IIETA International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Journal Européen des Systèmes Automatisés
Vol. 59, No. 10, October, 2026, pp. 2877-2891
Journal homepage: http://iieta.org/journals/jesa

Artificial Intelligence-Powered Assistive System with Object Detection and Speech Recognition System for Visually Impaired People

Priyadarshini Shanmugasundaram[1*], Kala Iyapillai[2], Thamaraikannan Paneerselvam[3], Sahana Selvaraju[1], Karthick Myilvahanan Jothivel[1], Chinnarasu Kannan[4]

[1] Department of Computer Science and Engineering, PPG Institute of Technology, Saravanampatti, Coimbatore 641035, India
[2] Department of Computer Science and Engineering, PSG Institute of Technology and Applied Research, Neelambur, Coimbatore 641062, India
[3] Department of Computer Science and Engineering (Artificial Intelligence and Machine Learning), PPG Institute of Technology, Saravanampatti, Coimbatore 641035, India
[4] Department of Agricultural Engineering, PPG Institute of Technology, Saravanampatti, Coimbatore 641035, India

Corresponding Author Email: priyadarshiniscse.it@ppg.edu.in

Copyright: ©2026 The authors. This article is published by IIETA and is licensed under the CC BY 4.0 license (http://creativecommons.org/licenses/by/4.0/).

https://doi.org/10.18280/jesa.591010

Received: 25 August 2026
Revised: 24 September 2026
Accepted: 21 October 2026
Available online: 31 October 2026

Keywords:
object detection, You Only Look Once-Convolutional Neural Network hybrid, model, speech recognition, visually impaired, assistive technology, recipe recommendation, real-time systems

ABSTRACT

Smart assistive technologies have become increasingly important in recent years, particularly for supporting visually impaired individuals in performing their daily activities independently, safely, and efficiently. Food recognition and cooking guidance are still areas with limited support. Most traditional systems are not hands-free, do not offer real-time feedback, and are difficult for blind users to use. The proposed AI-Powered Food Assistance System combines You Only Look Once (YOLO)-based object sensing with Convolutional Neural Network (CNN) based speech detection to create an accessible and easy-to-use Kitchen Assistance System. The system identifies fruits and vegetables using a live camera feed and interprets user voice commands, recommends appropriate recipes based on the detected ingredients and delivers step-by-step cooking instructions through speech output. The complete pipeline operates as a closed-loop, automated system in which visual and voice inputs are continuously acquired, pre-processed and passed through detection, recognition and recommendation modules for confidence-based decision-making, enabling hands-free, real-time interaction without manual intervention. The system was evaluated across five real-world kitchen scenarios of increasing complexity such as ideal indoor lighting, low-light conditions, noisy backgrounds, cluttered scenes and multiple-object scenarios. In the ideal lighting conditions, the system achieved an object detection precision of 98.8% and a speech accuracy of 97.4%, while the overall average performance across all scenarios remained strong, with a mean object detection precision of 94.6%, recall of 91.6%, and mean Average Precision(mAP)@0.5 of 91.5%. System reliability was further validated through repeated- trial statistical testing, a model complexity and cross-platform deployment feasibility analysis across Graphics Processing Unit (GPU), embedded and mobile hardware and a small-scale usability evaluation with visually impaired participants assessing task completion time, voice command success rate and subjective usability ratings. By integrating computer vision, natural voice interaction and context -aware recipe suggestion into a single automated framework, the proposed Food Assistant System provides a significant improvement to kitchen usability, supports greater independence, and enhances overall cooking experience for visually impaired individuals.

1. INTRODUCTION

Object detection is a common application in modern technologies like autonomous navigation, smart surveillance systems, facial recognition, and assistive tools by Arsalwad et al. [1]. The existing object detection methods for visually impaired people rely on manual feature extraction, and the standard templates are often unable to meet the requirements due to the lack of flexibility as well as accuracy. To overcome these challenges, this project proposes an Artificial Intelligence (AI) based object detection and speech recognition system for Blind persons, Food Assistant. The system is intended to assist visually impaired people in the kitchen, offering a smart and interactive tool that makes cooking easier and more fun. A model for an object detector was developed by Sagana et al. [2] that can identify objects at a specific distance, which is helpful for visually impaired individuals and other important uses.

This object detection model connects with a Text-To-Speech (TTS) system, allowing users to hear the names of detected objects in their preferred language, along with relevant details in their native language by Varshini et al. [3].

The object is shown on the screen, but this may not be very helpful for individuals with visual challenges. To address this, the study uses Audio Assist, which converts text to speech. The Audio Assist feature is also used to provide an audio description of the identified object, including information about its location. This research uses a large dataset of objects like mobile phones, water bottles, cups and many others. This model is trained not only to detect objects but also to identify people. This technology has great potential to improve the independence and safety of visually impaired users by helping them to explore and interact with their environment more confidently and effectively by Islam et al. [4].

Despite growing You Only Look Once (YOLO)-based assistive prototypes, most systems are either standalone object detection modules with audio narration or separate speech tools. This manuscript positions its contribution against a two-part gap. The absence of a unified detection-to-speech recommendation pipeline and the lack of systems designed for cluttered, close-range, multi-object kitchen environments. Rather than combining independent modules, the proposed system is architected as a single automated pipeline where visual and voice inputs are continuously processed and resolved into recipe-aware spoken guidance without manual coordination. Section 1.1 details the technical specification, Hybrid YOLO-Convolutional Neural Network (CNN) architecture, confidence-based fusion, and cosine similarity recommendation, differentiating the proposed framework from prior assistive systems.

1.1 Research gap and summary of contributions

To clearly delineate the technical positioning of the proposed work relative to the prior work, Table 1 presents a feature-level comparison against three representative YOLO based assistive approaches discussed in the literature survey, examined specifically along the architectural and functional dimensions that distinguish a fully integrated, kitchen-oriented assistive pipeline from generic object announcement systems. The comparison highlights four key dimensions, such as detection backbone design, speech or language capability, presence of recommendation and target application domain, across which the proposed system introduces substantive departures from existing approaches.

As shown in Table 1, earlier systems such as the object recognition framework of Sagana et al. [2] rely on standard, general purpose detection backbones paired with unidirectional TTS output, offering no Automatic Speech Recognition (ASR) or dialogue capability and no downstream task layer beyond object announcement. Similarly, the EchoVision system proposed by Varshini et al. [3] employs a MobileNet-Solid State Detector (SSD) with a caffeModel pipeline, again limited to one-way TTS narration without voice-command interpretation or recommendation functionality. The VisionAssist system by Swetha et al. [5] extends this paradigm by incorporating monocular depth estimation for proximity categorization, representing a meaningful advance for navigation and obstacle-avoidance use cases, yet remains similarly constrained to TTS-only output with bidirectional speech interaction or any task-specific recommendation capability. These existing approaches were designed for close-range, visually cluttered, occlusion-prone conditions characteristic of kitchen environments. They did not incorporate any mechanism for personalized, context-aware task guidance beyond object identification.

The proposed Food Assistant System introduces the following contributions.

1. A domain-adaptive hybrid detection backbone: The YOLO detector is augmented with residual blocks and Intersection over Union (IoU) - refined bounding-box regression, specifically tuned to address the detection of small, occluded and visually similar kitchen items such as fruits, vegetables, utensils, and packaged foods. This architectural specialization directly targets a challenge left unaddressed by the generic assistive navigation and general object announcement systems.

2. A tri-modal closed automation loop: Rather than the loosely coupled detection-TTS pipeline characteristic of prior systems, the proposed framework fuses object detection, with CNN-based ASR, and cosine similarity-based recipe recommendation into a single, autonomous decision cycle. This tri-modal integration enables continuous bidirectional interaction encompassing visual perception, spoken intent recognition, and task-oriented response generation with one closed-loop pipeline.

Table 1. Feature-level comparison of the proposed system against the existing You Only Look Once (YOLO) based assistive systems

System	Detection Backbone	Speech or Language Module	Recommendation or Task Layer	Target Domain
Object Recognition System, Sagana et al. [2]	Standard object detector (generic)	TTS only (no ASR/dialogue)	None	General object announcement
EchoVision, Varshini et al. [3]	MobileNet-SSD and CaffeModel	TTS only	None	General object voice assistance
VisionAssist, Swetha et al. [5]	SSD-MobileNet and monocular depth	TTS only	Proximity categorization	Navigation or obstacle distance
Proposed Food Assistant System	Hybrid YOLO-CNN with residual blocks and IoU-refined bounding-box regression	CNN-based ASR and intent parsing (bidirectional dialogue)	Cosine-similarity recipe recommendation and dietary personalization	Food preparation, closed-loop automation

Note: Convolutional Neural Network (CNN), Solid State Detector (SSD), Text-To-Speech (TTS), Intersection over Union (IoU), Automatic Speech Recognition (ASR).

3. Targeted generative adversarial network (GAN)-based data augmentation: To mitigate class imbalance and lighting variation effects specific to food and kitchen item categories, the training pipeline incorporates GAN-based augmentation targeted at the kitchen items class distribution, rather than the generic augmentation strategies applied to broad, general-

purpose object sets as seen in comparable systems.

4. A personalization layer: Built directly on top of the recommendation engine, this layer supports dietary-preference filtering and portion adjustment, enabling the system to tailor its guidance to individual user needs. This capability is not present in the existing raw object detection approaches.

5. An explicit deployment-oriented evaluation: This evaluation assesses whether the trained model can execute effectively on mobile or embedded edge hardware, rather than assuming exclusive server -side or desktop Graphics Processing Unit (GPU) inference, as is implicitly the case in the majority of prior systems that report performance without addressing real-world, portable deployment constraints.

