



VISISENSE: A COMPREHENSIVE IOT-BASED ASSISTIVE TECHNOLOGY SYSTEM FOR ENHANCED NAVIGATION SUPPORT FOR THE VISUALLY IMPAIRED

BHASHA PYDALA*, T. PAVAN KUMAR† AND K. KHAJA BASEER‡

Abstract. The field of visually impaired assistive technology looks for novel approaches to enhance independence and navigation. In this field, systems have to reliably identify and transmit environmental data in order to facilitate visually impaired users' safe and effective navigation. Developing a sophisticated framework for assistive technology that significantly enhances visually impaired navigation is the goal of this research. Make object detection and environmental awareness more efficient, dependable, and intuitive. This study introduces VISISENSE, "A Comprehensive IoT-Based Assistive Technology System for Enhanced Navigation Support for the Visually Impaired." VISISENSE is an IoT-based system with multiple components that enhances object detection. For primary environmental sensing, a handstick with implant sensors, a visual capture and transmission unit for processing visual data, and edge computing for object detection and classification are used. The system makes use of the R-CNN global computer vision model hosted on a cloud server, Mobinet computer vision models, and Logistic Regression with Iterative Learning. VISISENSE's effectiveness is demonstrated by a performance analysis of its object detection accuracy, processing speed, resource utilization, energy consumption, latency, and false positive rate. In all of these categories, VISISENSE performs better than Smart Stick and Smart Navigation. The data includes the fastest processing time of 17 ms, the most efficient resource utilization of 41%, and object detection accuracy of up to 99% at 2 Mbps load. Across all load conditions, VISISENSE has the lowest false positive rate, energy consumption, and latency. The VISISENSE assistive technology system is developed for the visually impaired. Its excellent object detection and navigation accuracy, speed, and efficiency enhance user experience and hold the potential to increase the independence and quality of life for visually impaired people. This research contributes to a significant advancement in assistive devices—smart, responsive technologies for the visually impaired.

Key words: VisiSense, Artificial Intelligent, R-CNN, Cloudsim, MobileNet, IoT, Binary Classifier.

1. Introduction. People who are blind or visually impaired frequently have mobility issues. To address these issues, there is growing interest in creating IoT-based assistive technologies. [1]. This piece introduces "VisiSense," a state-of-the-art navigation aid. The objective of VisiSense is to develop a simple navigation system for those who are blind. VisiSense provides complete support and real-time assistance to visually impaired users as they navigate foreign environments. A network of devices that gathers and evaluates environmental data includes wearable sensors, smartphones, and environmental sensors. For accurate navigation assistance, VisiSense uses edge computing, cloud computing, and AI algorithms for advanced object detection and recognition.

The accuracy of object detection, processing speed, resource utilization, energy efficiency, latency, and false positive rates are all areas where VisiSense's performance analysis reveals promising results.

Due to the system's precise and timely obstacle information, visually impaired people can now confidently navigate their surroundings.

VisiSense, a navigation aid for people with visual impairments, advances this field through IoT, edge computing, cloud computing, and AI. The design, implementation, and evaluation of this comprehensive assistive navigation system demonstrate its potential to increase the freedom, security, and quality of life of blind and visually impaired people. VisiSense can increase inclusivity and accessibility for people who are blind or visually impaired.

*Research Scholar, Department of CSE, KLEF (Deemed to be University), Guntur, India -522 302 (Corresponding author, basha.chanti@gmail.com)

†Professor, Department of CSE, KLEF (Deemed to be University), Guntur, India -522 302 (pavankumar_ist@kluniversity.in).

‡Professor, Department of Data Science, Mohan Babu University (Erstwhile Sree Vidyanikethan Engineering College), Tirupati, India-517102 (baseer.kk@vidyanikethan.edu).

2. Literature Survey. This review looks at studies that suggest novel ways to improve navigation for people with vision impairments, ultimately raising their quality of life. Each paper examines a different topic and technology, highlighting its advantages and potential for development.

Islam et al.'s [1] discussion of walking assistants includes systems based on sensors, computer vision, and smartphones. In developing assistive tools, the authors stress the value of reliability, lightweight design, and simplicity. The article thoroughly examines the benefits and drawbacks of various methodologies, emphasizing the demand for further development of current systems.

A visually impaired person can use an AI-based assistive technology for smart navigation, according to Joshi et al. [2]. They use a deep learning model in their system to identify objects and provide immediate auditory feedback. The system shows promising results, achieving high accuracy rates for object detection and recognition. Notably, the addition of a distance-measuring sensor improves its performance.

Using ultrasonic sensors for obstacle detection, Saisubramanyam et al. [3] present a cheap and lightweight smart walking stick. Within a 2-meter radius, the stick can detect obstacles, including water and dark areas. The advantages of this system are its low cost, compact design, precise detection abilities, and distinctive features. The paper, however, could benefit from more in-depth programming explanations and system comparisons.

Vaidya et al. [4] develop a real-time object detection system for visually challenged individuals using image processing and machine learning techniques. The system successfully detects objects and notifies blind users through audio output. Noteworthy advantages include independent navigation for visually impaired individuals and real-time object detection. However, no specific disadvantages are mentioned in the paper.

Ramesh et al. [5] propose a low-cost walking stick for blind people equipped with ultrasonic sensors and an SMS message system for emergency situations. The stick detects obstacles within a 4-meter range and provides voice alerts. This navigation aid stands out for its affordability, efficient design, artificial vision capabilities, portability, and the inclusion of an SMS message system. No disadvantages are mentioned.

Farooq et al. [6] design an IoT-enabled smart stick incorporating ultrasonic sensors, a water sensor, a high-definition camera, and object recognition capabilities. The stick offers obstacle detection and recognition modes, providing vibration and voice feedback to users. Additionally, GPS/GSM modules enable live location tracking and emergency assistance. The system's merits include obstacle detection and recognition, voice feedback, live location tracking, and energy efficiency. No specific demerits are mentioned.

Kuriakose et al. [7] review smart navigation solutions for blind and visually impaired individuals, encompassing various tools and technologies. The article highlights the core features lacking in existing navigation systems and offers recommendations for future design. Despite the development of numerous assistive navigation technologies, there remains a need for a seamless system that functions effectively both indoors and outdoors, enables dynamic interactions, and adapts to changes.

Bhasha et al. [15], S. Maidenbaum et al. [30] and K. Patil et al. [25] designed a smart navigation stick through sensor technology. Kim et al. [16] proposed a model for improving visually impaired people experience through deep neural networks. Rahman et al. [17] proposed one IoT based smart device through IoT and Deep Learning techniques. Bhavani et al. [18] and Ashraf et al. [19] suggested a Smart Walking Stick through machine learning approach. Kim et al. [20] and Marzec et al. [21] build a healthcare device with optimal power consumption navigation system for blind people through infrared sensors.

Salama et al. [22] and Sangpal et al. [23] Proposal of Smart Stick for blind People through Arduino and AIML. Elmannai et al. [24, 26, 28] proposed an extremely precise and consistent data synthesis Framework through sensor based 6technology for Supervisory the Visually Impaired people. G.F. Lourenço et al. [27] designed an assistive device for VIP based on user satisfaction. Rajapandian et al. [29] proposed an innovative tactic AID for blind, deaf and dumb people. [31] Foley A et al. presented a review on Technology for people, not disabilities: ensuring access and inclusion.

D. Dakopoulos et al. [32] and Topknot et al. [33] presented a survey about Wearable obstacle avoidance electronic travel aids for blind. A. Iqbal et al. [34] proposed a low cost AI vision system for visually impaired people. Chirayu Shah et al. [35] proposed a Tactile Feedback based Navigation System for VIP. L. Ran et al. [36], R. Imrie et al. [38] and Mountain G et al. [37] proposed an integrated indoor/outdoor blind navigation system and quality assistive technology services.

