TYPE Original Research
PUBLISHED 22 April 2025
DOI 10.3389/fcomp.2025.1535775

Computer vision and AI-based cell phone usage detection in restricted zones of manufacturing industries

Uttam U. Deshpande[1]*, Supriya Shanbhag[1], Ramesh Koti[1], Ameet Chate[2], Sudhindra Deshpande[3], Rudragoud Patil[4], Pavan G. Kulkarni[1], Neha S. Ganiger[3] and Varad A. Rasane[1]

[1]Department of Electronics & Communication Engineering, KLS Gogte Institute of Technology, Belagavi, India, [2]Department of MBA, KLS Gogte Institute of Technology, Belagavi, India, [3]Department of Information Science and Engineering, KLS Gogte Institute of Technology, Belagavi, India, [4]Department of Computer Science and Engineering, KLS Gogte Institute of Technology, Belagavi, India

Check for updates

OPEN ACCESS

EDITED BY
Francisco Gomez-Donoso,
University of Alicante, Spain

REVIEWED BY
Ruiheng Zhang,
Beijing Institute of Technology, China
Felipe Trujillo-Romero,
Universidad de Guanajuato, Mexico

*CORRESPONDENCE
Uttam U. Deshpande
✉ uttamudeshpande@gmail.com

RECEIVED 27 November 2024
ACCEPTED 07 April 2025
PUBLISHED 22 April 2025

CITATION
Deshpande UU, Shanbhag S, Koti R, Chate A, Deshpande S, Patil R, Kulkarni PG, Ganiger NS and Rasane VA (2025) Computer vision and AI-based cell phone usage detection in restricted zones of manufacturing industries. *Front. Comput. Sci.* 7:1535775. doi: 10.3389/fcomp.2025.1535775

Phone calls are strictly forbidden in certain locations due to the potential security threats. Mobile phones' growing capabilities have also increased the risk of their misuse in places that are restricted, like manufacturing plants. Unauthorized mobile phone use in these environments can lead to significant safety hazards, operational disruptions, and security breaches. There is an urgent need to develop an intelligent system that can identify the presence of individuals as well as cellphone usage. We propose an advanced Artificial Intelligence and Computer Vision-based real-time cell phone detection system to detect mobile phone usage in restricted zones. Modern deep learning approaches, such as YOLOv8 for real-time object detection to accurately detect cell phone usage, are combined with dense layers of ResNet-50 to perform image classification tasks. We highlight the critical need for such detection systems in manufacturing settings and discuss the specific challenges encountered. To support this research, we have developed a custom dataset of 2,150 images, which features a diverse array of images with varying foreground and background elements to reflect real-world conditions. Our experimental results demonstrate that YOLOv8 achieves a Mean Average Precision (mAP50) of 49.5% at 0.5 IoU for cellphone detection tasks and an accuracy of 96.03% for prediction tasks. These findings underscore the effectiveness of our AI and CV-based system in detecting unauthorized mobile phone usage in restricted zones.

KEYWORDS

artificial intelligence, computer vision, YOLOv8, ResNet-50, cell phone detection, mobile phone, manufacturing plants, restricted zones

1 Introduction

In today's digitally connected world, the proliferation of mobile devices has transformed the way we live and work. In the modern era of fast network expansion and extensive smartphone use, voice calls have ingrained themselves into our everyday lives. Cell phone use in restricted areas, such as government buildings, military installations, and sensitive industrial zones poses a serious risk to security and privacy (Jegham et al., 2020). The unauthorized use of cell phones in these areas can lead to the leakage of sensitive information including Intellectual Property (IP) and compromise national security. However, there are specific circumstances in which using a mobile phone poses a significant risk to one's safety. According to recent studies, using a phone in a restricted place has resulted in numerous accidents. Surveillance cameras are placed in some areas, like gas stations, industries, vehicles, etc. to help control and constrain human behavior and address the serious safety risk that handheld phone use poses. Nonetheless, psychological factors like carelessness can still cause people to act in ways that are unsafe and can result in safety incidents (Farmer et al., 2010; Shukla et al., 2013). Many governments and nations have approved laws prohibiting the use of mobile phones in designated places due to the dangers they pose to public safety and property. The study in Wang et al. (2014), Berri et al. (2014), Jiménez et al. (2016), and Ziebinski et al. (2017) details precise mobile phone detection utilizing a variety of techniques, including gesture recognition.

Despite the implementation of traditional security measures, the detection of cell phones in restricted zones remains a significant challenge. Thus, it is essential to implement a computer-vision and artificial intelligence (AI) based mobile phone detection system in restricted areas, particularly in manufacturing plants where security and safety are of the highest importance. By proactively identifying and preventing unauthorized mobile phone usage, the project significantly enhances workplace safety, minimizes operational disruptions, and fortifies sensitive data integrity. This initiative not only fosters a culture of compliance and accountability but also bolsters overall operational efficiency and productivity within manufacturing facilities. Conventional methods of cell phone detection, such as manual searches and metal detectors, are often time-consuming, invasive, and prone to errors. Moreover, the increasing sophistication of cell phone designs and the widespread use of concealment methods have made it even more difficult to detect these devices using traditional approaches. As a result, there is a growing need for innovative solutions that can effectively detect cell phones in restricted zones without compromising individual privacy or security protocols.

The primary motivation behind the proposed research is to tackle a significant issue prevalent in manufacturing industries. The integration of Artificial Intelligence (AI) and Computer Vision (CV) technologies offers a promising solution to this problem. By leveraging the capabilities of AI-powered algorithms and CV-based image analysis, it is possible to develop a system that can accurately detect cell phones in restricted zones in real time. Such a system can be integrated with existing surveillance infrastructure, providing a cost-effective and efficient means of enhancing security in sensitive areas. We propose an AI and CV-based cell phone detection system in restricted zones, which aims to address the limitations of traditional detection methods and provide a robust solution for ensuring security and confidentiality in sensitive environments.

2 Related work

Conventional object detection algorithms that rely on human feature extraction have performed poorly and made slow progress. In 2012, the topic of object detection reached new heights with the development of Convolutional Neural Networks (CNNs). Despite this, many object detection network studies have ignored model, computation, and parametric size and concentrated only on increasing accuracy. Cai and Vasconcelos (2018) developed an R-CNN model which was a region recognition-based object detection system that embraced the sliding window principle and used a Region region-generating network (RPN). To overcome R-CNN networks' speed and accuracy constraints, the Fast R-CNN network was developed in 2015 by Ren et al. (2015). Even though its performance had improved, it was still dependent on the selective search technique to find Regions of Interest (ROI) and could not achieve real-time capabilities. To accomplish real-time end-to-end object identification, Faster R-CNN was subsequently proposed (Girshick, 2015). It demonstrated real-time performance that was most comparable to deep learning detection techniques. Despite being two-stage algorithms that are faster and more accurate than conventional algorithms, both models' slow detection rates fall short of real-time performance standards due to their sophisticated network architecture and computational redundancy. Mask R-CNN (He et al., 2017) was presented as a solution to this problem where a faster R-CNN structure was improved to accomplish the segmentation process and then the ROI pooling operation was aligned to provide greater object localization performance. Single-stage object detection methods immediately sample the image's dense features for classification and regression rather than generating candidate regions. Many efforts have been made in the field of CNN-based Infrared (IR) image enhancement in surveillance systems (Zhang et al., 2025; Zhang et al., 2024) to preserve image quality, and have produced promising results.

The YOLO (You Only Look Once) technique (Redmon et al., 2016) was described by Redmon et al. (2016) as a solution for the sluggish detection speeds that are typically encountered in two-stage target detection systems. Although the class of objects and their position in the image can be accurately predicted by this technique, small targets could not respond well to it. To overcome this problem, Liu et al. presented SSD (Single Shot multi-box Detector) (Liu et al., 2016), which detects targets on feature maps with various visual fields using initial frames that were utilized by Anchors in Faster RCNN. Feature-based fused SSD was proposed by Cao et al. (2018). It makes use of both layer-level and global-feature fusion techniques. While the latter improves the detection capability for small targets, the former preserves the symbolic data of each layer to increase the accuracy of the object detection task. The YOLO family has undergone significant evolution since its founding in 2016 and continues to do so. Using the foundational basis of YOLOv1, and YOLOv2 models, Redmon and Farhadi proposed the YOLOv3 model (Redmon and Farhadi, 2018). The backbone is the 53-layer (Darknet) feature-extraction-efficient CNN layers. This Darknet network creates three different feature map sizes, and multi-scale feature fusion is used to retrieve small target feature information. Despite improving the accuracy of small target recognition, the model is dense. The YOLOv4 network model was introduced by Bochkovskiy et al. (2020). The YOLOv4 algorithm integrates optimization techniques from the developments in CNN (Convolutional Neural Network) (LeCun et al., 1998) and is based on the original YOLO architecture (Redmon et al., 2016). An object detector's performance is highly dependent on the quality of features extracted. Different experimentations were carried out by the researchers on different backbones including EfficientNet (Google-Brain). This led to the creation of DenseNet (Huang et al., 2017), which lowered the number of network parameters by addressing the vanishing gradient issue and promoting feature propagation and reuse. As the result of another experiment, EfficientNet-B3 (Tan and Le, 2019) offers the best choice for parameter selection if CNN's scaling is accomplished using a search technique.