2. LITERATURE SURVEY

Earlier assistive frameworks for visual impairments relied on computer vision along with TTS to translate detected objects into spoken words. For example, Varshini et al. [3] created a system using CaffeModel, which incorporated a pre-trained model and used MobileNet SSD to transform recognized text into speech.

Recent systems have added more functionalities such as proximity estimation, Optical Character Recognition (OCR), scene narration, and multimodal interaction. Islam et al. [4] proposed a cost-effective system that utilized the object detection Application Programming Interface (API) and Single Shot Detector Lite to report the surroundings using multilingual TTS in Tamil, English, and Hindi. Swetha et al. [5] combined object detection, distance calculation, and voice feedback into their system. They employed SSD MobileNet object detection and a monocular depth estimation model to calculate depth from a single camera. The detected objects were categorized according to their distance and the information was conveyed through a TTS system. Building on this, Archana et al. [6] introduced a camera-based system that categorized objects and delivered audio feedback through earphones. Ashvitha et al. [7] related smart assistive sticks further demonstrated how machine learning can help visually impaired individuals in performing daily tasks and enhancing their independence.

Later development focused on creating more intelligent and context-aware assistive systems. Raja et al. [8] created an intelligent assistive system that employed AI to detect obstacles, pedestrians, vehicles and traffic signals in real-time by processing live video from a camera. Joshi et al. [9] created a system for real-time object detection with scene understanding and audio feedback, which achieved high accuracy in classifying the environment.

More advanced systems combined object recognition with navigation support. Niranjana et al. [10] developed a completely automatic guidance system that combined deep learning for object recognition with a distance sensor and auditory feedback.

In related research, Bharathi et al. [11] built a system that employed TensorFlow object detection to identify specific objects and translate the environmental data into spoken descriptions. Thilanka et al. [12] organized their framework into separate parts: collecting environmental information and representing it through speech, focusing on real-time object detection and delivering spoken feedback.

Bharath Kumar et al. [13] built upon this by integrating computer vision, deep learning, OCR, and sensor-based technologies on a Raspberry Pi device to enhance environmental awareness and safety. Shanmugapriya et al. [14] further included webcam input, Bootstrapping Language Image Pre-training (BLIP)-based scene description, YOLOv8-based object and traffic-sign recognition, and multilingual audio feedback supported by a voice-controlled hands-free interface. Anitha et al. [15] introduced a cool-down mechanism to reduce repetitive alerts.

Recent studies continue to emphasize portability and practical deployment. The work of Rajamani et al. [16] created a compact Raspberry Pi-based system that enables real-time object identification and converts TTS feature using YOLO and TTS technologies. Madhu et al. [17] utilized OCR and object detection on Raspberry Pi to help users access printed text and nearby objects through audio feedback. A study conducted by Nandwani et al. [18] presents an assistive system based on a camera that uses SSD technology for detecting objects, measuring distance and processing audio instructions. This system is designed as a complete navigation tool that delivers clear audio feedback to individuals who are visually impaired. The work of Reddy et al. [19] developed a system that detects objects and selected currency notes and relays the result through audio feedback to a blind user. This is a specific support system intended to assist in recognizing objects and determining the camera's approximate angle or distance. Subramanian et al. [20] used TTS to provide spoken object labels and locations to blind or visually impaired users. The abstract presents it as an accessibly packaged, on-demand assistive device rather than a large-scale evaluated study. Yadav et al. [21] present a real-time object detection system that announces detected items through audio output. It focuses on threaded audio announcements, non-maximum suppression, and a 0.5 confidence threshold to make sure feedback is still helpful in noisy or busy environments. Jambhulkar et al. [22] developed a system that uses YOLOv3 along with the Microsoft Common Objects in Context (MS COCO) dataset to provide real-time object recognition combined with audio feedback via Google Text-To-Speech (GTTS). Aralikatti et al. [23] developed a real-time assistive system based on Android that detects the nearby objects from the phone's video camera and converts them into audio to help visually impaired users to move around safely. The system integrates OpenCV, YOLO and FaceNet and has unique features such as identifying the person and announcing objects. Iqbal et al. [24] proposed a multi-purpose assistive model for visually impaired users that includes indoor navigation, object detection and recognition, distance estimation, text-to-speech output and speech recognition for user's queries. S. Maheswari et al. proposed an audio-tactile assistive device to provide vibrant feedback to enhance navigation confidence. Maheswari et al. [25] introduced a smartphone-based object detection system for the visually impaired. The system was designed for simplicity, with reliable audio feedback through headphones by Ravi Kumar et al. [26]. It combines LLM-based semantic normalization, BERT contextual embeddings, temporal modeling, and cross-system correlation by Swaminathan et al. [27]. Kala et al [28] Uses CNN and LSTM models to recognize toddler activities from video, with predicted results viewable locally or uploaded to the cloud for remote access. Kala et al. [29] discusses constrained IoT devices and proposes secure multipath routing. Kala et al. [30] describes IoT-enabled edge computing for farm-level real-time processing and decision-making.

Kuriakose et al. [31] demonstrated DeepNavi, a lightweight deep learning navigation assistant that can be deployed on a smartphone and was validated through pilot testing. The study showed that a compact on-device model can achieve rapid inference without compromising usability, a finding relevant to mobile deployment. Ajitha Gladis et al. [32] proposed In-Out YOLO Glass, a YOLO network that uses adaptive spatial pooling and attention augmentation for both indoor and outdoor object detection. This architectural refinement informed the design of the attention mechanism in the hybrid approach proposed in the present study. Nagarajan and Gopinath [33] combined metaheuristic optimization with deep learning detection architectures for indoor object detection and distance estimation tailored to visually impaired users. Obayya et al. [34] reported a YOLOv12-based indoor assistive framework and assessed its computational efficiency in terms of parameter count and inference time. Zhang et al. [35] deployed an improved YOLOv5 model on a Raspberry Pi equipped with a depth camera to provide indoor visual assistance to blind users. Their hardware implementation provides a precedent for edge deployment.

3. PROPOSED METHODOLOGY

3.1 Overview of the proposed framework

The Proposed AI-Powered Food Assistant System is designed with a clear and meaningful goal to help visually impaired individuals feel more confident and capable in their own kitchen. By using computer vision and speech recognition, the system works in real time to identify food items and kitchen tools, allowing users to act more independently and stay safe when using the kitchen tools by Islam et al. [4].

The system is based on YOLO detection model, which is trained to recognize a wide range of kitchen objects like vegetables, fruits, meals, and cooking ingredients. The live camera captures pictures, divides them into separate frames, and uses the YOLO model to identify and group many objects at the same time, even if they look alike. Once the objects are recognized, the system provides audio feeback through a CNN-based speech generation that informs the user about the names and positions of the objects, providing real-time, hands-free assistance. An interesting feature of this system is ability to provide intelligent recipe recommendations. The system looks at what is around it to find out which items and tools are often used. Based on the detected items and user-specific requests, it provides relevant guidance and recommendations to support the cooking process. When a user selects a recipe, the system reads out the full recipe, including the list of ingredients, step-by-step instructions and cooking tips using speech to text technology. This enables blind people to make recipes on their own without seeing or reading. Additionally, the system provides users with the ability to tailor recipe recommendations according to their diet such as selecting vegetarian options or modifying ingredient amount. The information of the recognized objects is stored in the database, which enables the system to learn continuously, personalize the experience for individual users and improve the model over time. Besides the text-to-speech conversion, the system also has strong speech recognition capabilities so that the user can control the system with voice commands. User can talk to system by asking questions about the items, following the recipe step by step, or hearing instructions through voice. consequently, the system makes the overall work while providing users with a more convenient and engaging experience. By adapting its guidance to individual requests, it helps users to carry out tasks with greater confidence, comfort and independence.

The system is designed for real-time and uses a hybrid YOLO CNN model with Residual Blocks and IoU for high detection accuracy in different lighting and kitchen environments. Bounding Box Regression increases the accuracy of object detection by preserving precision and minimizing errors.

For use on mobile devices and in embedded systems, the design is made to work well with fast processing right on the device, which helps save power and keeps energy use low. Generative Adversarial Network techniques are used to enrich the training dataset, to improve the system's ability to recognize a wide range of foods and to generalize better. The system works well with tools like screen readers, Braille displays and smart home devices makes it very easy to use for everyone by Shanmugapriya et al. [14]. It uses OpenCV to process the image and pyttsx3 for TTS conversion. It contains helpful tools like threading, queues, and logging to make tasks easier and manage data smoothly. A key thing is user privacy, since all AI-based processing is encrypted to make sure data is handled securely and follows privacy rules. To summarize, this smart kitchen helper helps visually impaired individuals to feel confident and able to use their cooking area on their own. By using all the models, it provides real-time performance, fast voice replies, and customized support, which makes cooking simpler and more empowering.

3.2 Automation process and real-time response mechanism

Figure 1 illustrates the overall workflow of the proposed Detection and Recognition system for AI-Powered Food Assistance. The system is designed for visually impaired individuals. It acquires visual and spoken input in parallel, interprets both and returns spoken guidance without requiring manual steps. Solid arrow in the figure denotes data flow, a dashed arrow denotes model or service call and bold return path denotes the real-time automatic loop.

1. Input Acquisition: After initialization, an auto trigger schedules capture times and activates the two input channels at the same time. The camera supplies live RGB Frames of the food items in view, and the microphone records the user's voice command. The capture is scheduled by the system; the user does not need to press any control to start an interaction.

2. Preprocessing: The Visual and audio streams are conditioned separately before recognition. Frames are denoised and normalized so that the detector receives consistent input under varying kitchen lighting conditions. Audio is transformed into frequency domain using Fast Fourier Transform (FFT) and passed through a noise filter to suppress background sounds, which improves the reliability of speech recognition.