Scherer MJ et al. [39] done a review of Living in the State of Stuck: How Technology Impacts the Lives of

Table 2.1: The Contemporary Contributions and Proposed Framework VisiSense

Reference	Using IoT	Objects on Different Sides	Edge Computing	Cloud Computing	Using AI
Joshi, Rakesh Chandra, et al. [2]	✓			✓	
Saisubramanyam, Kaladindi et al. [3]	✓				
Vaidya, Sunit et al. [4]				✓	
Ramesh, Gopisetty et al. [5]	✓				
Farooq, Muhammad Siddique et al. [6]	✓	✓			
VisiSense	✓	✓	✓	✓	✓

People with Disabilities. Md. Milon Islam et al. [40] presented a review on Development of Walking Assistants for VIP. Sreenu Ponnada et al. [41] designed a smart system for detecting Manhole and Staircase through sensor based technology. Nur Syazreen Ahmad et al. [42] developed a Multi-sensor Obstacle Detection System via Model-based State-Feedback Control in Smart Cane Design for VIP. J. Bai et al. [43] simulated a smart system model through Virtual-Blind-Road Following-Based Wearable Navigation Device for Blind People.

In assistive technologies for the visually impaired, the review addresses deep learning models, IoT, sensor-based systems, and AI applications. Reliability, ease of use, real-time processing, and affordability are necessary for assistive devices to be effective. By going over several models, the review draws attention to the expanding usage of cutting-edge technologies like AI and IoT in navigation aids. VISISENSE, which combines IoT, edge computing, cloud computing, and AI, is supported in this environment. The need for seamless indoor-outdoor functionality and user-centered design is met by this integration, which also suggests a strong and adaptable system. VISISENSE is a comprehensive solution that has the potential to improve assistive devices for the visually impaired and close many gaps in current technologies. The review demonstrates how, if it preserves user accessibility and practicality in real-world settings, its holistic approach might offer a better user experience.

3. The Proposed Methodology. The VisiSense framework is a comprehensive system designed to assist visually impaired individuals in detecting objects in their surroundings. It encompasses several key components that work together to provide object detection and alert capabilities. The Handstick with Implant Sensors serves as the primary sensing unit, detecting obstacles in the environment and relaying this information to the system. The Visual Capturing and Transmission Unit, consisting of a mobile camera and a processing unit, captures visual data from the surroundings and performs initial processing tasks. The Edge Computing component plays a crucial role in object classification and detection. It includes a binary classifier, specifically the Logistic Regression with Iterative Learning, which classifies visuals as "known" or "new". Known visuals are forwarded to the Mobinet computer vision model, which leverages its learned knowledge to detect objects and provide object details.

The Cloud Server component houses a global computer vision model, specifically the R-CNN (Region-based Convolutional Neural Network) [8], trained on images from multiple similar sources of the VisiSense prototype. New visuals captured by the mobile camera are transferred to the cloud server for processing. The global computer vision model applies its knowledge and detects objects on these visuals, extracting object details.

The functional flow within the framework starts with the implant sensors detecting obstacles and alerting the mobile camera. The camera captures visuals, which are processed by the unit and classified by the binary classifier. If classified as known, the visuals are sent to the computer vision model for object detection, and the processing unit alerts the user through a hearing aid about the detected objects. If classified as new, the binary classifier alerts the processing unit, which transfers the new visuals to the cloud server. The global computer vision model in the cloud server detects objects on the new visuals and sends the object details back to the processing unit. The processing unit, along with the computer vision model in the Edge Computing component, alerts the user through the hearing aid, providing information about the detected objects. VisiSense prototype is a system designed to assist visually impaired individuals in detecting objects in their surroundings (figure 3.1).

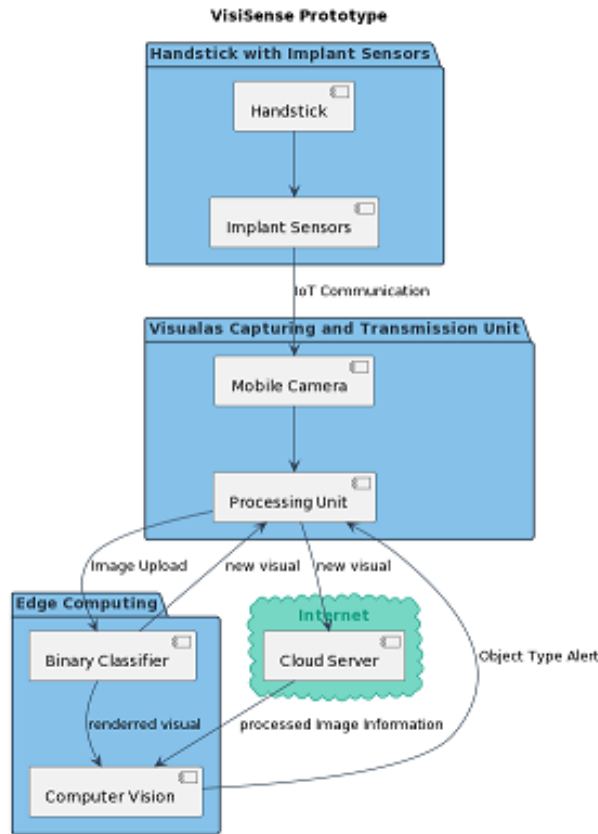


Fig. 3.1: The Flow Diagram of the Visisense Prototype Components

3.1. Functional Flow between Components. The implant sensors on the handstick detect obstacles in the environment and alert the mobile camera. The mobile camera captures visuals of the surroundings and transfers them to the processing unit. The processing unit forwards the visuals to the binary classifier for classification. If the visuals are classified as known, the binary classifier sends them to the computer vision model. The computer vision model detects objects on the known visuals and sends the object details back to the processing unit. The processing unit alerts the user through a hearing aid, providing information about the detected objects.

If the visuals are classified as new, the binary classifier alerts the processing unit about the new visuals and learns from them for future classification tasks. The processing unit transfers the new visuals to the cloud server.

The global computer vision model in the cloud server detects objects on the new visuals, extracting object details. The processing unit and Computer Vision Model of the Edge computing receives the object details from the cloud server and alerts the user through the hearing aid.

3.2. The Binary classification Strategy. Logistic regression [9] performs well and is simple for binary classification in the VisiSense prototype. Visual data are categorized as "already known / processed" (positive) or "new" using logistic regression. Logistic regression is especially suitable for the VisiSense system because it is well known for its effectiveness in handling binary classification problems. A critical function of the VisiSense framework is determining whether visual data has been processed or is new. It is well-suited for this task due to its probabilistic interpretations and adept handling of linearly separable data.

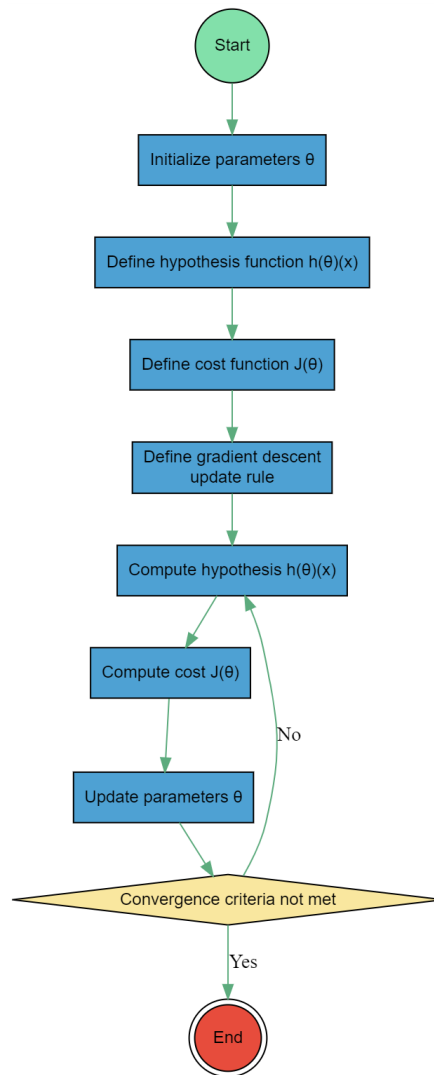


Fig. 3.2: Logistic regression algorithm

Data Preparation. Gather a labeled dataset of visual data, where each instance is associated with a binary label indicating whether it is "already known/processed" (positive) or "new" (negative).

