ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Vol 11, Issue 12, December 2024

Innovative Object Recognition, with Real-Time Text to Speech Conversion Using YOLOv8

[1] Ashwani Kumar, [2] Anita Devi, [3] Er. Pooja

[1] [2] [3] Department of Computer Science and Engineering, Chandigarh University, Punjab, India
Corresponding Author Email: [1] Ashwani152003@gmail.com, [2] thakur20052005@gmail.com, [3] pooja.e11313@cumail.in

Abstract*— Object detection is a computer vision technique that allows a system to locate and recognize an object in an image or video streams by plotting a rectangular box around it. This work describes a real-time object detection model that employs deep learning techniques and text-to-speech conversion. Yolov8 is renowned for its accuracy and speed of processing. The model broadcasts audio feedback about the detected object using gTTS. OpenCV and Python are used in the model's implementation, providing a broad assortment of techniques for computer vision uses. COCO is the dataset used to train YOLO. The algorithm recognizes the item, shows its label on the screen, and gives verbal output via using Google Text-to-Speech to convert the label to speech (gTTS) API, after which the Playsound library is used to play the audio. The integrated system's efficiency and versatility make it perfect for assistive technologies, smart environments, and interactive systems. It also enhances accessibility and interactivity in real-world applications.

Keywords*— Computer vision, Object Detection, Python, OpenCV, YOLO, Google Text-to-Speech, Playsound.

I. INTRODUCTION

Artificial intelligence and machine learning have advanced so quickly that they have drastically changed several industries, including computer vision and natural language processing. In recent years, computer vision has expanded efficiently and quickly. A critical stage in visual computing is object detection. as it allows the further recovery of information about the objects that have been detected allowing for additional analysis.

To improve comprehension of visual objects, we investigated the integration of aural perception in this research paper. Because hearing and sight are so similar, we developed an object detecting system in real time. that uses voice feedback to offer an audible description of items that are identified. There are many applications for object detection in today's world, such as for visually impaired individuals, object detection systems can aid by identifying and navigating obstacles in their path, enhancing mobility and independence. These systems are used in security and surveillance to keep an eye on and evaluate actions, hence enhancing safety and preventing crimes. Retail managers can use object detection to study consumer behavior and purchasing trends, which helps them with marketing and inventory planning. Furthermore, in order to ensure safe navigation, object detection in autonomous cars is essential for recognizing and reacting to traffic signs, pedestrians, and other vehicles.

Furthermore, by continuously observing and tracking things across video frames, real-time object tracking, which is crucial for applications like interactive systems, autonomous navigation, and video surveillance, enhances the capabilities of object recognition. These diverse applications highlight the value of object detection technology in improving a number of aspects of daily life and corporate operations. The two types of object detection techniques are single-shot and two-stage detectors. One-shot detectors, like YOLO use a fully convolutional neural network (CNN) to provide great computational efficiency and speed predictions on item existence and placement in a single pass of the picture. Two-stage detectors consist of two passes: a first pass produces candidates for possible objects, while a subsequent pass refines these candidates into final predictions.

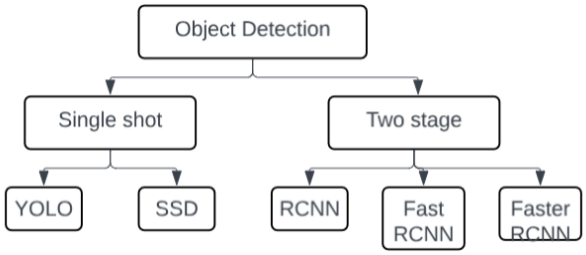

Fig 1: Single and multiple shot object detection

II. LITRATURE REVIEW

Tejaswini et al.[1] show how to combine TTS technology with YOLOv7, a high-performance object identification model. Because of its real-time detection capabilities, YOLOv7 is a good option for applications that need quick input, such helping visually impaired people navigate their surroundings. The accuracy and efficiency of the system are further enhanced by the incorporation of deep learning techniques.

The process of turning observed items into text and giving voice feedback via TTS is the main topic of **Rahima et al.** [2]The significance of preserving real-time capabilities and refining object detection algorithms is emphasized by their work. For real-world applications, where delayed input can

negatively impact user experience—particularly for those who depend on prompt responses—this solution is essential.

Jithendra et al.[3] provide a cognitive model that combines speech-to-text conversion and object detection to enable conversational interaction between the user and the system. A more dynamic and responsive interface is made possible by this cognitive approach, which also improves the user experience. This model is distinct because to its emphasis on speech-based interaction; in addition to offering optical object identification, it can also comprehend and process spoken input.

Burta et al.[4] provide a useful viewpoint by creating a mobile-based method for TTS conversion and object detection. Their emphasis on Android devices for those with visual impairments makes a substantial contribution to the field of accessible technology. They examine the difficulties in implementing mobile technology, such as hardware constraints and the requirement for real-time computing, and they suggest ways to get around these difficulties.

