SSRG International Journal of Electronics and Communication Engineering
ISSN: 2348-8549/ https://doi.org/10.14445/23488549/IJECE-V10I9P106

Volume 10 Issue 9, 52-62, September 2023

Original Article

Design and Evaluation of Image Quality Enhancement to Augment Object Detection and Tracking Framework to Support Visually Impaired Persons

S. Sajini[1], B. Pushpa[2]

[1,2]*Department of Computer and Information Science, Faculty of Science, Annamalai university, Chidambaram, Tamil Nadu, India.*

Received: 10 July 2023 Revised: 20 August 2023 Accepted: 10 September 2023 Published: 30 September 2023

Abstract - *Object Tracking after Prediction for the Visually Impaired was intricate, especially when identifying the right object with the right identity. The Image Quality Enhancement from captured images acquired from external devices significantly impacts the prediction of objects and tracking of their movement in Visual impairment devices. The foremost objective of the study was to design a framework for enhanced predictions in object tracking after prediction for visually impaired people. Initially, the captured image was applied with image quality enhancement measures in terms of filtering, segmentation and feature extraction to ascertain a quality image that could be used for object detection and tracking with enhanced prediction efficiency like accuracy, sensitivity, specificity, etc. The experimental analysis showed that the captured frame had been enhanced to a quality level that enhanced the prediction evaluation measures based on the confusion matrix assessment techniques. The research showed an enhanced accuracy level of 98.3% in comparison to benchmark models like Detectron2 (97.3%), YOLOv5 (84%), Faster RCNN (46%), HFYOLO (95.5%) to prove that when the image quality was augmented, Object Tracking after Prediction could be augmented being directly proportional to each other.*

Keywords - *Object tracking after prediction, Visually impaired, Evaluation measures, Image quality enhancement, Confusion matrix.*

1. Introduction

According to the World Health Organisation (WHO), Visual Impairment has become one of the growing visual problems leading to financial losses [1] estimated at around US$ 411 billion in all the world's major countries. Visual Impairment (VI) occurs in humans due to errors in the refractive index [2] and cataract problems of human beings. VI occurs in people over 50 years of age, and around 2.2 billion people have been affected with VI problems with near or distance impairment [3], as reported by WHO in 2022. It was also found that the VI problem would have been solved in almost all of these cases if the defect had been identified at the earliest stage possible. The chief aim of this research is to create an Object Detection and Tracking Evaluation Framework (ODTEF) to fine-tune the performance of the YOLO model on the COCO dataset. This improvement will involve augmenting the evaluation process by incorporating enhancements in image quality and prediction methods. Specifically, we will focus on enhancing image quality through various techniques such as filtering, segmentation, clustering, region of interest, and feature extraction. This research project has a specific focus on assessing the enhancement in both the quality and prediction accuracy of object detection and tracking for visually impaired individuals using the proposed model. The core problem we address is the significance of combining computer vision and image processing methods to demonstrate that we can enhance the overall results of object detection and tracking by improving image quality through segmentation, filtering, and feature extraction.

Computer Vision is a domain where computers can generate meaningful information from the image or video captured from any internal or external sources like cameras, smart devices or any image or video capturing device. Image Processing within the realm of computer science involves studying image patterns and enhancing them through various pre-processing and extraction techniques. Especially in captured images of Visually Impaired systems, the quality of images in terms of clarity, sharpness and visual uniformity has to be managed with no distortion, noise and irregular visuals. Consequently, an enhanced model should address both image quality for Object detection and tracking and facilitate the evaluation of the experiment through different assessment measures. The Image quality enhancement employed Image Processing techniques, while Object detection and tracking relied on Computer Vision (CV) techniques, and the evaluation criteria were grounded in Machine Learning models.

2. Related Works

The research aimed to analyse existing methods and frameworks to improve outcomes. One such technological advancement facilitating the extraction of valuable information from photos, videos, or a combination of both is computer vision [4]. Machine learning and artificial intelligence models have the potential to be harnessed for transforming this data into valuable insights. Within the field of computer vision, there exist three fundamental objectives, namely object tracking, object recognition, and object detection [5]. Object detection, a prominent computer vision challenge, entails the identification of visual entities categorised into predefined visual classes, such as animals, furniture, structures, and Individuals. These objects are identified using images, frames, or video clips in both static and dynamic scenarios [6]. Various types of object detection, such as face detection, crowd detection [7], pedestrian detection, text detection, pose detection, and recognition of significant objects like plants or license plates [8], are essential for individuals with visual impairments. A specific technique within object detection called individual identification [9] is employed to recognise people from input images and video frames. While deep learning methods are suitable for video detection, image processing techniques are recommended for static image detection. Deep learning algorithms have been applied to develop models tested for both frontal and oblique data perspectives. The literature encompasses traditional and contemporary approaches to object detection, encompassing both deep learning methodologies and image processing [10].

Object detection, using image processing as its foundation, depends on archived records due to the unsupervised nature of image data analysis. Python, along with OpenCV tools, is employed for this purpose, and one advantage of this method is that it does not necessitate object annotations. However, this approach has primarily been successful in scenarios with complex backgrounds, shadows, and multiple colors. In contrast, contemporary literature emphasises the use of deep learning techniques in computer vision models. Deep learning can handle supervised and unsupervised models simultaneously and is well-suited for complex images, including those with varying lighting conditions and occlusions. However, it requires sophisticated datasets like MS-COCO, KITTI, PASCAL VOC, and others. In their studies, several researchers have elucidated their understanding of tracking and detection of objects. [11] reviewed 50 articles on object recognition research, encompassing various models such as Fast RCNN, gradient-based algorithms and machine learning. They formulated analysis stemming from the datasets, methodologies, and evaluation metrics deployed to assess performance. The gaps identified in the literature suggest that hybrid models could improve object prediction and image enhancement. Building on this review, [12] devised a framework integrating YOLO with existing one-stage object detection models and deep learning algorithms. The model was evaluated using network images of rural roads and proved effective in accurately detecting objects, even under challenging lighting, visibility, and occlusion conditions. To classify and diagnose rickets in pediatric wrist radiographs, [13] conducted tests with a limited dataset. They achieved an accuracy rate of 86%, with 7% incorrect predictions and 7% yielding no results. [14] compared one and two-stage rooted from computer vision crop damage detection representations to assess the effectiveness of crop damage detection algorithms.