Figure 3.2 describes for logistic regression algorithm with continuous learning: Represent each visual instance as a feature vector. Let X be the matrix of feature vectors, where each row corresponds to a visual instance, and let y be the vector of corresponding labels.

Model Initialization: Initialize the parameter vector θ to small random values. Let θ be a column vector with dimensions $(n + 1) \times 1$, where n is the number of features. The additional 1 is for the bias term.

Hypothesis Function: Define the hypothesis function $h\theta(x)$ that models the probability of a visual instance belonging to the positive class as: $h\theta(x) = \text{sigmod}(\theta^T x)$ where $\text{sigmod}(z)$ is the sigmoid function given by Eq 3.1

$$\text{sigmod}(z) = 1 / ((1 + \exp(-z))) \quad (3.1)$$

Cost Function. Define the cost function $J(\theta)$ that measures the discrepancy between the predicted probabilities and the actual labels as: Eq 3.2

$$J(\theta) = (1/m) * \text{sum}(-y * \log(h\theta(x)) - (1 - y) * \log(1 - h\theta(x))) \quad (3.2)$$

where m is the number of training examples.

Gradient Descent. Update the parameters iteratively using gradient descent to minimize the cost function is Eq 3.3.

$$J(\theta) : \theta = \theta - * (1/m) * X^T * (h\theta(x) - y) \quad (3.3)$$

Continuous Learning. During the system's operation, continuously monitor the predictions and feedback received from false positives, false negatives, and true negatives.

For each misclassification, update the parameter vector θ by further adjusting the parameters using the gradient descent update rule.

The logistic regression algorithm optimizes the parameters θ to maximize the likelihood of the observed labels given the input visual features. By iteratively updating the parameters based on the gradient descent algorithm and continuously learning from misclassifications, the model improves its ability to classify "already known/processed" and "new" visuals more accurately.

Mathematical Model. The logistic regression algorithm can be formulated as follows.

By iteratively updating the parameters using gradient descent and continuously learning from misclassifications, the logistic regression model in the VisiSense prototype adapts to new visual data, improving its classification performance over time.

1. Hypothesis Function: $h\theta(x) = \text{sigmoid}(\theta^T x)$
2. Cost Function: $J(\theta) = (1/m) * \text{sum}(-y * \log(h\theta(x)) - (1 - y) * \log(1 - h\theta(x)))$
3. Gradient Descent Update Rule: $\theta := \theta - * (1/m) * X^T * (h\theta(x) - y)$

In the above equations, X represents the matrix of feature vectors, y represents the vector of labels, θ represents the parameter vector, and $*$ represents the learning rate. The sigmoid function is given by Eq 3.4.

$$\text{sigmoid}(z) = (1/((1 + \exp(-z)))) \quad (3.4)$$

By iteratively updating the parameters using gradient descent and continuously learning from misclassifications, the logistic regression model in the VisiSense prototype adapts to new visual data, improving its classification performance over time.

3.3. Computer Vision Model as Edge Computing Service . The constrained resource environment of the VisiSense prototype is ideal for MobileNet's [10] compact convolutional neural network (CNN) architecture. Without sacrificing precision, depth-wise separable convolutions reduce the number of parameters and computational complexity. MobileNet allows for customization based on the size of the model and the required computational power. Real-time processing and user alerts are feasible due to the system's lightweight design. Transfer learning improves performance with limited labeled data in MobileNet. Because of its efficacy, adaptability, and real-time processing, MobileNet is ideal for the VisiSense prototype's efficient and accurate object recognition, which improves the perception and safety of the blind and visually impaired (see Figure 3.3).

The steps involved in object detection model follows:

Step 1. Model Initialization and Configuration. Load the MobileNet architecture with a specific configuration suitable for the VisiSense prototype, considering the balance between model size, accuracy, and computational efficiency.

Step 2. Input Preprocessing. Preprocess the visual data captured by the wearable mobile camera to meet the input requirements of MobileNet. This typically involves resizing the images to the input dimensions expected by MobileNet and normalizing the pixel values.

Step 3. Convolutional Layers. Pass the preprocessed visual data through the convolutional layers of MobileNet, which consist of depth-wise separable convolutions. For each layer, compute the weighted sum of inputs and apply a non-linear activation function, such as ReLU: a. Depth-wise Convolution:

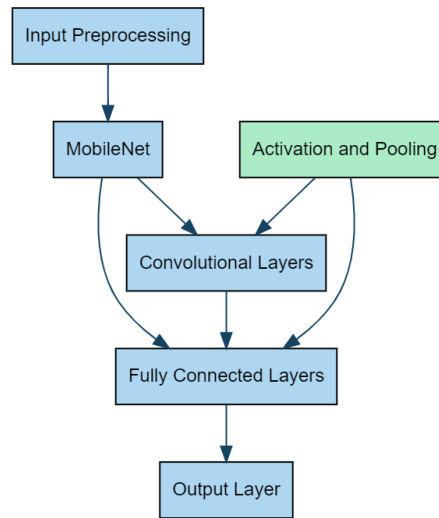


Fig. 3.3: The MobileNet for VisiSense

1. Perform depth-wise convolution for each channel i : $z_i = W_i * x$: Apply the activation function(e.g., $ReLU$): $a_i = ReLU(z_i)$.
- 2.
3. Point-wise Convolution:
 - (a) Perform point-wise convolution across channels: $z = W * a$
 - (b) Apply the activation function: $a = ReLU(z)$

Step 4: Activation and Pooling. Apply non-linear activation functions, such as $ReLU$, after each convolutional layer to introduce non-linearity into the model and capture complex features. Perform down sampling using pooling layers, such as max pooling or average pooling, to reduce the spatial dimensions of the feature maps while preserving important information.

Step 5: Fully Connected Layers. Flatten the output feature maps from the convolutional layers into a 1D vector. Connect the flattened vector to one or more fully connected layers, which process the extracted features and capture higher-level representations. Apply an activation function, such as $ReLU$, to the outputs of the fully connected layers.

Step 6: Output Layer. Connect the output of the fully connected layers to the final output layer. Apply an appropriate activation function based on the task, such as softmax for multi-class classification or sigmoid for binary classification. Obtain the predicted probabilities or class labels from the output layer.

3.4. Mathematical Model. Let's denote the input visual data as x , the output probabilities as y_{pred} , and the parameters of MobileNet as W_i , W and W_{fc} .

Convolutional Layers. For each layer, compute the weighted sum of inputs and apply the activation function:

1. Depth-wise Convolution:
 - (a) Perform depth-wise convolution for each channel i : $z_i = W_i * x$.
 - (b) Apply the activation function (e.g., $ReLU$): $a_i = ReLU(z_i)$.
2. Point-wise Convolution:
 - (a) Perform point-wise convolution across channels: $z = W * a$.
 - (b) Apply the activation function: $a = ReLU(z)$.

Activation and Pooling. Apply non-linear activation functions, such as $ReLU$, after each convolutional layer. Perform down sampling using pooling layers, such as max pooling or average pooling.

Fully Connected Layers.

1. Flatten the output feature maps into a 1D vector: $a_{flat} = Flatten(a)$.

2. Compute the weighted sum of the flattened vector: $z_{fc} = W_{fc} * a_{flat}$.
3. Apply the activation function: $a_{fc} = \text{ReLU}(z_{fc})$.

Output Layer.

1. Compute the weighted sum of the fully connected layer output: $z_{out} = W_{out} * a_{fc}$.
2. Apply the appropriate activation function based on the task (e.g., softmax for multi-class classification or sigmoid for binary classification): $y_{pred} = \text{activation}(z_{out})$.

The parameters W_i , W , W_{fc} , and W_{out} represent the weights of the respective layers, and ReLU is the activation function used throughout the model.

By following these algorithmic steps and applying the mathematical model, MobileNet within the VisiSense prototype efficiently processes visual data captured by the wearable mobile camera. It produces predicted probabilities or class labels, indicating the presence and type of objects within the captured visual data. The lightweight and efficient nature of MobileNet allows for real-time object identification on edge computing devices, enhancing the accuracy and responsiveness of the VisiSense system for the benefit of visually impaired individuals.