CSPDarknet-53 was identified as the official backbone for YOLO-v4 after further examination.

They experimented with several neck-level integration strategies for feature extraction, such as Spatial Attention Mechanism (SAM) (Vaswani et al., 2017), Feature Pyramid Networks (FPN) (Lin et al., 2017), and Path Aggregation Networks (PANet) (Liu et al., 2018). In the end, PANet was chosen as the best feature aggregator. An improved variant of FPN called PANet adds a shortcut connection to connect fine-grained features from high-to-low-level layers while working on a reverse augmentation path (bottom-up) in addition to the top-down FPN approach. To separate the key features coming from the backbone and expand the receptive field, CSPDarknet-53 was added. While training with a single GPU, the YOLOv4 model not only validates the effects of several cutting-edge target detector training techniques but also adapts and enhances them. Similar to YOLOv4, YOLOv5 (Ultralytics, n.d.) concentrates on integrating and refining various computer vision techniques to enhance performance. YOLOv5 divides the input picture into several grid cells, and each grid cell is in charge of predicting a set of bounding boxes including the likelihood estimate of a target class within it. PyTorch (Jocher et al., 2021) served as the foundation for the YOLOv5 architecture, which was created to offer the required infrastructure to assist in the deployment of portable handheld devices.

To design an object detector with an industry application focus, Li et al. published the first codebase of the YOLOv6 (Li et al., 2022) network in 2022. The architecture is required to be extremely fast and accurate while maintaining high performance on a variety of hardware options to satisfy the requirements of industrial applications. YOLOv6 employs a complex model with good resolution on a large training set to increase the detection accuracy. Through adaptive training and auto-hyperparameter settings, the model can efficiently strike a compromise between speed and accuracy.

Performance was improved by using a redesigned reparametrized YOLOv5 backbone (EfficientRep) and neck (Ding et al., 2021) (Rep-PAN neck) with extra layers isolating features from the final head. The YOLOv7 was introduced by Wang et al. to focus more on GPU speed enhancements, particularly inferencing (Wang et al., 2022). To preserve high detection speeds and improve accuracy, YOLO-v7 recommends certain architectural improvements.

In January 2023, Ultralytics released a new upgrade to the YOLO family called YOLOv8 (Jocher et al., 2023) to satisfy the demands of automated quality inspection in the industrial surface defect detection domain—such as the need for quick detection, high accuracy, and deployment onto edge devices. YOLOv8 provides state-of-the-art real-time and high-classification performance with a small number of effective computational parameters. Figure 1 illustrates how YOLO-v8 outperforms its predecessors in terms of throughput on identical parameters after being trained on COCO images (Lin et al., 2014). YOLOv5 provides exceptional real-time performance, while YOLOv8 is the ideal option for applications that require high inference speed and constrained real-time edge device deployment.

Several application computer-vision-based applications have been developed using these frameworks. Ahmad et al. proposed a mobile phone usage detection system based on the YOLOv5 algorithm to administer the online test (Al-Allaf and Asker, 2022). The Makesense website was used to categorize a custom dataset of phone photos in various positions and orientations. The examinee's computer's webcam records live footage which was subsequently analysed YOLOV5 algorithm. Real-time mobile phone usage detection with a maximum accuracy of 92% and a False acceptance rate (FAR) of 4% was reported. The YOLOv8n (Shen et al., 2024) based model was introduced by Qian et al. to identify distracted driving. They included StarNet into the model's core to boost feature extraction performance while achieving a notable decrease in computational complexity. To lower the detection head's computational load and parameter size, the shared convolution layers were incorporated. An accuracy of 99.6% on a dataset of 100 drivers showed a notable improvement in distracted driving behavior.

A method for detecting mobile phone usage was developed by Rehman et al. (2021). For feature extraction, they used the Speeded Up Robust-Features (SURF) and Histogram of Oriented-Gradients (HOG) approaches. Classification tasks are completed by Support Vector Machine (SVM), Nearest Neighbour (K-NN), and Decision Tree classifiers. High-speed detection was possible with HOG and SVM classifiers, but accuracy was compromised. Talib et al. (2024) presented "YOLOv8-CAB" as an improvement to the YOLOv8 object identification framework. Contextual-Attention-Block offers multi-scale feature maps and incremental feedback, which greatly enhances

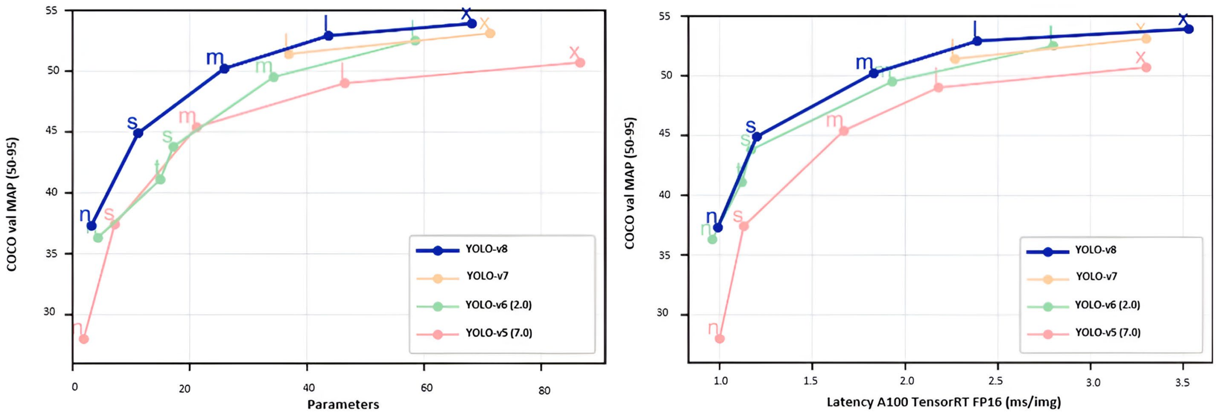

FIGURE 1
YOLOv5 to v8 performance comparison (Hussain et al., 2023).

small object identification performance. As a result, feature-fusion, contextual information preservation, and enhanced weak feature extraction are made possible by this architecture. When tested on the COCO dataset, the model demonstrated a mean average precision of 97% of detecting rate, which is a 1% improvement over traditional models.

As discussed in this section, the YOLOv8 has improved over its predecessors to provide state-of-the-art real-time performance with high accuracy and speed—two essentials for real-time object recognition applications. Its capacity to precisely identify smaller objects has been significantly improved by anchor-free architecture, multi-scale prediction capabilities, and optimizations. Most recent research focuses on driver distraction behavior, mobile usage detection in exams, or object detection using YOLO models. Research on mobile phone recognition in restricted areas, particularly in industrial settings, utilizing the YOLOv8 model is still in its early stages. Using the cutting-edge YOLOv8 object detector, we proposed an "automated cell phone detection system for manufacturing plants" to bridge this gap. The remainder of the paper is organized as follows: The implementation details are covered in Section 3. The research findings and analysis are discussed in section 4. The section wraps up by providing a summary of the work and outlining the scope of future work.

