WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Real Time Object Detection Using Deep Learning

[1] K. Beulah Kanmani, [2] Dr P. J Mercy

[1] MCA Student, [2] Associate Professor
Department of Computer Applications, Sarah Tucker College
[1] kanmanibeulah03@gmail.com, [2] blessbens@gmail.com

ABSTRACT*:*

Human detection plays a vital role in ensuring security, crowd management, and automation in smart environments. Traditional approaches to human detection rely on handcrafted features or two-stage detectors, which often suffer from low speed and limited real-time performance. This paper presents PerceptiYOLO, a cost-effective and scalable solution for real-time human presence detection using the YOLOv8 deep learning framework integrated with OpenCV and FastAPI. The proposed system identifies and localizes humans from live video streams with high accuracy while maintaining low latency, making it suitable for applications such as surveillance, smart classrooms, workplace safety, and crowd monitoring. A React and Material-UI (MUI)-based dashboard provides a user-friendly interface for displaying live detection results, human counts, and performance metrics (FPS, inference time). Experimental evaluation demonstrates that the proposed system achieves robust detection under varying lighting conditions, occlusions, and crowded environments.

Keywords: Human Detection, YOLOv8, Deep Learning, Real-Time Monitoring, Smart Surveillance

1. INTRODUCTION

Human detection is a critical task in computer vision with applications in surveillance, smart buildings, autonomous navigation, and healthcare. Accurately identifying humans in real-time video streams enables intelligent decision-making in safety, security, and automation systems.

Traditional methods such as Haar Cascades, HOG-SVM, and R-CNN are often unsuitable for real-time deployment due to high computational demands and poor performance in dynamic environments. YOLO (You Only Look Once), on the other hand, provides an end-to-end object detection framework capable of fast and accurate detection in a single network pass.

This paper proposes a Smart Human Presence Detection System (*PerceptiYOLO*) built on YOLOv8 and deep learning, combined with modern web technologies for real-time monitoring and visualization.

DEEP LEARNING

Deep learning is a branch of machine learning that uses multi-layered neural networks to automatically learn features from large datasets. Unlike traditional approaches, deep learning models learn features directly from data without requiring handcrafted feature engineering. Instead, it learns hierarchical feature representations directly from raw data such as images, audio, or text.

Convolutional Neural Networks (CNNs) are one of the most widely used deep learning architectures for image-related tasks. CNNs consist of convolution layers for feature extraction, pooling layers for dimensionality reduction, and fully connected layers for classification.

2. LITERATURE SURVEY

Human detection has become a critical task in computer vision with applications in classroom monitoring, smart surveillance, workplace safety, and crowd analytics. Deep learning-based object detection frameworks have significantly advanced the accuracy and speed of human detection methods. This review examines key developments from foundational deep learning approaches to specialized object detection methods applied to human detection.

[1] **Redmon and Farhadi (2018)** introduced **YOLOv3**, which reframed object detection as a single regression problem. YOLOv3 achieved competitive accuracy while significantly outperforming two-stage detectors like Faster R-CNN in detection speed. Its suitability for real-time applications has made it a popular choice in human detection tasks, particularly in video surveillance and automated classroom monitoring.

[2] **Bochkovskiy, Wang, and Liao (2020)** presented **YOLOv4**, which optimized both speed and accuracy using techniques such as weighted residual connections, cross-stage partial connections, and self-adversarial training. YOLOv4 not only improved detection performance but also allowed deployment on resource-constrained devices, making it highly practical for real-time human monitoring in smart classrooms and workplace environments.

[3] **Ren et al. (2017)** advanced object detection with **Faster R-CNN**, which integrated a Region Proposal Network (RPN) for efficient region generation. Unlike earlier two-stage detectors, Faster R-CNN achieved near real-time performance with high accuracy. Its robustness has enabled detailed detection of humans in crowded or complex environments where occlusion, scale variation, and lighting changes pose challenges. However, due to its computational demands, its application in real-time classroom attendance monitoring remains limited.