3.5. Global Computer Vision Model as Cloud Service. The R-CNN (Region-based Convolutional Neural Network) [8] algorithm holds significant significance as the Global Computer Vision Model in the VisiSense prototype. Figure 3.4 R-CNN provides accurate and robust object detection by employing a two-stage approach, with a region proposal network (RPN) [11] generating potential object proposals and a subsequent classification network performing object classification. This two-stage process ensures precise localization and classification of objects in visual data. The RPN efficiently generates region proposals using anchor boxes and convolutional features, enabling the model to handle objects of various sizes and aspect ratios. Additionally, R-CNN offers flexibility, adaptability, and real-time processing capabilities. Its popularity in the computer vision community provides access to pre-trained models, resources for fine-tuning on specific datasets, and an active support community, making it a valuable choice for enhancing object detection in the VisiSense prototype and empowering visually impaired individuals with improved situational awareness. R-CNN (Region-based Convolutional Neural Network) is a two-stage algorithm for object detection. It consists of a region proposal network (RPN) and a subsequent classification network. Here's a high-level description of the R-CNN algorithm and its mathematical model.

3.5.1. Region Proposal Network (RPN). The RPN generates region proposals by sliding a small network (typically a convolutional neural network) over the input image. It predicts potential object bounding boxes, known as region of interests (RoIs), and their associated objectness scores. The RPN operates on a set of anchor boxes of different scales and aspect ratios, which act as reference boxes for proposing potential object locations. The RPN predicts the objectness score (probability of an anchor containing an object) and the coordinates (x , y , width, height) of the bounding boxes relative to the anchors.

Inputs: Image (I)

Anchor boxes: $A = \{a_1, a_2, \dots, a_n\}$ (n anchor boxes with different scales and aspect ratios)

Output: Region proposals (R), Objectness scores (O)

RPN predicts the objectness score for each anchor box:

$O = \{o_1, o_2, \dots, o_n\}$ (where o_i is the objectness score of the i -th anchor)

RPN predicts the coordinates for each anchor box: $R = \{r_1, r_2, \dots, r_n\}$ (where $r_i = (dx_i, dy_i, dw_i, dh_i)$ represents the bounding box coordinates relative to the i -th anchor)

3.5.2. Classification Network. The classification network takes the proposed RoIs from the RPN and extracts fixed-sized feature vectors using a shared convolutional backbone network.

These feature vectors are then fed into a classifier (typically a fully connected network) to classify the object contained within each proposed RoI. The classifier predicts the object class probabilities and, optionally, refines the bounding box coordinates of the proposed RoIs.

Inputs: Proposed RoIs (R')

Output: Class probabilities (P), Refined bounding box coordinates (B)

The classification network takes the proposed RoIs R' and extracts feature vectors using a shared convolutional backbone network.

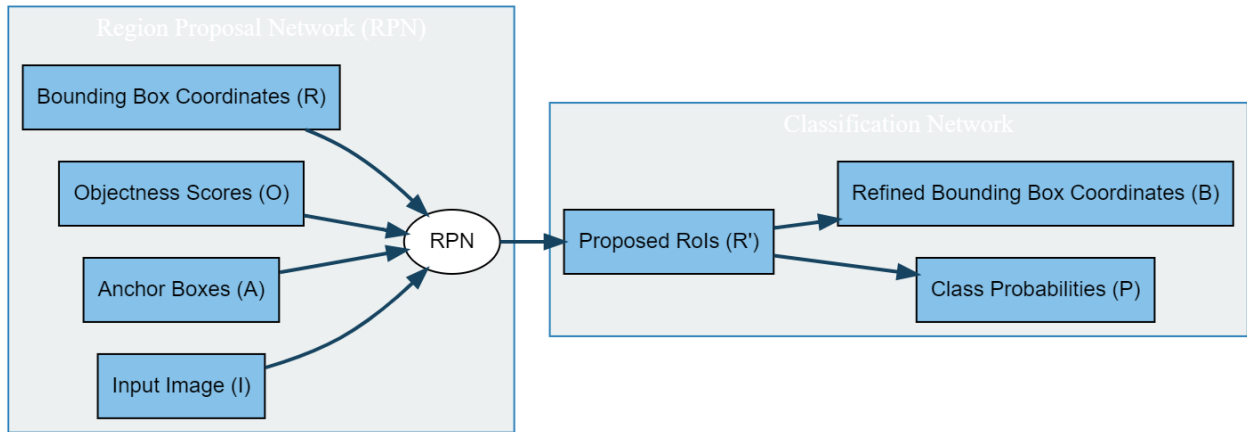


Fig. 3.4: R-CNN for global computer vision model

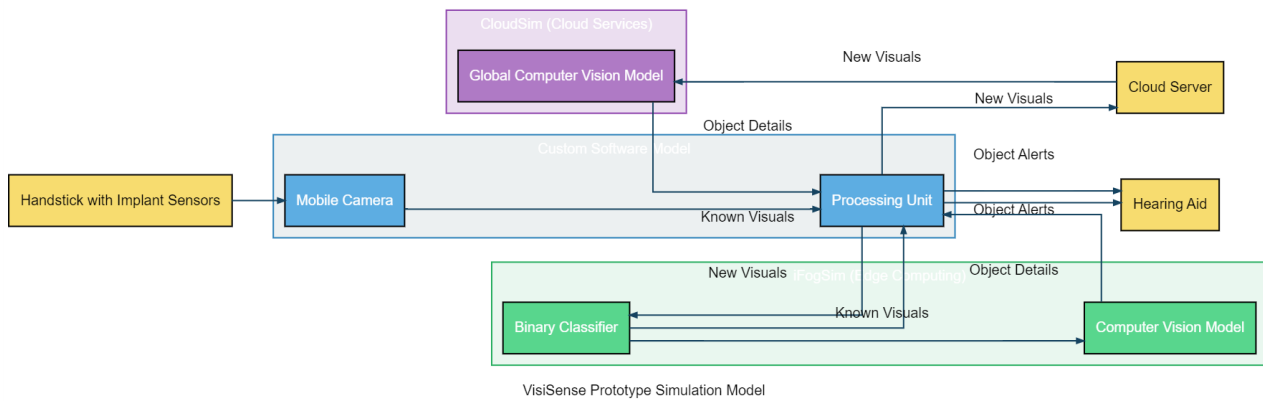


Fig. 4.1: VisiSense prototype simulation model

The feature vectors are fed into a classifier to predict the class **probabilities**: $P = \{p_1, p_2, \dots, p_m\}$ (where p_i is the probability of the i -th RoI belonging to a specific object class).

The classifier may also refine the bounding box coordinates of the proposed RoIs: $B = \{b_1, b_2, \dots, b_m\}$ (where $b_i = (dx'_i, dy'_i, dw'_i, dh'_i)$ represents the refined bounding box coordinates of the i -th RoI).

4. Experimental Study. The Simulation Model depicted in figure 4.1 is defined to test the performance of VisiSense. The simulation model that integrates Custom Software Model, iFogSim [12] for Edge Computing, and CloudSim [13] for Cloud Services. The Custom Software Model comprises the "Mobile Camera" and "Processing Unit" components. The "Mobile Camera" captures visuals from the environment, which are then processed by the "Processing Unit." The processed visuals are further passed to the "Binary Classifier" for classification, and the resulting visual classification is sent to the "Hearing Aid" for generating object alerts.

In Figure 4.1, the iFogSim module represents the Edge Computing component of the VisiSense Prototype Simulation Model, playing a critical role by enabling low-latency, real-time processing crucial for assisting visually impaired users. Edge Computing's proximity to data sources allows for rapid processing and decision-making, enhancing user safety through immediate object recognition and alerts. It reduces the system's reliance on cloud connectivity, ensuring consistent performance even with variable network quality.

This local processing capability not only optimizes bandwidth and energy efficiency, crucial for handheld assistive devices, but also bolsters privacy by minimizing the transmission of sensitive data. Furthermore, the Edge Computing layer, through its Binary Classifier, provides a scalable and adaptive mechanism that learns from new visual data, thereby continuously refining the system's accuracy and responsiveness. Overall, Edge Computing significantly contributes to the VisiSense system's robustness, making it an indispensable element in the proposed research work for real-time navigational support to visually impaired individuals.

