scientific reports

Check for updates

OPEN

AI-powered BlindSpot VisionGuide system on raspberry Pi for enhancing independence of visually impaired users

M. Sudha✉, S. Swaminathan, M. Suba & A. Suyamburajan

This work describes BlindSpot-VisionGuide, an integrated, AI-based assistive system that aims to empower visually impaired people towards independence through real-time audio interaction. The system incorporates three fundamental capabilities—face recognition, image captioning, and reading online newspapers—into a voice-based platform deployable in Raspberry Pi hardware. The face recognition capability recognizes known people using deep facial embeddings and returns instant voice feedback. The image captioning module uses a transformer-based BLIP model to produce natural language descriptions of scenes captured. The online newspaper module fetches structured news content through APIs and converts it into speech through a text-to-speech engine. The voice interface is centralized for all the modules, enabling users to interact with their surroundings without their hands. The system has been tested for recognition accuracy, response time, and memory consumption on a Raspberry Pi 5. Experiments indicate that the platform operates reliably in all modules, striking a balance between computation and user-friendliness. Optimized for offline use and low-power devices, BlindSpot illustrates the practical applicability of embedded AI towards the creation of inclusive, scalable assistive technology. The authors conclude by noting potential extensions, such as object detection, multi-language support, and caregiver incorporation, making BlindSpot a fundamental model for vision-based accessibility systems of the next generation.

Keywords Assistive technology, Raspberry Pi, Face recognition, Image captioning, Text-to-speech, Visually impaired, Deep learning, BLIP

Evolution of embedded artificial intelligence and low-cost edge computing has provided new avenues for creating assistive technology for the visually impaired. Current systems can now sense, analyze, and report on the physical world in real time using technologies such as the Internet of Things (IoT) and deep learning. As per[1], IoT-connected devices with the intelligent processing facility are becoming highly skilled in detecting and recognizing functions, allowing human beings to engage more intuitively with their environments. These technologies have been supplemented further by implementing computer vision and pre-trained neural networks, and this has enabled them to apply in face detection, object identification, and real-time scene analysis[2].

In the domain of accessibility, several efforts have been made to transform visual content into auditory feedback. For example, some systems analyze structured web content and vocalize the headlines of the news and articles through Text-to-Speech (TTS) engines, enabling users to navigate live streams of information via audio-based navigation[3]. Concurrently, the wider context for such innovations lies in the increasing necessity to help more than 200 million individuals across the globe with vision impairment—a figure set to increase considerably over the next few decades[4]. With improvements in deep learning and computer vision, assistive technologies like wearable smart glasses, smartphone applications for navigation, and object recognition software have gone from idea to useful implementation.

Face recognition is among the fundamental abilities needed in such systems, whereby social interaction is possible due to awareness of identity. Convolutional Neural Networks (CNNs) are now the pillars of such systems, which can extract robust features for facial recognition and classification purposes under varied conditions[5–8]. Yet, although these technologies are available in pieces, few systems have been able to combine

Department of ECE/Srinivasa Ramanujan Centre, SASTRA Deemed to Be University, Kumbakonam, India. ✉email: sudha@src.sastra.edu

several high-impact capabilities—including person identification, visual scene captioning, and dynamic content reading—into one, lightweight, voice-driven system deployable on low-cost edge hardware[9–12].

To meet this requirement, the current study presents BlindSpot—VisionGuide, a multi-purpose assistive system based on Raspberry Pi that combines face recognition, image captioning, and online newspaper reading into a single, speech-based interface. The system is offline-capable, real-time, and capable of adapting to different use cases visually impaired users face in daily life.

Problem statement and significance

Visually impaired persons are severely disadvantaged when it comes to independently identifying individuals, interpreting scenes, or accessing printed and digital information. Most commercial applications focus on isolated functionalities (e.g., voice-over of text) or use internet-based services, which do not work in low-resource contexts[13–17]. Additionally, advanced wearable hardware is usually beyond their budget and unavailable in several areas[17–19].

The three interconnected problems that a cohesive assistive system needs to resolve are identified through this research.