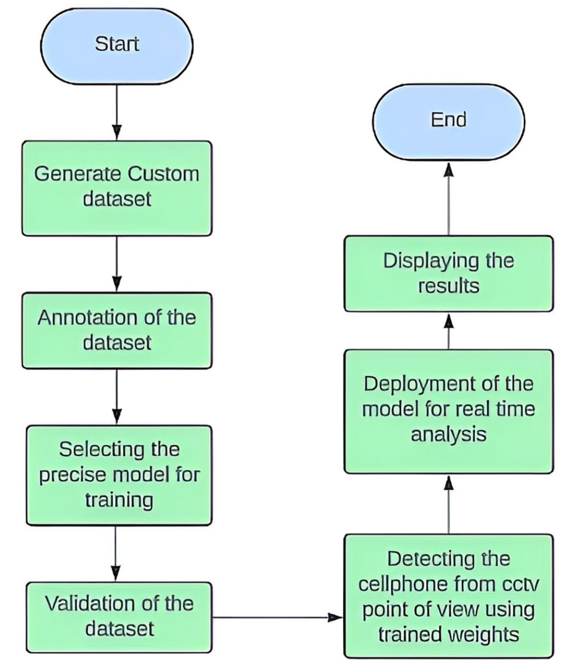
Start
Generate Custom dataset
Annotation of the dataset
Selecting the precise model for training
Validation of the dataset
Detecting the cellphone from cctv point of view using trained weights
Deployment of the model for real time analysis
Displaying the results
End

FIGURE 2
Steps followed in the implementation of the proposed cellphone detection system in restricted zones.

3 Implementation

To ensure a cell-free environment within restricted zones of manufacturing industries, there is a need to develop inexpensive and accurate solutions that can be seamlessly integrated into existing infrastructure. To address this issue, we have proposed an AI and Computer-vision-based cell phone detection system in restricted zones, detecting and preventing cell phone usage in restricted zones. The system leverages cutting-edge advancements in AI and CV that can be integrated with existing surveillance camera systems to identify cellphone misuse and alert authorities in real-time. The flowchart in Figure 2 illustrates the process of the proposed system. The steps are explained next.

3.1 Dataset collection

Our study started by tackling the issue of the absence of publicly available datasets on mobile usage in industrial environments. By collecting images from numerous manufacturing facilities in the Belagavi (Karnataka) and Kolhapur (Maharashtra) regions under varying lighting circumstances and CCTV camera angles, we produced a high-quality dataset. Additionally, as the resolution of CCTV footage varies across different industries, we developed a dataset derived from online videos to simulate CCTV point-of-view perspectives.

This allowed our model to be trained and become proficient in identifying cell phones even from various CCTV angles. The sample collected dataset images are indicated in Figure 3. We have created a custom dataset containing 4,500 images by capturing images from various industries and collecting images of people carrying cell phones from online repositories. After the data cleaning process, we carefully chose 2,150 images for implementing the model. The dataset is divided into two classes and contains images of persons carrying a cell phone (With cellphone) and another without one (Without cellphone). To maintain the person's privacy and confidentiality, we have blurred their faces in the generated dataset.

3.1.1 Ground truth dataset and annotation

Now our next step is to create ground truth by labeling and segmenting the collected dataset. For this we chose a popular annotating tool called "LabelImg" for cell phone detection. LabelImg is a user-friendly tool with a straightforward interface that typically involves drawing bounding boxes around the phones in images. LabelImg excels at this task, allowing us to efficiently create precise annotations and labelling as shown in Figure 4. After labeling the data set using people with and without cellphone labels, we perform segmentation tasks on the specific object of interest, i.e., the phone itself. This allows us to focus solely on the phone's presence or absence in the images without any distractions from other objects or background elements. Segmenting the phone helps standardize the dataset by removing variations in background, lighting conditions, or other environmental factors that may be present in the original images. This ensures consistency across the dataset and facilitates more accurate analysis and classification. We have segmented up the dataset using the Python code, so we get good quality. Following labeling and segmentation, the information is saved in XML (eXtensible Markup Language) format to give structured annotated information, such as the location and characteristics of objects in images. It contains tags that represent many aspects of the labeled data, including class labels, object bounding boxes, and picture metadata. In addition to being compatible with a variety of annotation tools and frameworks, this structured representation enables effective storage and retrieval of annotated data. After the labeling process, the XML file is then converted to YOLO format.

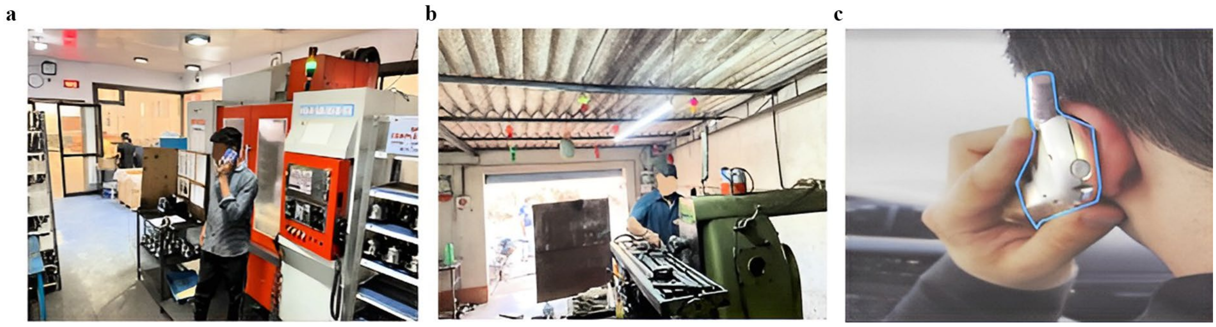

FIGURE 3
The sample dataset used in our proposed work. **(a)** Custom developed dataset image with a cellphone **(b)** a Custom dataset image without a cellphone **(c)** a sample person with a cellphone image from the Roboflow 2.0 (Talib et al., 2024) dataset.

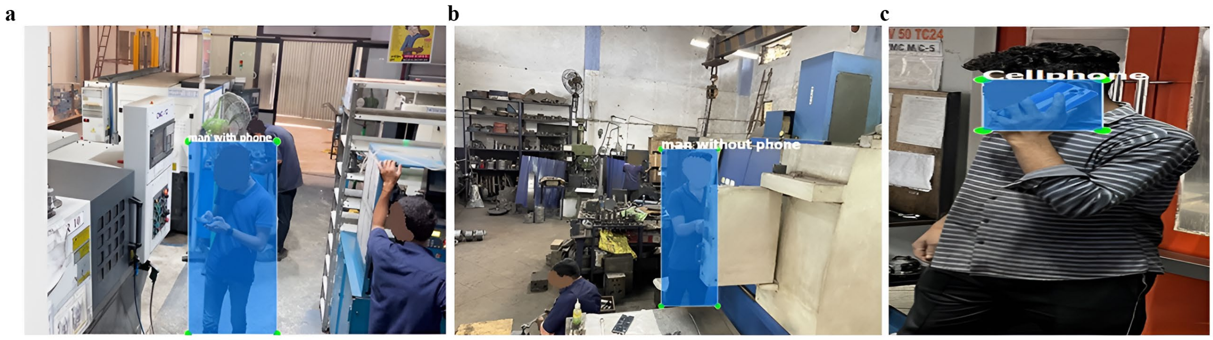

FIGURE 4
Ground truth dataset labeling and segmentation. **(a)** With a cellphone **(b)** without a cellphone **(c)** segmenting cellphone object.

3.1.2 Data augmentation

Data augmentation techniques are employed to enhance the variability of the dataset, thereby improving the model's robustness. These techniques include rescaling, shearing, zooming, flipping, rotation, and adjusting brightness. By exposing the model to a wider range of variations in the data, data augmentation helps to enhance the model's performance and prevent overfitting.

3.1.3 Data splitting and generators

The dataset is split into separate training and validation sets using a train-test split. This step ensures that we have distinct datasets for training and evaluating the performance of our cellphone detection model. Data generators are created for both the training and validation data, responsible for loading batches of data during model training. This facilitates the efficient processing of large datasets and ensures that the model receives properly formatted input data during training, thereby enhancing the model's performance and accuracy.

3.2 Model selection

The detection of cell phones in restricted zones is a critical task for ensuring compliance with regulations and maintaining security. Computer vision techniques can be employed to accurately detect cell phones in such environments. These techniques involve the use of machine learning algorithms, such as deep learning architectures based on convolutional neural networks, to recognize cell phones from visual data. The problem of object detection is a rapidly evolving research area in contemporary AI, with several well-recognized strategies available. These include:

- Single-shot Multi-Box Detector (SSD)
- YOLO
- ResNet (Residual Network)
- Region Based Convolutional Neural Network (R-CNN)
- DetectNet
- RetinaNet
- CenterNet

Since the proposed work requires a balance between detection accuracy and real-time performance, we chose the YOLOv8 framework for cell phone detection operations. Since the cellphone is a small object to be detected under diverse lighting and background scenes, after accurate cellphone object detection, accurate recognition plays a very crucial operation in the success of this proposed work. In addition to YOLO, we utilize ResNet-50 (He et al., 2016), a state-of-the-art deep learning algorithm to recognize and validate the unseen dataset as well as assess the model's performance during training. ResNet-50, a variant of the ResNet (Residual Network) architecture, is frequently used in object detection tasks since it can train networks with hundreds of layers, which is extremely deep. Because information from previous layers can be preserved, residual blocks and skip connections are used to make this possible. Getting state-of-the-art results in a variety of image-related tasks, including object identification, image classification, and picture segmentation, is another benefit of ResNet-50. This dual-layer approach of using YOLOv8 for real-time object detection and ResNet-50 for cellphone recognition ensures a high level of accuracy and robustness in detecting unauthorized cellphone usage.