They crafted a model to evaluate and contrast a couple of algorithms (YOLO and YOLO3) for cautioning individuals near an object. Following ongoing training, the detect model demonstrated a persuasive 97% efficacy plus exceptional accuracy. They converted the facts into discourse using Google Text-to-Speech's audio feedback technique. [16] Assembled a dataset of Pakistan's vehicle registration plates and employed the YOLO exemplar within the Automatic Number Plate Recognition (ANPR) architecture to address object detection issues, specifically with regard to license plates. The results were highly efficient, with YOLOv3 achieving 94.3% and YOLOv4 reaching 99.5% accuracy.

The model exhibited a 73% accuracy rate and outperformed other models regarding inference speed. [17] introduced a simulation model, CARLA (CAR Learning to Act), which employed object detection concepts to identify objects and promote autonomous driving. The real-time implementation of computer vision algorithms allowed for seamless autonomous driving simulation. Likewise, [18] formulated an end-to-end framework for Enhanced multi-task environment detection that facilitated object detection during autonomous driving. The results indicated that the specimen outperformed competing paradigms by a factor of two to six. Post-simulation, it was evident that 3D-based object detection was also feasible. These prevalent standards demonstrated that object detection and tracking were typically executed as independent tasks, with the influence of cost-effective deep-learning strategies and algorithms. These research gaps highlight areas that must be addressed in the suggested model.

To assess the proficiency of the results in a controlled experimental setting, performance metrics and assessment measures are executed within a prognostic system. Digital image processing is a technique utilised to enhance the quality of photographs [8] sourced from known or unknown origins through a range of techniques, including filtering, normalisation, outlier removal, segmentation, and noise reduction. Computer vision is a subset of DIP that uses real-time object identification, tracking, and recognition to make sense of the improved images that are being captured [9]. Many literary works were launched in these models based on evaluation metrics for Object detection in the COCO dataset, as indicated in Table 1.

Table 1. Performance metrics for evaluation of object detection in COCO dataset

S.No	Major Metric	Performance Metric	Evaluation Threshold
1	Average Precision(AP)	Average Precision	IoU = 0.5-0.05-0.95 cI Primary Challenge Metric
2	AP	Average Precision @ IoU =0.5	IoU =0.5 PASCAL VOC Metric
3	AP	Average Precision @ IoU =0.75	IoU =0.75 Strict Metric
4	AP Across Scales	AP Small	Small Objects @ area < 322
5	AP Across Scales	AP Medium	Small Objects @ 322 < area < 962
6	AP Across Scales	AP Large	Small Objects @ area > 962
7	Average Recall(AR)	Max=1	1 Detection per Image
8	AR	Max=10	10 Detection per Image
9	AR	Max=100	100 Detection per Image
10	AR Across Scales	AP Small	Small Objects @ Area < 322
11	AR Across Scales	AP Medium	Small Objects @ 322 < Area < 962
12	AR Across Scales	AP Large	Small Objects @ Area > 962

Table 1 shows four prediction levels, which are further grouped into two evaluation metrics named Precision and Recall. We computed the Average Precision and AP Across Scales within the precision category.

Likewise, we used the average recall to determine recall and calculate AR across scales. Following object detection, we could derive various insights based on the available performance metrics to enhance recognition performance. The following list outlines some anticipated outcomes from the current COCO model:

- The Average Precision (AP) [11] was computed with IoU thresholds of 0.5, 0.05, or 0.95, allowing for the use of up to 10 prediction thresholds.
- AP represented an improved version of mean Average Precision (mAP), which was consistently used among the 12 metrics and exhibited minimal variation. These metrics were chosen primarily based on their relevance to Digital Image Processing (DIP) refinement methods and Computer Vision evaluation measures, aiding process comprehension. In COCO (Common Objects in Context), we selected twelve metrics at the prediction level, providing evaluation models for object detection assessment [10]. Table 1 summarises the COCO dataset's object detections and performance metrics, including thresholds.
- The actual evaluation of AP depended on the number of IoU thresholds and the object category to be detected over time. The COCO evaluation method heavily relied on these factors.
- Generally, the COCO dataset was found to contain more small objects than large ones. Specifically, 41% of COCO's objects were categorised as small, with an image area of less than 322; 24% were very large, with an image area exceeding 962; and 34% were medium-sized, with an image area between 322 and 962.
- Recall [12] indicated that the assessment parameter was grounded by achieving the optimal possible number of recalls with IoU criteria categorised by object type. The recall performance was computed using the top 100 objects spotting among the tested models. This indicator played a crucial role during the detection phase, facilitating the identification of bounding boxes and segmentation masks for recognised objects. In accordance with the aforementioned performance indicators for object detection, it was revealed that this model's evaluation had some research gaps, as outlined below:
    - Out of the six different metrics used to evaluate confusion matrix performance, only Precision and Recall were considered.
    - The performance evaluation range was predetermined prior to object identification and often encountered difficulty accommodating objects in motion. Additionally, this scenario did not assess important proficiency indicators like Accuracy, Sensitivity, and Specificity.
    - Objects were not evaluated in various contexts, such as indoor and outdoor environments, which could have improved the results.
    - Image clarity for object detection was not determined by deploying appropriate image processing techniques aimed at upgrading object recognition.