Redmon et al.[5] Deep learning approaches have brought about a substantial evolution in object detection. Redmon et al. (2016) proposed the YOLO architecture, which transformed real-time object recognition by treating the task as a single regression problem and predicting bounding boxes and class probabilities directly from whole photos in a single assessment. Owing to its remarkable precision and quickness, YOLO has gained popularity across multiple industries, such as robotics, healthcare, and surveillance.

The understanding of how object identification can be integrated with text and voice outputs is advanced by **Sarkar and Gupta**[6] . Their research focuses on how to accurately and efficiently provide voice feedback for things that are detected. With an emphasis on the significance of user-friendly interfaces for accessibility applications, the article provides practical advice on how to optimize voice output systems for everyday use.

Related Work

The development of deep learning techniques has had a significant impact on the object detection. Convolutional neural networks (CNNs) are used in modern object detection systems to attain great accuracy and efficiency. Among these, real-time object identification has greatly benefited from the YOLO (You Only Look Once) architecture. With versions like YOLOv3, YOLOv4, and YOLOv7 exhibiting notable gains in speed and accuracy, the YOLO family of models has had a significant impact. By predicting bounding boxes and class labels in a single forward pass over the network, these models are engineered to efficiently conduct object detection.

The integration of text-to-speech (TTS) technology has become an important research field, concentrating on improving user engagement and accessibility, alongside advances in visual detection. The benefits of combining object detection and TTS systems to provide aural feedback have been demonstrated by recent research. In order to help visually challenged users, [3] developed a model that combines speech synthesis and object detection to provide real-time explanations of items that are observed. Expanding on this strategy, [6] investigated the use of TTS to pronounce the names of items that were recognized, hence enhancing the accessibility of detection systems. By giving audio feedback instantly via Google Text-to-Speech (gTTS), [2] expanded on these ideas and demonstrated the usefulness of TTS in improving user experience.

In this paper, we build upon these previous works by implementing YOLOv8 for real-time object detection, offering improved accuracy and processing speed. Additionally, we integrate Google Text-to-Speech (gTTS) to convert object labels into voice feedback, enhancing the user experience. Our approach aims to bridge the gap between visual object recognition and auditory feedback, offering a comprehensive solution for real-time interaction in diverse applications, including assistive technologies and smart environments.

III. RESEARCH GAP

Object detection has emerged as a crucial component of computer vision in today's world of quickly evolving technology, allowing machines to detect and identify things inside an image or video. Still, a great deal of work needs to go into developing solutions that can seamlessly provide real-time object detection and audible feedback to people who are visually impaired. This gap could be filled by combining object detection with text-to-speech translation, which would allow systems to translate object detections into auditory descriptions in addition to detecting and identifying them.

Integration of Advanced Models: Previous research primarily concentrates on conventional architectures such as YOLOv4 (Alahmadi et al., 2024)[7] and YOLOv7 (Tejaswini et al., 2024)[1]. Nevertheless, there is a dearth of thorough research on the advantages of combining these architectures with more sophisticated models, like ResNet101, particularly in real-time applications for visually impaired people (VIPs). The effectiveness and precision of detection might be enhanced by this integration.

Real-Time Performance: A lot of implementations prioritize accuracy over processing speed. Burta et al.'s research from 2024 has promise for mobile applications, but it doesn't offer specific methods for attaining real-time performance without compromising speech synthesis and object recognition accuracy.

User-Centric Design: Although there have been proposals for voice representations (Sarkar & Gupta, 2024)[6] and cognitive models (Pavuluri et al., 2024)[4], these methods frequently ignore user-specific requirements and contextual characteristics. In order to improve usability and happiness, future research should concentrate on tailoring systems to accommodate a variety of user profiles,

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Vol 11, Issue 12, December 2024

especially those of VIPs.

Comprehensive Testing: Several research, such as Rahima et al. (2024)[2] and others, mostly use benchmark datasets such as COCO for assessment. To ensure reliability and robustness under real-world situations, there is an urgent need for more thorough testing across a variety of locations and scenarios, especially for apps targeted at accessibility.

Error Analysis: Error analysis receives less attention in speech recognition and object identification systems. Gaining more knowledge into the kinds of errors that commonly arise in real-time scenarios could help to strengthen the resilience and dependability of these models.

Better, more approachable solutions that enhance object identification abilities and accessibility for those with visual impairments can be created once these gaps are closed. Through comprehensive testing, creative feedback systems, and the integration of sophisticated detection algorithms with user-friendly designs, this effort seeks to close these gaps.

Methods and Meterials

The goal of this research article is to integrated system that combines google text-to-speech (gTTS) feedback with real-time object identification. The methodology uses OpenCV for image processing and YOLOv8 for object detection and gTTS for object name translation. It is all implemented in a Python environment. The specific procedures and technologies used in the system are described in detail in the ensuing sections.