3.2.1 YOLOv8 architecture

In the year 2025, Redmon et al. (2016) presented their object-detection algorithm known as “You Only Look Once (YOLO)” which outperformed its predecessor Region-based Convolutional Neural Network (R-CNN) model in terms of real-time object detection capabilities. It is called a single-shot detector that uses a single neural network to predict bounding boxes and class probabilities from the same image to perform classification tasks in a single pass. The YOLO architecture is illustrated in Figure 5. Over the earlier YOLO models, the recently released YOLOv8 computer vision algorithm is the best example of a cutting-edge model. The head, neck, and backbone layers make up the architecture.

3.2.1.1 Backbone

It is sometimes referred to as the feature extractor that is responsible for obtaining edge and texture feature maps from images through a pre-trained CNN.

3.2.1.2 Neck

The neck performs feature fusion operations and integrates contextual information using Feature Pyramid Network (FPN) path aggregation blocks. This layer serves as a bridge between the head and the backbone. It is responsible for predicting bounding boxes and classification of objects before passing them onto the head layer.

3.2.1.3 Head

It is the last component of the network and is in charge of producing the outputs in the form of object detection confidence scores in the bounding box representing the likelihood that an object is present.

Each of these components is essential to our YOLO cell phone detection process, and the network’s architecture is tailored to effectively and efficiently gather complex visual information, producing the quick and precise predictions required for real-time applications.

3.2.2 Pre-trained ResNet-50

Figure 6 highlights the key components of the popular ResNet-50 model (He et al., 2016) where each block has a different composition, convolution sizes, and feature maps. ResNet50 is pre-trained with images from the ImageNet dataset. It presents a novel residual learning idea that makes training deeper networks easier and mitigates the vanishing gradient issue (see Figure 7).

The modified architecture after removing the original classification layers also has proven to have remarkable feature extraction capabilities, which makes it a good fit for challenging classification tasks like cell phone detection in complex environments. In our work, we use ResNet50 as a feature extractor without fully connected layers. We freeze its convolutional layers to retain pre-trained knowledge. Later we add new dense layers with ReLU activation and dropout to prevent overfitting and finally use a sigmoid activation in the final layer for binary classification. To ensure the precision of the detection process, the output layer is therefore essential in pinpointing the precise locations of the pure bounding boxes of the objects that have been detected. This is especially crucial for accurately detecting cell phones in photos processed by the ResNet-50 and YOLOv8 combination.

3.2.3 Proposed YOLOv8 with ResNet-50 based cellphone detector

Figure 7 illustrates the proposed YOLOv8 and ResNet-50 based cellphone detector system. YOLO is a popular object detection model known for its speed and accuracy. It is an end-to-end neural network architecture that makes bounding box and class probability predictions all at once. It differs from the approach taken by previous object detection algorithms, which repurposed classifiers to perform detection. By taking a radically different approach to object detection, YOLO outperformed existing real-time object detection algorithms and produced state-of-the-art results. A custom-trained deep learning model, specifically designed for cellphone detection, forms the core of the system. High-resolution CCTV cameras or webcams

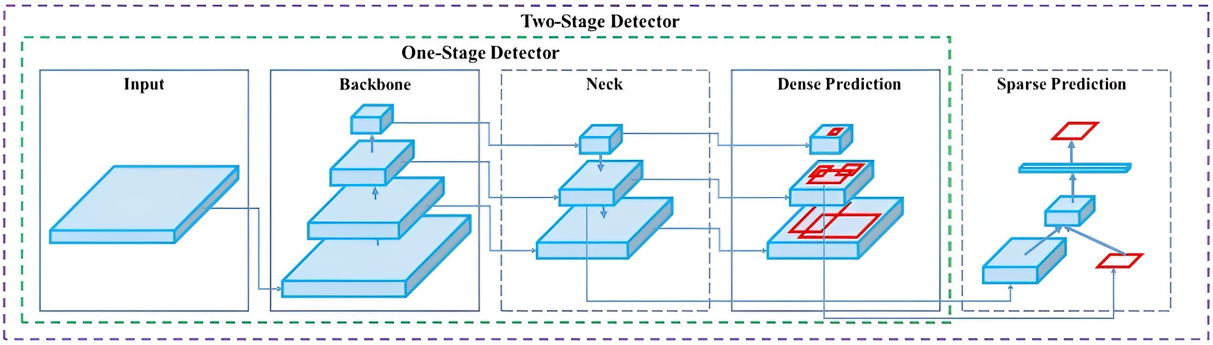

FIGURE 5
YOLO object detector architecture (He et al., 2016).

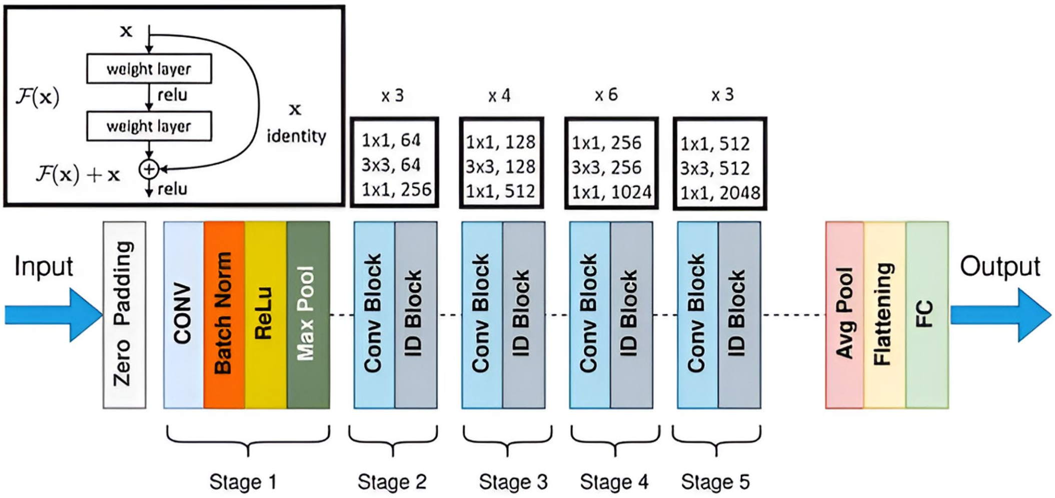

FIGURE 6
ResNet-50 architecture (He et al., 2016).

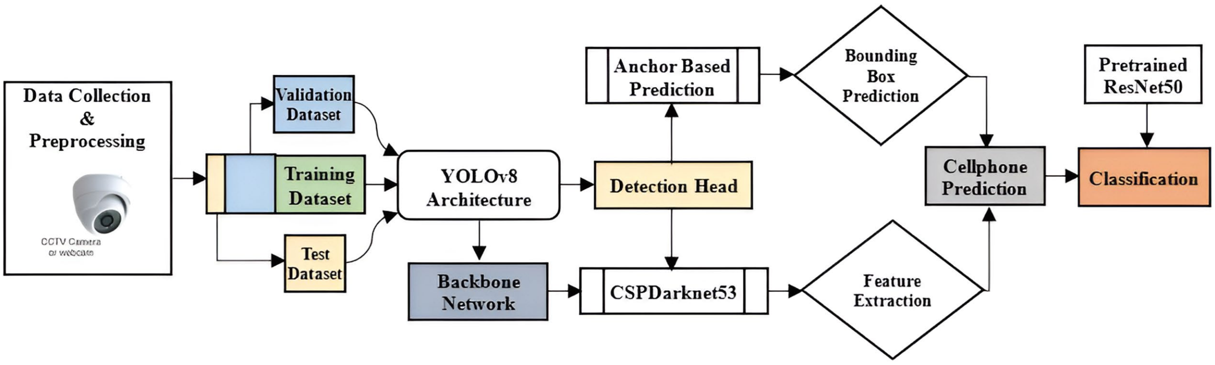

FIGURE 7
YOLOv8 with ResNet-50 dense layer architecture for cellphone detection.

strategically placed within the restricted zone capture real-time video footage of the manufacturing environment. The acquired video stream undergoes preprocessing to optimize it for subsequent analysis. This may involve frame rate adjustment, resizing, and noise reduction to enhance image quality and computational efficiency. The model is trained on a diverse dataset comprising images of cellphones in various orientations, lighting conditions, and backgrounds, ensuring robustness and adaptability to real-world scenarios.

