IJNRD.ORG ISSN : 2456-4184
INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD) | IJNRD.ORG
An International Open Access, Peer-reviewed, Refereed Journal

SMART OBJECT DETECTION FOR THE VISUALLY IMPAIRED USING YOLO AND VOICE ASSISTANCE

[1]Ann Jeba Jovitha M, [2]Caroline Grace L, [3]Gladys J, [4] Sivakumar K

[1]Student, [2]Student, [3]Student, [4]Assistant Professor,
1Department of Computer Science and Engineering,
[1]Francis Xavier Engineering College, Tirunelveli, Tamil Nadu, India

Abstract: The primitive machine learning algorithms that are present break down each problem into small modules and solve them individually. Nowadays requirement of detection algorithm is to work end to end and take less time to compute. Real-time detection and classification of objects from video records provide the foundation for generating many kinds of analytical aspects such as the amount of traffic in a particular area over the years or the total population in an area. In practice, the task usually encounters slow processing of classification and detection or the occurrence of erroneous detection due to the incorporation of small and lightweight datasets. To overcome these issues, YOLO (You Only Look Once) based detection and classification approach (YOLOv8) for improving the computation and processing speed and at the same time efficiently identify the objects in the video records. In addition to the advancements brought by YOLOv8, introduces a novel scheme to further enhance operational speed by incorporating distance calculation. Utilizing a formula based on the focal length and known width of objects, the system estimates the distance of detected objects from the camera. This distance information enriches the contextual understanding of object annotations, providing valuable insights for various analytical tasks. Moreover, the Smart Object Detection System extends its capabilities by integrating Optical Character Recognition (OCR) for character recognition. By combining YOLOv8 with distance calculation and OCR, the system aims to offer a comprehensive solution for real-time object detection, classification, distance estimation, and character recognition in video streams.

KEYWORDS
YOLOv8,Real-time Object Detection,Object Classification,Distance Estimation,Optical Character Recognition (OCR),Video Analytics, Focal Length Calculation,End-to-End Detection Systems,Computer Vision,Lightweight Models,Deep Learning,Context-Aware Detection,Real-time Video Processing,Intelligent Surveillance,Smart Traffic Monitoring,Object Tracking,Scene Understanding,Edge Computing,Camera-based Detection,Annotation Enrichment

INTRODUCTION

The process of identifying and locating instances of real-world objects—like cars, bikes, TVs, flowers, and people—in pictures or videos is known as object detection. By recognizing, localizing, and detecting multiple objects within an image, an object detection technique enables you to comprehend the details of a picture or video. Applications such as advanced driver assistance systems (ADAS), security, surveillance, and image retrieval typically use it.

These days, the main task in video surveillance applications is object detection from a video. The object detection technique is used to cluster pixels of required objects in video sequences.

In many applications, particularly video surveillance applications, the ability to detect objects in a video sequence is crucial. Pre-processing, segmentation, foreground and background extraction, and feature extraction are some methods that can be used to detect objects in a video stream. Objects in an image are easily detectable and identifiable by humans.

The human visual system is quick, precise, and capable of handling complicated tasks like recognizing several objects with little conscious thought. We can now easily train computers to detect and classify multiple objects within an image with high accuracy thanks to the availability of large amounts of data, faster GPUs, and improved algorithms.

An open source software library for numerical computation with high performance is called Tensor Flow. Because of its adaptable design, it enables straightforward computation deployment across a variety of platforms (CPUs, GPUs, and TPUs) on desktops,

server clusters, mobile devices, and edge devices. Researchers and engineers from Google's AI division, the Brain team, created Tensor Flow. It has strong support for deep learning and machine learning, and its flexible numerical computation core is utilized in a number of other scientific fields.

TensorFlow makes it simple to build, train, and implement object detection models. It also offers a set of detection models that have already been trained on the COCO, Kitti, and Open Images datasets. One of the many detection models is the combination of mobile net architecture and single shot detectors (SSDs), which is fast, effective, and requires little processing power to detect objects.

Real-time object detection is known as YOLO. It divides the image into regions using a single neural network and predicts possibilities and bounding boxes for each region. These bounding boxes are weighted according to predicted probabilities. Bounding boxes and class possibilities are directly predicted from full images by a single neural network in a single evaluation. The entire detection pipeline can be directly optimized for detection performance from beginning to end because it is a single network.