The "Binary Classifier," a component of the iFogSim package, examines the images received from the "Processing Unit" to determine whether they are old or new images. The "Computer Vision Model" then processes known images for object detection, sending the resulting object details back to the "Processing Unit."

When there are new visuals, the "Binary Classifier" alerts the "Processing Unit" to their presence. The "Processing Unit" then sends these images to the "Global Computer Vision Model" inside the CloudSim package, which carries out object detection. The "Processing Unit" and the "Computer Vision Model" are both given access to the object details discovered from the fresh visuals.

The "Processing Unit" creates object alerts by combining the object details obtained from the "Global Computer Vision Model" and the "Computer Vision Model." Users can learn more about the discovered objects thanks to the "Hearing Aid [14]," which receives these alerts.

4.1. Performance Analysis. By examining important performance indicators like object detection accuracy, processing speed, resource utilization, energy consumption, latency, and false positive rate, this section evaluates VisiSense. VisiSense proved its superiority and efficacy. Accurate object detection, real-time data processing, resource optimization, energy efficiency, and low false positive rates are just a few of its strong points. The section also contrasts VisiSense with other current models to show its benefits and potential influence on the assistive technology market. The performance analysis aids in decision-making and ongoing development while showcasing VisiSense's capabilities and directing improvements. In the end, VisiSense shows promise in enhancing accessibility and quality of life for people with visual impairments.

4.1.1. Object Detection Accuracy. This metric evaluates the precision of VisiSense's object detection. It evaluates the precision of object identification in the recorded images. It's crucial to evaluate both overall accuracy and object type accuracy in order to gauge the system's object detection capabilities.

The Figure 4.2, Table 4.1 shows the results for "Object Detection Accuracy". The values in Mbps in the table compare the object detection accuracy of VisiSense, Smart Stick [6], and Smart Navigation [7] under various load conditions.

VisiSense achieves the highest object detection accuracy of 99% at a load of 2 Mbps. The Smart Stick comes in second place at 91% and Smart Navigation comes in at 84%. VisiSense maintains a high accuracy of 97% as the load increases to 4 Mbps, while the Smart Stick and Smart Navigation achieve 90% and 83% accuracy, respectively. VisiSense consistently outperforms the other solutions in terms of object detection accuracy as the load increases, and this trend continues.

VisiSense achieves 94% accuracy at 10 Mbps, compared to 88% and 82% accuracy for the Smart Stick and Smart Navigation, respectively. As the load grows, accuracy gradually declines, but VisiSense consistently keeps its accuracy higher than the other two solutions.

4.1.2. Processing Time. In this metric, VisiSense analyzes visual data and identifies objects. It takes time to capture visuals and to perform preprocessing, classification, and object detection. People with visual impairments require quicker responses and real-time performance Figure 4.3.

Table 4.2 shows the processing time in milliseconds (ms) for VisiSense, Smart Stick, and Smart Navigation under various load conditions, denoted by the values in Mbps.

VisiSense displays the quickest processing time of 17 ms at a load of 2 Mbps, followed by the Smart Stick at 26 ms and Smart Navigation at 32 ms. The processing time for VisiSense slightly increases to 19 ms at 4 Mbps load, while the processing times for the Smart Stick and Smart Navigation are 28 ms and 34 ms, respectively. As the load continues to rise, this pattern persists.

VisiSense processes data at a rate of 24 ms at 10 Mbps, compared to 35 ms and 40 ms for the Smart Stick and Smart Navigation, respectively. As the load increases, processing time gradually increases, but VisiSense consistently shows faster processing times than the other two models. Overall, these findings suggest that

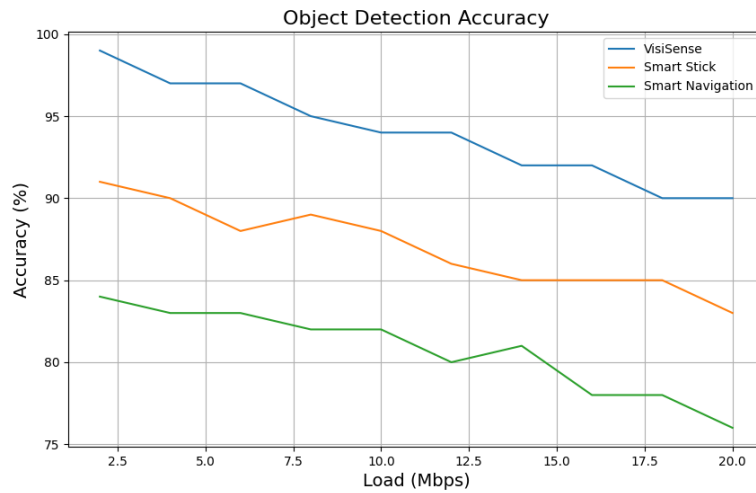


Fig. 4.2: The Object detection accuracy for VisiSense, Smart Stick, and Smart Navigation

Table 4.1: Comparison of Performance Metrics at Different Loads

Load (Mbps)	VisiSense	Smart Stick	Smart Navigation
2	99%	91%	84%
4	97%	90%	83%
6	97%	88%	83%
8	95%	89%	82%
10	94%	88%	82%
12	94%	86%	80%
14	92%	85%	81%
16	92%	85%	78%
18	90%	85%	78%
20	90%	83%	76%

VisiSense processes information more quickly than the Smart Stick and Smart Navigation under various load conditions. This suggests that VisiSense can process information more quickly and effectively, leading to faster response times. Although slightly slower than VisiSense, the Smart Stick and Smart Navigation also display acceptable processing times.

These results highlight how quickly and effectively VisiSense processes information, making it a promising assistive technology for people who are blind or visually impaired. The system is more usable overall because of the quicker object detection and recognition and more seamless user experience that results from the shorter processing time.

4.1.3. Resource Utilization. This metric evaluates the efficiency with which VisiSense makes use of the CPU, memory, and network bandwidth. It assesses the effectiveness of visual processing and resource allocation. Cost and efficiency are both increased by using fewer resources.

The Figure 4.4, Table 4.3 representation of the resource utilization for VisiSense, Smart Stick, and Smart Navigation in percentages (%) for various load conditions.

VisiSense shows a resource utilization of 41% at a load of 2 Mbps, compared to resource utilizations of 50% and 42% for the Smart Stick and Smart Navigation, respectively. VisiSense's resource utilization rises to 47% at 4 Mbps, while the resource utilization for the Smart Stick and Smart Navigation is 57% and 54%,

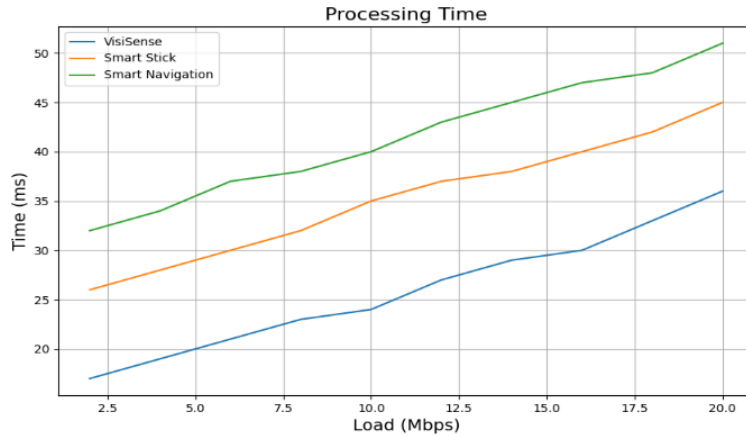


Fig. 4.3: The processing time in milliseconds (ms) for VisiSense, Smart Stick, and Smart Navigation

Table 4.2: Response Time Analysis Across Different Systems

Load (Mbps)	VisiSense	Smart Stick	Smart Navigation
2	17	26	32
4	19	28	34
6	21	30	37
8	23	32	38
10	24	35	40
12	27	37	43
14	29	38	45
16	30	40	47
18	33	42	48
20	36	45	51

respectively. As the load continues to rise, this pattern persists.

VisiSense uses 58% of the resources at 10 Mbps, while the Smart Stick and Smart Navigation use 68% and 66% of the resources, respectively. As the load increases, resource utilization gradually rises, with VisiSense consistently showing lower resource utilization than the other two models.