The deep learning model analyzes each preprocessed frame, identifying potential Regions of Interest (ROI) that may contain a cellphone. Within these ROIs, the model performs fine-grained analysis, extracting features like shape, texture, and color to determine the likelihood of a cellphone being present. Upon detection, the system accurately localizes the cellphone within the frame by drawing a bounding box around it, providing visual feedback to operators. Regions of interest are further scrutinized by ResNet-50 for dataset validation. Post-processing refines detections, triggering alerts, or real-time visual displays upon confirmation of mobile phone presence. All detections are logged in the serial frame monitor and display the result of cellphone detections. These backbones include, for example, the well-known Darknet-53 or CSPDarknet-53.

3.3 Implementation

A head network, neck network, and backbone network make up the YOLOv8 architecture as indicated in Figure 5. These steps are explained below.

3.3.1 Dataset preprocessing

This process involves removing irrelevant images, resizing images to maintain uniformity, and enhancing image quality to improve the system's accuracy. Different transformation operations such as rotation, scaling, and flipping are applied to increase the diversity of the training data and prevent overfitting. The images are labeled to create a ground truth dataset for training and validation purposes. Our model receives an input image, typically in the form of pixels, and preprocesses it to a format suitable for neural network inference. The image is resized to a fixed input size (e.g., 416×416 pixels) to ensure consistency across different images.

3.3.2 Neural network architecture

YOLOv8 employs a convolutional neural network (CNN) architecture, specifically designed for object detection tasks. The network consists of multiple convolutional layers followed by detection layers responsible for predicting bounding boxes and associated class probabilities.

3.3.3 Anchor box and grid cell

The principle of anchor boxes forms the basis of YOLOv8's operation. To predict the location and class of objects in a picture, anchor boxes with preconfigured bounding boxes of varying sizes and aspect ratios are used. YOLOv8 utilizes anchor boxes, which are predefined bounding boxes of different sizes and shapes, to predict object locations and sizes. The network predicts bounding box coordinates (x, y, width, height) relative to each anchor box, along with confidence scores indicating the likelihood of an object being present and class probabilities for each detected object class. The feature maps are divided into a grid of cells, typically with a size of 13×13 or 19×19. Each grid cell is responsible for detecting objects whose centre falls within its boundaries. The model gains the ability to modify the anchor boxes to better fit the image's objects during training.

3.3.4 Feature extraction backbone network

The task of extracting features from the input image falls to the backbone network. The backbone network collects hierarchical features by methodically examining input images, revealing important information from different levels of abstraction. The neck network lowers the feature maps' spatial resolution by aggregating features from several scales. Predicting the bounding boxes and class probabilities for every object in the picture is the responsibility of the head network. The input image is passed through the CNN, where successive convolutional layers extract features at different scales. Feature maps are generated at multiple resolutions, allowing the network to detect objects of various sizes and aspect ratios.

3.3.5 Bounding box prediction

For each grid cell, YOLOv8 predicts multiple bounding boxes (usually 3 or 5) using anchor boxes of different scales and aspect ratios. The network outputs confidence scores for each bounding box, representing the likelihood that the box contains an object and class probabilities for each object class. Let the image be divided into a S*S grid by this architecture. This grid identifies the object if the object's bounding box centre is situated inside it. Bounding boxes are predicted by each grid using its confidence score. Each confidence score indicates how precisely the bounding box coordinates are predicted about the ground truth prediction and how likely it is that the bounding box will include an object.

We multiply the individual box confidence predicted by the conditional class probabilities at test time. Our confidence score is defined in the Equation 1.

$$\text{Confidence Score} = \Pr\left(\text{Object}\right) * \text{IOU}^{\text{truth}} \tag{1}$$

The confidence score in object detection tasks is essential for identifying whether items are present inside bounding boxes. The overlap between the predicted and ground truth bounding boxes is measured by the Intersection over Union (IoU) metric, which is used to calculate this score. The confidence score is assigned to 0 when there is no object in the grid cell, demonstrating the model's uncertainty about object presence. The degree of agreement between the predicted and ground truth bounding boxes is reflected in the confidence score when an item is discovered, on the other hand; larger IoU values signify greater confidence in the detection. The existence of an object in the grid cell determined the condition of this likelihood. Each grid cell forecasts a single set of class probabilities, regardless of the number of boxes. The class-specific confidence scores for each box are then obtained by multiplying the conditional class probabilities by the individual box confidence forecasts as given below in Equation 2.

$$\begin{aligned} &\Pr\left(\text{Class}|\text{Object}\right) \\ &* \Pr\left(\text{Object}\right) * \text{IOU}^{\text{truth}} = \Pr\left(\text{Class}\right) * \text{IOU}^{\text{truth}} \end{aligned} \tag{2}$$

Next, when multiple boxes are predicted for the same item, we use non-maximal suppression (NMS) to suppress the non-max outputs. Finally, our final forecasts are produced. NMS eliminates overlapping boxes with lower confidence scores, ensuring that each object is detected only once.

3.3.6 Post-processing and cell phone classification

The identified bounding boxes are post-processed to extract the coordinates and class labels of detected objects, including cell phones. ResNet-50 is trained with ImageNet-pretrained weights, and the model is fine-tuned for cell phone detection by adding dense layers for classification. Following the precise cell phone detection, the algorithm draws a bounding box around the phone to localize it methodically. These ROIs are further investigated attentively by ResNet-50 layers and are categorized as individual "with a cell phone" or "without a cell phone." The model is compiled with appropriate loss and optimization functions, and its performance is evaluated using validation data. The system performance parameters and results are discussed in the next section.

4 Results and discussions

Meticulous selection and dataset preparation for training the model coupled with thorough preprocessing and augmentation tasks play an important role in creating the foundation for a reliable and successful "YOLOv8-based cell phone usage detection in an industrial environment." This carefully curated dataset will provide a solid foundation for training and evaluating our AI model, ensuring its ability to accurately detect mobile phone usage in the complex and varied settings of manufacturing plants.

Depending on the industry sector and region, manufacturing facilities might have a variety of machinery, lighting, and layouts. These variations are included in our dataset by collecting data from various places as tabulated in Table 1. This is due to the possibility that our model, which was trained exclusively on data from one place, may not be able to generalize and function accurately in other plants with diverse backgrounds.

4.1 Experimental setup

Several assessment metrics and hyperparameters are used by our unique YOLOv8 + ResNet-50, method as demonstrated by the following:

1. To ensure uniformity among images, the image is reduced to a predetermined input size (e.g., 416×416 pixels). High-quality images would increase the detection accuracy but at a significant computational expense. However, employing low-quality photos could result in each image losing important details. Previous YOLO versions have shown experimentally that the specified resolution yields meaningful results.
2. 300 epochs are used in the training process because this quantity of epochs enables the model to learn key characteristics without overfitting the training set.
3. A set of 30 batch samples is used in each cycle because deep learning techniques generally use this batch size to strike a reasonable balance between computational viability and model update frequency.
4. We set the initial learning rate of 0.001 and it reduces by 50% for every 3 cycles. This is a common setting that enables the model to cover at a moderate pace to optimally utilize the computational resources. The trials were carried out Google Colab platform that uses NVIDIA's Tesla T4 and V100 18 GB GPUs with pre-installed libraries such as TensorFlow, PyTorch, and Open CV.

The efficiency of the suggested model was assessed by a series of comparative tests carried out on the custom-built dataset. To evaluate the detection capabilities of the YOLOv8 + ResNet-50 algorithm in real-world scenarios, a variety of complex scene photos in different scenarios were used.