3. Recognition: The pre-processed frames are passed to a YOLO detection model, which identifies the ingredients in the scene and returns the detected items with their confidence scores. In parallel, the pre-processed audio is passed to the ASR module, which converts the utterance into text and extracts the user's intent with an associated score. The dashed arrows indicate that the preprocessing stages invoke these models as services.

4. Decision Making: The detected item with scores and recognized intent with its score are merged at a decision node labelled “Result Clear?”. This node determines whether the combined evidence is reliable enough to proceed. If it is not, the system generates a clarify prompt that asks the user to retry the command or reposition the camera. This prevents the system from acting on uncertain detections or misrecognized commands, which is important for a user who cannot visually verify the output.

5. Recipe matching and response generation: when the result is clear, the recommendation module retrieves candidate recipes from the recipe database, which stores ingredients and preparation steps. It ranks them using cosine similarity between the detected ingredients and each recipe’s ingredients profile. The ranked item or clarify prompt from the previous stage is then passed to the response module. The response module produces the textual reply, which is either the next recipe step or a request for further input.

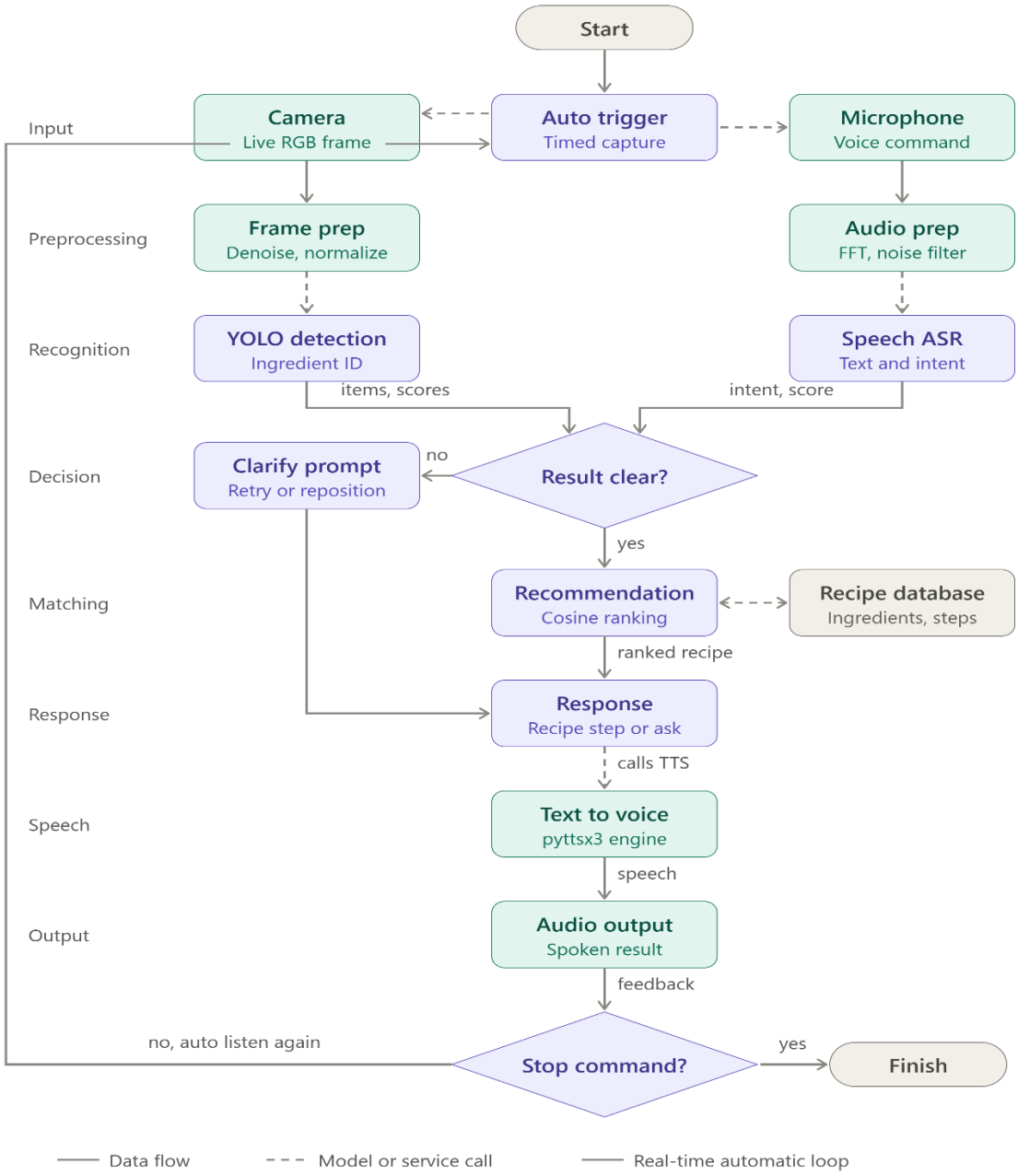


Figure 1. Workflow of object detection and speech recognition system for AI-powered Food Assistance

6. Speech Output and Feedback: The response text is converted into speech by the text-to-voice module, implemented with the pyttsx3 engine. The audio output delivers the spoken result to the user, which forms the feedback stage of the system. Since all guidance is delivered as audio, the user can follow recipe instructions hands-free.

7. Real- time automatic Response: After each spoken response, the system evaluates whether the user has issued a stop command. If not, the control flow returns automatically to the auto trigger, and capture and listening resume without user intervention. If a stop command is detected, the workflow terminates. This continuous loop, combined with parallel processing of two input channels, lets the system respond to the user’s changing kitchen situation and follow-up commands as they occur.

The workflow integrates automated input acquisition, confidence-based decision making, recipe recommendation and voice feedback into single closed loop, enabling hands-free kitchen assistance for visually impaired users.

The system works smoothly with a camera setup to

recognize object in real time, offering useful support to visually impaired users in a kitchen setting. At first, the user sets the camera through the application to obtain appropriate view for object detection. Once it is installed, the AI based system constantly monitors the surrounding environment and gives voice feedback whenever it detects any relevant object. The webcam continuously records video of the selected area, enabling the system to analyze the environment on the fly. The captured video is then broken down into separate frames at regular intervals, allowing the system to examine each frame carefully and accurately detect objects. This frame-based processing approach helps the system identify commonly used kitchen tools, food items and other important objects. The use of proposed model framework facilitates fast and reliable object detection, making it suitable for real time applications and improving the overall responsiveness and performance of the system.

The system is designed to maintain reliable object recognition under different kitchen settings. Variation in lighting, object positioning and background clutter may affect detection accuracy. By incorporating machine learning techniques, the system can gradually improve its capability to identify objects and deliver suitable audio-based instructions.

3.3 System architecture for AI-Powered Assistive System

As illustrated in Figure 2, the architecture of the proposed AI-Powered Food Assistive System for visually impaired individuals. The system is organized as a set of modules and processing steps.

1. User interaction layer: The user is the origin and final recipient of the workflow. The user points the camera at the food items and issues a voice command through the microphone. The system needs no screen interaction, and the user only points, speaks and listens. The camera supplies live RGB frames and the microphone supplies the spoken command and both are passed to their respective preprocessing modules.

2. Automatic scheduling and preprocessing: when the user starts the system, the auto scheduler triggers timed capture on both channels. In the vision channel, the frame preprocessing module denoises and normalizes each frame to reduce the effect of lighting variation. In the audio channel, the audio preprocessing applies FFT and noise filtering to suppress the background sound. The dashed arrow shows that the scheduler invokes these modules automatically without further user action.

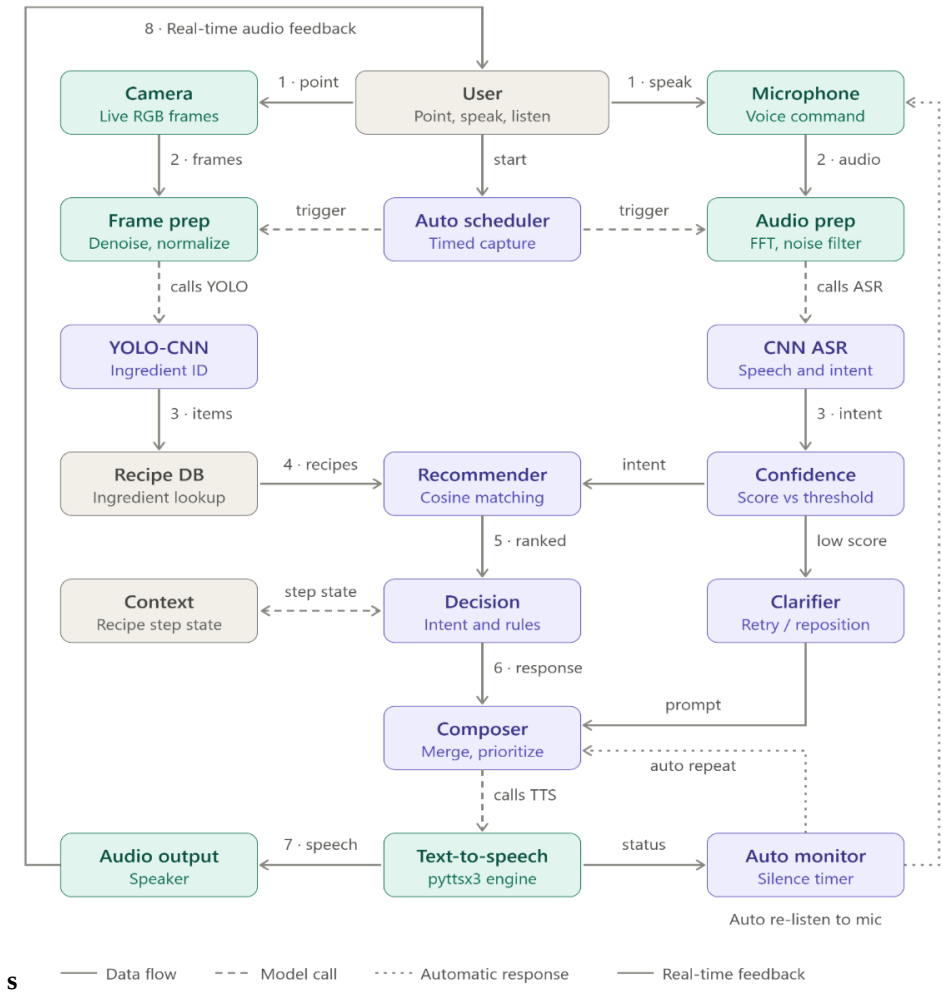


Figure 2. Architecture diagram of the AI-Powered Food Assistive System for visual impairments

3. Model calling relationships: the two recognition models are invoked as a service by their preprocessing modules. Frame preprocessing calls the YOLO-CNN model, which identifies the ingredients present in the frame and returns the detected items. Audio preprocessing calls the CNN-based ASR model, which converts the speech to text and extracts the user's intent. These calls are shown as dashed arrows to separate them from the flow of data.

