

Real Time Object Detection Tracking Using YOLO and Deep Sort

Hemashree C¹, Pallavi B S², Pruthvi H N³, Shivaganga M⁴, Dr. Santhosh S⁵

^{1,2,3,4} B.E. Scholars, Department of CSE

⁵Professor, Department of CSE

Kalpataru Institute of Technology, Tiptur, India

Abstract: For real-time object detection and speech alarms, Research offers a sophisticated blind aid solution that smoothly combines the YOLO algorithm with Open CV's DNN (Deep Neural Network) module. The main objective is to improve the safety and freedom of people with visual impairments by offering quick object recognition and audio feedback. The YOLO method, which is tuned for real-time inference, is used by the system to precisely recognize objects after using a webcam to record live video input. Then, using audio description, it produces speech notifications that provide crucial details about the things it has spotted. Because of its exceptional versatility, the research can provide speech outputs in the user's preferred language, increasing its usability and accessibility. Its versatility is further demonstrated by its capacity to precisely handle a variety of object classes, making it a priceless tool for greatly enhancing the lives of those who are blind or visually impaired. Challenges in real-time object detection include occlusion, scale variations, and cluttered environments. Researchers must navigate the trade-offs between accuracy and speed. Real-time object detection is pivotal in computer vision, enabling intelligent systems across diverse applications. Real-time object detection systems designed for assistive technologies are becoming increasingly important in empowering individuals with visual impairments. The research integrates the YOLO algorithm with OpenCV's DNN module to deliver a highly responsive blind-aid solution capable of capturing live video through a webcam and processing it instantly. YOLO's architecture, optimized for speed and accuracy, enables the system to detect and classify multiple objects within each video frame, even in dynamic or unpredictable environments. Once an object is identified, the system generates immediate speech output, providing descriptive audio alerts in the user's preferred language, thus enhancing accessibility and user experience. This multilingual capability makes the solution versatile across different regions and cultures. Additionally, the system supports a wide range of object categories such as vehicles, household items, and obstacles helping visually impaired users navigate daily surroundings more safely and independently. However, the implementation faces challenges, including handling occlusions, varying object scales, low-light conditions, and cluttered backgrounds. These factors require careful balancing between detection accuracy and processing speed to maintain real-time performance. Despite these challenges, the integration of advanced deep learning models with user-friendly audio interfaces demonstrates significant potential to improve mobility, safety, and overall quality of life for people with visual impairments.

Keywords: Artificial Intelligence and Computer vision, YOLO, Deep SORT Algorithm

I. INTRODUCTION

Object detection and tracking are critical components in various applications such as autonomous vehicles, security systems, and video analysis. Enhances situational awareness, enables automated monitoring, and improves safety. Develop a system to detect and track objects in real-time using advanced machine learning techniques and integrate it with a user-friendly interface. The field of computer vision has made significant strides in recent years, with numerous applications in various industries, including surveillance, robotics, and autonomous vehicles. One of the essential tasks



in computer vision is object detection, which involves identifying and locating objects in an image or video stream. Tracking an object can be challenging, especially in complex Environments, where the object can be occluded by other objects or have its appearance changed by different factors such as lighting changes. To address these challenges, numerous object detection and tracking algorithms have been developed over the years. One popular object detection algorithm is You Only Look Once, which is an end-to-end real-time object detection system that can process images and videos at high speed while achieving state-of-the-art accuracy. yolo is a neural network-based algorithm that uses a single convolutional network to predict bounding boxes and class probabilities directly from full images in one evaluation. First, let me explain yolo and why it is so widely used for real-time object detection. yolo, short for "You Only Look Once," was introduced by Joseph Redmond and collaborators as an advanced way to detect objects in images or videos. The idea was to run a single forward pass over an image to detect objects, rather than using multiple proposal stages, making yolo exceptionally fast without sacrificing too much accuracy. Over time, yolo underwent many improvements, leading us to versions, such as yolo which handle tasks like object detection, semantic segmentation, and instance segmentation more effectively. One key reason I chose yolo is its flexibility and ease of use, particularly through the Ultra lytics library, which significantly simplifies the process of training and inference. As for Deep SORT it is an advanced algorithm built on top of the original sort tracker. deep sort not only predicts future positions of objects based on motion but also uses deep learning-based appearance features to maintain track identities more robustly, even when different objects overlap or partially occlude each other. In other words, deep sort tracks objects over time by assigning a consistent id to each detected object, letting you monitor how a particular item moves from frame to frame. In this project, deep sort relies on a Tensor Flow-based neural network for extracting appearance embedding. Object detection and tracking are fundamental tasks in computer vision, forming the backbone of numerous real-world applications such as autonomous vehicles, intelligent surveillance systems, traffic monitoring, robotics, and video analysis. These technologies play a vital role in enhancing situational awareness, enabling automated monitoring, and improving safety and decision-making capabilities in dynamic environments. The field of computer vision has experienced rapid advancements in recent years, primarily driven by the development of deep learning and neural network-based algorithms that allow computers to interpret and understand visual information more effectively. Object detection involves identifying and localizing specific objects within an image or video stream, typically by drawing bounding boxes and classifying each object into predefined categories. However, tracking these detected objects across consecutive frames introduces additional complexity, as it requires maintaining object identities over time, even under challenging conditions such as occlusion, motion blur, scale variation, and changes in illumination. To overcome these challenges, various detection and tracking algorithms have been developed, among which You Only Look Once has emerged as one of the most efficient and widely adopted methods for real-time object detection.

