Ensure proper brightness before capturing.
- **Guidelines for Image Capture:** Display an on-screen guideline (e.g., “Ensure the whole classroom is visible”).

B. Post-Capture (After Taking the Picture)

- **Blur Detection:** Use OpenCV’s Laplacian variance method to check for blur:
- **Face & Head Detection Check:** If the detected count is low or unreliable, prompt the user to retake the picture.

Would you like a more detailed breakdown of integrating these techniques into your Flutter app? 🚀

If a student keeps their head down, the AI model might face difficulty detecting them. However, there are ways to improve accuracy and handle such cases effectively. Let's break it down:

1. Handling Students Keeping Their Heads Down

AI-based models primarily rely on detecting **head features** (e.g., face, hair, or shape). If a student lowers their head, their face may be partially or fully obscured, making detection challenging. Some solutions include:

A. Multi-Angle Detection

- Instead of only using **face detection**, implement **head detection** models trained on various head orientations (top view, side view, back view).
- **Solution:** Use a deep-learning-based object detector like **YOLO** or **Faster R-CNN**, which can detect human heads from different angles.

B. Body Part Detection

- If the head is not visible, try detecting **upper body features** (shoulders, torso, posture).
- **Solution:** Pose estimation models like **OpenPose** or **MediaPipe Pose** can identify body key points, even if faces are hidden.

C. Multiple Image Capture

- Take **multiple images** or allow slight movement to detect students who might initially be hidden.
- **Solution:** Implement an **auto-capture** system that takes a few frames instead of a single snapshot.

---

2. How Do We Know It's a Child's Head?

To classify a detected object as a **child's head**, AI models use several techniques:

A. Feature-Based Head Recognition

- The model looks for specific features like:
  - **Head shape**
  - **Hair texture**
  - **Facial proportions**
  - **Skin tone differences**
  - **Ear and forehead detection**
- **Solution:** Train the model on a dataset containing various **age groups**, ensuring it learns how to differentiate between adults and children.

B. Size & Proportion Analysis

- Children's heads are generally **smaller** and **rounder** than adults'.
- **Solution:** Use **bounding box dimensions** and compare proportions relative to other detected heads.

C. Contextual Information

- If the classroom has students of a certain age range, the model can assume **most detected heads belong to children**.
- **Solution:** Implement AI models with a **predefined classroom environment** to improve accuracy.

D. Age Estimation Models

- **Face detection + Age estimation AI** can analyze detected faces and determine an approximate age.
- **Solution:** Use pre-trained models like:
  - **DeepFace**
  - **AgeNet**
  - **Face++ Age Detection**

Final Approach: Combining Techniques

To make detection as accurate as possible:

1. **Use multiple AI models** (head detection, pose estimation, face detection).
2. **Train with diverse datasets** (including different head angles and positions).
3. **Apply pre-processing techniques** (enhancing image clarity, contrast adjustments).
4. **Allow re-scanning** if the initial detection is unclear.

Would you like a **flowchart or visual diagram** to explain this more clearly? 🚀

A project proposal for partial fulfilment of the course unit CSC3232- Group Project for the degree of Applied Mathematics and Computing

University Student Attendance Monitoring App

Name of the students:

Name & Reg. No

Supervisor:

Department of Physical Science

Faculty of Applied Science

University of Vavuniya

March 2024

Declaration

I hereby declare that the project proposal submitted for evaluation of course module CSC3232 leading to the award of a Bachelor of Science in Applied Mathematics and Computing is entirely our own work, and the contents taken from the work of others have been cited and acknowledged within the text. This proposal has not been submitted for any degree at this University or any other institution.

……..

[Students Name & Reg. No]

I recommend the project to be carried out by the students,

[Name]

Supervisor,

Lecturer / Senior Lecturer,

Department of Physical Science,

Faculty of Applied Science.

Introduction

The University Student Attendance Monitoring App is designed to scan photos taken in classrooms, detect students, and count the number of attendees. This application leverages AI-based image processing techniques to automate attendance tracking, reducing manual effort and improving accuracy.

Objectives

- Develop a mobile application using Flutter for both Android and iOS.
- Implement image capture and gallery selection functionality.
- Integrate AI-based image processing for detecting and counting students.
- Provide an intuitive and user-friendly interface.
- Ensure efficient performance with on-device or cloud-based AI models.
- Store attendance records securely for future reference.

Benefits of this Research

- Eliminates manual attendance tracking, saving time and effort.
- Reduces human errors in attendance counting.
- Provides automated and real-time attendance monitoring.
- Enhances transparency and security of attendance records.
- Future scalability for real-time video processing.

Background

Review on the Existing Systems

Traditional attendance monitoring methods include manual roll calls and RFID-based or biometric systems. However, these methods have limitations such as time consumption, risk of proxy attendance, and the need for expensive hardware. AI-powered image processing provides a more efficient and cost-effective solution.

Materials and Methods

Brief Description of Proposed System Design

1. **Photo Capture & Upload:** Users (professors or administrators) can take a photo using the camera or select an existing image from the gallery.
2. **Student Detection & Counting:** AI-based detection to identify the number of students present in the classroom.
3. **Attendance Record Storage:** Securely save attendance data for future reference.
4. **User-Friendly Interface:** Clean UI with simple navigation.
5. **Results Display:** Shows the detected student count along with the processed image.
6. **Optional Cloud Storage:** Ability to save images and detection results for later verification.

Technologies Used

- **Frontend Development:** Flutter (Dart)
- **AI & Image Processing:** Google ML Kit, TensorFlow Lite, or OpenCV
- **Backend (Optional):** Firebase for image storage and attendance records
- **Cloud Services (Optional):** Google Vision API or AWS Rekognition

Expected Results

- A fully functional mobile application capable of detecting and counting students in classroom images.
- An accurate AI-powered solution with minimal processing time.
- A scalable platform with potential for future enhancements like real-time video processing.

Timeline of the Research

Phase	Task	Duration
Phase 1	Research & Planning	1 Week
Phase 2	UI/UX Design & Development	2 Weeks
Phase 3	AI Model Integration	2 Weeks
Phase 4	Testing & Optimization	1 Week
Phase 5	Deployment & Launch	1 Week

References

[List any sources used for research, AI model references, or related work here.]