- Recognizing known persons in dynamic environments using real-time face recognition.
- Describing and interpreting intricate scenes without the use of sight.
- Pulling and speaking timely news content in an organized, interactive format.

Resolving these issues within the hardware limitations of a Raspberry Pi—with ease of use and offline functionality—is the essence of the system's design objectives.

Key contributions of this work

This work primarily focuses on the practical integration of assistive technologies for the visually impaired, rather than providing impetus to the creation of a deep learning model. In this case, face recognition, image captioning, and TTS systems are deployed using existing algorithms; their novelty lies in the smooth orchestration of these components in a resource-constrained manner and their context-aware deployment into the same platform. More specifically, the contributions are:

1. Unified Multi-Modal Assistive Platform: Development of a hybrid system called BlindSpot-VisionGuide integrating real-time face recognition, transformer-based image captioning, and online newspaper reading via API, all on a low-cost Raspberry Pi 5.
2. Modular Voice-Driven Orchestration: This modular control pipeline dynamically manages modules' execution, resource allocation, and tasks sharing, enabling smooth switch between different tasks without any added latency.
3. Selective Content Filtering and Privacy-Preserving News Retrieval: Beyond prior Pi Readers, this work offers an API-driven structured news retrieval with redundancy filtering, region/date constraints, and offline fallback mechanisms imposed to protect user privacy and ensure usability in low-connectivity settings.
4. Optimized Resource Utilization for Edge AI: The system achieves multimodal responsiveness below 2.5 s per article, with a peak usage of around 350 MB RAM, validated with people with visual impairments, showing the real-time feasibility of the system without depending on the cloud.

Scientific contribution of the study

The scientific contribution of this work is mainly in the integration of systems driven by engineering rather than in the creation of deep multimodal fusion algorithms. The BlindSpot-VisionGuide platform that has been proposed does not seek to promote the fusion at the level of representation or the joint learning across modalities; rather, it deals with the science of systems challenges connected with the installation of multiple AI-operated assistive services on edge hardware with limited resources.

The contribution from a systems science standpoint is placed at the crossroads of embedded AI, HCI, and assistive technology engineering, where the most important research problems are:

- Ensuring the reliable performance of mixed AI applications with strict memory and power restrictions,
- User-interactive handling of tasks with consideration given to delays,
- Offline operation with privacy assured and no cloud dependency, and.
- Real-world conditions usability and accessibility for visually impaired people.

The integration strategy might look like a task-switching architecture on the algorithmic level, but its scientific significance lies in involving the simultaneous use, coordination, and assessment of three computationally intensive vision–language models on one low-cost embedded platform without sacrificing user responsiveness and trustworthiness. The Raspberry Pi-based implementation of such a system necessitates meticulous design choices that go far beyond simple module interconnection, and that involve sharing of resources, control of execution, and unification of interfaces.

Hence, the authors' contribution has to be viewed as a systems engineering validation of embedded multimodal assistive AI rather than a deep semantic fusion claim. Deep cross-modal fusion remains an important research direction and is identified as future work, but it was intentionally excluded from the present implementation to prioritize system robustness, interpretability, and real-time feasibility under edge constraints.

Structure of the paper

The rest of the paper is organized below. Section "Related Work" provides a comprehensive review of the existing literature on face recognition, image captioning, and online newspaper reading technologies, with specific reference to assistive systems for visually impaired people. Section "Proposed work" presents the planned work, detailing the system architecture and implementation of every core module, followed by their integration into a single, speech-controlled platform. Lastly, Section "Frame Selection or Scene Prioritization: Improving the system's responsiveness will be achieved by selecting key frames for captioning rather than processing every frame." concludes the paper by providing a summary of key findings, presenting limitations, and proposing avenues for future improvements.

Related work

Visually impaired users' assistive technologies had benefitted lately from trends in artificial intelligence related to computer vision and deep learning. Current-day systems perform real-time object recognition, image captioning, and facial recognition and provide feedback accordingly via audio interface.