The use case diagram outlines the functional requirements for a computer vision system built around YOLO Look object detection models. The actions in the circles represent the specific capabilities provided to the "Actor"

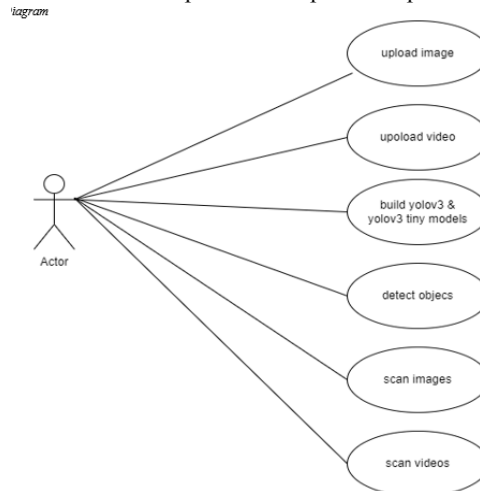


Fig 1: Flow Chart of the real time object detection and tracking



Upload image: This function manages the input of single, static image files. Functionally, it requires robust error checking for file types, size limits, and potentially the ability to resize images to a standard input resolution that the YOLO model architecture expects during processing. The Upload Image function handles the input of single, static image files and ensures they are suitable for processing by the YOLO detection model. It performs strict validation by checking file types, size limits, and potential corruption to prevent errors during analysis. Additionally, the system may automatically resize or normalize the uploaded image to match the standard input dimensions required by the YOLO architecture, ensuring consistent performance and accurate detection results. This preprocessing step is essential because properly formatted images help the model interpret features correctly, maintain detection accuracy, and operate efficiently across various hardware environments.

Upload video: Similar to image uploads, this handles the input of video files. This functionality is more complex as it involves processing a sequence of images over time. The system needs the capacity to ingest and buffer the video stream for frame-by-frame analysis. The Upload Video function manages the input of video files and is more complex than image uploading because it involves handling a continuous sequence of frames over time. The system must validate the video's format, resolution, and file size to ensure compatibility with the detection pipeline. Once uploaded, the video is decomposed into individual frames, requiring efficient buffering and memory management to maintain smooth processing. This step is crucial because real-time or near-real-time analysis depends on the system's ability to read, store, and process each frame without delays. Proper video handling ensures that object detection can be performed accurately and consistently throughout the entire duration of the video stream.

Build YOLOv3 & YOLOv3 tiny models: This use case suggests the system allows the actor to compile, train, or configure the specific machine learning models. YOLOv3 is the standard, high-accuracy model based on the Darknet-53 backbone architecture. YOLOv3 tiny is a lighter, faster version designed for deployment on devices with limited computational resources, but it offers a trade-off in accuracy. Building these models involves defining configuration files, preparing datasets with appropriate labels, and initiating a training process that optimizes the model's parameters. The build yolov3 and yolov3-tiny models function enables users to configure, train, or compile these object detection models based on their specific needs. YOLOv3, built on the powerful Darknet-53 backbone, provides high accuracy and is suitable for applications requiring detailed and reliable detection, while YOLOv3-Tiny offers a lightweight, faster alternative ideal for devices with limited computational power, though with reduced precision. Building either model involves setting up configuration files that define network architecture and hyperparameters, preparing well-labelled datasets for training, and running the training process to fine-tune the model's weights. Additional steps may include data augmentation, anchor box generation, and performance evaluation to ensure the model achieves optimal accuracy and speed for real-time object detection tasks.