We developed the proposed model and examined the assessment measures based on these identified gaps in previous research.

3. Materials and Methods

The construction of an Image Quality Enhanced Object Detection and Tracking Evaluation (IQEODTE) model for visually impaired people is proposed to counteract the research gaps of the existing models designated in the literature study. This model combines Image Quality Enhancement, Object Detection, Object Tracking, and Evaluation Measures on a movable object, and it is the methodology applied in this study. The existing framework MonoGRNet, as suggested by [19], proposes the recognition of 3D objects through the dissection of 3D items in photographs using geometric projections. To validate this model, it underwent testing and practical implementation using the COCO dataset [20] on cityscapes and KITTI. Employing supervised learning techniques, the pre-processing and post-processing phases were employed to eliminate extraneous elements within the images, focusing solely on the model's critical components. However, due to the necessity for enhanced efficiency and predictability, this approach was found to be both challenging and time-consuming. The Image Quality Enhanced Object Detection and Tracking Evaluation (IQEODTE) model was introduced to address these challenges. It functions as a hybridisation of Image Quality Enhancement from image processing, object detection, and object tracking based on Computer Vision, along with Machine Learning techniques for Evaluation Measures. This multifaceted approach seeks to offer a budget-friendly solution to the issue of detecting objects, with a specific focus on meeting the needs of visually impaired individuals. The methodology is structured into four distinct components, as depicted in Figure 1, encompassing image quality enhancement, object detection, object tracking, and prediction evaluation console.

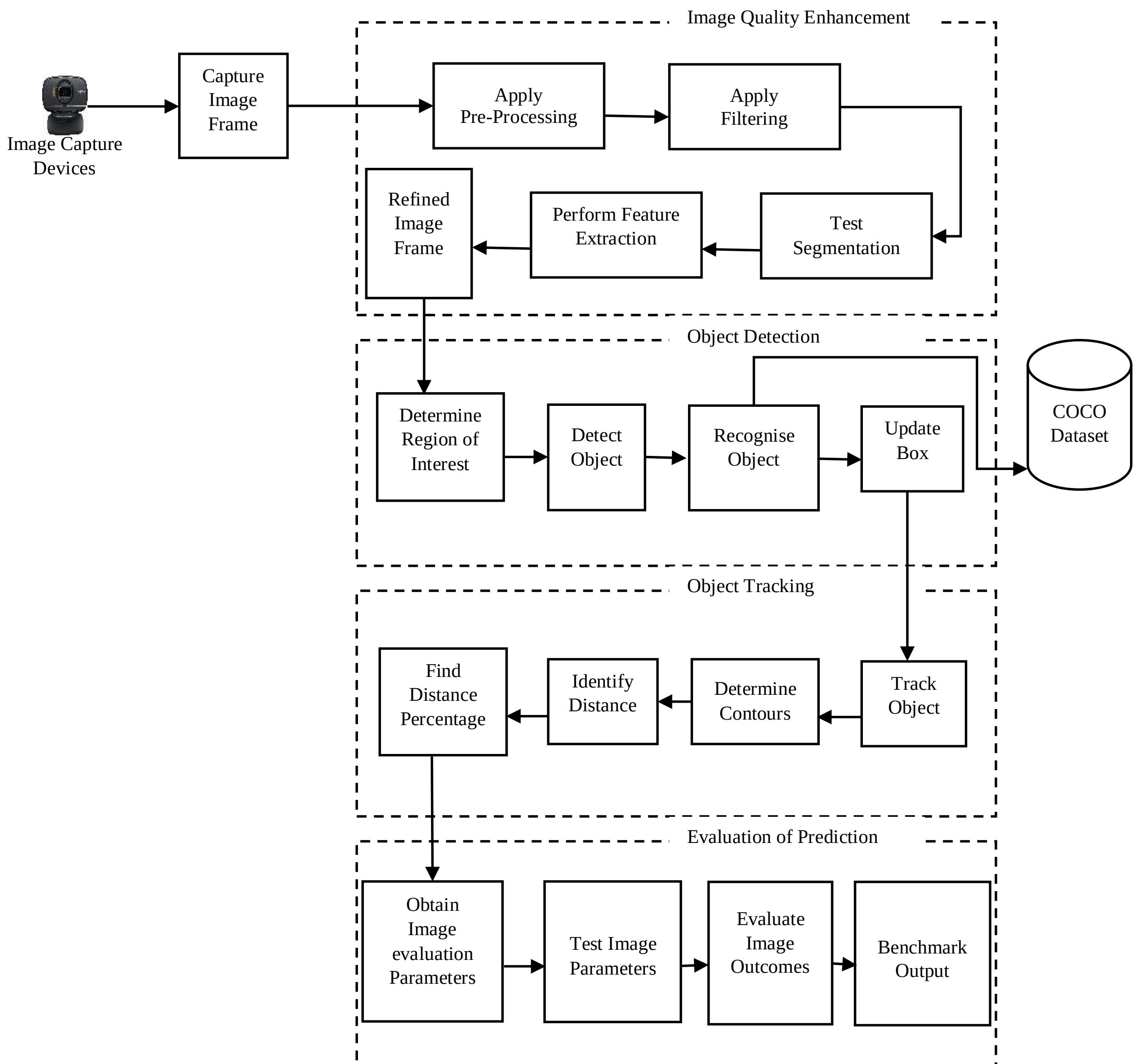


Fig. 1 Proposed Image Quality Enhanced Object Detection and Tracking Evaluation (IQEODTE) model