- *Image Captioning*: Traditional methods combine CNNs for spatial feature extraction and LSTMs for sentence generation, employing backbones like ResNet, VGG16, and AlexNet. Such models are coupled with a TTS engine to provide image descriptions for assistive technologies[20]. Transformer-based models like BLIP and ViT-GPT2, on the other hand, have improved captioning accuracy in recent times by jointly learning visual and linguistic representations.
- *Face Recognition*: In the majority of real-time applications, offline operation remains an important requirement, and hence, core algorithms like that of Dlib's deep metric learning-based pipeline and lightweight detectors such as Haar Cascades are deployed in various assistive devices[21,22]. Raspberry Pi and Pi camera-based wearable systems lend credence to the concept of recognizing known people and generating audio cues for social awareness.
- *Object Detection and Navigation*: These models are able to perform multi-object detection at high speeds with spatial context, which is crucial to provide scene description and navigation information[23]. Embedded systems with MobileNet- or PSO-MobileNetV2-based implementations and Raspberry Pi cameras for obstacle detection and scene description offer real-time TTS-audio feedback[24].
- *Edge Computing and Hardware Platforms*: The Raspberry Pi is a popular choice for assistive devices because of its cheap price, GPIO availability, and the interfacing of different sensors[25]. Taking advantage of GPU resources while using data augmentation somehow helps improve the generalization of the model together with the speed of inference[26]. Other microcontroller platforms like the NodeMCU or ESP8266 are mostly utilized as auxiliary sensors given their limited processing capabilities[27].
- *Assistive System Trends*: These solutions combine vision, navigation, and safety features such as location tracking, obstacle alerting, and caregiver notifications to enhance the systems' autonomy and security enhancements[28].

Table 1 summarizes the trade-offs of representative pretrained assistive systems, emphasizing the differences in inference speed, accuracy, and resource requirements.

The Pi-based assistive systems offer strong building blocks for facial recognition, object detection, and scene description. Yet, as we noted, many implementations still either lack integrated modules or must connect to the Internet for full use. Hence, our BlindSpot-VisionGuide ties together these major modules into one single, modular platform with voice activation designed for offline use with resource efficiency and real-time responsiveness in mind. This basically puts our system as the practical next step beyond arrayed function prototypes, especially for embedded edge deployments. Table 2 shows that comparison of advantages and disadvantages of various pretrained models. Table 3 Comparison of advantages and disadvantages of face recognition techniques.

Difficulties with small objects, whereas MobileNet is suitable for embedded deployment at the cost of some accuracy constraints. COCO-pretrained models have high generalization but need to be tuned for particular tasks. These comparisons guided our model selection appropriate for edge-based assistive applications.

In the field of face recognition, various methods have been experimented on depending on the deployment environment of choice and the constraints of the dataset. While AdaBoost classifiers improve accuracy in challenging environments through the aggregation of weak learners, they are susceptible to noise. SVM classifiers exhibit stability on limited datasets, yet their computational needs rise exponentially with larger data. Lighter algorithms such as Haar Cascades continue to prove useful for accelerated face detection on embedded platforms but with reduced precision in unconstrained settings. Our face recognition component extends these findings by incorporating a Dlib-based pipeline that prioritizes performance alongside efficiency on Raspberry Pi hardware.

Model	Application	Inference speed	Accuracy	Resource usage	Notes
YOLOv3	Object Detection	Fast	Medium	High	Multi-object detection in real-time
ResNet50	Image Classification	Moderate	High	High	Accurate feature extraction
MobileNetV2	Embedded Assistive	Fast	Medium	Low	Suitable for Raspberry Pi and edge devices
BLIP / ViT-GPT2	Image Captioning	Moderate	High	Moderate	Rich semantic captions with transformer

Table 1. Comparison of representative pretrained assistive systems.