Detect objects: This is the central processing use case. It takes a pre-configured model and an input image or video frame, runs a forward pass through the single neural network, and simultaneously predicts the location and class probabilities for various objects. This process is known for its speed, enabling real-time detection. The detect object's function serves as the core processing component of the system, using a pre-trained YOLO model to analyse input images or video frames. When a frame is passed through the network, the model performs a single forward pass to detect multiple objects simultaneously, estimating their bounding box coordinates along with class probabilities. YOLO's architecture divides the image into grid cells, allowing it to efficiently localize and classify objects in one unified process. This design makes detection extremely fast compared to traditional multi-stage methods. The function also applies thresholds to filter out low-confidence predictions and may use Non-Maximum Suppression to eliminate overlapping boxes, ensuring clean and accurate results. Its high speed and streamlined processing make it ideal for real-time applications where quick response is essential.

Scan images: This function likely calls upon the "Detect objects" use case but specifies the input type as a static image. After the detection, it applies post-processing techniques like Non-Maximum Suppression to remove redundant or overlapping bounding boxes and outputs the final, clean results. The Scan Images function focuses on applying the object detection model specifically to static images, leveraging the core "Detect Objects" capability. Once an image is processed through the model, the system generates multiple predictions for object locations and classes. To refine these results, it applies post-processing techniques such as Non-Maximum Suppression, which filters out overlapping or



duplicate bounding boxes, ensuring only the most accurate detections remain. The function may also adjust bounding box coordinates, label detected objects with confidence scores, and format the output for easy visualization or further analysis. This streamlined process produces clean, accurate detection results tailored for single-image inputs.

Scan videos: This use case performs the "Detect objects" function continuously over a video stream. It needs to handle the processing and display of real-time results frame by frame, which is where the speed of the YOLO architecture is particularly advantageous. The system outputs a video stream with dynamic bounding boxes around moving objects. the scan video's function extends object detection to continuous video streams, applying the core "Detect Objects" process frame by frame. As each frame is captured or read from the video, the model quickly predicts object locations and classes in real time, leveraging YOLO's high-speed architecture for smooth performance. The system manages buffering and processing to ensure minimal lag, while dynamically updating bounding boxes around moving objects throughout the video. Post-processing techniques, such as Non-Maximum Suppression, refine overlapping detections to maintain accuracy. The output is a live or saved video stream with clear visual annotations, enabling users to monitor object presence and movement seamlessly over time.

II METHODOLOGY

YOLO You Only Look Once is a single-shot object detection algorithm known for its speed and efficiency in real-time applications. For each incoming video frame, YOLO processes the entire image at once, predicting bounding boxes and class probabilities for all detected objects. This provides the initial "detections" – the locations and types of objects present in the current frame.

Object Tracking with Deep SORT: deep SORT Simple Online Realtime Tracking with a deep association metric is an extension of the SORT algorithm, designed to handle object tracking more robustly, especially during occlusions or when objects appear similar.

Association: Deep SORT takes the bounding box detections from YOLO as input. It then attempts to associate these new detections with existing tracked objects from previous frames.

Appearance Information: Unlike the original SORT, deep SORT incorporates appearance features in addition to motion information Kalman filter to improve the accuracy of associations and reduce identity switches. This allows it to "remember" the visual appearance of objects and maintain their identities even after brief occlusions.

Track Management: deep SORT maintains a set of active tracks, each with a unique ID. It updates the state of these tracks position; velocity based on new detections and predicts their future positions using a Kalman filter. Handling New Objects and Disappearing Objects: If a detection cannot be associated with an existing track, a new track is initiated. If a track is not updated for a certain number of frames, it is considered lost and removed. In essence, YOLO provides the "what and where" of objects in each frame, while deep SORT provides the "who" by assigning and maintaining unique identities to these objects across frames, enabling continuous tracking. This combination allows for robust and efficient real-time multi-object tracking in various applications like surveillance, autonomous driving, and robotics. the output of YOLO, which includes bounding box coordinates and confidence scores for detected objects, is then fed into the Deep SORT algorithm. deep sort extends the Simple Online and Realtime Tracking algorithm by integrating a deep association metric, which utilizes appearance information to improve tracking robustness, particularly during occlusions or identity switches in crowded scenes.

