EduVision: A Deep Learning-Based Computer Vision System for Detecting Observable Classroom Actions to Support Data-Informed Teaching

Author: Sanjay Chhetri — High School Math Teacher & Data Science Practitioner
GitHub Repository: https://github.com/sanjaykshetri/edu-vision-action-detection

Executive Summary

EduVision is a deep learning capstone project designed to demonstrate the application of computer vision in the K-12 education context. The system classifies observable student actions in classroom images (e.g., hand-raised, writing, looking at board) to support future analytics in teaching effectiveness. The project intentionally avoids mental-state inference and instead focuses solely on ethical, objective, observable actions.

1. Problem Statement

Measuring classroom engagement is challenging due to heavy reliance on subjective teacher perception. A teacher cannot continuously monitor visible participation signals for an entire classroom. This project seeks to explore whether a computer-vision model can classify observable physical actions from static images, enabling future classroom analytics and decision support.

2. Stakeholders

Stakeholder	Value Proposition
Teachers	Receive supplemental analytic support to observe participation trends.
Administrators	Gain insight into classroom instructional efficacy at scale.
Students	Benefit from more responsive, adaptive instruction.
EdTech Developers	Opportunity to integrate model into dashboards and products.

3. Constraints & Ethical Boundaries

• Privacy: No minors or identifiable real students will be used in dataset collection.
• Dataset must be manually curated due to lack of preexisting aligned data.
• GPU compute may be required for training (Paperspace recommended).
• The model will NOT attempt emotional inference or automated engagement scoring.

4. Project Timeline (M3 – Visual Progress Graphic)

Phase	Description

Data Collection	Scrape images via Pexels API and hand-label (~2 days)
Model Training	Train ResNet-18 model using PyTorch (~1–2 days)
Evaluation	Compute validation metrics and confusion matrix (~1 day)
Deployment	Deploy via Streamlit web app (~1 day)
Packaging	GitHub documentation + PDF + walkthrough (~1 day)

5. Data Sources

• Pexels API – royalty-free classroom images
• Unsplash / Pixabay for additional variety
• Optional: staged images using consenting adults
Target Dataset Size: ~300 images (3-class), 80/20 train–validation split

6. Planned Technical Approach

1■■ Collect dataset using automated python scraper
2■■ Label dataset into 3 categories: hand_raised, writing, looking_board
3■■ Train CNN using transfer learning (ResNet-18) in PyTorch
4■■ Evaluate with accuracy %, loss curve, confusion matrix
5■■ Deploy inference system using Streamlit

7. Success Criteria

• Validation accuracy ≥ 70%
• Functional Streamlit demo—image upload returns predicted class
• GitHub repo includes install instructions and code
• Case study PDF and notebook clearly tell the story

8. Final Deliverables

• Trained PyTorch model (.pt file)
• GitHub repository
• Streamlit app deployment
• Project notebook and walkthrough
• Final PDF case study & README summary