4. Recipe retrieval and confidence evaluation: The detected items are used to query the recipe database, which returns candidate recipes to the recommender. The recommender ranks the candidates by cosine similarity between the detected ingredients and each recipe's ingredients profile. In parallel, the recognized intent passes through the confidence module, which compares the recognition score with a threshold. When the score is sufficient, the intent is forwarded to the recommender. When the score is low, the request is sent to the clarifier, which prepares a prompt asking the user to retry or reposition. This keeps the system from acting on uncertain recognition results.

5. Decision making and context management: The ranked recipes are passed to the decision module, which combines the intent with a set of rules to choose the appropriate action, such as presenting a recipe or advancing to the next step. So that later commands like "next" or "repeat" are interpreted correctly.

6. Response generation and speech output: The decision output and any clarification prompt from the clarifier are received by the composer. This merges and prioritizes them into a single response. The composer calls the text-to-speech engine implemented by pyttsx3, which converts the text response to speech. The audio output module then delivers the spoken result through the speaker.

7. Real-time feedback and automatic response: The spoken result returns to the user as real-time audio feedback, which closes the interaction loop and allows the user to respond with the next command. The text-to-speech module also reports its status into an auto monitor, which runs a silence timer. If the user does not respond within the allowed time, the monitor sends an automatic repeat request to the composer, and it also reopens the microphone for auto re-listening. These automatic responses keep the system active without manual intervention.

In addition, the system enables continuous interaction via audio commands, allowing the user to ask for more information, move through the steps, or switch to different recipes without any obstacles. This process of spotting, reacting, and communicating creates a helpful and easy to use cooking assistant that's especially good for people who can't see. The proposed hybrid algorithm is a strong method that uses three main things. Residual blocks split a picture into parts that are S × S in size, making it simpler for the network to handle by looking at certain areas of the image. Bounding box regression helps the system precisely locate each detected object by surrounding it with a box. Each bounding box is defined by four parameters, its width (b w), height (b h), class (c), and center position (bx, by).

This ensures that objects are correctly placed in the image. IOU method is a way that helps YOLO get better finding objects by checking how much the boxes it guesses match up with real ones. By utilizing a one box prediction method, YOLO quickly guesses the size and type of objects in one go, making it fast, precise, and able to work in real time for different uses.

3.4 AI-Powered Assistive System with You Only Look Once-Convolutional Neural Network-based hybrid structure

All Food Assist for Blind People is an AI system that aims to help visually impaired users become more independent and safe when cooking. The app uses computer vision, deep learning, speech recognition, and real-time audio feedback to help users to identify food items, utensils and guide them while they cook. At the center of this system is YOLO technology, which takes in live camera video by processing and analyzing each frame to achieve fast and accurate detection of items such as vegetables, fruits, meals, and utensils. This is supported by a hybrid structure that includes residual blocks, bounding box regression, and optimization, which helps to ensure reliable detection even in different lighting conditions or busy backgrounds. Once objects are identified, the system responds right away with spoken feedback using CNN-based TTS methodology to announce what was detected and where it is located by Shanmugapriya et al. [14]. The application also includes speech recognition, built on various Fourier transform techniques, so users can ask for instructions to be repeated or request more details through simple voice commands. Together, these features create a natural, back-and-forth interaction between the user and the system. One of the standout features is the recipe recommendation engine, which uses cosine similarity to suggest dishes based on whatever ingredients the camera pickup. For example, spotting tomatoes and onions on the counter might prompt the system to suggest tomato soup or curry. Once a recipe is chosen, the system reads out the full recipe, preparation steps, and instructions so the user can get complete hands-free guidance in the kitchen. We can also customize recipe recommendations to individual dietary needs and preferences.

All detection information and user interactions are stored securely in a database. This stored data is used in the learning process of the system such that the object recognition model improves over time. The review of past detections combined with GAN-based data augmentation makes the model increasingly accurate and well adapted to real-world conditions. Real-time detection, voice interaction, personalized recipe recommendations, and secure data handling make this cutting -edge Food Assistant System a safer, more accessible, and empowering experience for visually impaired individuals cooking every day. The convolution operation is one of the basic operations for image processing, especially in object detection tasks. It is a process of sliding a convolution filter over an image to extract features such as edges, textures, or shapes.

$$O(a,b) = \sum_{m}\sum_{n} I(a-x, b-y) \times K(x,y) \tag{1}$$

where, $O(a, b)$ shows the result at position (a, b) by multiplying each value in the kernel K with the matching pixel values $I(a\text{-}x, b\text{-}y)$, then adding all those multiplied values together. This process is performed several times for each pixel of the image. As this happens, the model begins to learn important features of the image such as edges, textures and shapes. This helps the model to recognize objects, which is key for the recipe-based system to identify the ingredients correctly.

In voice recognition tasks, the system uses the Discrete Fourier Transform to convert an audio signal from time phase into the frequency domain. This process is important step for identifying phonetic elements of speech.

$$X(K) = \sum_{n=0}^{N-1} x(n) e^{-j\frac{2\pi kn}{N}} \tag{2}$$

where, $X(K)$ represents the frequency phase format, $x(n)$ is the

time phase audio signal, and N indicates the number of points in the transformation. The Fourier transform breaks the signal into sinusoidal components, which can be examined to identify phonemes, words, or other spoken commands, allowing systems to process speech with precision.

In recipe recommendation systems, cosine similarity is used to determine how similar two food items are by comparing their feature vectors. This similarity measure is calculated using the following formula,

$$(X, Y) = \frac{\sum_{i=1}^{n} X_i Y_i}{\sqrt{\sum_{i=1}^{n} X_i^2} \times \sqrt{\sum_{i=1}^{n} Y_i^2}} \quad (3)$$

where, X and Y are feature vectors that represent two distinct food items, and the denominator reflects their respective magnitudes. The similarity measures the angle between these two vectors, offering a way to assess how alike the items are. This makes it especially useful recipe recommendations, since it allows the system to suggest dishes that share similar ingredients with what is been detected.

3.5 Model configuration and training details

Table 2 summarizes the architectural configuration and training hyperparameters adopted for both branches of proposed hybrid system. The detection branch uses modified YOLOv8n backbone with residual blocks and IoU refined bounding box regression, comprising 53 convolutional layers and 3 detection heads, operating on 640 × 640 × 3 input. It was trained using the Adam optimizer over 150 epochs with a batch size 16 on 12500 GAN augmented kitchen item images.

Table 2. Model configuration and training parameters

Parameter	YOLO-CNN Detection Branch	CNN-Based Speech Recognition Branch
Base architecture	YOLOv8n Hybrid with residual blocks	1-D CNN over MFCC features-6 layer convolutional stack
Backbone modification	Residual blocks and IoU-refined bounding-box regression	Batch-normalized convolutional blocks
Number of layers	53 convolutional layers and 3 detection heads	6 convolutional layers and 2 fully connected layers
Input size	640 × 640 × 3	16 kHz audio, 25ms frame length, 10ms hop, 40 MFCC coefficients per frame
Optimizer	Adam	Adam
Learning rate	0.001 cosine annealing decay	0.0005, step decay
Batch size	16	32
Number of training epochs	150	100
Training/validation/test split	70%/15%/15%	70%/15%/15%
Framework	PyTorch 2.0 YOLOv8 implementation	TensorFlow 2.12 keras
Total training dataset size (images)	12,500 annotated kitchen item images with post-GAN augmentation	Audio dataset 5000 labelled voice commands

Note: You Only Look Once (YOLO), Convolutional Neural Network (CNN), Intersection over Union (IoU), generative adversarial network (GAN).

The speech recognition branch employs a 6-layer 1-D CNN over MFCC features followed by two fully connected layers. It was trained using Adam over 100 epochs with a batch size 32 on 5000 labelled voice commands. Both Branches used a 70%, 15%, 15% of train, validation and test split. The detection branch was implemented in PyTorch while the speech branch used TensorFlow, balancing accuracy with real-time deployment efficiency.

3.6 Dataset description

Table 3 summarizes the sourcing and pre-processing characteristics of the dataset used to train and evaluate the proposed hybrid YOLO-CNN detection model. The dataset was constructed by combining multiple fruits and vegetable detection sources with kitchen item categories to provide sufficient class diversity and visual variability for robust model training.