These findings collectively show that VisiSense uses resources more sparingly than the Smart Stick and Smart Navigation under various load conditions. This suggests that VisiSense makes better use of system resources through optimization, which enhances performance and stability. While slightly higher than VisiSense, the Smart Stick and Smart Navigation also show acceptable resource usage.

These results demonstrate VisiSense’s effectiveness and resourcefulness, making it a viable and useful assistive technology for people who are blind or visually impaired. Lower resource utilization guarantees improved system performance and scalability, enabling more seamless operation and better user experience.

4.1.4. Energy Consumption. This metric assesses the energy required for processing visual data by VisiSense. It takes into account the processing power, edge computing hardware, cloud servers, and mobile camera’s energy consumption.

Energy optimization reduces system energy requirements and increases the battery life of mobile devices Figure 4.5, Table 4.4. The provided table shows VisiSense, Smart Stick, and Smart Navigation energy consumption in millijoules for various load conditions, represented by the values in Mbps.

VisiSense uses 127 millijoules of energy at a load of 2 Mbps, compared to 182 millijoules and 151 millijoules

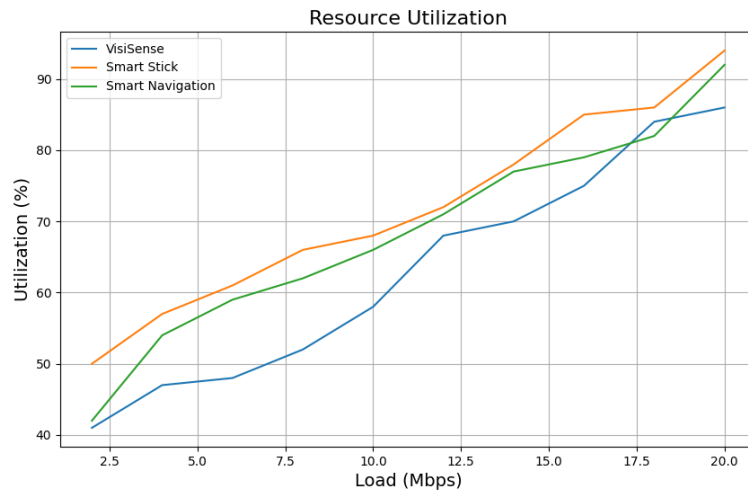


Fig. 4.4: Resource utilization for VisiSense, Smart Stick, and Smart Navigation in percentages (%) for various load conditions

Table 4.3: Comparison of Load Distribution Efficiency

Load (Mbps)	VisiSense	Smart Stick	Smart Navigation
2	41%	50%	42%
4	47%	57%	54%
6	48%	61%	59%
8	52%	66%	62%
10	58%	68%	66%
12	68%	72%	71%
14	70%	78%	77%
16	75%	85%	79%
18	84%	86%	82%
20	86%	94%	92%

used by the Smart Stick and Smart Navigation, respectively. VisiSense's energy consumption increases to 152 millijoules at 4 Mbps load, while the Smart Stick and Smart Navigation use 206 millijoules and 182 millijoules of energy, respectively. As the load rises even higher, this pattern persists.

VisiSense uses 228 millijoules of energy at 10 Mbps, compared to 280 millijoules and 257 millijoules for the Smart Stick and Smart Navigation, respectively. As the load increases, the energy consumption rises gradually, with VisiSense consistently showing lower energy consumption than the other two models.

These findings show that VisiSense uses considerably less energy than the Smart Stick and Smart Navigation under various load conditions. This implies that VisiSense is more energy-efficient and optimized, resulting in lower power usage and longer battery life. Both the Smart Stick and Smart Navigation exhibit reasonable energy usage, albeit a little bit more so than VisiSense.

These findings highlight VisiSense's sustainability and energy efficiency, making it an appealing assistive technology for people who are blind or visually impaired. Because of the lower energy consumption, devices can run for longer periods of time and require fewer battery changes or recharges overall. In the end, VisiSense provides a dependable and cost-effective solution to increase the independence and mobility of people with visual impairments.

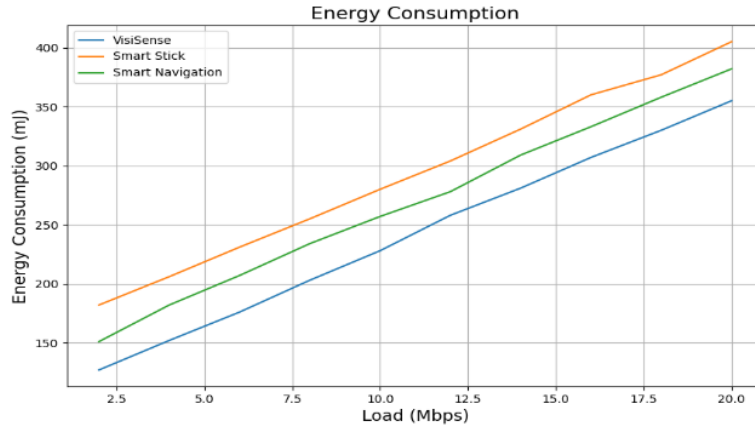


Fig. 4.5: Energy Consumption for VisiSense, Smart Stick, and Smart Navigation

Table 4.4: Performance Metrics Across Different Load Levels

Load (Mbps)	VisiSense	Smart Stick	Smart Navigation
2	127	182	151
4	152	206	182
6	176	231	207
8	203	255	234
10	228	280	257
12	258	304	278
14	281	331	309
16	307	360	333
18	330	377	358
20	355	405	382

The VisiSense system is compared at a 10Mbps data rate. In order to balance quick data transfer with the bandwidth available in locations where such a system could be used, this data rate was selected. If you want to know how well the system works in scenarios that users might encounter frequently, the 10Mbps rate is a good place to start. Figure 4.5 shows the processing efficiency of VisiSense at 10Mbps, showing how fast it can process large amounts of data in real time—a critical feature for applications requiring users of assistive technology. The system’s response time is displayed by this rate, which is helpful in circumstances that change quickly. Figure 4.6 displays system optimization and resource utilization at 10Mbps. Compared to similar systems, ViaSense uses CPU, memory, and network bandwidth more efficiently. Maintaining the system’s smooth and efficient operation depends heavily on resource allocation efficiency. Figure 4.7 illustrates the system’s lower energy consumption compared to other models for VisiSense at 10Mbps. Energy efficiency improves user experience for devices that are used for extended periods of time without being charged.

4.1.5. Latency. This metric assesses the communication overhead of VisiSense’s handstick, mobile camera, processing unit, edge computing devices, and cloud servers.

The provided Figure 4.6, Table 4.5 displays the VisiSense, Smart Stick, and Smart Navigation latency in milliseconds for various load conditions, represented by the values in Mbps. VisiSense exhibits a latency of 13 ms at a load of 2 Mbps, compared to latencies of 33 ms and 24 ms for the Smart Stick and Smart Navigation, respectively. The latency for VisiSense increases to 16 ms at 4 Mbps of load, while the latencies for the Smart Stick and Smart Navigation are 42 ms and 29 ms, respectively. As the load continues to rise, this pattern

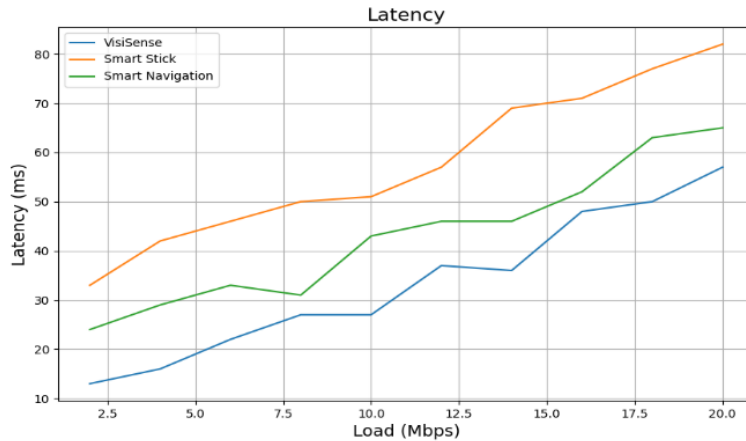


Fig. 4.6: The VisiSense, Smart Stick, and Smart Navigation latency in milliseconds for various load conditions

Table 4.5: Comparative Analysis of Load Distribution

Load (Mbps)	VisiSense	Smart Stick	Smart Navigation
2	13	33	24
4	16	42	29
6	22	46	33
8	27	50	31
10	27	51	43
12	37	57	46
14	36	69	46
16	48	71	52
18	50	77	63
20	57	82	65

persists.