NEED OF THE STUDY

In today's data-driven world, accurate and real-time object detection plays a critical role across various domains such as surveillance, agriculture, inventory management, healthcare, and assistive technologies. Traditional methods involving manual inspection or basic image processing techniques are often time-consuming, error-prone, and lack adaptability in dynamic environments. With the advancement of deep learning, especially models like YOLOv8, there is a growing opportunity to enhance detection accuracy and efficiency.This study is necessary to explore how cutting-edge object detection models can be integrated with real-time feedback systems and accessibility features to create inclusive, scalable, and intelligent automation solutions suitable for real-world deployment.

ALGORITHMS

The object detection system in this study is based on YOLOv8 (You Only Look Once), OpenCV, and Pyttsx3, integrating deep learning, computer vision, and speech synthesis to provide real-time object detection and distance estimation. The algorithm is designed to detect objects in real-time, estimate their distance from the camera using a pre-defined focal length, and provide auditory feedback indicating whether an object is within a safe or unsafe range.

The YOLOv8m model, a mid-sized variation of the YOLO family that is renowned for its quickness and precision in object detection, is loaded first. The model can identify common objects in the environment because it has been pre-trained on the COCO dataset, which comprises 80 distinct object classes. A static image or a live camera feed is used as the detection system's input. The YOLO model is used by the system to identify objects and provide information like bounding boxes, confidence scores, and class labels after a frame is captured. A confidence threshold (set at 0.8) is used to weed out less trustworthy detections because object detection models occasionally produce false positives or identify objects with poor accuracy. To maximize Once an object has been detected and selected, the algorithm estimates its distance using the focal length distance formula. Distance estimation is a crucial aspect of the system, as it helps determine the proximity of detected objects relative to the camera. The formula used is based on the principles of pinhole camera projection, where the real-world distance of an object is computed as: Distance = (Known Width × Focal Length)/Width in Pixels.

This setup uses a known width of 60 cm as a reference, and the focal length has been pre-calibrated at 360 pixels. The width of the detected object in pixels is provided by the bounding box dimensions that are taken from YOLO detections. This enables the computation of the actual distance in centimeters. The system dynamically updates the perceived position of objects in relation to the camera by continuously measuring distance values.

The system incorporates Pyttsx3-based text-to-speech synthesis to improve usability, especially in safety-critical applications. Spoken words are generated from the detected object's distance and class name. The system warns and indicates that the situation is "Unsafe" if an object is discovered to be closer than 50 cm. If not, it confirms that the object is at a safe distance. The system is helpful for autonomous navigation, robotics, and visually impaired users because of its auditory feedback.

In order to overlay detection results on the captured frame, the system additionally incorporates a visual annotation mechanism. Bounding boxes are drawn around identified objects using OpenCV, along with class labels and the calculated distance displayed as text on the screen. While cv2.putText() adds textual information to make the output easier to interpret, cv2.rectangle() is used to highlight detected objects. These visual improvements give users an intuitive depiction of object proximity and aid in their understanding of the system's real-time predictions.

Rather than processing each frame, object detection is done at predetermined intervals (e.g., every 5 seconds) to guarantee computational efficiency. By lowering the computational load, this improves system efficiency without sacrificing real-time functionality. The system continuously updates the visual and aural outputs, performs the detection and distance estimation algorithm, and takes pictures from the camera. In order to prevent pointless calculations, the system outputs "No detection" if no object is detected. A subfield of computer vision called object detection draws bounding boxes around objects it detects in pictures or videos. For accessibility, the Smart Object Detection System combines voice responses, positional awareness, and object detection. The You Only Look Once (YOLO) algorithm is used to train the model on the Common Objects in Context (COCO) dataset.Together, these algorithms produce individualized, superior interior designs that maximize functionality, space efficiency, and aesthetics while requiring the least amount of human labour possible.