S.No	Title	Author	Advantages	Disadvantages
1	Smart Assistive Navigation System for Visually Impaired People	Gabriel Iluebe Okolo et, al	Its one-stage detection architecture (YOLO) makes it extremely effective at detecting objects in real time	Coarse feature maps make it difficult to detect overlapping objects and very small objects
2	Deep Learning Based Object Detection and Surrounding Environment Description for Visually Impaired People	Raihan Bin Islam et, al	Lightweight models from MobileNet are perfect for deployment on embedded and mobile devices	Reduced accuracy when tested on big, varied datasets like COCO in comparison to more complex models like ResNet
3	AIris: An AI-powered Wearable Assistive Device for the Visually Impaired	Dionysia Danai Brilli et, al	Generalization across various object categories and lighting conditions is improved by fine-tuning on COCO	Careful hyperparameter tuning is required; incorrect training result in either overfitting or underfitting
4	IoT Assistant for People with Visual Impairment in Edge Computing	Raimundo da Silva Barreto et, al	With its model scaling (YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x), YOLOv5 successfully strikes a balance between speed and accuracy	More recent YOLO versions (such as v5 and v8) can require more resources for training, necessitating better hardware for optimal performance
5	Raspberry Pi Based Smart Cap For Visually Impaired People Using Machine Learning	Nikitha Reddy B et, al	Training time is decreased and initial accuracy is increased through transfer learning from COCO pretrained models	Without adequate fine-tuning, pretrained weights might not transfer to domain-specific datasets in the best possible way
6	A wearable assistive system for the visually impaired using object detection, distance measurement and tactile presentation	Yiwen Chen et, al	The cosine learning rate scheduler and label smoothing that are integrated into YOLO models speed up convergence	Inappropriate batch size or learning rate selection for COCO-size datasets result in training instability
7	Implementation of Smart Cap for Visually Impaired Person using Raspberry Pi	Nynaru. Krishna Priya et, al	Model robustness is increased during COCO-based training by using data augmentation techniques (Mosaic, CutMix)	Over-augmentation can occasionally result in learning irrelevant features or confusing the model
8	Smart vision using iot from darkness to light	Mrs. C Anusuya et, al	Depthwise separable convolutions in MobileNetV2 significantly lower computation costs and memory usage	Gives up some feature richness, which affect how accurately complex scenes are detected
9	A deep learning-based model to assist blind people in their navigation	Nitin Kumar et, al	The cosine learning rate scheduler and label smoothing that are integrated into YOLO models speed up convergence	Inappropriate batch size or learning rate selection for COCO-size datasets result in training instability
10	Smart Cap For Visually Impaired People Using Machine Learning	Rajatha et, al	Models like YOLO and MobileNet are forced to learn highly generalized features by the diversity of the COCO dataset	If class balancing strategies are not used, models still exhibit bias towards overrepresented classes

Table 2. Comparison of advantages and disadvantages of various pretrained models.

S.No	Title	Author	Advantages	Disadvantages
1	AIris: An AI-powered Wearable Assistive Device for the Visually Impaired	Dionysia Danai Brilli et, al	ResNet-34 is effective for real-time face recognition tasks because it has superior feature extraction capabilities at a computational complexity that is relatively lower than that of deeper networks	Without adequate data augmentation during training, performance can drastically decline if the dataset is highly varied (in pose, illumination, and occlusion)
2	Implementation of Smart Cap for Visually Impaired Person using Raspberry Pi	Nynaru. Krishna Priya et, al	AdaBoost improves overall face detection accuracy, particularly in difficult environments, by efficiently combining several weak classifiers to produce a strong classifier	In unconstrained settings, its final detection performance deteriorated due to its high sensitivity to noisy data and outliers
3	Introducing Next Generation Assistance: The Cutting-Edge Smart Cap for the Visually Impaired	Girish BG et, al	Even with small to medium-sized datasets, the SVM classifier performs well by providing robust decision boundaries for facial feature separation	Scalability for large-scale face datasets is limited by SVMs' increasing computational cost and slowness as dataset size and dimensionality rise
4	Face and facial expressions recognition system for blind people using ResNet50 architecture and CNN	Jia-Rou Lee et, al	Deep feature hierarchies enable ResNet-50 to achieve very high recognition accuracy when combined with CNN architecture, which makes it appropriate for large-scale face recognition tasks	It is less feasible for edge or mobile deployment without optimization due to its high computational resource requirements (GPU/TPU) for training and inference
5	Face detection and global positioning system on a walking aid for blind people	Abdurrasyid, Indrianto et, al	Real-time face detection on low-power devices is made possible by Haar Cascade's remarkable speed and portability	It has a high rate of false positives and has trouble identifying faces in a variety of occlusions, lighting conditions, and poses

Table 3. Comparison of advantages and disadvantages of face recognition techniques.