The Smart Stick and Smart Navigation have latencies of 51 milliseconds and 43 milliseconds, respectively, while VisiSense has a latency of 27 milliseconds at 10 Mbps. As the load rises, the latency gradually increases, but VisiSense consistently maintains a lower latency than the other two models.

In conclusion, these results show that VisiSense has lower latency than the Smart Stick and Smart Navigation under various load conditions. This suggests that VisiSense enables real-time interaction and navigation for people who are blind or visually impaired by offering quicker response times and more effective data processing. Even though they have slightly longer latencies than VisiSense, the Smart Stick and Smart Navigation also exhibit acceptable latencies. These findings demonstrate VisiSense’s effectiveness as an assistive technology for people who are blind or visually impaired. Lower latency guarantees accurate and timely information delivery, improving user experience and facilitating fluid navigation. In the end, VisiSense provides a dependable and adaptable solution to increase the independence and mobility of people with visual impairments.

4.1.6. False Positive Rate. These metrics gauge the frequency of false positives and negatives in VisiSense’s object detection. When an object is wrongly classified or overlooked, false positives and negatives are produced. Improved system dependability and trustworthiness result from fewer false positives and negatives. The Figure 4.7, Table 4.6 gives the percentage false positive rates for VisiSense, Smart Stick, and Smart Navigation under various load conditions, represented by the values in Mbps.

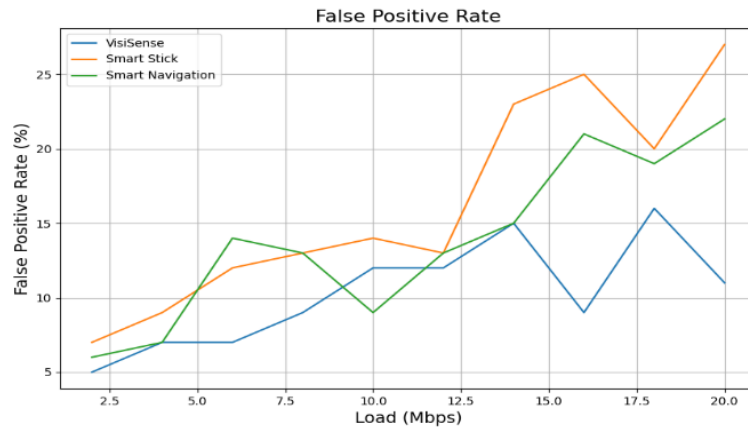


Fig. 4.7: The False positive rates for VisiSense, Smart Stick, and Smart Navigation under various load conditions

Table 4.6: Performance Metrics at Various Network Loads

Load (Mbps)	VisiSense	Smart Stick	Smart Navigation
2	5	7	6
4	7	9	7
6	7	12	14
8	9	13	13
10	12	14	9
12	12	13	13
14	15	23	15
16	9	25	21
18	16	20	19
20	11	27	22

The false positive rates of VisiSense, the Smart Stick, and Smart Navigation were evaluated across different load conditions. At a load of 2 Mbps, VisiSense exhibited a 5% false positive rate, while the Smart Stick and Smart Navigation had rates of 7% and 6%, respectively. As the load increased to 4 Mbps, VisiSense's false positive rate rose to 7%, with the Smart Stick and Smart Navigation showing rates of 9% and 7%. As the load increased further, this trend persisted at varying rates.

VisiSense's false positive rate at 10 Mbps was 12%, which was lower than that of the Smart Stick (14%) and the Smart Navigation (9%). False positive rates also fluctuated between models and loads.

The comparison as a whole demonstrates that VisiSense maintains a low false positive rate across a wide range of workloads. This shows that it detects objects more precisely and with fewer false positives than competing models. The Smart Stick and Smart Navigation performed adequately in object detection, despite having slightly higher false positive rates. Maintaining a low false positive rate is a key component of VisiSense's reliability and usability as an assistive technology, allowing it to provide visually impaired users with a trustworthy and reliable navigation solution. As a result of its ability to reduce false positives, VisiSense helps people with visual impairments move around more securely and with greater assurance.

Using the false positive rate, Figure 4.7 determines the object detection precision of the VisiSense system under different network load scenarios. False positives are situations in which the system incorrectly recognizes an object in visual data in the graph. This rate is critical to system reliability because a lower false positive rate denotes higher accuracy and reliability. VisiSense performs better than Smart Stick and Smart Navigation,

as evidenced by its false positive rate being lower under various loads.

The ratio of false positive detections to the total number of objects—true and false—is known as the false positive rate. This is a typical performance metric for object detection systems. By maintaining a lower rate of false positives, the graph in Figure 4.7 illustrates the dependability of the system and shows how effective VisiSense is at assisting visually impaired users with navigation. For users who depend on the system for safe navigation, the graph illustrates the model’s accuracy in object detection.

5. Conclusion. Quantitative gains over current models demonstrate that VisiSense represents a significant advancement in assistive technologies for the visually impaired. To enhance navigational aids, the system integrates cutting-edge technologies such as edge computing, cloud computing, Internet of Things, and artificial intelligence. With a 2Mbps object detection accuracy of 99%, VisiSense outperforms both Smart Stick (91%), and Smart Navigation (84%). VisiSense processes in 17 milliseconds, while the Smart Stick and Smart Navigation take 26 and 32 milliseconds, respectively. VisiSense exhibits superior efficiency by using only 41% of its available bandwidth at 2Mbps, while its competitors use 50% and 42%. The system uses 228 millijoules of energy at 10Mbps, which is significantly less than the 280 and 257 millijoules used by the Smart Stick and Smart Navigation, respectively. These figures show not only how much VisiSense has improved, but also how dependable it is—a feature that makes it a dependable tool for users. According to the study, VisiSense can significantly enhance the security, autonomy, and quality of life of people with visual impairments. With continued research and development, VisiSense has the potential to transform assistive navigation and empower people who are visually impaired.