YOLOv8 Architecture

The newest member of the YOLO (You Only Look Once) family of object detection models is called YOLOv8 , builds on the strengths of its predecessors while incorporating enhancements to improve accuracy and performance.

Yolo Architecture is divided into three main parts:

Backbone Network

The backbone network of YOLOv8 is a Darknet version known as CSPDarknet53, which includes a unique Cross-Stage Partial (CSP) link to improve information flow across the network's stages and gradient flow during training.

Neck and Head Structures

YOLOv8's neck structure uses a Path Aggregation Network (PANet) to efficiently capture features at multiple scales. By projecting bounding boxes, class probabilities, and objectless ratings on various scales, each detecting head of PANet helps to enable information flow across spatial resolutions.

Detection Head

The real breakthrough in YOLOv8 is found in its detection head. It uses a modified YOLO head with a dynamic anchor assignment and a new IoU (Intersection over Union) loss mechanism. These enhancements boost bounding box forecasting accuracy and enhance handling of overlapping items.

Performance Metrics

The better performance of YOLOv8 has been demonstrated on popular benchmark datasets such as COCO and VOC. It is a desirable choice for many applications, such as robotics and object detection in photos and videos, due to its precision and real-time processing skills.

While YOLOv8 performs mediocrely in some criteria (such mAP@0.5:0.95) [8]in comparison to more specialized models, its efficiency, speed, and real-time detection capabilities make it a well-liked model.

YOLOv8 delivers significant improvements over previous YOLO versions, particularly with regard to accuracy (mAP), real-time performance, and adaptation to different datasets. Its high IoU, low latency, and rapid inference speed make it the ideal choice for applications that require low-latency detection and high precision, such as autonomous driving, surveillance, and assistive technologies. Despite certain challenges with small or inconspicuous objects, YOLOv8 remains a remarkably versatile paradigm for object recognition.

Table1: Performance Metrics of YOLOv8

Metric	YOLOv8 Performance
mAP@0.5 (Mean Average Precision at IoU 0.5)	50-55%
mAP@0.5:0.95 (Mean Average Precision at IoU 0.5 to 0.95)	35-40%
Inference Speed (FPS)	30-60 FPS (varies by hardware)
Model Size	Varies (depending on model variant, e.g., YOLOv8n, YOLOv8m, YOLOv8l)
Number of Parameters	6M (YOLOv8n) to 70M+ (YOLOv8x)

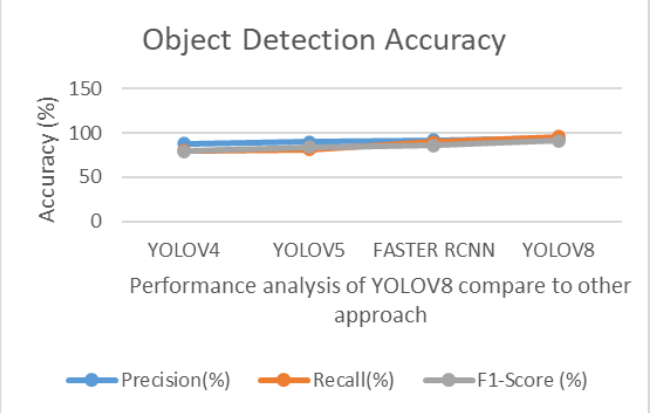

Fig 2: Performance analysis of YOLOV8 compared to other approach

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Vol 11, Issue 12, December 2024

YOLOv4, YOLOv5, Faster R-CNN, and YOLOv8 are the different object detection models that are compared in this logically organized chart. Current research indicates that the newest model, YOLOv8, should outperform older models, such as YOLOv4 and Faster R-CNN, as seen in the chart.

COCO Dataset

In computer vision, the COCO (Common Objects in Context) dataset is frequently used for object detection, segmentation, and captioning. It's one of the largest datasets for object detection model training, with over 1.5 million object instances and over 330,000 photos. There are 80 object categories in the COCO dataset, which span a variety of commonplace objects like:

- **People, for example**
- **Vehicles (such as a bicycle, bus, or car)**
- **Animals such as horses, dogs, and cats**
- **Home furnishings (such as a chair, table, and cup)**

Because YOLOv8 was pretrained on the COCO dataset, it can accurately identify and detect items in these categories. The model's ability to function properly in a variety of settings is guaranteed by the diversity of items in COCO as well as by the variety of backgrounds and real-world scenarios.

Google Text-to-Speech (gTTS):

To translate text input into spoken words, gTTS is a Python package that communicates with Google's Text-to-Speech API. The process of translating identified item names—such as "person" or "bottle"—into speech is handled by gTTS. Following object detection and identification, gTTS receives the object labels and uses them to create an audio file containing the associated voice output. In this step, the user receives spoken response from the system, making it interactive.