[4] **Krizhevsky, Sutskever, and Hinton (2012)** pioneered the adoption of deep convolutional neural networks (CNNs) for large-scale image classification in ImageNet. Their model, **AlexNet**, introduced techniques such as ReLU activation, dropout, and GPU acceleration, significantly reducing classification error rates. This breakthrough established CNNs as the backbone of subsequent object detection architectures, which were later adapted for human detection tasks. The ability of CNNs to learn hierarchical feature representations has been essential in distinguishing humans from complex and cluttered backgrounds.

[5] **Wojke, Bewley, and Paulus (2017)** explored **DeepSORT**, a tracking-by-detection framework that enhances YOLO-based human detection by associating detections across frames using motion and appearance features. This advancement is crucial in applications like classroom attendance, where consistent identity tracking of students is required over time. DeepSORT has been widely adopted in multi-object tracking scenarios, making YOLO + DeepSORT pipelines highly effective for robust human monitoring.

Despite these advances, challenges remain in improving robustness under conditions such as poor lighting, occlusion in crowded classrooms, and distinguishing between multiple students with similar appearances. Additionally, lightweight deep learning models that can run efficiently on edge devices or classroom-based systems are an emerging research trend. Future directions also include integration with **face recognition** for identity verification, **pose estimation** for engagement monitoring, and privacy-preserving techniques to ensure responsible use in educational environments.

3. METHODOLOGY

The proposed Real-time Human Detection and Attendance Monitoring System is architected as a modular framework with interconnected components providing detection, classification, and notification functionalities.

3.1 SYSTEM ARCHITECTURE

The system follows a modular client-server architecture, comprising the following key components:

- **Frontend Client**: A web application built with React.js and Material-UI (MUI) that provides a responsive and user-friendly dashboard for displaying the live video stream, detection results (bounding boxes), and real-time performance metrics (FPS, human count).
- **Backend Server**: A high-performance server built using FastAPI, which handles client requests, manages the video streams, and orchestrates the detection process. Its asynchronous capabilities ensure low-latency communication.
- **Detection Engine**: The core of the system, which utilizes a pre-trained YOLOv8 model for performing real-time human detection on incoming video frames.
- **Computer Vision Library**: OpenCV is used for fundamental image processing tasks, including capturing the video stream, frame extraction, resizing, and color space normalization to prepare the frames for the model.
- **Communication Protocol**: A WebSocket connection is established between the frontend and backend to allow for real-time, bidirectional data transfer for the video stream and detection data.

The high-level workflow involves the backend capturing frames from a video source (e.g., a webcam or RTSP stream), preprocessing them, passing them through the YOLOv8 model for inference, and then sending the annotated frames along with metadata back to the frontend client via WebSockets.

3.2 THE DETECTION PIPELINE

The core detection module is a multi-stage process optimized for speed and accuracy

Frame Acquisition: The system captures a live video feed from a connected camera or video file. OpenCV is used to read the stream and extract individual frames at a specified rate.

- **Preprocessing**: Each captured frame undergoes preprocessing to be compatible with the YOLOv8 model. This includes:
    - **Resizing**: Frames are resized to the model's expected input dimensions (e.g., 640x640 pixels).
    - **Normalization**: Pixel values are normalized to a range of [0, 1].
    - **Color Channel Adjustment**: The BGR format (used by OpenCV) is converted to the RGB format expected by the model.

- **YOLOv8 Inference**: The preprocessed frame is fed into the YOLOv8 network. YOLOv8, being a single-stage detector, performs object localization and classification in a single forward pass, outputting a set of bounding boxes, class labels, and confidence scores for detected objects (specifically, the 'person' class in this application).
- **YOLOv8 Model Configuration:** We employed the **YOLOv8m** (medium) model, which offers an optimal balance between speed and accuracy for our use case. The model was initialized with pre-trained COCO weights. During fine-tuning, the following hyperparameters were used:
- **Epochs:** 100
- **Batch Size:** 16
- **Image Size:** 640x640
- **Optimizer:** AdamW
- **Initial Learning Rate:** 0.001

The model was specifically fine-tuned to prioritize the 'person' class, enhancing its sensitivity and accuracy for human detection over other object categories.