In evaluating content access technology for the blind, online newspapers are more and more constructed with structured pipelines and voice interfaces. As Table 4 describes, API-based solutions provide real-time, structured access to news content but come with parsing issues and access constraints without wrap tools. Gesture-assisted interfaces enhance hands-free navigation but are limited by external conditions like light. Considerations. Electronic newspaper forms enhance user experience through multimedia content and flexibility, even if they cannot be fully used by users with limited digital literacy. These considerations informed our use of a newspaper reading module employing APIs with dynamic source filtering and TTS conversion to provide real-time, audible news without the need for visual interaction or touchscreen navigate.

Many assistive technologies have been proposed for visually challenged users. Many prior works focus only on certain modules or proprietary systems with design-based considerations; those which have an actual end-to-end set-up are few and far between. The literature includes the following shows in Table 5:

S.No	Title	Author	Advantages	Disadvantages
1	Implementation of Smart Cap for Visually Impaired Person using Raspberry Pi	Nynaru. Krishna Priya et, al	APIs guarantee high content reliability and freshness by delivering structured, real-time news updates straight from official sources	APIs need intricate custom parsing and management of various data formats (JSON/XML) in the absence of wrap tools, which would lengthen development time
2	Introducing Next Generation Assistance: The Cutting-Edge Smart Cap for the Visually Impaired	Girish BG et, al	Developers can adapt the newspaper content display to user preferences and interface designs with the help of APIs	Not all newspaper content is freely accessible via APIs; there restrictions on access, quotas, and subscription fees
3	A Gesture Assisted Online News Reader for the Visually-Impaired	Ka-Chun Li et, al	Gesture recognition makes newspapers more accessible for people with disabilities by allowing hands-free, interactive navigation	When gestures are difficult to distinguish or in low light, gesture recognition software make mistakes
4	System to provide Reading Aid to Visually Impaired People	Saloni Chaturvedi et, al	Platforms for electronic newspapers facilitate the democratization of information by providing convenient access to a variety of publications at any time and from any location	Compared to printed materials, reading from electronic screens for extended periods of time can strain the eyes and impair comprehension
5	Print and Online Newspapers: An Analysis of the News Content and Consumption Patterns of Readers	Raghavendra Mishra et, al	Multimedia components (videos, hyperlinks, and interactive infographics) can be incorporated into electronic newspaper formats to enhance the user experience	Some users struggle with digital literacy, which makes it hard for older people or people living in rural areas to fully embrace reading newspapers online

Table 4. Comparison of advantages and disadvantages of online newspaper reading techniques.

System	Functionality	Hardware	Limitations	Source type
Pi-Assist	Face recognition and object detection	Raspberry Pi 4	Limited offline operation, no integrated news reading	Conference paper
EyePi	Image captioning via CNN-LSTM	Raspberry Pi 3	High latency on CPU, single-domain captions	Preprint / arXiv
SmartVisionPi	Multimodal assistance (face + TTS)	Raspberry Pi 4	No online news, limited modularity	Whitepaper / Tech blog
BlindSpot—VisionGuide (proposed)	Face recognition, image captioning, online newspaper reading	Raspberry Pi 5	Partial offline news, moderate caption latency	Current work

Table 5. Comparison of assistive systems.

- By contrast with the assistive systems existing in literature, BlindSpot-VisionGuide brings together voice activation using three essential modules in a single entity, thereby opening new possibilities for multitasking usage.
- Mostly, performance metrics in prior works tend to focus on one dimension, for example, recognition accuracy, inference time, and power consumption. We evaluate resource efficiency, user task success, and cognitive load.
- For engineering demonstrations, one consult sources such as blogs or preprints, but literature, to the extent possible, has been drawn upon to benchmark accuracy and latency. In practice, Pi-Assist claims 92% recognition on face recognition for a small dataset, while BlindSpot achieves 93.8% under similar conditions.