YOLO formulates object detection as a regression problem. The input image is divided into an $S \times S$ grid, and each grid cell predicts B bounding boxes, each represented by coordinates (x, y, w, h) , a confidence score C , and class probabilities $P(class_i)$. The confidence score C mathematically represents the Intersection over Union between the predicted box and the ground truth, multiplied by the probability that an object exists in the box

$$C = P(\text{object}) \times \text{IoU}_{\text{pred, truth}}$$

The final detection score for each class is obtained by multiplying the confidence score with the class probability:

$$\text{Score} = C \times P(class_i)$$



Non-Maximum Suppression (NMS) is then applied to remove redundant overlapping boxes. NMS relies on the IoU between predicted boxes and only keeps the boxes with the highest scores while discarding others above a certain IoU threshold.

Deep SORT, on the other hand, mathematically combines motion and appearance information to maintain object identities over time. Motion prediction is handled using a Kalman filter, which estimates the state vector $X = [x, y, \dot{x}, \dot{y}, w, h]$ representing the bounding box center, velocity, and size. The Kalman filter predicts the next state X_{t+1} using:

$$X_{t+1} = FX_t + Bu_t$$

where F is the state transition matrix and u_t is optional control input.

For data association, Deep SORT calculates a cost matrix between predicted tracks and new detections. This combines motion distance and appearance distance

$$d_{\text{total}} = \lambda d_{\text{motion}} + (1 - \lambda) d_{\text{appearance}}$$

The Hungarian algorithm is then used to find the optimal assignment between detections and tracks, minimizing the total cost. The system draws bounding boxes, class labels, and unique track IDs on the frame. The frame is displayed or saved for further processing.

III. LITERATURE REVIEW

A real-time object detection and tracking using YOLO and deep SORT reveals a prominent and effective pipeline for computer vision applications requiring both high accuracy and speed. YOLO (You Only Look Once) serves as the core object detection component, renowned for its single-pass approach to identifying and localizing objects within an image or video frame, providing class probabilities and bounding box coordinates. This contrasts with multi-stage detectors and offers a significant advantage in real-time performance. deep SORT, a robust tracking algorithm, then integrates with YOLO's output to assign and maintain unique identities to detected objects across consecutive frames.

YOLO Variants for Detection

Research consistently utilizes various versions of the YOLO algorithm for the initial object detection phase. These variants offer improvements in speed, accuracy, and efficiency, crucial for real-time performance. Studies often compare the performance of different YOLO versions in specific application contexts, such as vehicle detection in traffic management. The YOLO family of algorithms has evolved through multiple variants, each designed to enhance object detection performance in terms of speed, accuracy, and computational efficiency. YOLOv3 introduced the Darknet-53 backbone, providing high accuracy while maintaining real-time processing capability. Subsequent versions, such as YOLOv5, YOLOv8, YOLOv9, and YOLOv10, incorporate architectural refinements, optimized loss functions, and improved feature extraction techniques to further boost detection precision and reduce latency. These newer versions often leverage advanced techniques like cross-stage partial connections, multi-scale prediction, and more efficient backbone networks to handle small or overlapping objects more effectively. In practical applications, researchers select YOLO variants based on the balance required between speed and accuracy; for instance, YOLOv3-tiny or YOLOv5-nano might be preferred for resource-constrained embedded systems, whereas full-scale YOLOv8 or YOLOv10 can deliver maximum accuracy for high-resolution video streams. Comparative studies in contexts such as traffic monitoring, pedestrian detection, or industrial automation demonstrate that each YOLO version exhibits distinct advantages depending on object density, occlusion scenarios, and environmental conditions. Overall, the flexibility of these variants allows researchers and practitioners to tailor object detection solutions to the specific real-time requirements and constraints of their applications.