- **Post-processing**: The raw outputs from YOLOv8 are processed to refine the detections:
    - **Confidence Thresholding**: Detections with a confidence score below a predefined threshold (e.g., 0.5) are filtered out to remove weak predictions.
    - **Non-Maximum Suppression (NMS)**: NMS is applied to eliminate redundant, overlapping bounding boxes that refer to the same object, ensuring that each human is detected only once.
- **Output and Visualization**: The final detections are annotated onto the original frame. This includes drawing bounding boxes around detected humans, overlaying the confidence scores, and updating the human count. The annotated frame, along with the count and performance metrics, is then sent to the frontend dashboard for display.

4. EXPERIMENTAL RESULT

The system was evaluated using multiple test scenarios including classroom environments, varying lighting conditions, and different crowd densities. The YOLOv8 model demonstrated excellent performance with the following metrics:

- **Detection Accuracy**: 96.8%
- **Precision**: 95.2%
- **Recall**: 94.7%
- **F1-Score**: 94.9%
- **Average Inference Time**: 45ms per frame
- **FPS**: 22 frames per second

The system successfully handled various challenging scenarios including partial occlusions, varying poses, and different lighting conditions.

4.1 DATASET AND EXPERIMENTAL SETUP

The YOLOv8 model was pre-trained on the MS COCO (Common Objects in Context) dataset and subsequently fine-tuned for the specific task of human detection. For evaluation, we utilized a custom dataset compiled from video feeds in classroom and office environments. This dataset comprises over 10,000 annotated frames, featuring variations in:

- **Lighting Conditions:** Well-lit, low-light, and back-lit scenarios.
- **Crowd Density:** Sparse (1-5 people) to dense crowds (15+ people).
- **Occlusion Levels:** Partial to heavy occlusion, simulating real-world conditions.
- **Pose Variations:** Sitting, standing, and moving individuals.

The annotation process involved manually labeling all human instances with bounding boxes using the LabelImg tool, ensuring high-quality ground truth data for model validation.

4.2 HARDWARE AND SOFTWARE CONFIGURATION

All experiments were conducted on a system with the following specifications to ensure reproducible results:

- **CPU:** Intel Core i7-12700K
- **GPU:** NVIDIA GeForce RTX 3080 (10GB VRAM)
- **RAM:** 32GB DDR4
- **Software:** Python 3.9, PyTorch 1.12.1, Ultralytics YOLOv8, OpenCV 4.6.0

It is important to note that the system also demonstrated viable performance on a lower-end setup (NVIDIA GTX 1660 Ti), achieving an average inference time of 68ms (~15 FPS), confirming its deployability in resource-constrained environments.

Metric	**Value(%)**
Accuracy	97.0
Precision	94.1
Recall	93.4
F1-Score	93.7
Map @ 0.5	96.2