4.2 Model training and validation

The training process is divided into two main parts:

i. Dataset validation: This involves validating the labeled dataset to ensure that it is accurate and reliable. This step is essential for ensuring that the training process is based on a high-quality dataset, which in turn is critical for achieving accurate and reliable results. This process encompasses various techniques and steps aimed at ensuring that the data is accurate, consistent, and representative of the real-world phenomenon it is intended to describe or analyze. The validation process involves evaluating various parameters, including accuracy, loss, validation accuracy, validation loss, and learning rate. These metrics provide valuable insights into the performance of the model and enable us to make informed decisions about whether the model is suitable for deployment in a manufacturing plant setting. By validating the dataset, we can ensure that the model is trained on high-quality data that accurately represents the real-world phenomenon of cell phone usage in a manufacturing plant.

ii. Cell phone detection: This is the primary training step, where the model is trained to detect the presence of a cell phone in the images. The YOLOv8 model is particularly well-suited for this task, as it is designed to detect objects in real-time, even in complex and dynamic environments. Object detection involves identifying and localizing multiple objects within an image or video frame. Unlike image classification (where we classify the entire image), object detection pinpoints the location of each object. Further, a pre-trained ResNet-50 model plots a bounding box around the phone to categorize persons "with a cell phone" or "without a cell phone" class.

We divided our custom dataset in a 70:20:10 ratio, resulting in 1522 training, 387 validation, and 241 test images. Throughout the training process, we will closely monitor the model's performance and make any necessary adjustments to ensure that it is accurately and reliably detecting the presence of a cell phone in the images. This will enable us to develop a highly accurate and reliable cell phone detection system that can be deployed in a manufacturing plant setting. The training and validation performance of our proposed model is indicated in graphs as shown in Figure 8. These are training and validation metrics for a machine learning model, likely related to object detection Here's what each graph represents:

4.2.1 Training metrics (top row)

i. Train/box_loss: The loss related to bounding box predictions during training is displayed in this graph. This loss is decreasing as the model is gaining knowledge.

ii. Train/cls_loss: It refers to the loss associated with class (object category) predictions. It is declining throughout epochs, like box loss.

iii. Train/dfl_loss: Both box and classification losses are included in the total loss. Smoothing makes trends easier to see.

iv. Precision (at B): The precision metric quantifies the proportion of true positives that are predicted. The model achieved higher, i.e., 97% of detection accuracy. This suggests the model is good at avoiding false positives.

v. Recall (at B): The graph shows the proportion of true positive cases that were accurately predicted. The model is missing some true positives, so it's less sensitive to detecting all relevant objects. The model achieved moderate recall accuracy of 65%.

4.2.2 Validation metrics (bottom row)

i. val/box_loss: Like training box loss, but with a metric that is decreasing over time and assessed on the validation set.

ii. val/cls_loss: Similar to training box loss but measured on the validation set the values are expected to decline over time.

iii. val/dfl_loss (Smoothed): It is the measure of overall loss on the validation set and the values are reducing over time.

iv. mAP50 (at B): Mean Average Precision (mAP) at an IoU threshold of 0.5. It combines precision and recall across different confidence thresholds. The mean average precision (at top 50 predictions) of mAP50 = 0.495 indicates the model produces decent overall performance but this value can further be improved.

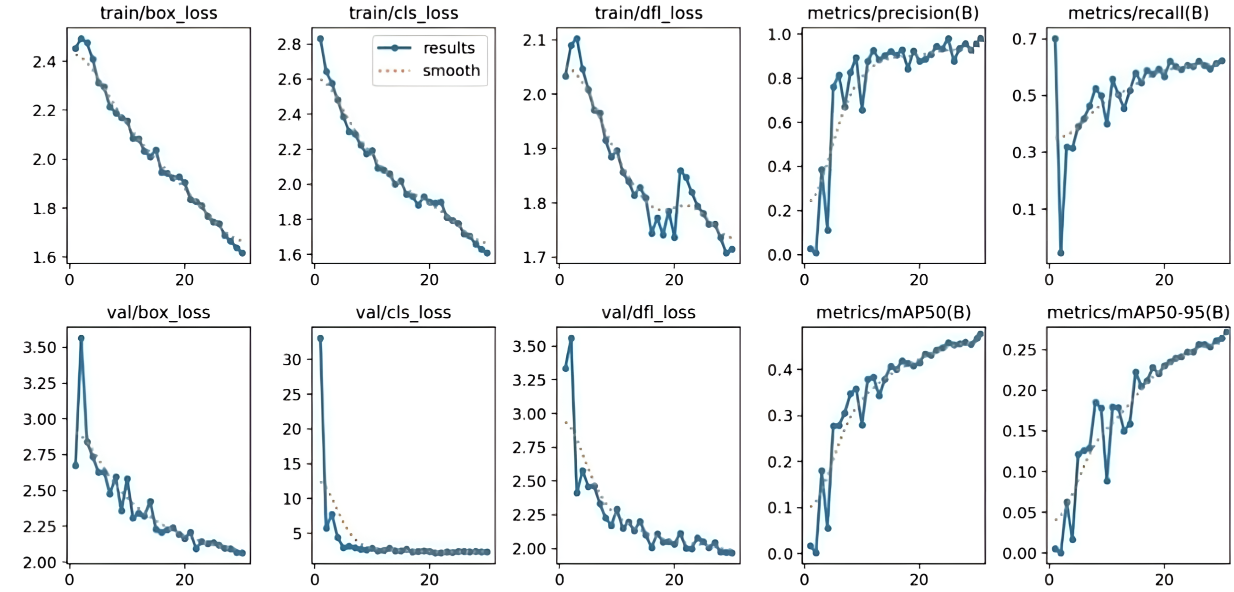

FIGURE 8
Training and validation loss graphs.

TABLE 1 Custom dataset created to test the proposed cell phone detection system.

Sr. No	Dataset source	Number of images
1.	Custom cell phone usage dataset collected from Belagavi (Karnataka, India), Kolhapur, and Pune (Maharashtra, India) industries.	1,000
2.	Roboflow Cellphone Dataset Computer Vision Project (Talib et al., 2024).	800
3.	Cell phone usage images from various online sources.	350
Total Images		2,150

v. mAP50_95 (at B): Similar to mAP50 but considers a wider range of IoU thresholds (from 0.5 to 0.95). mAP50_95 of 0.289 indicates the model seeks performance improvement.

After successfully training and validating the model, we proceed with model testing and demonstrate the results next (see Table 1).

4.3 Model testing for cell phone detection

Testing entails utilizing an independent test set that was not presented to the model during training. This guarantees an objective assessment of the model's capacity to generalize to unknown data. A variety of lighting settings, phone types, backgrounds, and possible occlusions should all be included in the test set to make it typical of real-world situations. We have used 241 samples for testing our YOLOv8 (without ResNet-50) and YOLOv8 (with ResNet-50) model of which 151 samples contain "person with cellphones" and 90 are "person without cellphones." The confusion matrix showing the performance of our model utilized for cell phone detection in restricted zones is described in Table 2. The confusion matrix contrasts the model's predictions with the actual labels, or ground truth and it is explained next.

True-Positive (TP) instances: 145 images containing a person with a cell phone are correctly predicted by Model A, and 73 images are correctly predicted by Model B.

True-Negative (TN) instances: In 9 instances where the person was without cellphones Model A predicted correctly and Model B correctly predicted 3 instances.

False-Positive (FP) instances: Model A predicted a person with cellphones 6 times and Model B 30 times, when there were none.

False-Negative (FN) instances: There were 81 images of people using cell phones, but Model A failed to detect them. Similarly, Model B missed 106 instances of people using cell phones.

Model B (YOLOv8 without ResNet-50) produced 70.87% precision compared to 96.03% of Model A (YOLOv8 with ResNet-50). The inclusion of ResNet-50 is certainly helping in reducing false detection rates. We consider Model A for further discussion and performance comparisons. Based on the confusion matrix in Table 2, several performance measures are calculated to assess the model's performance. The performance measure metrics and their values for our suggested system are listed in Table 3. The Roboflow 2.0-based Cellphone Object Detection (Fast) (Talib et al., 2024) and YOLOv8-CAB (Talib et al., 2024), Nanodet (Talib et al., 2024) models are developed on the YOLOv8 framework and tested on COCO datasets (Lin et al., 2014). Another cellphone detection

TABLE 2 Confusion matrix of the proposed YOLOv8 detector with and without ResNet-50.

N = 241		YOLOv8 with ResNet-50 (Model A)		YOLOv8 without ResNet-50 (Model B)	
		Actual		Actual	
		Person with cell phone	Person without cell phone	Person with cell phone	Person without cell phone
Predicted	Person with cell phone	145	6	73	30
	Person without cell phone	81	9	106	3

TABLE 3 Experimental results and performance comparison.