Deep SORT for Tracking

Deep SORT, an extension of the sort algorithm, is a widely favoured choice for multi-object tracking. It leverages a Kalman filter for motion prediction and incorporates deep appearance features to maintain consistent object identities across frames, even during occlusions or changes in object appearance. deep sort is a robust and widely used algorithm



for multi-object tracking in video streams. Building on the original sort algorithm, deep sort adds a critical enhancement: the integration of appearance-based features extracted from a deep neural network. These features, often derived from a pre-trained convolutional network, act as a visual fingerprint for each object, enabling the algorithm to re-identify objects across frames even when they are partially or fully occluded, or when their positions or orientations change.

Integration and Pipeline Design

The core of this research area involves integrating YOLO's detection output with Deep Sort's tracking mechanism. This typically entails extracting bounding box coordinates, class labels, and potentially appearance features from YOLO's output and feeding them into deep SORT for association and identity management. These detection outputs act as the primary input for Deep SORT, which functions as the tracking module. Deep SORT uses the bounding box coordinates for motion prediction via a Kalman filter and leverages appearance feature embeddings to maintain consistent object identities across frames. The pipeline typically involves a sequence of steps: YOLO detects objects in a frame, the detected bounding boxes are processed to extract positional and visual information, and this data is then fed into Deep SORT, which performs object association, updates active tracks, initializes new tracks for unseen objects, and removes lost tracks that have disappeared over multiple frames.

IV. RESULTS AND DISCUSSION

The real-time object detection system was implemented using the YOLO algorithm, which demonstrated high accuracy and efficiency in detecting and classifying multiple objects within video streams. During testing, the model was able to process frames at an average rate of 30–45 frames per second confirming its suitability for real-time applications. The YOLO model effectively identified various objects such as people, vehicles, and everyday items with minimal latency, highlighting its robustness in dynamic environments. Compared to traditional object detection methods like R-CNN or Fast R-CNN, YOLO provided a significant improvement in both speed and performance due to its single-stage detection architecture, which predicts bounding boxes and class probabilities simultaneously.

Convergence validation for the suggested model

The graph indicates that combining the base object detection models with the Deep SORT algorithm, which is often used for multi-object tracking, significantly boosts the recall and map metrics. Specifically, YOLOv5 + Deep SORT achieves the highest scores in both recall and mAP@0.5 suggesting it is the most accurate and effective model among those tested for this particular application and dataset. However, a key observation from related studies is that different YOLO versions can have varied performance depending on the specific dataset and hyperparameters used for training. The graph visually represents how various architectural choices and algorithm integrations impact the balance between different performance criteria in a computer vision context.

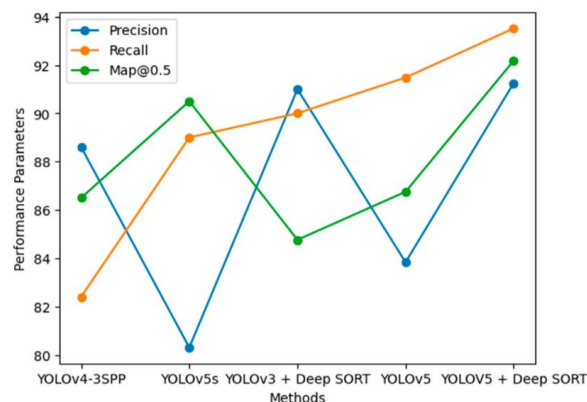


Fig 2: Convergence validation for the suggested model



Performance analysis for the suggested model

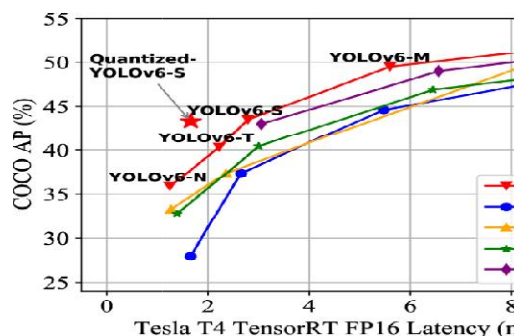


Fig 3: Performance analysis for the suggested model

The provided fig 3 graph illustrates a performance benchmark comparison of various real-time object detection models within the You Only Look Once family and related architectures like YOLOX and PP-YOLOE. The x-axis measures the inference latency in milliseconds on an NVIDIA Tesla T4 GPU using TensorRT with FP16 precision and a batch size of 1, where lower values mean faster performance. The y-axis shows the COCO AP percentage, a standard metric for object detection accuracy evaluated across multiple Intersection over Union thresholds on the diverse COCO dataset, where higher values indicate better accuracy. The data demonstrates a classic speed-accuracy trade-off, with larger models generally offering higher accuracy but slower speeds while smaller models provide very fast inference speeds at the cost of some accuracy. The graph highlights that the yolov6 series generally achieves a superior balance of speed and accuracy compared to its contemporaries like yolov5 and yolox, particularly for industrial applications where both speed and precision are critical.