The original input data was taken from an image dataset that could have come from a web camera or CCTV surveillance recordings. The image that was captured is loaded for additional pre-processing. Pre-processing is carried out using filter methods to distinguish design patterns and create frames for diverse objects within the collected images. Subsequently, an object detection technique is employed to cross-reference the identified frames with each feature in the MS-COCO dataset [24], encompassing determination for a total of 80 different commodities. An object is deemed recognised if it matches any entry in the MS-COCO dataset. Then, an image-tracking technique is applied to establish box-based framing outlines for the detected objects. The group of identified objects from the dataset is scrutinised for proximity using their (x, y) coordinates. These coordinates enable the identification of nearby objects. By comparing the discovered XY coordinates to predefined threshold values, objects are categorised as either close by or distant based on the magnitude of the difference. Evaluation methods are employed to assess performance and compete with the top-performing object identification and object tracking models.

Table 2. Objects detected in the COCO dataset based on algorithms

Toothbrush	Hair drier	Teddy bear
Book	Refrigerator	Sink
Cell phone	Keyboard	remote
Remote	Bed	Potted plant
Dining table	Donut	carrot
Pizza	broccoli	banana
Hot dog	Fork	Cup
Sports ball	Surfboard	skateboard
Handbag	Snowboard	skis
Bicycle	Person	Traffic light
Motorbike	Car	Fire hydrant
Aeroplane	Train	Stop sign
Bus	Truck	Parking meter
Boat	Bench	Scissors
Vase	Clock	toaster
Oven	Microwave	Lapto
Tv monitor	Toilet	Sofa
Chair	Cake	Orange
Sandwich	Apple	Bowl
Spoon	Knife	Wine glass
Bottle	Tennis racket	Baseball glove
Baseball bat	Kite	frisbee
Suitcase	Tie	Bird
Dog	Horse	Cat
Sheep	Cow	elephant
Bear	Zebra	zebra
Zebra back pack	umbrella	

The MS-COCO dataset, known as common objects in context, not only contains a comprehensive list of 80 object categories but also boasts an additional 91 categories of objects and over 1.5 million object instances. It includes various features such as segmentation, recognition, and super pixel-form segmentation with high-quality images exceeding 200k. The dataset provides extensive resources for both normal and masked modes to separate an object's multiple components. Furthermore, it incorporates diverse annotations, including image captions, panoptic data in JSON format, dense pose information, key point detection, object detection, stuff segmentation, and dense pose data. According to Table 2, the computer vision model has been trained on these 80 object categories. The COCO dataset encompasses a vast array of resources, including 80 object categories, an additional 91 object categories, and an impressive 1.5 million instances of objects. This dataset offers diverse features, such as segmentation, recognition, and super pixel-based segmentation, with image quality exceeding 200k. Notably, the COCO dataset provides versatile tools employed in both normal and masked modes to separate an object's various components effectively. Furthermore, it includes a rich set of annotations, encompassing image captions, panoptic data in JSON format, dense pose information, key point detection, object detection, stuff segmentation, and dense pose data. Table 2 highlights the training of the computer vision model on these 80 object categories.

The You Only Look Once (YOLO) algorithm stands out as a top performer among various computer vision algorithms, surpassing others like Region Convolutional Neural Networks (RCNN) and Fast RCNN. YOLO operates as a two-stage detector, initially identifying regions of interest and subsequently classifying significant and less significant image portions based on these regions. In contrast, RCNN and Fast RCNN function as one-stage detectors, detecting objects in the entire image without needing region selection. Combining elements from both one and two-stage object detection models could potentially enhance identification capabilities. Furthermore, YOLO has been enhanced with a unique YOLOR model, which leverages both direct and indirect knowledge to improve multi-tasking object detection, increase prediction accuracy, and achieve spatial alignment within the kernel. It is worth noting that these algorithms serve distinct purposes. For the development of an on-the-fly detection system based on the PASCAL VOC dataset, [22] conducted a parallel study of various computer models. Deep neural networks based on Convolutional Neural Networks outperformed traditional real-time systems. They introduced the Hybrid Face-lifted YOLO (HFYOLO) algorithm to boost the model, combining elements from You Only Look Once and Fast RCNN.

Table 3. Image Quality Enhanced Object Detection and Tracking Evaluation (IQEODTE) algorithm