Table 3. Dataset description

Item	Details
Data source(s)	LVIS Fruits-and-Vegetables Detection Dataset(github.com/henningheyen/Fruits-And-Vegetables-Detection-Dataset), Fruits and Vegetable Detection for YOLO (datasetninja.com/fruits-and-vegetable-detection), kaggle.com/datasets/moltean/fruits, kaggle.com/code/danielwiszowaty/fruit-and-vegetable-detection-using-yolov8.
Total number of images	12,500(post augmentation), 8200(pre augmentation) for proposed system execution.
Number of object/food classes	25 classes (fruits, vegetables, packed foods and common kitchen utensils) for execution.
Samples per class (range/table)	220-650 images per class.
Train/validation/test split	70%/15%/15%.
Are test images independently captured?	Yes, the test set images are captured in separate sessions using different lighting setups, backgrounds, and camera angles than the training/validation sets to avoid scene overlap and ensure unbiased evaluation.
Data augmentation methods used	Rotation, flipping, brightness/contrast jitter, and 3,000 GAN-based synthetic samples were added.

Note: You Only Look Once (YOLO).

This multi-source composition was done to broaden the range of object appearances, backgrounds and imaging conditions represented in the training data, thereby improving the model's ability to generalize across the varied visual characteristics of real-world kitchen scenes.

The dataset compiled from these sources for our proposed model comprises 12,500 images post-augmentation expanded from 8,200 original images, spanning 25 classes covering fruits, vegetables, packed foods, and kitchen utensils with 220-650 images per class. Data was partitioned using a standard 70% ,15%, 15% of train, validation and test split.

Test images were independently captured in separate sessions with distinct lighting, backgrounds, camera angles, preventing scene overlap with training data and ensuring unbiased evaluation of real-world generalization. To enhance the robustness and improve the recognition of visually similar or infrequent items, rotation, flipping, brightness or contrast jitter were applied along with 3,000 GAN-based synthetic samples added to the training set.

3.7 Pseudo-labeling algorithm

Algorithm 1 describes the operational workflow of the AI powered Food Vision Assistant System, which integrates real-time object detection, speech interaction, recipe recommendation, and voice-guided navigation to assist visually impaired users during food preparation. The system integrates computer vision with natural language processing to provide a smart, hands-free cooking experience.

The system begins by taking a live picture with the device's camera called captureImage. Then, it goes through a preprocessing step called preprocessFrame to fix the lighting, lower the noise in the image, and make the image clearer so that the analysis can be done accurately. The improved image is then given to the trained YOLO model that detects and labels different food items in the picture. After detecting the items, each one is shown with a box around it for visual confirmation. The names of these items are then turned into spoken words using the text formatting module and a TTS engine. This helps visually impaired users to know what food items are available without needing to see them.

Algorithm 1. Food Vision Assistant System

```
liveFrame ← captureImage (imageInput)
processedFrame ← preprocessFrame (liveFrame)
detectedItems ← objectModel.detectObjects(processedFrame)
annotatedItems ← drawBoundingBoxes (detectedItems)
speechText ← formatDetectedItems (detectedItems)
speakOut (speechModel, speechText)
userVoice ← captureVoiceInput ()
parsedCommand ← processVoiceCommand (userVoice)
ingredientList ← extractIngredients (detectedItems)
recommendedRecipes ← matchRecipes (recipeDB, ingredientList, parsedCommand)
personalizedRecipes ← filterByPreferences (recommendedRecipes, preferenceData)
speakOut (speechModel, personalizedRecipes.step1)
for each step in personalizedRecipes do
wait for voiceCommand "Next" or "Repeat"
if voiceCommand == "Next" then
speakOut(speechModel, nextStep)
else if voiceCommand == "Repeat" then
speakOut(speechModel, currentStep)
end if
```

Once the food items are detected, the system listens for the user's voice command using the ASR module. The spoken words are converted into text, which is then checked to understand the user's request, like asking for a recipe or mentioning dietary conditions. At the same time, the identified food items are put together into a list of ingredients, which is used as the main input for the recipe recommendation system.

Algorithm 2 explains how the system finds recipe match in the Food Vision Assistant System based on the ingredients found by the object detection and the user's verbal preferences. The algorithm uses a way to represent the ingredients as vectors to find similar ones, offering recipe suggestions that are relevant and match the user's needs.

Algorithm 2. Match_Recipes (itemList, recipeDB, parseCmd)

```
Match_Recipes (itemList, recipeDB, parseCmd)
vectorIngredients ← vectorize(itemList)
bestMatches ← []
for recipe ϵ recipeDB do
vectorRecipe ← vectorize(recipe.ingredients)
similar ← cosineSimilarity (vectorIngredients, vectorRecipe)
if similar >= threshold
then
bestMatches.add((recipe, similar))
end if
end for
sorted_Recipes ← sort (bestMatches by similar)
return filterByCommand(sorted_Recipes, parseCmd)
```

The list of detected ingredients is turned into a number-based format using a vectorization function. This helps to organize the ingredient information in a way that makes it easier to compare with other recipe vectors stored in the recipe database. Each recipe is also converted into an ingredient vector so that all data in the same format, which helps when calculating how similar they are.

The algorithm then checks each recipe in the database by comparing the vector of each recipe. It uses cosine similarity, which measures how similar two vectors are by looking at the angle between them. This method works well because it shows how well the ingredients match, no matter how long the vectors are.

To keep the recommendations relevant, only recipes that meet a similarity score above a certain level are considered. These recipes, along with their similarity score, are saved in a list of possible matches. Using this threshold helps remove recipes that do not share enough ingredients, making the search more accurate by cutting out unnecessary suggestions.

After all possible recipes are checked, they are sorted from the most similar to the least similar. This way, recipes that are closest in ingredient come first, making it easier to find ones that need just a few more items.

Finally, the sorted list of recipes is adjusted using the voice command that was recognized by the speech system. This step takes into account user preferences like favourite cuisine, dietary limits, meal types, or cooking styles, to create recommendations that are both personalized and fit the situation. The final list of recipes is then sent as the output, allowing the assistive system to suggest recipes that match

what is available and what the user needs.

The proposed recipe matching algorithm works well by combining ingredient similarity checks with context-based filtering using voice input, which helps provide better recipe suggestions. Its organized structure makes it simple to connect with the other parts like object detection and speech interaction. It also runs efficiently, allowing real-time performance in smart kitchen settings. Because of this, the algorithm creates an accurate, flexible, and user-centered recommendation system that makes the Food Vision Assistant System more helpful for people who are visually impaired.

4. RESULTS AND DISCUSSION

The Food Vision Assistant System leverages computer vision, voice recognition, and intelligent recipe recommendations to provide precise and tailored dish suggestions. It employs an advanced object detection model that uses convolution operations to accurately identify food items in images and extract key features. Voice recognition interprets user commands accurately, enabling users to operate easily. The similarity algorithm efficiently links the detected food items with voice inputs, thus ensuring recipe-relevant suggestions. This AI-based system enhances user satisfaction with accurate food identification, natural language understanding and smart decision-making.

The Food Vision Assistant captures pictures of the food items in real time, and the advanced object detection model is applied to process the image. This system guarantees correct identification of various food items so that users can get accurate suggestions for possible meals. Each detected food item has a confidence score so that users can verify the accuracy of the system and build trust in its recommendations.

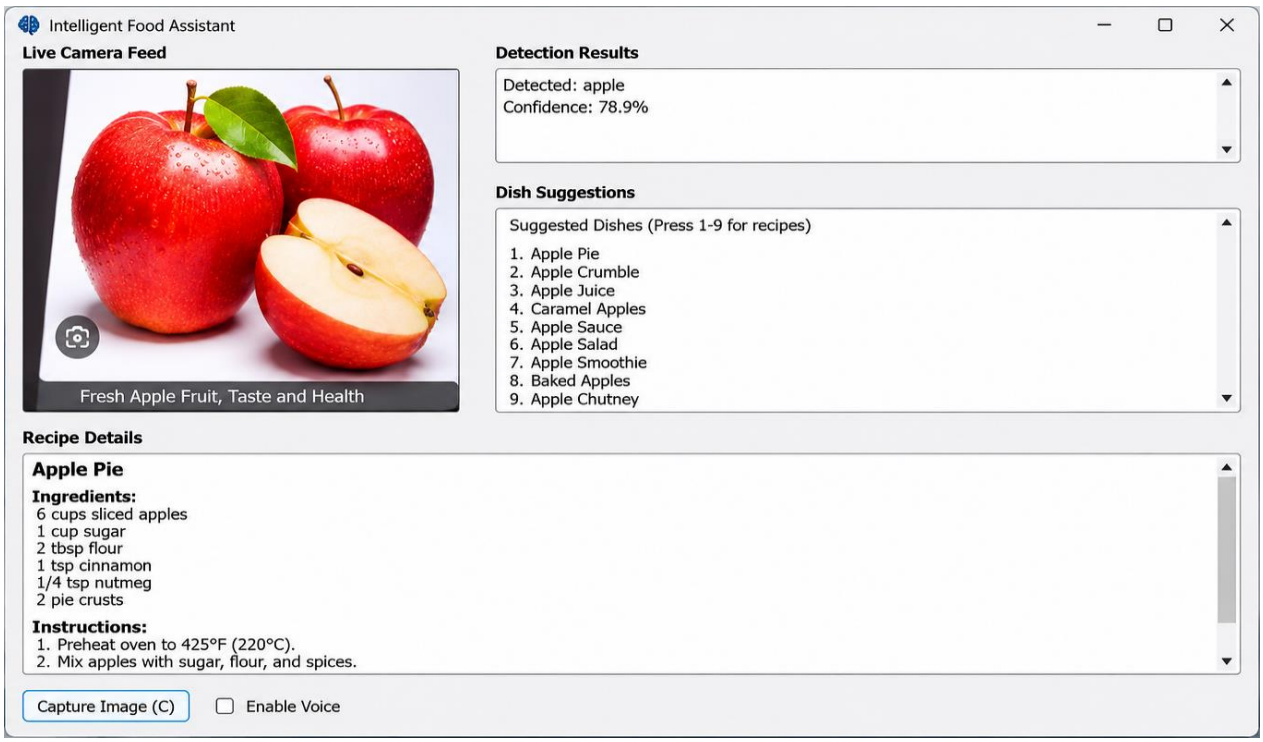


Figure 3. Food Vision Assistant

The AI powered assistant system in Figure 3, constantly improves its accuracy of machine learning. It evolves with new food types and variations over time. The system uses a convolutional neural network and deep learning techniques to effectively recognize multiple food items in a single image, making it especially useful in complex or cluttered environments. The ability to give suggestions at the moment makes the cooking process easier and allows users to experiment with a variety of recipes. Through an intuitive interface, users can refine searches for recipes, request alternative ingredients, and follow step-by-step cooking instructions via voice commands. Users can take photos of ingredients, verbally communicate what they want and get personalized recipe suggestions immediately.

