Exam Monitoring System: Real-Time Cheating Detection Using YOLOv11

May 2025

Abstract

The Exam Monitoring System employs the YOLOv11 model to detect cheating behaviors in real-time via webcam, classifying actions as "cheating" or "good." This report outlines the system's development, including dataset acquisition, model training, and live inference, achieving high accuracy and low latency for scalable proctoring.

1 Introduction

Academic integrity is paramount in examinations, yet traditional proctoring is resource-intensive. The Exam Monitoring System automates cheating detection using YOLOv11, processing live webcam feeds to identify suspicious behaviors. Objectives include real-time detection, high accuracy, and user-friendly proctoring.

2 System Architecture

The system comprises input processing, inference, and visualization, optimized for GPU performance.

2.1 Input Processing

Webcam frames are captured via OpenCV, resized to 640x640, and normalized for YOLOv11 input.

2.2 Inference

YOLOv11 classifies behaviors as "cheating" or "good" with a 0.25 confidence threshold, leveraging a Tesla P100 GPU for 8.0 ms inference per frame.

2.3 Output Visualization

Results are annotated with bounding boxes and displayed in real-time, with logs saved for review.

3 Technical Implementation

Implemented in Python 3.11.11 on Kaggle, using YOLOv11, OpenCV, Roboflow, NumPy, Pandas, and PyTorch.

3.1 Dataset

Sourced from Roboflow ("Exam-Monitoring-System-v2," version 1), with annotated images and videos.

3.2 Training

YOLOv11 nano was fine-tuned over 100 epochs:

```
!yolo task=detect mode=train data={dataset.location}/data.yaml
   model="yolo11n.pt" epochs=100 imgsz=640
```

3.3 Inference

Live inference code:

```
from ultralytics import YOLO
import cv2

model = YOLO('best.pt')
cap = cv2.VideoCapture(0)

while True:
    ret, frame = cap.read()
    if not ret:
        break
    results = model.predict(source=frame, conf=0.25, save=False,
       verbose=False)
    annotated_frame = results[0].plot()
    cv2.imshow("YOLOv8 Live Inference", annotated_frame)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

cap.release()
cv2.destroyAllWindows()
```

3.4 Hardware

Tesla P100-PCIE-16GB, CUDA 12.6, 16 GB GPU memory.

4 Performance

Evaluated on 'videoplayback.mp4' (5119 frames): - Preprocessing: 1.8 ms. - Inference: 8.0 ms. - Postprocessing: 1.1 ms. - Total: 10.9 ms (90 FPS). Accurate classification of "cheating" (frames 1–4967) and "good" (frames 4968–5119).

5 Challenges and Solutions

- **Dataset Quality**: Enhanced via Roboflow augmentation. - **Latency**: Optimized with 640x640 input and GPU. - **Lighting**: Addressed through normalization and diverse training data.

6 Future Enhancements

- Multi-camera support. - Temporal behavioral analysis. - Cloud deployment. - Proctor dashboard.

7 Conclusion

The system achieves real-time cheating detection with YOLOv11, offering a scalable solution for academic integrity.

Acknowledgments

Thanks to Kaggle, Ultralytics, and Roboflow for resources and support.

References

[1] Ultralytics, "YOLOv11 Documentation," `https://docs.ultralytics.com`, 2025.

[2] Roboflow, "Roboflow Documentation," `https://docs.roboflow.com`, 2025.