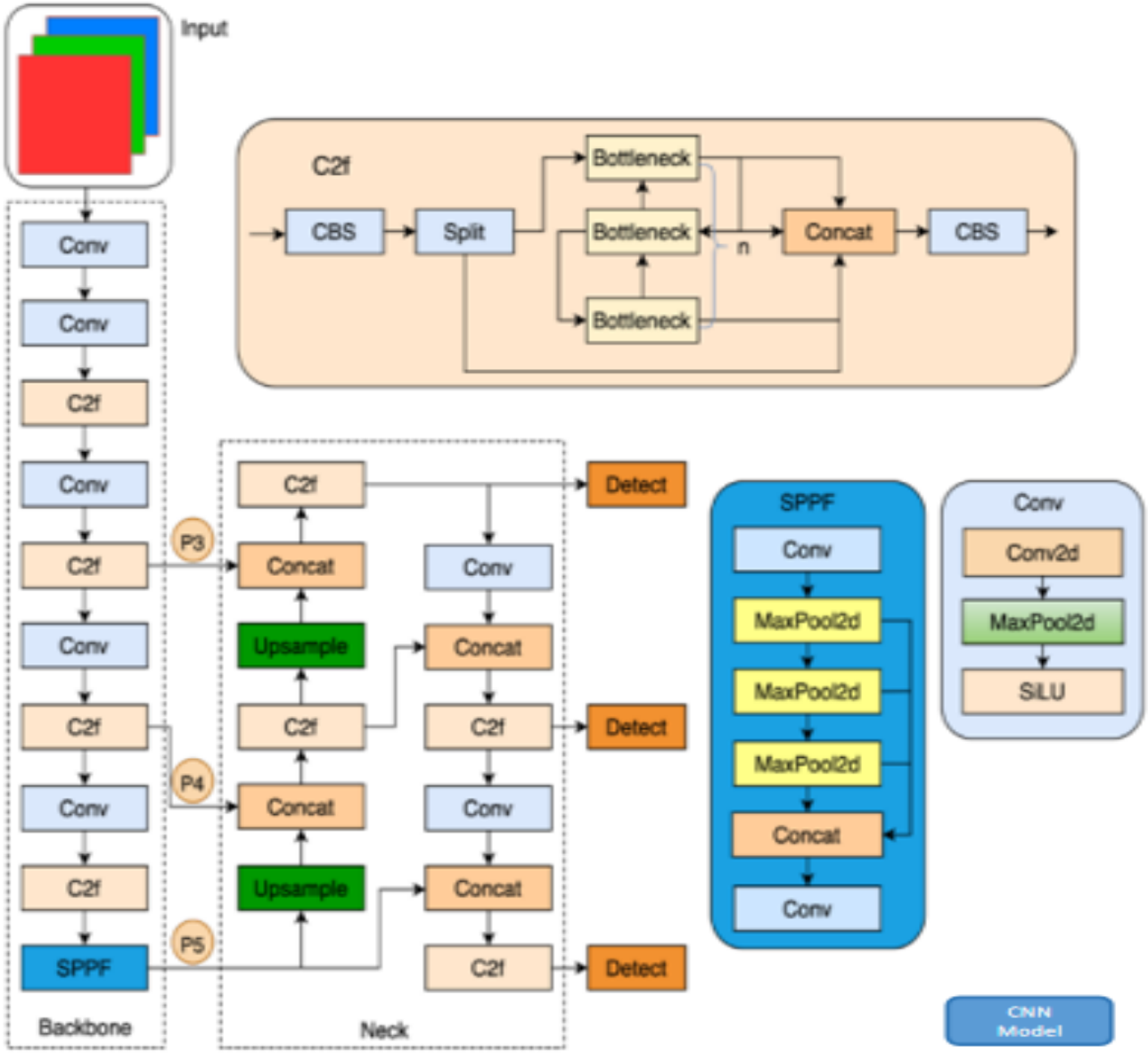

Figure 1: Architecture for Convolutional System

The **Figure 1** presents a Convolutional Neural Network (CNN)-based object detection model architecture, comprising three main components: Backbone, Neck, and Detection Head.The backbone extracts hierarchical features from the input image through multiple convolutional layers. It includes convolutional blocks (Conv), C2f modules, and a Spatial Pyramid Pooling-Fast (SPPF) block for enhanced feature extraction.The neck further processes the extracted features using upsampling and concatenation operations to fuse multi-scale feature maps. It contains C2f blocks, upsampling layers, and concatenation steps to refine feature representation.The detection head consists of convolutional layers and pooling operations, responsible for predicting object locations and classifications at different scales.A detailed view of the C2f module is also provided, showcasing its internal structure, including bottleneck layers and concatenation operations. The model follows a feature pyramid approach, allowing effective multi-scale object detection.