The ingredients assistant module, as shown in Figure 4, helps the user to find recipes based on the ingredients they have, encouraging creativity and reducing food waste. It uses Natural Language Processing to identify the available ingredients accurately. It demonstrates how the system can translate the detected ingredients into actionable cooking instructions by generating recipe instructions based on the recognized food items.

The interactive features allow users to navigate through the instructions smoothly, with voice guided assistance or text-based step tracking. This feature enhances user satisfaction by making the cooking process simpler and requires less effort to follow recipe, resulting in more efficient, enjoyable and personalized meal preparation. If user miss or does not understand an instruction, they can simply repeat the current or previous steps by using ad voice command. They can also be moved to the next step with a voice command or a simple click, so manual navigation is less needed. Figure 5 shows the Classification and Detection Application. This indicates the system's capability of employing deep learning for classification, allowing for accurate identification of a wide variety of food items.

The model is trained with a wide variety of images, such as raw ingredients, cooked meals, packaged foods, so it can tell

the difference between different types of food. It uses convolutional neural networks to look at detail characteristics like texture, color, and shape, which helps in recognize items correctly, even when they look alike. This way of sorting food is useful for keeping track of what someone eats, managing food supplies, and offering meal suggestions tailored to individual needs.

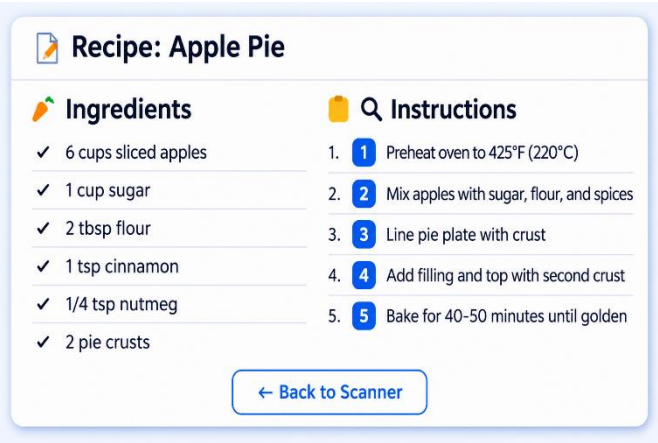


Figure 4. Ingredients Food Assistant

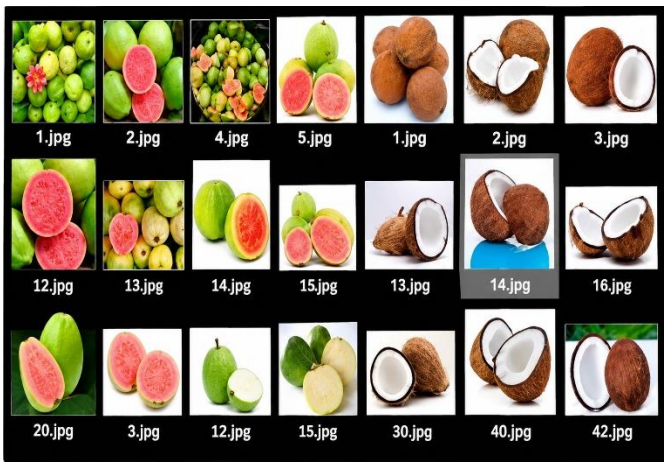


Figure 5. Classification and detection framework

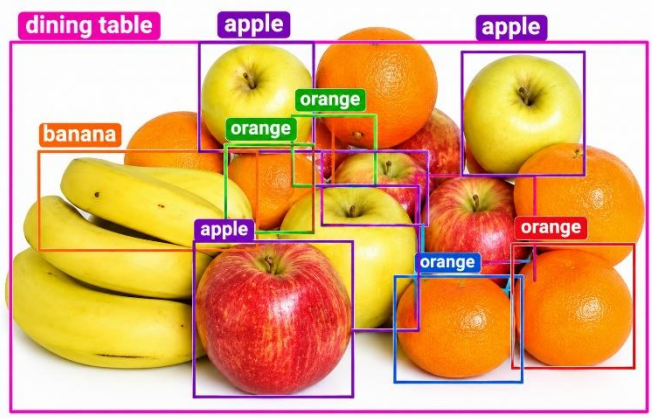


Figure 6. Deep learning-based fruit detection and classification system

The model gets better over time by learning from user feedback and by adding more images, allowing it to keep up new food trends and items as in Figure 6.

5. EXPERIMENTAL RESULT AND ANALYSIS

In this section, we evaluated the performance across four complementary dimensions. The detection accuracy under varying environmental conditions, statistical reliability and repeatability, real world usability with the target user population and cross-platform deployment feasibility. Section 5.1 shows the detection metrics across five environmental scenarios, with strong accuracy. Section 5.2 assessed the reliability through repeated trails, confirming consistent performance with rising reliability under challenging conditions. Section 5.3 evaluated usability with visually impaired participants with two tasks, demonstrating strong success rates and satisfaction. Section 5.4 benchmarked deployment across four platforms, confirming practical feasibility. These completed evaluations validate the system's accuracy, reliability and deployment readiness.

5.1 Experimental setup and evaluation metrics

Table 4 shows a comparative evaluation of the AI Powered Food Assistant System performs in different real-world conditions. The test was done to check how well the object detection and speech recognition parts work in various environments that visually impaired people often face in kitchen. Under ideal indoor lighting, the model achieved its highest performance with the precision of 98.8%, recall of 95.6%, mAP@0.5 of 96.5%, F1 Score of 98.1% confirming that the architecture performs well under controlled, well illuminated conditions with minimal background interference.

As environmental complexity increased, a consistent and expected degradation in performance was observed. In the low-light environment, precision and recall dropped to the 94.7% and 91.6% respectively, reflecting the reduced feature visibility and increased noise inherent to low illumination imaging. The noisy background scenario further reduced performance marginally mAP@0.5 is 91.1%, suggesting the model retains reasonable robustness to background clutter and sensor noise, though with a slight increase in false positives.

The most challenging conditions were the cluttered kitchen scene and multiple items detection scenarios, which recorded the lowest metrics. The multiple items detection scenario exhibited the highest gap between the precision and recall, indicating that as object density increases, the model is prone to missed detection.

In terms of computational efficiency, inference time increased steadily from 584ms in ideal conditions to 645ms under multiple item detection corresponding to a decrease in FPS from 1.71 to 1.55. This inverse relationship between scene complexity and processing speed is expected, as denser scenes require the model to process a greater number of candidate regions and additional non-maximum suppression operations.

Overall, the system achieved an Average Precision of 94.6%, recall of 91.6%, mAP@0.5 of 91.5%, F1 Score of 93% and an average inference speed of 1.62 FPS across all tested conditions. These results demonstrate that the proposed model maintains strong detection accuracy and reasonable robustness across diverse and challenging real-world scenarios.

This shows how well the YOLO-based object detection framework and the speech processing part work when the lighting is just right. These high scores show that the system can reliably identify kitchen items and provide accurate voice responses in controlled environment.

Table 4. Performance evaluation metrics of AI-Powered Food Assistance System for visually impaired users across varying test scenarios

Test Scenario	Precision (%)	Recall (%)	mAP @0.5(%)	F1 Score(%)	Inference Time(ms)	FPS
Ideal Indoor Lighting	98.8	95.6	96.5	98.1	584	1.71
Low Light Environment	94.7	91.6	92.5	94.0	610	1.64
Noisy Background	93.2	90.2	91.1	92.5	620	1.61
Cluttered Kitchen Scene	92.8	89.1	89.9	91.3	629	1.59
Multiple Items Detection	90.4	86.6	87.5	88.8	645	1.55
Overall System Performance	94.6	91.6	91.5	93.0	617	1.62

5.2 Comparative analysis with existing detection models

Table 5 presents a comparative analysis of the Proposed Hybrid YOLO-CNN model against two widely used baseline detection architectures, such as Vanilla YOLO and SSD MobileNet baselines.

The proposed model achieved the highest detection performance across all accuracy metrics, with a precision of 94.6%, recall of 91.6%, mAP@0.5 of 91.5% representing improvements. This performance gain is attributed to the integration of residual blocks and IoU-refined bounding box regression, which enhances the model's capacity to resolve occluded and visually similar kitchen items more reliably than the existing systems. This comparative analysis confirms that the architectural enhancements introduced in the proposed hybrid model yield a meaningful and consistent improvement in detection accuracy over conventional YOLO and SSD based baselines. Also, it provides more accuracy in real world kitchen environments.

5.3 System performance and reliability analysis

Table 6 presents the system performance and reliability testing across the different environmental conditions based on 20 independent trails per conditions. The coefficient of variation was computed for both object detection and speech recognition. The coefficient of variation is defined as the ratio of standard deviation to the mean, provides a normalized measure of relative dispersion. Under ideal indoor lighting, the system achieved peak performance 98.8% precision, 97.4% speech recognition accuracy with minimal variability, shows highly stable and repeatable outputs. Overall, results confirms that environmental complexity impairs both accuracy and reliability.