PlaySound:

PlaySound is a basic Python module designed to play audio files on a variety of devices. gTTS produces audio files, which PlaySound oversees playing. PlaySound outputs the audio once gTTS translates the identified object labels into speech, guaranteeing that the user gets verbal input about the objects instantly.

problem statement

Although the existing system's combination of object identification and text-to-speech (TTS) feedback allows for useful real-time interaction, its performance may be impacted by a number of issues and constraints.

Difficulty in Detecting Small or Overlapping Objects: Even with YOLOv8's advancements, it is still difficult to detect small or overlapping objects. The algorithm may have trouble accurately identifying items in complicated settings if they are close together or partially veiled, which could result in missing detections or incorrect classifications. This restriction may have an impact on the system's accuracy and dependability, particularly in situations where exact object recognition is necessary.

Audio Latency: There is some latency involved in translating object labels to voice and playing the audio, particularly when several things are recognized at once. Verbal feedback might be delayed due to text-to-speech conversion and audio playback times, which can cause issues in real-time applications where prompt reaction is essential.

Demands on Computation: Both gTTS and real-time object detection demand large amounts of processing power. When YOLOv8 is run on systems with little processing power (e.g., low-end hardware), the system's responsiveness may suffer from slower detection and inference times. TTS processing can also put additional pressure on the system, especially when handling several objects quickly one after the other.

IV. METHODOLOGY

The approach used in this study combines object detection with google text-to-speech (gTTS) technologies to get auditory support spoken feedback in real-time. To achieve higher accuracy and a broad variety of object detection capabilities, the system is constructed utilizing the COCO dataset, Google Text-to-Speech (gTTS), and the YOLOv8 object detection model. The steps that make up the methodology are as follows:

1. Data Preparation

COCO Dataset: The system makes use of a pretrained YOLOv8 model that was trained on the 80 object categories—such as people, cars, animals, and household items—found in the COCO dataset (Common Objects in Context). To support object detection tasks, the collection offers bounding boxes, class labels, and image annotations.

Class Labels: Detected objects are mapped to their corresponding categories using the class labels from the COCO dataset. Creating text descriptions and audio feedback requires these labels.

2. Object Detection Model

YOLOv8: The speed and precision of the YOLO (You Only Look Once) v8 model led to its selection. In a single pass, YOLOv8 predicts bounding boxes, class labels, and confidence scores by dividing an input image into grids. The steps involved are as follows:

Input Image Division: YOLO reduces the requirement for area proposal networks, which were employed in previous approaches, by treating object detection as a single regression issue. An S×S grid is created from the input image, and each grid cell's job is to forecast bounding boxes and class probabilities for objects whose centers are lies inside that cell.

Bounding Boxes Prediction: These are commonly expressed as the width (w), height (h), and center coordinates

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Vol 11, Issue 12, December 2024

(x, y). It is normalized these coordinates between 0 and 1 with respect to the grid cell's dimensions.

The absolute coordinates of the bounding box are calculated from the normalized values:

- xbox=(xcell+xpred)×image width*xbox=(xcell+xpred)*×image width
- ybox=(ycell+ypred)×image height*ybox=(ycell+ypred)*×image height
- widthbox=wpred×image width*widthbox=wpred*×image width
- heightbox=hpred×image height*heightbox=hpred*×image height

Class Probabilities: For every object detected, a class probability is predicted for each grid cell:

Class P(s1), Class P(s2), Class C,... The probability of every class that match the objects type.

Confidence Score: For every grid cell, a predetermined number of bounding boxes (usually two) and the confidence ratings corresponding to those boxes are anticipated. The confidence score indicates the model's degree of confidence that a box includes a particular object as well as its assessment of the correctness of the box.

This is how the confidence score is determined:

Confidence=P(Object)×IOU

The probability that an object is inside the box is denoted by P(Object).

Additionally, the intersection over union, or IOU, is a measurement of the overlap between the ground truth and the predicted bounding boxes. It is computed as follows: IOU= Area of Overlap/ Area of Union.

Output Vector:

Output = [x, y, w, h, Confidence, P(Class1), P(Class2), …, P(ClassC)]

Where:

x, y are coordinates of the center of the bounding box.

w, h are width and height of the bounding box.

Confidence score indicates the likelihood that an object is inside the bounding box.

P(Class1), P(Class2), …, P(ClassC) are the Probabilities of each class (e.g., Person, Dog, etc.).

And C is the total number of classes.

NMS (Non-Maximum Suppression): YOLO uses a technique called non-maximum suppression (NMS) at inference time to handle multiple detections of the same object.

- Initially, the bounding boxes are arranged according to their confidence ratings.
- Then the one with maximum confidence score is picked.
- With all the other boxes, calculate the IOU for the reference box.
- Suppress any boxes with IOU exceeding a specified level.
- Continue until every package has been handled.
- By doing this step, duplicates are minimized and just the most pertinent bounding boxes are retained.