Algorithm IQEODTE
Step-1: Initialise the dataset COCO with Objects, Names
Step-2: Initialize the data for Input as COCO[Names], Output as COCO [Names]-1, height, weight, region_of_Interest
Step-3: Initialize N as the number of features in COCO and **α** as Threshold
Step-4: Initialize boxid, score, IoU, confidence_level, tracker_points;
Step-5: Load the YOLO Model and RCNN Model with the COCO dataset
Step-6: Select Threshold as Contour_Area(COCO)
Step-7: Capture the Image frame from a device connected to the system
Step-8: Calculate the height and width of the Image Object Frame
Step-9: Perform Spatial Filtering of the image frame
Step-10: Perform Threshold segmentation and Region of Interest
Step-11: Perform Feature Extraction of the captured image.
Step-12: Compute fitness with region_of_interest and area as detected object in region_of_interest.
Step-13: Loop through the area that belongs to ContourArea and Test STEP-14 thru STEP-15
Step-14: Test whether area greater than threshold. If True STEP-15, else return.
Step-15: Draw the Contour region with region_of_interest, height and width
Step-16: Compute tracker_points as EucleadeanDistance of each pixel frame from 1 to N.
Step-17: Compute boxid as detected part of the image with height, width, x and y
Step-18: Loop through the Outer Layer to N from STEP-16 thru
Step-19: Test if tracker_points is less than or equal to **α.** If True, Perform STEP-17 through STEP-19
Step-20: Update Score as detected part of Tracker
Step-21: Update class id as the maximum of the score value
Step-22: Test if Confidence level is greater than0.5 perform STEP-23 thru STEP-28
Step-23: Compute xcenter as detected initial tracker * width
Step-24: Compute ycenter as detected second tracker * height
Step-25: Compute width as detected third tracker * width
Step-26: Compute height as detected fourth tracker * height
Step-27: Calculate x as the difference of xcenter and width divided by 2
Step-28: Calculate y as the difference of ycenter and height divided by 2.
Step-29: Test the Object detection and Tracking models with predictions
Step-30: Find Confusion Matrix values of predictions
Step-31: Compute accuracy as the sum of True Positives and False Negatives divided by Total samples.
Step-32: Compute sensitivity to test correct object detections and specificity to test wrong object detections in the test.
Step-33: Benchmark the CVModel and update box id, Confidence limits and class id
End IQEODTTE

The algorithm shown in Table 3 is divided into Four phases: Image Quality Enhancement, Object Detection, Object Tracking and Performance Evaluation.

The initial stage involves the computation of the region of interest, which defines the scope for object detection, thereby determining the template's effectiveness. The contour region, encompassing the selected visionary area, is meticulously examined to map the object to a specific threshold value (). If the mapped area surpasses this value, it indicates the entity's proximity to the source. Subsequently, the contour area, including the height and width of the region of interest, is utilised to draw object detection boundaries.

The object tracking phase ensues after the object detection phase, assigning a unique tracker-id to a specific object. This assignment is based on the Euclidean distance calculated between objects mapped among "N" objects. The (x, y) coordinates, in conjunction with the object's height and width, are employed to establish the box's values.

The outer layer iterates "N" times, testing the tracker-id against a threshold value to initialise the object's score, class id, and confidence levels, which are interconnected. If the confidence level of the detected object exceeds the average cutoff value of 0.5, the image's xcenter, ycenter, width, and height are updated accordingly.

Additionally, the x and y coordinates of the object within the moving object are utilised to update the rectangle's coordinates. This tracking process continues without interruption until the moving object stops. With each iteration of the looping procedure, the identified object is presented in box plots alongside its corresponding highest score.

4. Implementation

In testing the COCO dataset, user-created photos and videos and the OpenCV function from Python were all used simultaneously in constructing the HFYOLO method. The person utilising the camera gadget in the space captures the image. The film was recorded when city traffic was busiest, as seen in Figures 3(a) and 3(b), cars and other vehicles carrying people were common. The OpenCV model's Python interface was loaded using the chosen inputs. The source code was generated and put into use based on the Hybrid Face-lifted YOLO algorithm (HFYOLO) employed in this computer vision model. The Image Quality Enhanced Object Detection and Tracking Evaluation (IQEODTE) Algorithm was also programmed to test with the HFYOLO model. The area of interest was initially located. After that, the camera's objects were found and situated within the input variables. With YOLO and Fast RCNN, the Python modules cv2 and NumPy were imported as libraries. The experiment involved three different file kinds, including the following:

4.1. The Cfg File

This was a representation of the experiment's configuration file, which contained all of the model's initial settings.

4.2. The Weight File

This contained a trained version of the algorithm used to help identify objects in the dataset.

4.3. The Name File

This lists the 80 nomenclatures of potential experiment-detectable objects.

Python's cv2.dnn.readNet() function was used to conduct the initial loading of the input variables, as seen below:

$$Input = cv2.dnn.readNet("yolov3.weights", "yolov3.cfg")$$

When read-only access was enabled for coco.names, through the readlines() function, the border lines of the images under investigation were loaded with the class values. The names of the layers, including the inner and outer layers, are determined by randomising numbers between 0 and 255.

The process began by establishing the primary Region of Interest (ROI) through cv2. Subsequently, a subtractor was employed to eliminate irrelevant portions of the visual data, addressing the issue of unwieldy image sizes and consequently enhancing performance.

To load the image in blob format, the imread() function was utilised. YOLO demonstrated the ability to categorise images into three distinct types based on their dimensions. Images with high resolutions, such as (609x609), exhibited subpar accuracy, while those with resolutions of (320x320) or smaller demonstrated excellent accuracy. Consequently, object detection yielded more successful results with smaller image sizes. As indicated by the code, the image used for the experiment possessed a reasonable resolution of (416x416).

$$blob = cv2.dnn.blobFromImage(limg, 0.00392, (416, 416), (0, 0, 0), True)$$

The threshold value, which ranged from 0 to 1, was used to determine whether an image was selected and cropped. The output layers received the detected image. The outcomes of objects discovered throughout the object detection procedure were shown on the screen. The discovered objects were mapped to the confidence threshold. The object's accuracy was determined to be close to the threshold value. The enhancement of object tracking, incorporating class_id, confidence variables, and box plots, paralleled the progress made in object detection. A threshold was applied to map the confidence values, facilitating the determination of (x, y) centers. Utilising the image's width and height, the coordinates for center_x and center_y were established. These center_x and center_y coordinates were used to update the rectangle's coordinates. Throughout the picture analysis, box_ids, confidence levels, and classes were continuously updated. To execute object tracking and recognition for this image, the HFYOLO model was employed through the Python interface. The process commenced by selecting a movie within the Python platform and initiating Object Detection and Tracking. Initially, background subtraction was executed, followed by the segmentation of individual frames. The input image was then extracted from these frames, and an examination of the image was performed to identify any noise using a mask, as depicted in Figure 2.