REFERENCES

- [1] MD MILON ISLAM, ET AL., “Developing walking assistants for visually impaired people: A review,” *IEEE Sensors Journal*, 19.8 (2019): 2814-2828.
- [2] RAKESH CHANDRA JOSHI, ET AL., “Efficient multi-object detection and smart navigation using artificial intelligence for visually impaired people,” *Entropy*, 22.9 (2020): 941.
- [3] KALADINDI SAISUBRAMANYAM, AND A. JHANSI RANI, “IOT Based Smart Walking Stick for Blind or Old Aged People with GPS Tracking Location,” *Journal of Engineering Science*, 11.1 (2020): 501-505.
- [4] SUNIT VAIDYA, ET AL., “Real-time object detection for visually challenged people,” in *2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS)*, IEEE, 2020.
- [5] GOPISETTY RAMESH, NARENDRA B. MUSTARE, AND K. UDAY3 C. PRAMOD KUMAR, “Development of E-Stick for Blind Persons using IoT,” *I-Manager’s Journal on Embedded Systems*, 10.2 (2022).
- [6] MUHAMMAD SIDDIQUE FAROOQ, ET AL., “IoT Enabled Intelligent Stick for Visually Impaired People for Obstacle Recognition,” *Sensors*, 22.22 (2022): 8914.
- [7] BINEETH KURIAKOSE, RAJU SHRESTHA, AND FRODE EIKA SANDNES, “Tools and technologies for blind and visually impaired navigation support: a review,” *IETE Technical Review*, 39.1 (2022): 3-18.
- [8] ROSS GIRSHICK, ET AL., “Region-based convolutional networks for accurate object detection and segmentation,” *IEEE transactions on pattern analysis and machine intelligence*, 38.1 (2015): 142-158.
- [9] RAYMOND E. WRIGHT, “Logistic regression,” (1995).
- [10] HONG-YEN CHEN, AND CHUNG-YEN SU, “An enhanced hybrid MobileNet,” in *2018 9th International Conference on Awareness Science and Technology (iCAST)*, IEEE, 2018.
- [11] S. REN, ET AL., “Faster r-cnn: Towards real-time object detection with region proposal networks,” *Advances in Neural Information Processing Systems*, 28 (2015).
- [12] H. GUPTA, ET AL., “iFogSim: A toolkit for modeling and simulation of resource management techniques in the Internet of Things, Edge and Fog computing environments,” *Software: Practice and Experience*, 47.9 (2017): 1275-1296.
- [13] R. KUMAR, AND G. SAHOO, “Cloud computing simulation using CloudSim,” *arXiv preprint arXiv:1403.3253*, (2014).
- [14] B. EDWARDS, “The future of hearing aid technology,” *Trends in Amplification*, 11.1 (2007): 31-45.
- [15] P. BHASHA, DR. T. P. KUMAR, AND DR. K. K. BASEER, “A Simple and Effective Electronic Stick to Detect Obstacles for Visually Impaired People through Sensor Technology,” *Journal of Advanced Research in Dynamical & Control Systems*, 12, Issue-06, (2020).
- [16] Y. KIM, H. JEONG, J.-D. CHO, AND J. SHIN, “Construction of a Soundscape-Based Media Art Exhibition to Improve User Appreciation Experience by Using Deep Neural Networks,” *Electronics*, 2021, 10, 1170.
- [17] M. W. RAHMAN, ET AL., “The Architectural Design of Smart Blind Assistant Using IoT with Deep Learning Paradigm,” *Internet of Things*, 13, (2021), Article ID: 100344.
- [18] R. BHAVANI, ET AL., “Development of a Smart Walking Stick for Visually Impaired People,” *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, 12, (2021): 999-1005.
- [19] A. ASHRAF, ET AL., “IoT Empowered Smart Stick Assistance for Visually Impaired People,” *International Journal of Scientific & Technology Research*, 9, (2020): 356-360.

- [20] T. H. KIM, H. KOO, AND E. HAN, "Healthcare Use among Persons with Visual Impairment: Difference-in-Difference Estimations," *Disability and Health Journal*, 12(2) (2019): 302–9.
- [21] P. MARZEC AND A. KOS, "Low energy precise navigation system for the blind with infrared sensors," in *2019 MIXDES-26th International Conference Mixed Design of Integrated Circuits and Systems*, IEEE, 2019, pp. 394–397.
- [22] R. SALAMA AND A. AYOUB, "Design of Smart Stick for Visually Impaired People Using Arduino," *New Trends and Issues Proceedings on Humanities and Social Sciences*, 6 (2019): 58–71.
- [23] R. SANGPAL, T. GAWAND, S. VAYKAR AND N. MADHAVI, "JARVIS: An interpretation of AIML with integration of gTTS and Python," in *2019 2nd International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICT)*, Kannur, Kerala, India, 2019, pp. 486–489.
- [24] W. M. ELMANNAI AND K. M. ELLEITHY, "A Highly Accurate and Reliable Data Fusion Framework for Guiding the Visually Impaired," *IEEE Access*, vol. 6, 2018, pp. 33029–33054.
- [25] K. PATIL, Q. JAWADWALA, AND F.-L. C. SHU, "Design and construction of electronic Aid for visually impaired people," *IEEE Transactions on Human-Machine Systems*, vol. 48, no. 2, 2018, pp. 172–182.
- [26] W. ELMANNAI AND K. ELLEITHY, "Sensor-based assistive devices for visually-impaired people: current status, challenges and future directions," *Sensors*, 17(3) (2017): 1–42.
- [27] G. F. LOURENÇO, A. C. HONÓRIO, AND A. O. FIGUEIREDO, "Satisfaction of use of assistive devices for orientation and mobility of adults with visual impairment," *Journal of Occupational Therapy, University of São Paulo*, 28(3) (2017): 340–348.
- [28] W. ELMANNAI AND K. ELLEITHY, "Sensor-based assistive devices for visually-impaired people," *IEEE Access*, vol. 17(3), 2017, pp. 565–600.
- [29] B. RAJAPANDIAN, V. HARINI, D. RAKSHA, AND V. SANGEETHA, "A novel approach as an AID for blind, deaf and dumb people," *IEEE Access*, vol. 8, 2017, pp. 403–408.
- [30] S. MAIDENBAUM, S. ABOUD, AND A. AMEDI, "Sensory substitution: closing the gap between basic research and widespread practical visual rehabilitation," *Neuroscience & Biobehavioral Reviews*, 41 (2014): 3–15.
- [31] S. MAIDENBAUM, S. ABOUD, AND A. AMEDI, "Sensory substitution: closing the gap between basic research and widespread practical visual rehabilitation," *Neuroscience & Biobehavioral Reviews*, 41 (2014): 3–15.
- [32] A. FOLEY AND B. A. FERRI, "Technology for people, not disabilities: ensuring access and inclusion," *Journal of Research in Special Educational Needs*, 2012: 192–200.
- [33] D. DAKOPOULOS AND N. G. BOURBAKIS, "Wearable obstacle avoidance electronic travel aids for blind: a survey," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 40(1) (2010): 25–35.
- [34] V. TIPONUT ET AL., "Work directions and new results in electronic travel aids for blind and visually impaired people," *Proceedings of the 14th WSEAS International Conference on Systems*, Vol I, 2010, pp. 5346–5463.
- [35] A. IQBAL, U. FAROOQ, H. MAHMOOD, AND M. U. ASAD, "A low cost artificial vision system for visually impaired people," in *2009 Second International Conference on Computer and Electrical Engineering*, December 2009, pp. 474–479.
- [36] C. SHAH, M. BOUZIT, M. YOUSSEF AND L. VASQUEZ, "Evaluation of RU-Netra - Tactile Feedback Navigation System For The Visually Impaired," *2006 International Workshop on Virtual Rehabilitation*, 2006, pp. 72–77.
- [37] L. RAN, S. HELAL, AND S. MOORE, "Drishti: an integrated indoor/outdoor blind navigation system and service," in *Second IEEE Annual Conference on Pervasive Computing and Communications, 2004. Proceedings of the, IEEE*, 2004, pp. 23–30.
- [38] G. MOUNTAIN, "Using the evidence to develop quality assistive technology services," *International Journal of Integrated Care*, 2004: 19–26.
- [39] R. IMRIE AND P. HALL, *Inclusive Design: Designing and Developing Accessible Environments*, Spon Press, London, 2001.
- [40] M. J. SCHERER, *Living in the State of Stuck: How Technology Impacts the Lives of People with Disabilities*, Cambridge, MA, Brookline Books; ed 3, 2000.
- [41] MD. M. ISLAM, M. S. SADI, K. Z. ZAMLI, AND MD. M. AHMED, "Developing Walking Assistants for Visually Impaired People: A Review," *IEEE Sensors Journal*, vol. 19, no. 8, Jan 2019.
- [42] S. PONNADA, S. YARRAMALLE, AND M. R. T.V., "A Hybrid Approach for Identification of Manhole and Staircase to Assist Visually Challenged," *IEEE Access*, vol. 6, 2018.
- [43] N. S. AHMAD, N. L. BOON AND P. GOH, "Multi-sensor Obstacle Detection System via Model-based State-Feedback Control in Smart Cane Design for the Visually Challenged," *IEEE Access*, vol. 6, 29 Oct 2018, pp. 64182 - 64192.
- [44] J. BAI, S. LIAN, Z. LIU, K. WANG AND D. LIU, "Virtual-Blind-Road Following-Based Wearable Navigation Device for Blind People," *IEEE Transactions on Consumer Electronics*, vol. 64, no. 1, Feb. 2018, pp. 136–143.

Edited by: Anil Kumar Budati

Special issue on: Soft Computing and Artificial Intelligence for wire/wireless Human-Machine Interface

Received: Sep 22, 2023

Accepted: Dec 14, 2023