The algorithm effectively combines deep learning for object detection, classical vision-based distance estimation, and speech synthesis to create a robust, real-time detection system. It has potential applications in autonomous navigation, assistive technology for visually impaired individuals, and industrial automation, where understanding object distances and ensuring safety is critical. The integration of YOLO with OpenCV and Pyttsx3 creates an interactive and informative system that enhances situational awareness through both visual and auditory feedback.

PROPOSED SYSTEM

The proposed solution is a real-time, deep learning-based object detection and counting system using YOLOv8. It integrates OpenCV and TensorFlow to detect, classify, and count objects in live video feeds or images. The system also includes distance estimation and OCR modules, enhancing accessibility with voice feedback for visually impaired users. Designed for applications in security, agriculture, inventory, and smart surveillance, it delivers high-speed, accurate detection even in challenging environments. Custom datasets and hardware acceleration ensure adaptability and performance, making the solution scalable, efficient, and impactful across multiple industry use cases.

SYSTEM WORKFLOW

1. Input Acquisition

The system captures real-time images or video frames from a webcam or an image dataset. The input is processed at 80 frames per second (fps), ensuring smooth and efficient detection. To optimize performance, frame-skipping techniques are applied without compromising accuracy. The captured frames serve as the foundation for subsequent processing and detection tasks. By acquiring high-quality input, the system enhances detection precision and reduces processing latency.

2. Preprocessing

Before detection, the input images undergo preprocessing to enhance clarity and reduce noise. The frames are first converted to grayscale, which simplifies computations and improves feature extraction. Noise reduction techniques are applied to eliminate unwanted distortions, ensuring cleaner image processing. Contrast enhancement further improves object visibility, making detection more reliable. These preprocessing steps refine the input data, improving accuracy in object recognition.

3. Object Detection using YOLO Algorithm

The system employs the YOLOv8 algorithm, which performs object detection in a single pass for high efficiency. The model is trained on the COCO dataset, enabling it to recognize 80 different object categories. Each detected object is assigned a bounding box along with its positional information, such as top/middle/bottom and left/center/right. The model's deep learning framework ensures precise classification and quick response times. This approach significantly outperforms traditional region-based detection methods by reducing computational overhead.

4. Distance Estimation

To enhance accessibility, the system calculates the distance of detected objects from the camera. The distance is determined using the formula: **Distance = (Known Width × Focal Length) / Width in Pixels**. The known width of the object is referenced from predefined dataset values, while the focal length is derived from the camera specifications. This calculation helps in providing positional awareness, assisting users in navigating their surroundings. The estimated distance is used to generate real-time voice feedback for the user.

5. Optical Character Recognition (OCR) Module

The system includes an OCR module to extract and interpret text from detected objects. OpenCV and Tesseract OCR are used to process text embedded in images, making information more accessible. The process involves grayscale conversion, color inversion, brightness adjustment, noise reduction, and contour extraction. These steps ensure that text is accurately recognized, even under varying lighting conditions. The extracted text can then be converted into speech for voice-based output.

6. Prediction and Detection Mechanism

During the detection phase, input frames are resized to fit the YOLO model's 416×416 network format. The image is divided into an S×S grid, where each grid cell is responsible for detecting objects within its area. The model predicts bounding boxes along with confidence scores based on class probability and Intersection Over Union (IOU). Higher confidence scores indicate stronger detection reliability, while lower scores are filtered out. This process ensures efficient identification of objects with minimal computational cost.

7. Non-Maximum Suppression (NMS)

Multiple bounding boxes are initially predicted for each object in the image. To refine detection, the system selects the bounding box with the highest confidence score while suppressing overlapping boxes. If two boxes have an IOU greater than 0.5, the one with the lower score is removed. This step ensures that only the most relevant detection is retained for accurate classification. By applying NMS, redundant detections are eliminated, improving overall precision.