Performance metrics	Model scores				
	Proposed YOLOv8 + ResNet-50	Roboflow YOLOv8 (Fast) (Cellphone Dataset Computer Vision Project, 2022)	YOLOv8-CAB (Talib et al., 2024)	Nanodet (Talib et al., 2024)	SURF & SVM (Rehman et al., 2021)
Mean average precision (mAP50) %	49.5	40.3	47.4	39.5	--
mAP50 average precision (mAP50_95) %	28.9	--	28.2	--	--
Precision %	96.03	59.7	89.3	66	91
Sensitivity (recall) %	64.16	38.4	64.7	61	--
F1 score %	76.92	--	75.07	71	--

method (Rehman et al., 2021) is developed using a classical Speeded Up Robust Features (SURF) feature extraction and Support Vector Machine (SVM) based classification methods. We compare our proposed model's performance with these frameworks and list the values. YOLOv8-CAB details are already discussed in the literature survey reported in Section 2. Roboflow 2.0 is an object detection YOLOv8 model tested on 1901 images in which 1,525 images contain with cell phones and 376 without cell phones instances. Roboflow offers a subset of the 2017 COCO dataset and provides a good alternative to work on this research area. The results in the table show that our system produced Mean Average Precision (mAP50) and mAP50 average precision (mAP_95) of 49.5 and 28.9%, respectively. SURF and SVM-based methods (Rehman et al., 2021) utilize 1,000 images (500 with and the remaining 500 without cellphones). 91% classification accuracy of persons with cell phones was achieved by using this combination method.

The Precision, Recall, and F1 scores of our proposed YOLOv8 + Resnet-50 model are 96.03, 64.16, and 76.92%, respectively. A selection of scenes demonstrating people using cell phones and successfully detecting objects is shown in Figure 9. The model can recognize and distinguish different objects in challenging environments. Precise object boundary masks, even in situations with complex backgrounds and object occlusions, are among the notable features.

A key element of our cell phone detection system is real-time analysis using a webcam, which allows for quick and precise cell phone recognition in live video feeds, as demonstrated in Figure 10. The user may be able to monitor and view the real-time analysis system's performance, including frame rate, processing time per frame, and detection accuracy. This allows users to evaluate the effectiveness and dependability of the system while it is in use.

Ongoing real-time analysis of the webcam video stream is made possible by the looping processes of frame capture, preprocessing, object detection, annotation, visualization, performance monitoring, and interactivity. Rapid cell phone detection is made possible by the remarkable inference speed of 340 ms, which is essential for real-time application settings.

5 Conclusion and future scope

This work addresses the issue of enforcing cellphone restrictions in restricted areas of various industries by proposing a novel YOLOv8 object detection model combined with the layers of ResNet-50 for accurate cell phone detection and classification tasks. The work involved several key steps, including data collection, preprocessing, model configuration, training, evaluation, and deployment. Through extensive experimentation and fine-tuning of hyperparameters, we successfully trained a YOLOv8 model that demonstrated robust performance in detecting cell phones with high accuracy. At 0.5 IoU, the model's precision (mAP50) for cell phone identification tests was 49.5%, while it produced 96.03% accuracy for prediction tasks. This exhibits the generalization and efficiency of YOLOv8 architectures on object identification tasks, as well as their ability to precisely locate and classify cell phones in a range of scenarios. Furthermore, the real-world deployment of the trained model demonstrated its practical utility in identifying cell phones in images and videos with real-time performance, indicating its readiness for practical applications. The benefits of this research can be extended to various domains, including surveillance, security, and image analysis applications. The accurate detection of cell phones can enhance security measures in sensitive areas, facilitate content moderation in social media platforms, and enable automated analysis of visual data in research and industry settings.

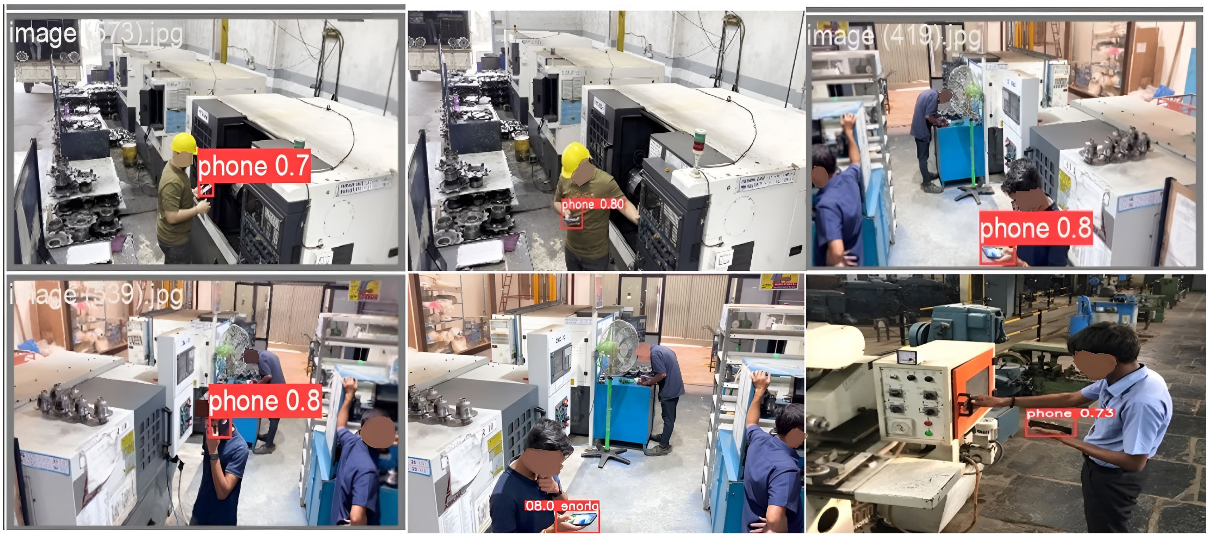

FIGURE 9
Proposed models' performance in detecting persons with cell phones.

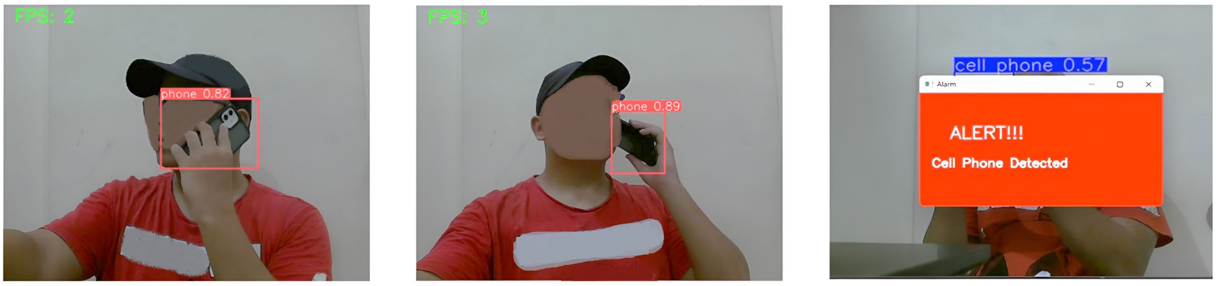

FIGURE 10
Real-time man with cell phone detection using a webcam.

Despite the system's strong prediction performance, false positives may occur when phones are partially or covered by objects, in low or bright illumination, or when objects resemble phones. False-negative results may result from tiny items that are comparable to or indistinguishable from a cell phone. Further research could focus on overcoming these limitations by integrating AI-based pose estimation methods, and deep learning models (Generative AI & GANs) to recognize the context and predict the missing object for occluded cell phones. Histogram equalization or exposure correction algorithms can help deal with varying lighting conditions while detecting cell phones. Multi-modal fusion (Infrared, RF detection, depth sensors) methods can detect a phone's heat emissions or thickness to differentiate between a cell phone and look-alike objects ex wallets, remote controls, etc. The YOLOv8 architecture for particular use cases can be improved by looking into real-time deployment strategies for large-scale applications and examining other data augmentation techniques to enhance model generalization. By deploying the YOLOv8 model on edge devices, such as NVIDIA's Jetson Nano, for on-device processing, latency and reliance on cloud connectivity can be decreased, increasing system resilience and making it appropriate for deployment in remote or bandwidth-constrained environments.

Data availability statement

The raw data supporting the conclusions of this article will be made available by the authors, without undue reservation.

Ethics statement

Written informed consent was obtained from the individual(s) for the publication of any potentially identifiable images or data included in this article.