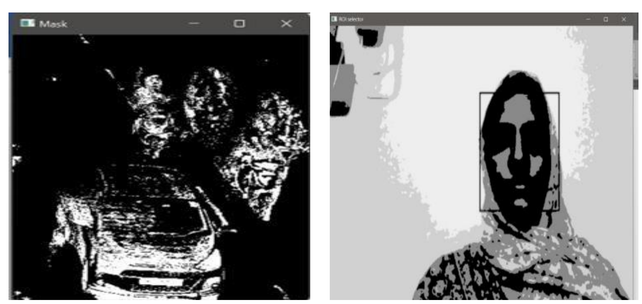
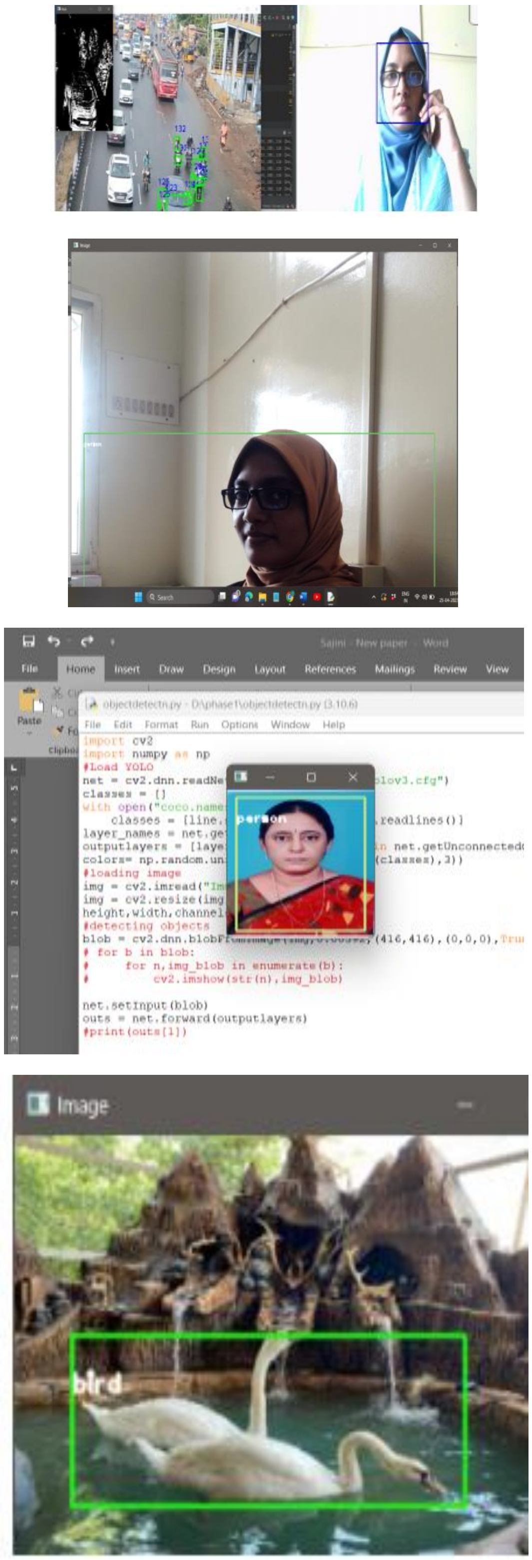


Fig. 2 Mask to determine the noise of the selected frame

Following the utilisation of image processing or deep learning models to eliminate noise from the image frame, the object detection process commenced. Drawcontours() were employed to identify the most suitable parts of the target object. Detection of frames was executed by comparing the chosen region with the image's cutoff value, determined by the object mask. This continuous mapping of the threshold value played a crucial role in simultaneously performing object detection and object tracking, effectively eliminating false positives or Type-II errors in the process. The introduction of the Euclidean distance measurement facilitated occasional object tracking. Due to the model's hybrid nature, object tracking and detection were conducted simultaneously. As depicted in Figure 3, the coordinates spanned from 0 to the final frame at 415, covering the entire region of interest.



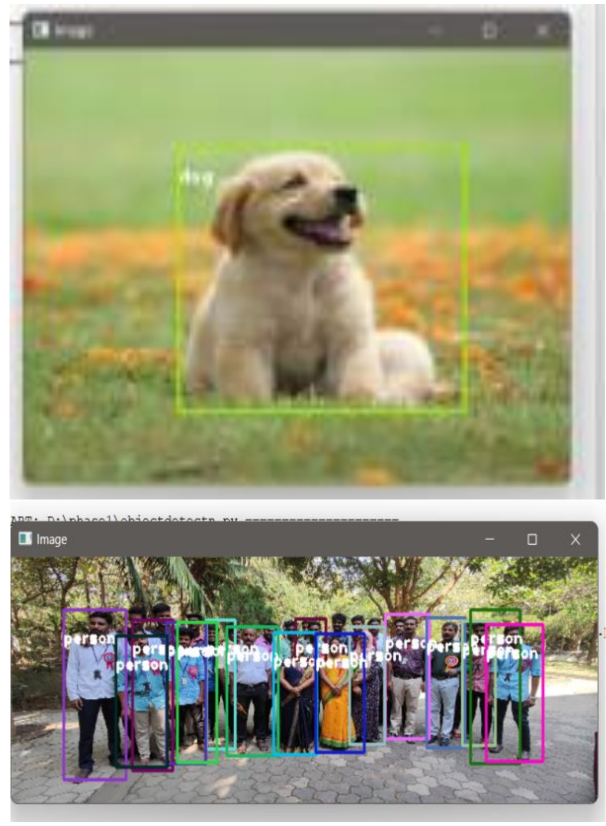


Fig. 3 Object tracking and detection simultaneously in hybrid form using HFYOLO and IQEODTTE

The values of (x, y, height, and width) for the objects undergo continuous changes and are updated in accordance with the outcomes of entity identification and tracing. This observation highlights the significance of performing object detection simultaneously with tracking to ensure precise object prediction.