5.4 User-based evaluation

To assess the practical usability and real worlds effectiveness of the proposed system for its intended user group, a small-scale evaluation was conducted with 10 visually impaired participants across two representative task such as Ingredients identification (Task A) and Full recipe Completion (Task B) as summarized in Table 7.

For Task A, participants completed the ingredient identification task rapidly, with a mean completion time of 7.9 seconds and a high voice command success rate of 96.7%, indicating that the system reliability interpreted user commands and returned accurate object identification with minimal delay. The low mean number of clarification requests 0.56 per session further supports the system's responsiveness and ease of interaction for single object recognition tasks, and participants rated usability 4.78 out of 5 shows reflecting strong user satisfaction and confidence in the system's performance. For Task B, which requires participants to complete a full recipe involving multiple sequential steps, showed the completion mean time of 93 seconds.

5.5 Comparative deployment analysis of proposed model across the various platforms

Table 8 benchmarks model complexity and cross-platform inference for the YOLO CNN hybrid detection model across four platforms, evaluated in terms of precision, model complexity such as parameters and size, interference efficiency such as latency and FPS, and power consumption. These are the factors that collectively determine platform suitability for real-world assistive deployment.

Table 5. Comparative analysis of detection models

Model	Detection Precision (%)	Recall (%)	mAP@0.5 (%)	FPS	Model Size (MB)
Vanilla YOLO	89.2	85.4	85.9	1.75	6.2
SSD-MobileNet	86.5	82.1	82.8	2.4	22.5
Proposed Hybrid YOLO-CNN (this work)	94.6	91.6	91.5	1.62	7.8

Note: You Only Look Once (YOLO), Convolutional Neural Network (CNN), Solid State Detector (SSD).

Table 6. System performance and reliability testing for different environments

Test Scenario	Trials (n)	Object Detection Precision: Mean ± SD (%)	Coefficient of Variation (%)	Speech Recognition Accuracy: Mean ± SD (%)	Coefficient of Variation (%)
Ideal Indoor Lighting	20	98.8	2.46%	97.4	3.65%
Low Light Environment	20	94.7	5.29%	94.8	5.24%
Noisy Background	20	93.2	6.04%	91.8	6.68%
Cluttered Kitchen Scene	20	92.8	6.23%	91.5	6.82%
Multiple Items Detection	20	90.4	7.29%	88.6	8.02%

Table 7. Small-scale user evaluation with visually impaired users

Metric	Task A: Ingredient Identification	Task B: Full Recipe Completion
Number of participants	10	10
Mean task-completion time (mean ± SD, seconds/minutes)	7.9s	93s
Operation/voice-command success rate (%)	96.7%	89.9%
Mean number of repeat/clarification requests	0.56	1.04
Mean usability rating (e.g., SUS score or Likert mean)	4.78	4.01

The Desktop/server GPU with parameters of 25-65M delivered the fastest inference but required high power, serving as a development benchmark. On embedded boards, the unoptimized model showed sharply increased latency. INT8 quantization reduced parameters and size, improving inference to 150-280 ms and 3.5-6.5 FPS with lower power, narrowing the gap towards GPU-level performance.

The mid-range smartphone platform offers the best portable, competitive latency, highest constrained platform and low power making more practical option for real-world assistive deployment.

Table 8. Model complexity and cross-platform inference benchmark

Platform	Precision	Parameters (M)	Model Size (MB)	Avg. Inference Time (ms)	FPS	Power Consumption
Desktop/Server GPU (development baseline)	YOLO CNN hybrid FP32	25-65	90-260	15-30	30-65	150-300
Embedded board (e.g., Raspberry Pi/Jetson Nano)	YOLO CNN FP16 unoptimized	25-65	90-260	580-650	1.5-1.7	5-10
Embedded board (quantized)	YOLO CNN int8 quantized	6-15	20-65	150-280	3.5-6.5	3-6
Mid-range smartphone (Android/iOS)	YOLO CNN (Mobile Optimized, TFLite/coreML)	3-10	10-40	80-180	5.5-12	2-4 (battery-based)

Note: You Only Look Once (YOLO), Convolutional Neural Network (CNN), Graphics Processing Unit (GPU).

5.6 Limitations and future work

The preceding results demonstrate good performance under various conditions. It remains usable for visually impaired users and can be feasibly deployed across a range of hardware platforms when appropriately optimized. The performance analysis has decreased in processing speed under complex environmental conditions, particularly in scenarios involving multiple overlapping items, indicating the current object detection architecture remains susceptible to occlusion and inter-object interference. The reliability analysis showed the accuracy degradation is accompanied by rising output variability. The user evaluation was limited to a small sample of 10 participants and two task types, constraining the generalizability of the usability findings across the broader diversity of visual impairment severity, kitchen layouts and user familiarity with assistive technology. The deployment analysis shows the quantized embedded and mobile-optimized configurations substantially narrow the performance gap with GPU-class hardware. hese findings point to three directions for future work. First, architectural improvements such as attention-based feature refinement or improved non-maximum suppression strategies to enhance robustness under occlusion and multi-tier clutter. Second, broader and longer-term user studies involving a larger and more diverse participant pool, extended usage periods and a wide range of real-world kitchen environments to validate usability findings at scale. hird, model compression techniques such as pruning, knowledge distillation, or neural architecture search to close the remaining performance gap on low-cost embedded platforms without compromising the accuracy required for safe, independent use.

6. CONCLUSIONS

The proposed AI-Powered Assistive System helps visually impaired people by recognizing kitchen tools and ingredients using object detection and speech recognition. It helps users to move around their kitchen confidently on their own without needing someone else to see for them. It gives real-time guidance and uses voice to provide step-by-step instructions, making sure cooking is done correctly. With Speech recognition, users can control the system using their voice to ask about ingredients, get recipe ideas, and check cooking steps. AI-powered object detection makes the system safer by correctly identifying tools and ingredients, which helps avoid mistakes and makes cooking faster. The system uses computer vision and deep learning algorithms to identify objects accurately, reducing the chance of misunderstanding. It uses YOLO for object detection and a CNN for speech feedback, giving clear and specific responses to keep cooking smooth and mistake-free. The AI works well in different lighting conditions and backgrounds, making it reliable in various kitchen settings. It also works with other assistive tools and making it easier to use and providing a user-friendly, interactive experience tailored for visually impaired users.

Future improvements will focus on making the AI more accurate by using integrated machine learning models, adding multiple languages to reach people around the world, and offering real-time feedback for quick corrections to help improve cooking skills.

Personalized AI assistants will help users save their preferences, dietary restrictions and favourite recipes, making the cooking experience more tailored to their needs. Connecting with smart kitchen devices that use IoT technology could handle tasks like controlling temperature, setting timers, and measuring ingredients, making cooking faster and easier. As the system gets better, it will help visually impaired individuals cook more independently, safely and confidently.

ACKNOWLEDGMENT

This work is supported by the Department of Computer Science and Engineering at PPG Institute of Technology and PSG Institute of Technology and Applied Research, Coimbatore, Tamil Nadu, India.

REFERENCES

[1] Arsalwad, G., Dabhade, S., Shaikh, K., D'silva, S. (2024). YOLOInsight: Artificial intelligence-powered assistive device for visually impaired using internet of things and real-time object detection. Cureus Journal of Computer Science. https://doi.org/10.7759/s44389-024-02353-2

[2] Sagana, C., Keerthika, P., Manjula Devi, R., et al. (2021). Object recognition system for visually impaired people. In 2021 IEEE International Conference on Distributed Computing, VLSI, Electrical Circuits and Robotics (DISCOVER), Nitte, India, pp. 318-321. https://doi.org/10.1109/discover52564.2021.9663608

[3] Varshini, P.H., Swati, V., Navya, D., Milan, A.K. (2025). EchoVision: RealTime object detection and voice assistance for the visually impaired. In 2025 3rd International Conference on Intelligent Systems, Advanced Computing and Communication (ISACC), Silchar, India, pp. 988-994. https://doi.org/10.1109/isacc65211.2025.10969403

[4] Islam, R.B., Akhter, S., Iqbal, F., Rahman, M.S.U., Khan, R. (2023). Deep learning based object detection and surrounding environment description for visually impaired people. Heliyon, 9(6): e16924. https://doi.org/10.1016/j.heliyon.2023.e16924

[5] Swetha, K., Asha, K.N., Santhosh, L., Potdar, V., Shekar, B.S. (2025). VisionAssist: A real-time multimodal navigation aid with object recognition, proximity estimation, and voice feedback for the visually impaired. In 2025 International Conference on Intelligent Computing, Information and Control Systems (ICOIICS), Lalitpur, Nepal, pp. 1-8. https://doi.org/10.1109/icoiics67115.2025.11390458

[6] Archana, G., Pavan kumar, K., Naaz, R., Bhavani, Y., Bharath, N. (2024). Echo guidance voice-activated application for blind with smart assistive stick using ML and IoT. Journal of Nonlinear Analysis and Optimization, 15(2): 709-717. https://doi.org/10.36893/jnao.2024.v15i2.120

[7] Ashvitha, S., Ashvitha, P., Judgi, T. (2026). Assistive system for visually impaired people using artificial intelligence. In 2026 International Conference on Smart Electronic Devices and Intelligent Systems (ICSEDIS), Chiramanangad, India, pp. 897-902. https://doi.org/10.1109/icsedis68157.2026.11518134

[8] Raja, B., Varsha, K.S., Pranav, L.K., Kuresan, H., Anilet Bala, A., Elavel Visuvanathan, G. (2025). AI enhanced spectacle for visually impaired people. In 2025 International Conference on Recent Advances in Electrical, Electronics, Ubiquitous Communication, and Computational Intelligence (RAEEUCCI), Chennai, India, pp. 1-6. https://doi.org/10.1109/raeeucci63961.2025.11048197