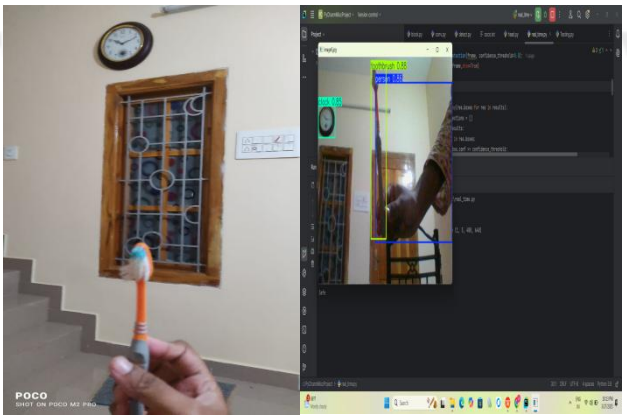

Figure: 2 Sample of Usage of non-max suppression

The **figure 2** illustrates a modular system architecture for object detection and text recognition, consisting of four main components: The system accepts inputs from a webcam or an image dataset, serving as the primary sources for visual data

processing. Performs grayscale conversion, noise reduction, and contrast enhancement to improve image quality for better detection.Utilizes the YOLO model for object detection, followed by Non-Maximum Suppression (NMS) to filter overlapping detections.Implements OpenCV and Tesseract OCR to extract textual information from detected objects.Computes object distances to enhance the interaction experience.The extracted and processed information is conveyed through a Voice Feedback Module and a User Interface, ensuring accessibility and usability.

Multi-Scale Predictions:

By utilizing multi-scale feature detection, Yolo v8 enhances the detection of small objects.Other grid sizes, such as 26×26 and 52×52, are used in place of just an 18×18 grid.This improves accuracy by enabling the recognition of objects at different scales.

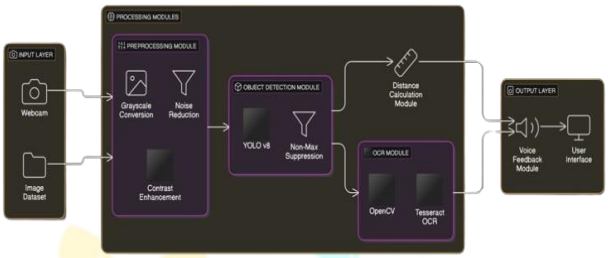

Figure: 3 Architecture Diagram of YOLO-Based Detection

Figure 4 The Object Detection module (YOLO v8) detects objects and their positions.The Distance Calculation module estimates object distances.The OCR module extracts text from images.

To improve accessibility, this system combines OCR, distance estimation, and YOLO-based object detection. Voice feedback, position awareness, and real-time bounding boxes guarantee a smooth experience for users, especially those who are blind or visually impaired. Because of its effectiveness and versatility, the model can be used for a wide range of applications, including security surveillance and assistive technologies.

RESULTS AND DISCUSSION

The system utilizes a webcam to detect and classify objects in real-time. The model accurately identifies objects, displaying their classification confidence and distance from the camera.

Model Creation and Training:

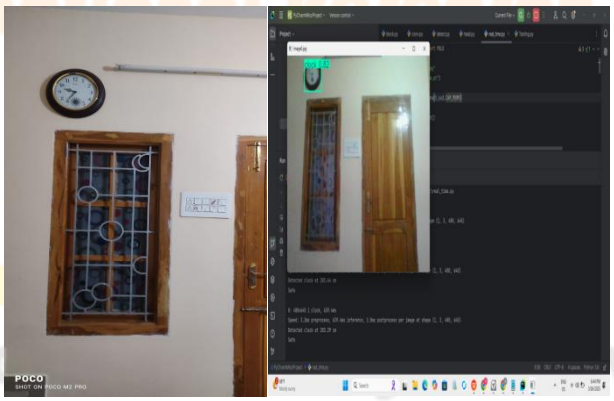

Figure: 4 Real-Time Object Detection Output

Figure 4 displays the real-time detection of a clock using the webcam. The model achieves 0.82 accuracy for object clock and detects the clock which at a distance of 285 cm.

CONCLUSION:

For visually impaired users, the use of YOLOv8 for voice assistance, distance estimation, and real-time object detection provides an extremely effective solution. In contrast to conventional techniques like RCNN, Fast RCNN, and SSD, which depend on region-based strategies, YOLOv8 allows for direct end-to-end detection with low latency, which makes it ideal for real-time applications. The model ensures smooth object tracking and recognition by processing video frames quickly.

With the help of precise bounding boxes, class labels, and confidence scores ranging from 80 to 99 percent, YOLOv8 effectively detects objects from the COCO dataset in Smart Object Detection. By figuring out whether objects are within a safe or dangerous range, the system incorporates a distance estimation algorithm based on focal length and object width, improving situational awareness.The detection speed rate varies between 2 and 4 ms, depending on the objects in the frame. For visually impaired users

who depend on real-time alerts to safely navigate their environment, this functionality is especially helpful.