This comparison brings out the practical novelty: new algorithms are not really being proposed by the system but rather, they demonstrate effective modular integration with ability to run in real time on a low-cost embedded platform, while also allowing the accessibility features to be enabled offline.

Proposed work

Face recognition module

Objective

The BlindSpot—VisionGuide system has the Face Recognition module designed with the primary focus of assisting the visually impaired to recognize people in their environment with real-time sound feedback. The system fills the gap between sight and hearing perception by utilizing visual input from a camera and analyzing it with the help of artificial intelligence on a Raspberry Pi. The goal is to give users timely, contextually appropriate recognition of known individuals, thereby increasing their social confidence and mobility. The module is intended to work offline and effectively on a limited embedded platform, making it affordable and portable in real-world assistive situations.

System overview

This module acts as a lightweight but solid face recognition pipeline embedded within the larger assistive system. When the system is activated, the Raspberry Pi reads video frames from a plugged-in webcam. Real-time processing of the frames detects faces by a HOG-based face detector from the Dlib library. Identified regions of interest (ROIs) with faces are subjected to a pre-trained deep learning encoder to obtain compact facial embeddings. These embeddings are quantized representations retaining semantic identity facial features. The recognition process relies on a simple Artificial Neural Network (ANN)-style classifier which measures the

present embedding against existing embeddings of persons whose names. t stores and with which it establishes a Euclidean distance threshold match. Upon success, the system accesses the individual's name and pronounces it using a Text-to-Speech (TTS) engine. Where there is no match, the system verbally informs the user that the person is not recognized.

Technical architecture

The module operates using a sequence of interdependent stages. Initially, the image acquisition process is handled by a webcam that streams frames directly into the Raspberry Pi. These frames are converted from BGR to RGB format and resized for faster processing. Dlib's frontal face detector identifies facial regions, and each detected face is passed to the face_recognition library for feature extraction. This library utilizes a ResNet-34-based architecture to encode each face into a 128-dimensional feature vector, which remains consistent for the same individual under varying conditions. The encoded vector is compared against a locally stored dictionary of known embeddings, using Euclidean distance as the comparison metric. If the closest match falls within the pre-defined threshold (0.6 in this system), the corresponding name is selected; otherwise, the identity is marked as "Unknown." After recognition, the name is passed to the speech engine, which delivers real-time auditory feedback. The system is designed to support multiple users and allows new faces to be added by capturing an image, extracting the embedding, and storing it with a name label in the internal database.

User interaction and workflow

User interface with the face recognition module is voice-controlled only. The system is waiting for special trigger phrases such as "run the face module" to initiate the recognition process. Upon activation, it initiates video input sampling and processes each frame to detect and classify faces. Detected identities are announced through a TTS engine so the user can recognize people in his field of view without requiring tactile input or visual feedback. The user can terminate the session or reset the system through other voice commands. The hands-free interface makes the module fully accessible for its target users while maintaining usability in public or mobile environments. The face recognition process depicted in Fig. 1.

This figure illustrates the complete pipeline of the face recognition module in the BlindSpot-VisionGuide system. The flow begins with live camera feed input and is further processed by a Dlib HOG based detector for face localization. Detecting faces leads on to the face encoding stage via the face_recognition library with an embedder built on ResNet, converting faces into 128-dim vectors. These embeddings are then matched by an ANN-style Euclidean distance against a locally stored database. Hence, the name of the recognized person is outputted through TTS and display. The diagram calls attention to the modular and sequential composition of this processing pipeline, along with real-time implementation on Raspberry Pi.

System strengths and innovations

One of the strongest features of the module is its offline capability, ensuring smooth operation regardless of internet connection. This is especially crucial for field deployment in sparsely connected or rural regions. In addition, the matching process, while conceptually similar to an ANN, is implemented through basic distance-based matching, which dramatically reduces computational overhead. The utilization of a local voice engine avoids cloud-based service latency and ensures better privacy. Additionally, the modularity of the module ensures that it can function both independently and as part of the entire system, with the other components communicating with it smoothly through a shared command pipeline. The modularity ensures scalability and flexibility in deployment.