5. Results and Discussions

After the experiment, an assessment was conducted to evaluate the training time and the detected coordinates. This evaluation aimed to determine the precision of the detection and investigate any errors in the experiment. Precision, a common performance metric in benchmark models, was calculated using the ratio of True Positives (TP) to the total number of Positive images, which includes both correctly identified and incorrectly identified ones, as defined in Eq. (1) [23].

$$Precision = \frac{True\ Positive + False\ Positive}{True\ Positive} \qquad (1)$$

In this equation, the term "image" in the precision measure signifies the accuracy of the entity tracking and detection carried out during the evaluation. Error [24] is identified in accordance with the False Positives (FP) nature of the detected images. The experiment involved processing the COCO dataset, and objects were recognised in image format, as illustrated in Figure 4.

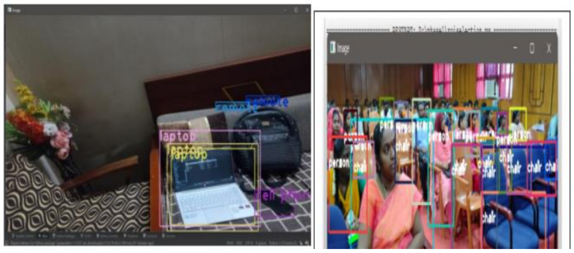


Fig. 4 Detecting and tracking objects within an image using HFYOLO

After consistent testing, the model accurately depicted the specified objects within the zone of interest, including the laptop, remote, handbag, and mobile phone. When the precision of the appropriately classified and detected objects was examined, it was discovered that 97% of them were accurately identified, whereas 16% of them were not. Similarly, after looking at the COCO dataset unveiled the arrangement of objects in diverse forms[25], the video has been evaluated and the precision factor examined, as shown in Figure 5.

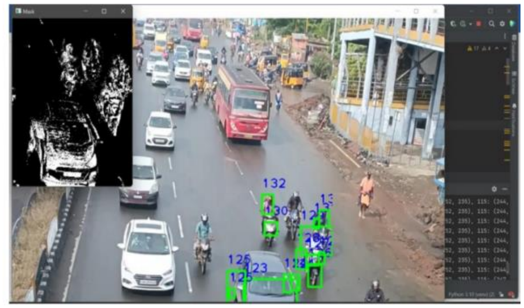


Fig. 5 A real-time video object tracking and detection using HFYOLO

As depicted in Figure 5, the video underwent an assessment, and the various types of vehicles, including two, four, and six-wheelers, were allocated their respective identification numbers. As a result, the previous models experienced a high rate of True Positives, resulting in an average precision of 63.2, along with a precision efficiency of 97.36% and a detection error of 0.33, respectively. Based on the results, the effectiveness benchmarks like accuracy, sensitivity, specificity, precision, recall and error rate are compared with the benchmark models and shown in Table 4 for comparison.

Table 4. The Comparative assessment of extant one and two-stage detector models against a hybrid model

	Detectron2	YOLOv5	Faster RCNN	HFYOLO	IQEODTE	Augment
Accuracy	96.3%	84%	46%	95.5%	97%	YES
Sensitivity	0.91	0.88	0.45	0.97	0.97	YES
Specificity	0.77	0.89	0.78	0.71	0.03	YES
Precision	97%	82%	45%	94.5%	97.36%	YES
Recall	0.76	0.78	0.88	0.91	0.93	YES
Error Rate	0.77	0.87	0.87	0.57	0.33	YES

In light of these findings provided in Table 4., the positive measures like Accuracy, Sensitivity, Precision, Recall and Error rate have improved in values and thus augmented. However, the negative predictive values expected to show a reduction have certainly shown a reduction in specificity and Error rate, respectively. In conclusion, the innovative IQEODTE model, combining image enhancement, object detection, and tracking [26], has proven successful. It has demonstrated improved predictive evaluation metrics compared to earlier models and achieved notable error downsizing throughout the entity identification process. The IQEODTE framework and algorithm generated in this study indicate the feasibility of simultaneously tracking and detecting objects, thereby enhancing prediction and object identification. Employing an HFYOLO method, which maps and plots coordinates in accordance with their vicinity to a visually impaired individual's location, enabled effective object recognition and tracking. The research's goals were met because the findings demonstrated good precision with rapid estimation. The comprehensive study asserted that this model was realistically viable in a real-time system, where visually impaired individuals might naturally employ the device, enhancing their vision through digital transformations in computer vision simulation devices.

6. Conclusion

The research work propelled the fact that prediction enhancement is possible for the Computer Vision Object Detection and Tracking model if the quality of the image is enhanced with three different stages, including Filtering, Segmentation and Feature Extraction process. The COCO dataset used in the experiment has broadened the impact of getting better results using the existing YOLO model.

The predictive evaluation measures also suggested that the measures from the confusion matrix, like accuracy, sensitivity, specificity, etc., have resulted in enhanced results even from unrealistic images captured from the devices. This research would ponder a contribution to the research world in a way that image processing and computer vision methods could be combined with machine learning evaluation models to provide a complete solution to the system's prediction and the augmented outcomes for assisting visually impaired people.