[9] Joshi, R.C., Yadav, S., Dutta, M.K., Travieso-Gonzalez, C.M. (2020). Efficient multi-object detection and smart navigation using artificial intelligence for visually impaired people. Entropy, 22(9): 941. https://doi.org/10.3390/e22090941

[10] Niranjana, R., Agash, K., Anbarasan, V., Lyngkhoi, B. (2026). Sight assist: Advanced deep learning for real-time object identification and assistance for the blind. International Research Journal on Advanced Engineering and Management, 4(5). https://doi.org/10.47392/irjaem.2026.0286

[11] Bharathi, P., Spurthy, B.L. D., Priya, A.B., Murthy, B.Y. K., Teja, C.C.S.V. (2023). Real-time object detection for blind people. International Journal of Advanced Research in Science, Communication and Technology (IJARSCT), 3(5): 262-266. https://doi.org/10.48175/ijarsct-9354

[12] Thilanka, S.A.N., Jinadasa, G.C.H.V.D., Bandara, P.M., Weerakoon, W.M.M.T. (2023). Vision-based real-time object detection and voice alert for blind assistance system. In 2023 IEEE 17th International Conference on Industrial and Information Systems (ICIIS), Peradeniya, Sri Lanka, pp. 507-512. https://doi.org/10.1109/iciis58898.2023.10253546

[13] Bharath Kumar, V., Padma Naidu, K., Kousalya, T., Ashok, M., Chandana, U., Fayaz, S.K. (2026). Real-time object recognition with voice feedback for visually impaired using raspberry Pi. International Journal of Scientific Research in Engineering and Management, 10(2). https://doi.org/10.55041/ijsrem58962

[14] Shanmugapriya, S., Hari Priya, G., Lokesh Sai, D., Jethin, G., Vinod Kumar, C. (2023). Audio assist: Enabling object detection through speech for the visually impaired. In 2023 2nd International Conference on Automation, Computing and Renewable Systems (ICACRS), Pudukkottai, India, pp. 1757-1762. https://doi.org/10.1109/icacrs58579.2023.10404820

[15] Anitha, T.G., Anagha, P., Rashmithaa Prabhu, G., Nair, G.B., Geetha, K. (2025). A survey on smart assistive system for visually impaired people obstruction avoidance through object detection and classification. International Journal for Research in Applied Science & Engineering Technology (IJRASET), 13(9): 2040-2043. https://doi.org/10.22214/ijraset.2025.74351

[16] Rajamani, S.P., Karuppanasamy, K., Thilagavathi, P., Chandru, T., Devi Dharsini, S., Kamini, P. (2026). Real-time object detection and text-to-speech system for enhancing accessibility of visually impaired. In 2026 6th International Conference on Trends in Material Science and Inventive Materials (ICTMIM), Kanyakumari, India, pp. 1006-1071. https://doi.org/10.1109/ictmim68190.2026.11507611

[17] Madhu, B.M., Raj, S., Das, S. (2025). Vision assistance system for visually impaired. In 2025 3rd International Conference on Smart Systems for Applications in Electrical Sciences (ICSSES), Tumakuru, India, pp. 1-5. https://doi.org/10.1109/icsses64899.2025.11009576

[18] Nandwani, P., Jain, G., Jain, S., Phulre, A.K., Joshi, S., Swaroop, R. (2024). Speech-based visual aid for the blind eye using deep convolution neural network. In 2024 International Conference on Emerging Trends in Networks and Computer Communications (ETNCC), Windhoek, Namibia, pp. 1-6. https://doi.org/10.1109/etncc63262.2024.10767556

[19] Reddy, K.K.S.N., Yashwanth, C., KVS, S., Tamidala Venkata Sai, P.A., Khetarpaul, S. (2020). Object and

currency detection with audio feedback for visually impaired. In 2020 IEEE Region 10 Symposium (TENSYMP), Dhaka, Bangladesh, pp. 1152-1155. https://doi.org/10.1109/tensymp50017.2020.9230687
[20] Subramanian, R.R., Ravikiran, L., Pavan Teja, K.V., Venkatesh Reddy, K., Nagendra Reddy, K. (2024). Voice guided object detection: Enabling independence for the visually impaired. In 2024 Third International Conference on Intelligent Techniques in Control, Optimization and Signal Processing (INCOS), Krishnankoil, Virudhunagar district, Tamil Nadu, India, pp. 1-6. https://doi.org/10.1109/incos59338.2024.10527768
[21] Yadav, N., Kumawat, S., Mishra, S.K., Jaseja, M., Khare, S. (2025). A real-time object detection system with audio feedback for visually impaired persons. In 2025 International Conference on Intelligent Control, Computing and Communications (IC3), Mathura, India, pp. 16-19. https://doi.org/10.1109/ic363308.2025.10957036
[22] Jambhulkar, A.R., Gajera, A.R., Bhavsar, C.M., Vatkar, S. (2023). Real-time object detection and audio feedback for the visually impaired. In 2023 3rd Asian Conference on Innovation in Technology (ASIANCON), Ravet, India, pp. 1-5. https://doi.org/10.1109/asiancon58793.2023.10269899
[23] Aralikatti, A., Appalla, J., Kushal, S., Naveen, G.S., Lokesh, S., Jayasri, B.S. (2020). Real-time object detection and face recognition system to assist the visually impaired. In First International Conference on Advances in Physical Sciences and Materials, Coimbatore, India, pp. 012149. https://doi.org/10.1088/1742-6596/1706/1/012149
[24] Iqbal, A., Akram, F., Ihtaram Ul Haq, M., Ahmad, I. (2022). A comprehensive assistive solution for visually impaired persons. In 2022 2nd International Conference of Smart Systems and Emerging Technologies (SMARTTECH), Riyadh, Saudi Arabia, pp. 60-65. https://doi.org/10.1109/smarttech54121.2022.00027
[25] Maheswari, S., Mangai, S., Aksharasree, J.R., Gobika, S., Janani, M., Janani, S. (2025). Object recognition and audio tactile system for visually impaired. In 2025 3rd International Conference on Intelligent Data Communication Technologies and Internet of Things (IDCIoT), Bengaluru, India, pp. 1008-1013. https://doi.org/10.1109/idciot64235.2025.10914756
[26] Ravi Kumar, D., Thakkar, H.K., Merugu, S., Gunjan, V.K., Gupta, S.K. (2020). Object detection system for visually impaired persons using smartphone. In ICDSMLA 2020. Lecture Notes in Electrical Engineering, Springer, Singapore, pp. 1631-1642. https://doi.org/10.1007/978-981-16-3690-5_154
[27] Swaminathan, S.S.A., Daniel, E., Iyapillai, K., Sivan, D. (2025). TALL-detect: A transformer and LLM-augmented framework for tamper-evident and explainable secure log analysis. Journal Européen des Systèmes Automatisés, 59(2): 455-475. https://doi.org/10.18280/jesa.590215
[28] Kala, I., Bhuvaneswari, P., Dharun, R., Dharshini, S. (2023). Toddler activity analysis using machine learning. In 2023 International Conference on Intelligent Systems for Communication, IoT and Security (ICISCoIS), Coimbatore, India, pp. 287-291. https://doi.org/10.1109/iciscois56541.2023.10100553
[29] Kala, I., Karthik, S., Srihari, K. (2021). Advanced hybrid secure multipath optimized routing in Internet of Things (IoT) based WSN. International Journal of Communication Systems, 34(8). https://doi.org/10.1002/dac.4782
[30] Kala, I., Praveen Barathi, K., Subhika, V., Siddiq, M. (2025). Farming smarter: Real-time agriculture with IoT enabled edge computing. In 2025 International Conference on Next Generation Computing Systems (ICNGCS), Coimbatore, India, pp. 1-7. https://doi.org/10.1109/icngcs64900.2025.11183161
[31] Kuriakose, B., Shrestha, R., Sandnes, F.E. (2023). DeepNAVI: A deep learning based smartphone navigation assistant for people with visual impairments. Expert Systems with Applications, 212: 118720. https://doi.org/10.1016/j.eswa.2022.118720
[32] Ajitha Gladis, K.P., Madavarapu, J.B., Raja Kumar, R., Sugashini, T. (2024). In-out YOLO glass: Indoor-outdoor object detection using adaptive spatial pooling squeeze and attention YOLO network. Biomedical Signal Processing and Control, 91: 105925. https://doi.org/10.1016/j.bspc.2023.105925
[33] Nagarajan, A., Gopinath, M.P. (2022). Hybrid optimization-enabled deep learning for indoor object detection and distance estimation to assist visually impaired persons. Advances in Engineering Software, 176: 103362. https://doi.org/10.1016/j.advengsoft.2022.103362
[34] Obayya, M., Al-Wesabi, F.N., Alshammeri, M., Iskandar, H.G. (2025). An intelligent optimized object detection system for disabled people using advanced deep learning models with optimization algorithm. Scientific Reports, 15: 16514. https://doi.org/10.1038/s41598-025-00608-z
[35] Zhang, K.K., Wang, Y.Y., Shi, S.Z., Wang, Q.Q., Wang, C., Liu, S. (2024). Improved YOLOv5 algorithm combined with depth camera and embedded system for blind indoor visual assistance. Scientific Reports, 14: 23000. https://doi.org/10.1038/s41598-024-74416-2

NOMENCLATURE

$O(a,b)$	Output feature map
$I(a-x, b-y)$	Input image pixel intensity
$K(x,y)$	Convolution kernel
$X(k)$	Frequency-domain representation
$x(n)$	Time-domain speech signal
X, Y	Feature vector
$X_i Y_i$	Feature components
N	Number of samples
i	Feature index
k	Frequency index
n	Time index
j	Imaginary unit
π	Mathematical constant
i	Feature index
k	Frequency index
n	Sample index
x,y	Kernel coordinates
a,b	Image coordinates