It is an essential method for improving the functionality and performance of object detecting systems. NMS is crucial in increasing the efficacy and efficiency of object detection by lowering redundancy and enhancing output quality.

Final Output: The final detection results are the bounding boxes that remain after NMS have been processed or shown.

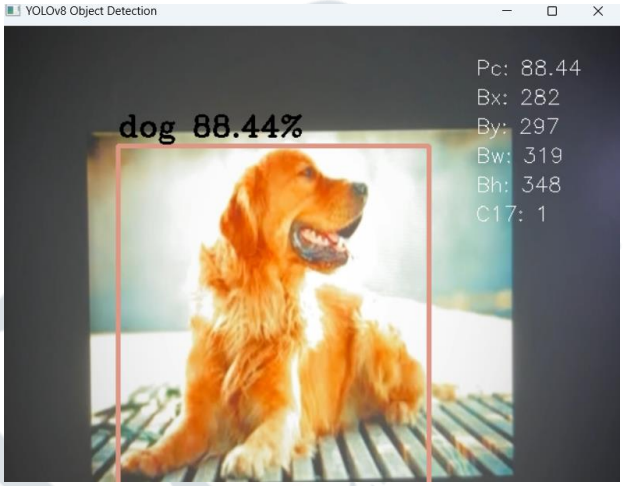

Fig 3: Processed and Labelled Images

3. Text-to-Speech (TTS) Conversion

Speech Generation: Once an object is detected and classified, the model convert the object's class label into a text format. For instance, the system will produce a statement like "**I see a dog**" if the detected object is a "dog".

gTTS (Google Text-to-Speech): The Google Text-to-voice engine receives the created text and converts it into audible voice. The user receives instant spoken feedback on the objects they have recognized by playing back a temporarily preserved audio file.

4. Optimization and Real-Time Performance

Several optimizations are put in place to guarantee that the system runs in real-time and delivers prompt feedback:

Frame Resizing: Each video frame is resized to 480x620 pixels before processing to reduce computational load without significantly sacrificing detection accuracy.

Confidence Threshold: A confidence threshold (like 0.45, for example) is specified to filter out low-confidence detections and ensure that only reliable items are detected and reported.

Dynamic TTS: When a new or changed object enters the frame, the text-to-speech system only sounds an audio response. By cutting down on pointless or repetitive speech outputs, this enhances user experience.

5. System Testing and Evaluation

To assess how well the system performs in actual settings, it is put to various tests:

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Vol 11, Issue 12, December 2024

Object Detection Accuracy: The accuracy of the YOLOv8 model is evaluated across multiple item categories in the COCO dataset. The system's rating is based on how well it can identify and label things under different lighting and environmental conditions.

Real-Time Responsiveness: The system's response time is tracked to ensure that there is never a noticeable delay between item detection and speech output. The frame rate, detection speed, and TTS generation time are tracked to guarantee a smooth, real-time user experience.

User Interaction: To test the system, users listen for audio feedback regarding the system's usability and clarity of speech output.