Furthermore, accessibility is guaranteed by voice assistance driven by pyttsx3, which provides spoken feedback on objects detected and their distances. By bridging the gap between human interaction and computer vision, this integration makes the system useful in everyday situations. The potential of YOLOv8 in assistive technology and intelligent object recognition systems is demonstrated by the way that real-time detection, precise classification, and voice output improve usability while preserving high accuracy.

Future Enhancements

Although the current system detects and classifies objects in real-time with effectiveness, there are a number of ways to improve its efficiency, accuracy, and usability. Future improvements consist of:

Increased Model Accuracy: In complex environments, more training with a wider range of datasets can improve detection accuracy and lower misclassifications.

Edge Device Deployment: Real-time detection without the need for powerful computers can be made possible by tailoring the model for edge devices such as the Raspberry Pi or Jetson Nano.

Integration with Cloud Storage: Data analysis and future retrieval are made possible by storing detected objects and their classification history in the cloud.

Support for Mobile Applications: Creating an object detection mobile application can enhance smartphone usability and accessibility.

Enhanced Distance Estimation: Accurate distance measurements can be achieved by utilizing sophisticated depth-sensing methods like LiDAR or stereo vision.

Voice Feedback Integration: Including voice output for objects that are detected can improve system usability, particularly for users who are blind or visually impaired.

The system can become more reliable, effective, and appropriate for a greater variety of real-world applications by implementing these improvements.

References

[1] Chen, X.; Ma, H.; Wan, J.; Li, B.; Xia, T. Multi-view 3D Object Detection Network for Autonomous Driving.In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Hawaii, HI, USA,21–26 July 2017; pp. 6526–6534.

[2] Christ, P.F.; Kaissis, G.; Ettlinger, F.; Kaissis, G.; Schlecht, S.; Ahmaddy, F.; Grün, F.; Menze, B.; Valentinitsch, A.;Ahmadi, S.-A.; et al. SurvivalNet: Predicting patient survival from diffusion weighted magnetic resonance images using cascaded fully convolutional and 3D Convolutional Neural Networks. In Proceedings of the IEEE International Conference on International Symposium on Biomedical Imaging, Melbourne, Australia,18–21 April 2017; pp. 839–843.

[3] Hanchinamani, S.R.; Sarkar, S.; Bhairannawar, S.S. Design and Implementation of High Speed Background Subtraction Algorithm for Moving Object Detection. In Proceedings of the IEEE International Conference on Advances in Computing, Communications and Informatics, Jaipur, India, 21–24 September 2016; pp. 367–374.

[4] Mao, J.; Xiao, T.; Jiang, Y.; Cao, Z. What Can Help Pedestrian Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Hawaii, HI, USA, 21–26 July 2017; pp. 3127–3136.

[5] M. Ponika, K. Jahnavi, P. S. V. S. Sridhar and K. Veena, "Developing a YOLO based Object Detection Application using OpenCV," 2023 7th International Conference on Computing Methodologies and Communication (ICCMC), Erode, India, 2023, pp. 662-668, doi: 10.1109/ICCMC56507.2023.10084075.

[6] Pedoeem, J.; Huang, R. YOLO-LITE: A Real-Time Object Detection Algorithm Optimized for Non-GPUComputers. In Proceedings of the IEEE International Conference on Big Data, Seattle, WA, USA, 10–13 December2018; pp. 2503–251

[7] Redmon, J.; Farhadi, A. YOLO9000: Better, Faster, Stronger. IEEE Trans. Pattern Anal. 2017,29, 6517–6525.

[8] Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region ProposalNetworks. IEEE Trans Pattern Anal Mach Intell. 2017,39, 1137–1149.

[9] Senicic, M.; Matijevic, M.; Nikitovic, M. Teaching the methods of object detection by robot vision.In Proceedings of the IEEE International Convention on Information and Communication Technology,Electronics and Microelectronics, Opatija, Croatia, 21–25 May 2018; pp. 558–563.

[10] S hafiee, M.J.; Chywl, B.; Li, F.; Wong, A. Fast YOLO: A Fast You Only Look Once System for Real-time Embedded Object Detection in Video. J. Comput. Vis. Image Syst. 2017,3, 171–173.