References

[1] John-Ross Rizzo et al., "The Global Crisis of Visual Impairment: An Emerging Global Health Priority Requiring Urgent Action," *Disability and Rehabilitation: Assistive Technology*, vol. 18, no. 3, pp. 240-245, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[2] Seid Worku et al., "The Magnitude of Refractive Error and its Associated Factors Among Patients Visiting Ophthalmology Clinics in Southern Ethiopia, 2022," *Clinical Ophthalmology*, pp. 1801-1811, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[3] Fabiana Sofia Ricci et al., "Virtual Reality as a Means to Explore Assistive Technologies for the Visually Impaired," *PLOS Digital Health*, vol. 2, no. 6, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[4] Varun Kumar Reja, Koshy Varghese, and Quang Phuc Ha, "Computer Vision-Based Construction Progress Monitoring," *Automation in Construction*, vol. 138, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[5] Fahad Ashiq et al., "CNN-Based Object Recognition and Tracking System to Assist Visually Impaired People," *IEEE Access*, vol. 10, pp. 14819-14834, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[6] Wei Chen et al., "2D and 3D Object Detection Algorithms from Images: A Survey," *Array*, vol. 19, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[7] Imran Ahmed et al., "Towards Collaborative Robotics in Top View Surveillance: A Framework for Multiple Object Tracking by Detection Using Deep Learning," *IEEE/CAA Journal of Automatica Sinica*, vol. 8, no. 7, pp. 1253-1270, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[8] Jingwei Liu et al., "Feature Rescaling and Fusion for Tiny Object Detection," *IEEE Access*, vol. 9, pp. 62946-62955, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[9] Zhengxia Zou et al., "Object Detection in 20 Years: A Survey," *Proceedings of the IEEE,* vol. 111, no. 3, pp. 257-276, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[10] K.V. Sriram, and R.H. Havaldar, "Analytical Review and Study on Object Detection Techniques in the Image," *International Journal of Modeling, Simulation, and Scientific Computing*, vol. 12, no. 5, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[11] Pingli Ma et al., "A State-of-the-Art Survey of Object Detection Techniques in Microorganism Image Analysis: From Classical Methods to Deep Learning Approaches," *Artificial Intelligence Review*, vol. 56, no. 2, pp. 1627-1698, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[12] Kunyi Li, and Lu Cao, "A Review of Object Detection Techniques," *5th International Conference on Electromechanical Control Technology and Transportation*, pp. 385-390, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[13] Xuesong Li, and Jose E. Guivant, "Efficient and Accurate Object Detection with Simultaneous Classification and Tracking under Limited Computing Power," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 6, pp. 5740-5751, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[14] Samed Şatir et al., "A Novel Approach to Radiographic Detection of Growth Development Period with Hand-Wrist Radiographs: A Preliminary Study with ImageJ Imaging Software," *Orthodontics and Craniofacial Research*, vol. 26, no. 1, pp. 100-106, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[15] Mansi Mahendru, and Sanjay Kumar Dubey, "Real Time Object Detection with Audio Feedback Using Yolo Vs. Yolo_V3," *2021 11th International Conference on Cloud Computing, Data Science and Engineering (Confluence)*, pp. 734-740, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[16] Muhammad Gufran Khan et al., "Development of ANPR Framework for Pakistani Vehicle Number Plates Using Object Detection and OCR," *Complexity*, pp. 1-14, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[17] D.R. Niranjan, B.C. VinayKarthik, and Mohana, "Deep Learning-Based Object Detection Model for Autonomous Driving Research Using Carla Simulator," *2021 2nd International Conference on Smart Electronics and Communication (ICOSEC)*, pp. 1251-1258, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[18] Shanglin Zhou et al., "An End-to-End Multi-Task Object Detection Using Embedded GPU in Autonomous Driving," *2021 11th International Conference on Cloud Computing, Data Science and Engineering (Confluence)*, pp. 122-128, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[19] Zengyi Qin, Jinglu Wang, and Yan Lu, "Monogrnet: A General Framework for Monocular 3D Object Detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 9, pp. 5170-5184, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[20] Barret Zoph et al., "Learning Data Augmentation Strategies for Object Detection," *European Conference on Computer Vision*, pp. 566-583, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[21] Silvia Rostianingsih, Alexander Setiawan, and Christopher Imantaka Halim, "COCO (Creating Common Object in Context) Dataset for Chemistry Apparatus," *Procedia Computer Science*, vol. 171, pp. 2445-2452, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[22] V.R.S. Mani, A. Saravanaselvan, and N. Arumugam, "Performance Comparison of CNN, QNN and BNN Deep Neural Networks for Real-Time Object Detection Using ZYNQ FPGA Node," *Microelectronics Journal*, vol. 119, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[23] Rafael Padilla, Sergio L. Netto, and Eduardo A.B. da Silva, "A Survey on Performance Metrics for Object-Detection Algorithms," *2020 International Conference on Systems, Signals and Image Processing (IWSSIP)*, pp. 237-242, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[24] Zicong Jiang et al., "Real-Time Object Detection Method Based on Improved Yolov4-Tiny," *arXiv*, pp. 1-11, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[25] Tsung-Yi Lin et al., "Microsoft Coco: Common Objects in Context," *European Conference on Computer Vision*, vol. 8693, pp. 740-755, 2014. [CrossRef] [Google Scholar] [Publisher Link]

[26] Ayushi Sharma et al., "Object Detection Using OpenCV and Python," *2021 3rd International Conference on Advances in Computing, Communication Control and Networking (ICAC3N)*, pp. 501-505, 2021. [CrossRef] [Google Scholar] [Publisher Link]