V. MODEL WORK FLOW

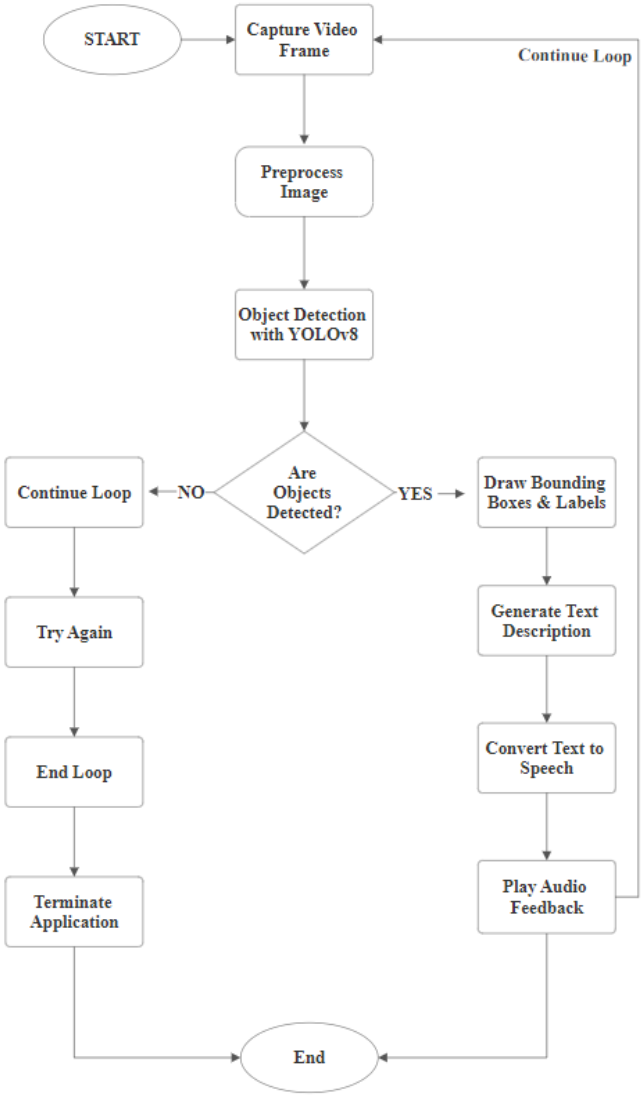

Fig 4: flow chart of working of model

The suggested system may identify objects in a video stream, provide text descriptions that match to those objects, and then speak the descriptions out to provide audio feedback. The flowchart (Fig. 3) depicting the system architecture describes the sequential phases and decision-making procedures involved in text-to-speech (TTS) conversion and real-time object detection. Here is a description of the major stages:

Start and Initialization

System startup is the first step in the process. The machine starts up and starts capturing frames of video from a live feed. This loop is appropriate for real-time applications since it enables ongoing monitoring and object detection.

Capture Video Frame

The first step in every iteration is for the system to grab a frame from the video stream. The input for the next item detection pipeline will be this frame. The system can function in real time if frames are captured at a constant rate.

Preprocess Image

Preprocessing is done on the acquired image before object detection.By doing this, you may be sure the frame is prepared correctly for the detection algorithm. Resizing the picture, levelling the pixel values, and employing filters to bring out the features of the objects are standard preprocessing approaches. Preprocessing that is effective shortens processing times and improves detection precision.

Object Detection using YOLOv8

The pre-processed image is fed into the YOLOv8 object detection algorithm. YOLOv8 is a deep learning-based model intended for real-time object recognition. When an object is detected, it identifies what is inside the frame and provides the bounding box coordinates and class labels for each one. YOLO's design allows it to be applied in scenarios such as autonomous systems or assistive technologies where low-latency object recognition is required.

Decision: Are Objects Detected?

The system checks to see if any items in the frame have been discovered after the detection phase. If **no objects** are found, the system moves on to the next frame, essentially restarting the detection loop, which is a critical decision point for managing the process flow. This ensures that the system will continue to scan until a valid detection is made.

The following step of the process involves the system visualizing and labelling any objects discovered so they may be processed further.

Draw Bounding Boxes and Labels

The system creates bounding boxes around the recognized objects in the frame after successful detection. Each bounding box is also given an item class label (such as "person," "car," or "dog"). The location and category of each identified object within the frame are made easier to understand with the aid of this visualization phase.

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Vol 11, Issue 12, December 2024

Generate Text Description

After that, the system uses the objects it has identified and their labels to create a text description. Usually, this description enumerates the items discovered within the frame, for example, "A person and a car are detected." The speech output in the following step is derived from the created text.

Convert Text to Speech

The generated text description is converted into speech using a Text-to-Speech (TTS) engine. The system can synthesis natural-sounding speech thanks to TTS technology, producing an aural output that can be used in assistive applications, especially for people who are visually impaired.

Play Audio Feedback

After that, the user hears the synthetic speech, giving them immediate audio feedback. In assistive circumstances, where users depend on audio to comprehend their environment, this step is crucial. The smooth transition from object identification to speech output guarantees that the system can successfully communicate visual information orally.

End or Continue Loop

As long as the system is operational, the process will continue. The system repeats the detection and voice creation cycle, looping back to collect the next video frame if no termination signal is received. Should the system be stopped, the program comes to an end, and it leaves the loop.

Result

The first time the model is run. First, the speech input is received, and it is subsequently translated into text. The object name that the model is supposed to detect is represented by this converted. If the user enters "watch" in this case (fig. 4), the output should be a bounded box with the name "watch" that appears on the output screen.

Input: The object name must be spoken into the microphone by the user.

```
Speed: 6.0ms preprocess, 120.4ms inference, 2.5ms postprocess per image
at shape (1, 3, 480, 640)
```

Fig.3: Output of the model after detected object is converted to text.

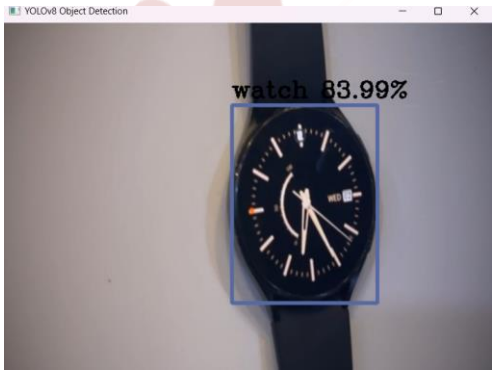

Fig.5: Output ‘Watch’ as given input object is ‘Watch’.