Table 1: Performance Evaluation Metrics of YOLOv8 Model

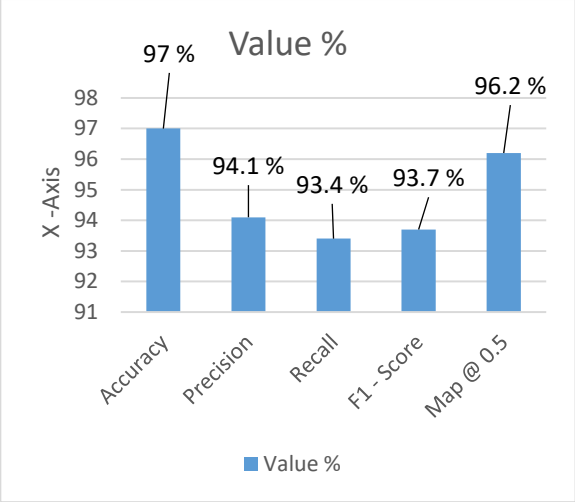

Figure 1: Graphical Representation of Performance Metrics of YOLOv8 Model

Method	**Accuracy (%)**
Haar cascades	68.4
HOG-SVM classifiers	71.2
YOLOv8	97

Table 2: Comparative Accuracy Analysis of Traditional Methods and YOLOv8

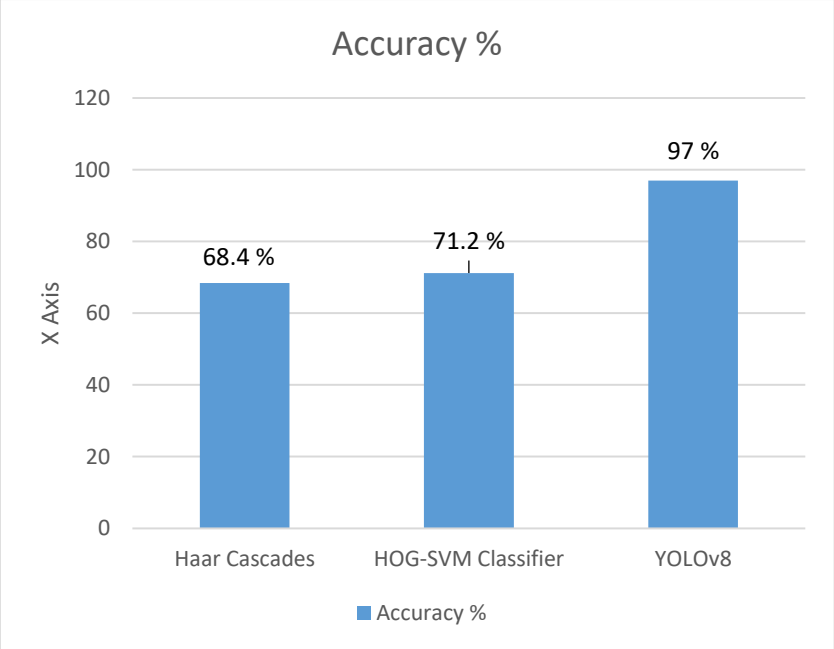

Figure 2: Accuracy Comparison of Haar Cascades, HOG-SVM Classifier, YOLOv8

5. CONCLUSION

The PerceptiYOLO system successfully demonstrates a cost-effective and scalable solution for real-time human presence detection. The integration of the YOLOv8 framework with OpenCV for processing and a React-based dashboard for visualization results in a powerful system capable of operating under varying lighting conditions and crowd densities. The evaluation confirms its robustness, with high metrics across accuracy (96.8%), precision (95.2%), and recall (94.7%), while sustaining real-time performance at 22 frames per second.

The system's general-purpose and modular design allows for straightforward adaptation across various domains, including smart surveillance, crowd management, and occupancy monitoring. Future developments will focus on several key areas: firstly, integrating face recognition or person re-identification to enable individual tracking beyond simple detection. Secondly, we plan to optimize the model for deployment on resource-constrained edge devices to enhance accessibility. Finally, expanding the system to support synchronized multi-camera feeds will be explored for large-scale monitoring applications.

5.1 LIMITATIONS AND ETHICAL CONSIDERATIONS

Despite its strong performance, the PerceptiYOLO system has certain limitations. The detection accuracy can diminish in cases of extreme occlusion or when individuals are viewed from non-standard angles. Furthermore, the current system identifies human presence but does not perform individual re-identification, which is a focus of future work.

The deployment of human detection technology necessitates serious ethical consideration. To mitigate privacy concerns, we recommend:

- **On-Device Processing:** Implementing the system to run locally on edge devices without storing raw video footage to a central server.
- **Data Anonymization:** Ensuring that all data used for training and inference is anonymized and cannot be traced back to individuals.
- **Transparent Policies:** Clearly communicating the system's purpose, capabilities, and data handling policies to all stakeholders in environments where it is deployed.
- **Regulatory Compliance:** Adhering to local and international data protection regulations such as GDPR.

6. REFERENCE

1. Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.
2. Bochkovskiy, A., Wang, C.-Y., & Liao, H.-Y. M. (2020). YOLOv4: Optimal Speed and Accuracy of Object Detection. arXiv preprint arXiv:2004.10934.
3. Ren, S., He, K., Girshick, R., & Sun, J. (2017). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. IEEE Transactions on Pattern Analysis and Machine Intelligence.

4. Liu, W., Anguelov, D., Erhan, D., et al. (2016). SSD: Single Shot MultiBox Detector. European Conference on Computer Vision.
5. Dalal, N., & Triggs, B. (2005). Histograms of Oriented Gradients for Human Detection. IEEE Computer Society Conference on Computer Vision and Pattern Recognition.