Author contributions

UD: Data curation, Formal analysis, Investigation, Methodology, Project administration, Supervision, Validation, Visualization, Writing – original draft, Writing – review & editing. SS: Investigation, Resources, Supervision, Writing – review & editing. RK: Data curation, Formal analysis, Investigation, Supervision, Writing – review & editing. AC: Project administration, Investigation, Writing – review & editing, Supervision, Formal analysis. SD: Data curation, Methodology, Visualization, Writing – review & editing, Supervision. RP: Data curation, Investigation, Software, Writing – review & editing. PK: Data curation, Methodology, Software, Validation, Writing – review & editing. NG: Data curation, Methodology, Validation, Writing – review & editing. VR: Data curation, Formal analysis, Methodology, Software, Validation, Visualization, Writing – review & editing.

Funding

The author(s) declare that no financial support was received for the research and/or publication of this article.

Acknowledgments

We sincerely thank Mr. Ajay Kabadi of DocketRun Tech. Pvt. Ltd. Hubballi, Karnataka, India for providing valuable guidance in carrying out this research work.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Generative AI statement

The author(s) declare that no Gen AI was used in the creation of this manuscript.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

Al-Allaf, A. F., and Asker, B. H. (2022). Detecting the usage of a mobile phone during an online test using AI technology. *Przeglad Elektrotechniczny* 1:140341, 69–72. doi: 10.15199/48.2022.11.11

Berri, Rafael, Silva, Alexandre, Parpinelli, Rafael, Girardi, Elaine, and Arthur, Rangel. (2014). A pattern recognition system for detecting use of mobile phones while driving. VISAPP 2014.

Bochkovskiy, A., Wang, C. Y., and Mark Liao, H. Y. (2020). YOLOv4: optimal speed and accuracy of object detection. *arXiv*.

Cai, Z., and Vasconcelos, N., Cascade R-CNN: delving into high quality object detection, in: Proceedings of the IEEE conference on computer vision and pattern recognition, (2018).

Cao, G., Xie, X., Yang, W., Liao, Q., Shi, G., and Wu, J. (2018), Feature-fused SSD: fast detection for small objects, in: Ninth international conference on graphic and image processing (ICGIP 2017).

Cellphone Dataset Computer Vision Project. Workspace, 'cellphone dataset dataset', Roboflow universe. Roboflow, (2022). Available online at: https://universe.roboflow.com/workspace-f5gtr/cellphone-dataset

Ding, X., Zhang, X., Ma, N., Han, J., Ding, G., and Sun, J. (2021). Repvgg: making vgg-style convnets great again. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, Nashville, TN, USA; pp. 13733–13742.

Farmer, C. M., Braitman, K. A., and Lund, A. K. (2010). Cell phone use while driving and attributable crash risk. *Traffic Inj. Prev.* 11, 466–470. doi: 10.1080/15389588.2010.494191

Girshick, R. (2015). Fast R-CNN, in: Proceedings of the IEEE international conference on computer vision.

He, K., Gkioxari, G., Dollár, P., and Girshick, R. (2017). Mask R-CNN, in: Proceedings of the IEEE international conference on computer vision.

He, K., Zhang, X., Ren, S., and Sun, J., (2016), "Deep residual learning for image recognition," 2016 IEEE conference on computer vision and pattern recognition (CVPR), Las Vegas, NV, USA, pp. 770–778.

Huang, G., Liu, Z., Van Der Maaten, L., and Weinberger, K.Q. (2017). Densely connected convolutional networks. In Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), Honolulu, HI, USA; pp. 4700–4708.

Hussain, M. (2023). YOLO-v1 to YOLO-v8, the rise of YOLO and its complementary nature toward digital manufacturing and industrial defect detection. *Mach. Des.* 11:677. doi: 10.3390/machines11070677

Jegham, I., Khalifa, A. B., Alouani, I., and Mahjoub, M. A. (2020). A novel public dataset for multimodal multiview and multispectral driver distraction analysis: 3MDAD, signal process. *Image Commun* 88:115960. doi: 10.1016/j.image.2020.115960

Jiménez, F., Naranjo, J. E., Anaya, J. J., García, F., Ponz, A., and Armingol, J. M. (2016). Advanced driver assistance system for road environments to improve safety and efficiency. *Transp. Res. Procedia* 14, 2245–2254. doi: 10.1016/j.trpro.2016.05.240

Jocher, G., Chaurasia, A., and Qiu, J. YOLO by Ultralytics. GitHub. (2023). Available online at: https://github.com/ultralytics/ultralytics.

Jocher, G., Stoken, A., Borovec, J., Christopher, S.T.A.N., and Laughing, L.C. (2021). Ultralytics/yolov5: v4.0-nn.SiLU() activations, Weights & Biases Logging, PyTorch hub integration. Zenodo. Available online at: https://zenodo.org/record/4418161

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proc. IEEE* 86, 2278–2324. doi: 10.1109/5.726791

Li, C., Li, L., Jiang, H., Weng, K., Geng, Y., Li, L., et al. (2022). YOLOv6: a single-stage object detection framework for industrial applications. *arXiv*. arXiv:2209.02976

Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., and Belongie, S. (2017). Feature pyramid networks for object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), Honolulu, HI, USA; pp. 2117–2125.

Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., et al. Microsoft coco: Common objects in context. In Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 6–12 (2014).

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., et al. (2016). SSD: single shot multibox detector, in: Computer vision–ECCV 2016: 14th European conference, Amsterdam, The Netherlands.

Liu, S., Qi, L., Qin, H., Shi, J., and Jia, J. (2018). Path Aggregation network for instance segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), Salt Lake City, UT, USA; pp. 8759–8768.

Redmon, J., Divvala, S., Girshick, R., and Farhadi, A., (2016). You only look once: unified, real-time object detection, in: Proceedings of the IEEE conference on computer vision and pattern recognition.

Redmon, J., and Farhadi, A. (2018). Yolov3: an incremental improvement. *arXiv*.

Rehman, Abdul, Zaib, Alam, Azmat, Shoaib, and Khattak, Shahid. (2021). Detection of cell phone usage in restricted areas.

Ren, S., He, K., Girshick, S., and Sun, J. (2015). Faster R-CNN: towards real-time object detection with region proposal networks. *Adv. Neural Inf. Proces. Syst.* 28.

Shen, Q., Zhang, L., Zhang, Y., Li, Y., Liu, S., and Xu, Y. (2024). Distracted driving behavior detection algorithm based on lightweight StarDL-YOLO. *Electronics* 13:3216. doi: 10.3390/electronics13163216

Shukla, Vipin, Singh, Gaurav, and Shah, Pratik. (2013). Automatic alert of security threat through video surveillance system.

Talib, M., Al-Noori, A. H. Y., and Suad, J. (2024). Yolov8-cab: improved yolov8 for real-time object detection. Karbala international journal of modern. *Science* 10. doi: 10.33640/2405-609X.3339

Tan, M., and Le, Q. (2019). EfficientNet: rethinking model scaling for convolutional neural networks. In Proceedings of the international conference on machine learning (ICML), Long Beach, CA, USA.

Ultralytics. YOLOv5 2020. Available online at: https://github.com/ultralytics/yolov5

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, J., Gomez, A. N., et al. (2017). Attention is all you need. *Adv. Neural Inf. Proces. Syst.* 30.

Wang, C. Y., Bochkovskiy, A., and Liao, H. Y. M. (2022). YOLOv7: trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. *arXiv*. arXiv:2207.02696

Wang, Dan, Pei, Mingtao, and Zhu, Lan. "Detecting driver use of Mobile phone based on in-car camera". (2014) Tenth International Conference on Computational Intelligence and Security.

Zhang, R., Liu, G., Zhang, Q., Lu, X., Dian, R., Yang, Y., et al. (2025). Detail-aware network for infrared image enhancement. *IEEE Trans. Geosci. Remote Sens.* 63:5000314, 1–14. doi: 10.1109/TGRS.2024.3504240

Zhang, R., Tan, J., Cao, Z., Xu, L., Liu, Y., Si, L., et al. (2024). Part-aware correlation networks for few-shot learning. *IEEE Trans. Multimed.* 26, 9527–9538. doi: 10.1109/TMM.2024.3394681

Ziebinski, A., Cupek, R., Grzechca, D., and Chruszczyk, L. (2017). Review of advanced driver assistance systems (ADAS), in: AIP conference proceedings 1906, AIP Publishing LLC.