In Fig,5, 6. "Watch" and "Person", a similar method is used to recognize the objects in the picture using speech input. Here, Google Speech Recognition API is used to transform speech to text. The Labelling software is used to label images. The photos are trained in the cloud using free GPU resources by means of the Darknet framework. The weights file is obtained once the image has been trained using the Darknet Framework. You should run the Yolo object detection model with all the training data and the Configuration file. The model verifies each pixel's confidence score in the image. A boundary box is drawn and the required object is located when an object gets close to the training image's score.

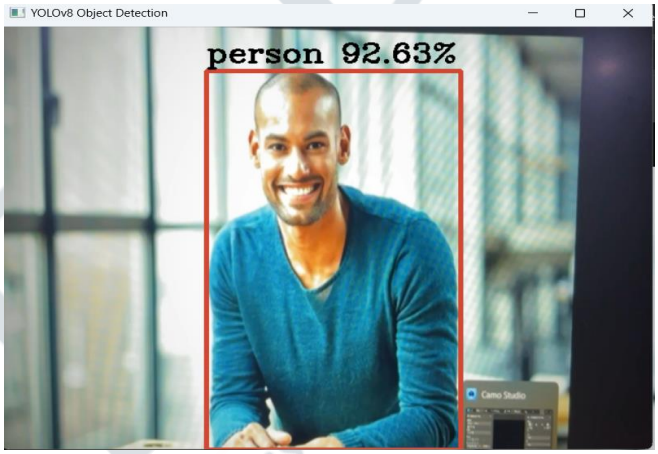

Fig.6: Output ‘Person’ as given input object is ‘Person’.

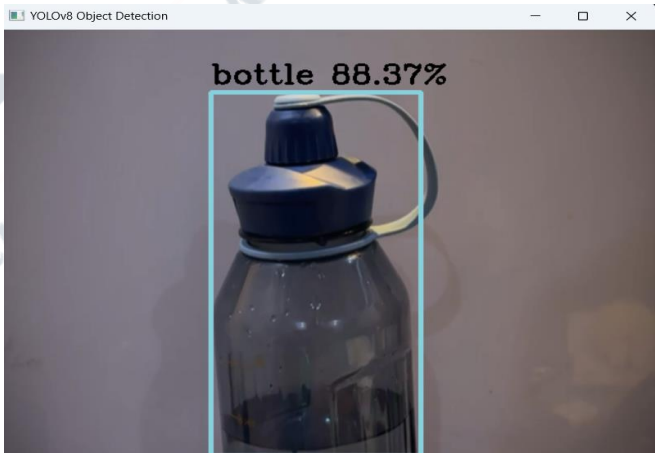

Fig.7: Output ‘bottle’ as given input object is ‘bottle’.

In the model training phase, the tagged item name, configuration file, and tagged image are used for training of this model using the Darknet Framework. The model determines the accuracy in percentage of the detected objects by comparing the given object with the pre-trained or current YOLO weights at each iteration.

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Vol 11, Issue 12, December 2024

Table 2: Average Accuracy and Response Time of objects

S. No.	Objects	Accuracy of object detected and recognized (in percentage)	Response Time
1	Person	92.63	0.19s
2	Cat	93.08	0.27s
3	Dog	91.81	0.21s
4	Cell Phone	95.12	0.19s
5	Keyword	94.54	0.22s
6	TV	96.75	0.18s
7	Refrigerator	93.44	0.24s
8	Watch	83.99	0.21s
9	Teddy Bear	89.62	0.17s
10	Bottle	88.37	0.14s
Average Response time and Accuracy		91.94	0.18s

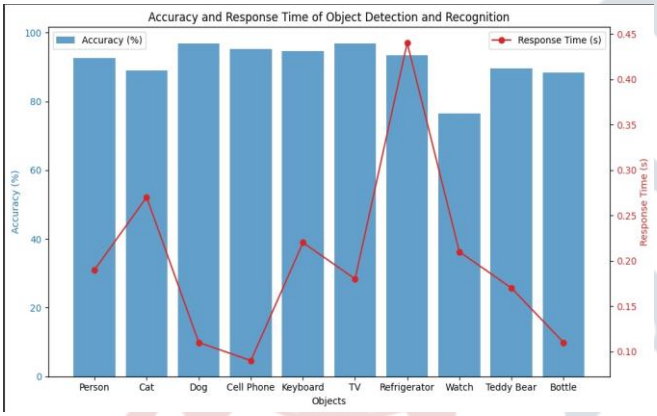

Fig 8: Accuracy and Response Graph

As observed by the Figures, we have conducted some real-time experiments on the objects in the path at the designated distances and directions. As we seen in Table 2, the entire procedure takes an average of 0.18 seconds, and the achieved accuracy of item detection and recognition is 91.94%. Accuracy (depend on the lighting and camera) and real-time reaction are the two main issues with our proposed schema that will have an impact on the smart cane's dependability and efficiency. Table 2 shows that we have met our goal and obtained the greatest