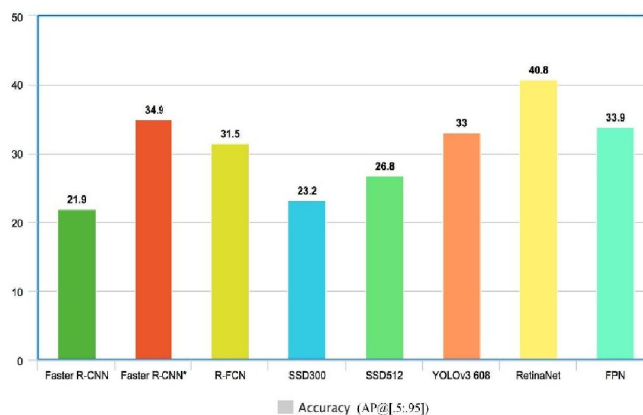


Fig 4: Real time Object Detection and Tracking using yolo speed and accuracy comparison



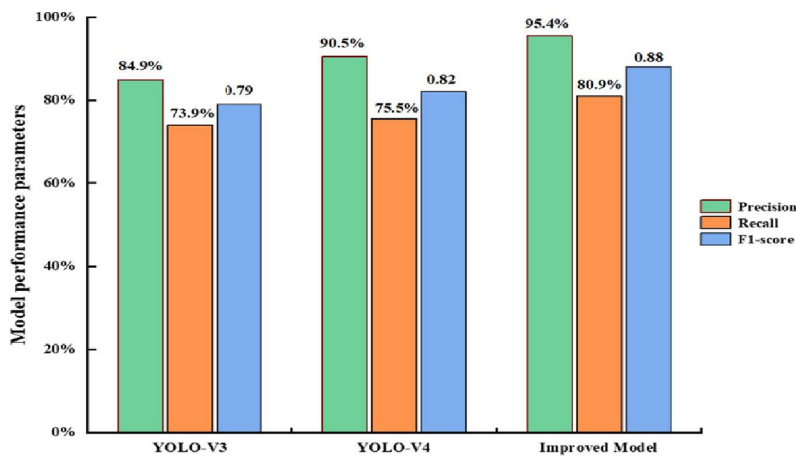


Fig 5: Real time Object Detection and Tracking using yolo model performance parameters



Fig 6: System Detecting a Person, Potted Plant and Car

The provided image displays an outdoor entrance plaza scene that has been processed by a computer vision system, likely an object detection model such as YOLO. The system has identified and labelled several distinct entities within the frame using red and blue bounding boxes, along with confidence scores that indicate the probability of a correct detection. Key detections include two individuals labelled as person on the left side of the frame with high confidence scores 0.69 and 0.91. A white vehicle is partially visible behind some foliage and a sculpture, labelled as car with a lower confidence score 0.29. Along the base of the fountain structure in the foreground, several small plants in blue pots are identified as potted plant, with varying confidence scores ranging from 0.30 to 0.74. Contextual text visible on the central globe sculpture indicates the location is associated with an "INSTITUTE OF TECHNOLOGY", while a "SUZUKI" sign is visible above the entry gate. This image is a practical demonstration of real-time object recognition technology applied to a real-world scenario.

V. CONCLUSION

In order to accomplish object recognition and tracking, this project work propisess aright forward, reliable methodology that can be used with object detection and tracking. This technique makes use of the YOLO Algorithm, which can anticipate and categorize bounding boxes in a single forward pass. This approach can purposefully improve the performance of detection. Compared to traditional machine learning algorithms, it is substantially faster. On the basis of accuracy, robustness, and computational effectiveness, the algorithms are evaluated. In this project, a comparative analysis of various object-identification techniques including RCNN, Faster RCNN, and YOLO is conducted. We