VI. CONCLUSION AND FUTURE WORK

In this research paper, we developed a computer vision model that can quickly and accurately recognize objects in real time with great precision and accuracy. We were able to effectively apply YOLOv8 for real-time recognition of objects. Yolov8 was selected for this task because of its fast processing and accuracy in real-time object detection. Using its improved design to properly sort objects into multiple categories, it identified and categorized the objects with efficiency. This development is crucial for the efficient and accurate use of object detection in technologies like smart surveillance systems and autonomous navigation systems. OpenCV's incorporation is essential for supporting efficient identification of objects. It made the image processing and manipulation easier, facilitating easy communication between the visual input and our detecting system.

Moreover, we enhanced the user experience by employing gTTS, and PlaySound to deliver audio feedbacks of labels that matched the objects that were identified. It is feasible to employ an interactive feature in a number of scenarios thanks to this functionality. Safety applications, for example object detection in outdoor areas, can be used to increase accessibility for those with vision impairments. When a security system notices objects or unidentified people near private property, for instance, it can improve situational awareness by warning users audibly. Putting these two technologies together, it demonstrates the versatility of Python frameworks for building complex, interactive apps and the strength of deep learning in computer vision.

Our system's modular architecture allows for future enhancements and integration of additional machine learning models for more complex tasks like facial or gesture recognition. Further study, can make it possible to adjust the model to operate on multiple hardware platforms, such as mobile devices or edge computing systems, or to obtain optimal performance for a particular type of site, such as indoor versus outdoor, by conducting more tests. This research demonstrates the potential connections between AI, computer vision, and audio processing, which might result in novel uses that have a big influence on many different industries and daily tasks. Collaborating with subject matter experts and obtaining user input will be essential to tailor the system to the specific needs of various applications. As technology advances, object detection systems in conjunction with AI and TTS still have a lot of potential and provide intriguing opportunities.

REFERENCES

[1] K. TEJASWINI, K. S. CHAITHANYA, and K. L. BAI, "OBJECT DETECTION AND TEXT TO SPEECH CONVERSION BASED ON YOLOV7 USING DEEP LEARNING," *INTERANTIONAL JOURNAL OF SCIENTIFIC RESEARCH IN ENGINEERING AND MANAGEMENT*, vol. 07, no. 04, Apr. 2023, doi: 10.55041/ijsrem18807.

[2] S. Rahima, Y. G. V Viswath, S. Vamsi, U. Y. V. Krishna, and M. K. Bhasker, "OBJECT DETECTION CONVERT OBJECT NAME TO TEXT AND TEXT TO SPEECH".

[3] P. Jithendra, T. V. Sai, R. K. Mannam, R. Manideep, and S. Bano, "Cognitive model for object detection based on speech-to-text conversion," in *Proceedings of the 3rd International Conference on Intelligent Sustainable Systems, ICISS 2020*, Institute of Electrical and Electronics Engineers

Inc., Dec. 2020, pp. 843–847. doi: 10.1109/ICISS49785.2020.9315985.

[4] *2020 Fourth World Conference on Smart Trends in Systems, Security and Sustainability (WorldS4)*. IEEE, 2020.

[5] M. I. Thariq Hussan, D. Saidulu, P. T. Anitha, A. Manikandan, and P. Naresh, "Object Detection and Recognition in Real Time Using Deep Learning for Visually Impaired People," *International Journal of Electrical and Electronics Research*, vol. 10, no. 2, pp. 80–86, 2022, doi: 10.37391/IJEER.100205.

[6] P. Sarkar and A. Gupta, "Object Recognition with Text and Vocal Representation", doi: 10.9790/9622-1005046377.

[7] T. J. Alahmadi, A. U. Rahman, H. K. Alkahtani, and H. Kholidy, "Enhancing Object Detection for VIPs Using YOLOv4_Resnet101 and Text-to-Speech Conversion Model," *Multimodal Technologies and Interaction*, vol. 7, no. 8, Aug. 2023, doi: 10.3390/mti7080077.

[8] S. Ronanki, O. Watts, S. King, and G. E. Henter, "Median-based generation of synthetic speech durations using a non-parametric approach," in *2016 IEEE Workshop on Spoken Language Technology, SLT 2016 - Proceedings*, Institute of Electrical and Electronics Engineers Inc., Feb. 2017, pp. 686–692. doi: 10.1109/SLT.2016.7846337.D

[9] YOLOv8 Architecture; Deep Dive into its Architecture -Yolov8

[10] J Redmon, S Divvala, R Girshick et al., "You Only Look Once: Unified," Real-Time Object Detection[J], p. 779-788, 2015.

[11] Teju, V., Bhavana, D., "An efficient object detection using OFSA forthermal imaging," International Journal of Electrical Engineering Education, 2020.