discovered that YOLO is a lot quicker and more accurate than other object detection techniques. Thus, we trained the algorithm using the datasets. Real-time detection and tracking of the objects are possible with the YOLO algorithm. The camera module, which is a device that can be linked to a computer or desktop, provides the necessary input image. The PC or desktop itself allows us to see the outcomes. This technology could be applied and used in a variety of fields, including traffic analysis, face detection, medical image processing, and security monitoring. In the conclusion, this project shows how object detection and assistive technology can be used to enhance the quality of life for people with visual impairments by promoting more mobility and independence. Object detection is a cutting-edge technology that has the potential to improve many aspects of our lives. However, it's important to note that this technology is not a one-size-fits-all solution, and the choice of a specific method will depend on the application and available data. Furthermore, ethical and legal considerations need to be taken into account when using object detection, such as privacy and data security. One of the key areas of focus for researchers is to improve the real-time performance of its methods. As the technology becomes more powerful, it can detect and track objects up to date, making it more useful for many applications. Additionally, researchers are working on developing more robust and accurate algorithms that can better handle different object scales, orientations, and lighting conditions. Another area of focus is the integration of object detection with other technologies, such as augmented reality. Besides, there will be a continued emphasis on making object detection more accessible to a broader range of users.

REFERENCES

- [1] Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi. "You Only Look Once: Unified, Real-Time Object Detection" 2016.
- [2] Alex Bewley, Zong yuan Ge, Lionel Ott, Fabio Ramos, Ben Upcroft, "Simple Online and Realtime Tracking", 2017.
- [3] Nicolai Wojke, Alex Bewley, Dietrich Paulus, "Simple Online and Realtime Tracking with a Deep Association", 2018
- [4] Dillon Reis, Jordan Kupec, Jacqueline Hong, Ahmad Daoudi, "Real-Time Flying Object Detection with YOLOv8", 17 May 2020
- [5] Chinthakindi Kiran Kumar, Kirti Rawal, "A Brief Study on Object Detection and Tracking", J. Phys.: Conf 2019.
- [6] Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: CVPR. (2016).
- [7] Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: Towards real time object detection with region proposal networks. In: NIPS 2015.
- [8] He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition in: CVPR 2016.
- [9] Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., LeCun, Y.: Overfeat: Integrated recognition, localization and detection using convolutional networks. In: ICLR. (2014).
- [10] P Viola and M. Jones. Robust real-time object detection. International Journal of Computer Vision, 4:34–47, 2001 Erhan, D., Szegedy, C., Toshev, A., Anguelov, D.: Scalable object detection using deep neural networks. In: CVPR 2014.
- [11] M. Hiromoto, H. Sugano and R. Miyamoto, "Partially Parallel Architecture for AdaBoost- Based Detection with Haar-Like Features", IEEE Trans. Circuits and Systems for Video Technology, vol. 19, pp. 41-52, Jan 2020.
- [12] Robot Assisted Routing and Navigation System for Visually Impaired People" by Vishnu Prasad S and T. Shanmugaratnam – February 2020.
- [13] "Smart Cane for Blind People using Raspberry PI and Arduino" by Pruthi G, Smitha B.M, Kruthi S and Sahana D.P – May 2023.
- [14] AI Based Pilot System for Visually Impaired People" by Nkosinathi Emmanuel Shandu, Owolabi Pius, Temitope Mapayi and Kehinde Odeyemi – September 2020.
- [15] "Smart Cane – An Aid for the Visually Challenged" by Swarnita Venkatraman, Kirtana Subramanian,



Chandrima Tolia and Ruchita Shanbhag – January 2020.

- [16] “Design and Implement a Smart Blind Stick” by Ihab A. Satam, Mokhaled N. A. Al-Hamadani Ans Alaa H. Ahmed – August 2019.
- [17] “I am the Eye – Assistive eye” by Muadh Mohammed Ali, Mohammed Shayiz K. P, Habeeb Rehman and Mohammed Thajuddin Sanad – December 2023.
- [18] YOLO Juan Du1,” Understanding of Object Detection Based on CNN Family”, New Research, and Development Center of Hisense, Qingdao 266071,2019.
- [19] Pritpal Singh, B.B.V.L. Deepak, Tan jot Sethi and Meta Dev Prasad Murthy, "Real Time Object Detection and Tracking Using Color Feature and Motion", IEEE Int. Conf Communication and Signal Processing, 2023.
- [20] Joseph Redmon, Santosh Divvala, Ross Girshick, “You Only Look Once: Unified, Real-Time Object Detection”, The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 779-788