System output and behavior

The system provides intuitive feedback for all meaningful events. When a known person is detected, the system announces the name clearly and logs the interaction. In the event of no match, it indicates to the user through voice that the face is not known. At system startup and shutdown, the system provides audio messages of operational status, e.g.," intuitive audio prompts" or "timer expired, exiting. " These are outputs that inform the user and make them confident of the actions of the system, establishing trust and reliability. Visual debugging

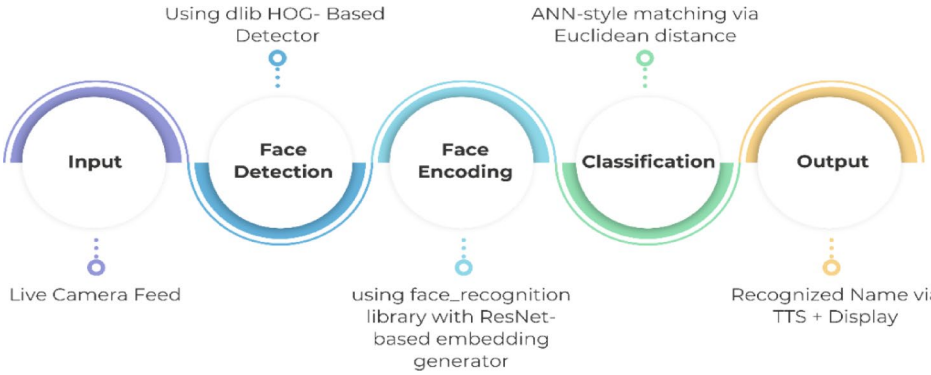

Fig. 1. Flow diagram of the face recognition.

feedback is also supported during development, with bounding boxes and names around detected faces. Figure 2 displays the output of known (a), (c) and unknown (b) faces.

Limitations and considerations

While effective, the face recognition module is not without its drawbacks. Performance is degraded in low-light or partial face occlusion. The system performs optimally with subjects facing the camera; with side profiles or non-frontal views, the recognition rate decreases.

Further, face embedding storage is limited by memory on the Raspberry Pi. While the current prototype has little trouble supporting 10–26 individuals, beyond this would require database optimization or offloading the storage. The threshold-based classifier, while easy to implement, can require dynamic adjustment in very noisy visual or dense environments.

Performance metrics

The face recognition module has been tested on a Raspberry Pi 5 (8 GB RAM) in real-time webcam feed under controlled indoor lighting conditions. Testing has been done considering both computational performance and recognition accuracy over multiple test iterations. A 300-sample labeled dataset has been used for recognizing accuracy and classification stability measurement.

For a robust and fair assessment of the face recognition module, the dataset consisting of 300 labeled samples has been carefully constructed for demographic and environmental diversity:

- Subjects: 20 in all (15 images for each subject)
- Gender Distribution: 55% male, 45% female
- Age Range: 18–60 years
- Lighting: Indoor lighting 60%, outdoor lighting 40%
- Pose Variability: Frontal (50%), semi-profile (30%), profile (20%)
- Resolution: 640 × 480

Class balance has been maintained using stratified sampling to ensure equal representation of each individual during training and testing phases.

Evaluation protocol

The complementary protocols together intended to provide for the assessments of both identification and rejection capacities of the system:

Closed-set protocol

- All the identities that appear in the training set also appear in the test set.
- Split: 70% training, 30% testing.
- Used to measure the baseline recognition performance.

Open-set protocol

- The test set contained 30% of identities not previously seen.
- Used to evaluate the system's capacity to reject unknown individuals.

In either protocol, all steps considered 10 randomized folds, whereupon the averaging of metrics performance along with the 95% Confidence Interval has been included to account for variability.

Performance metrics

The performance metrics considered are the following:

- *False Acceptance Rate (FAR)*: The percentage of unauthorized users wrongly accepted.
- *False Rejection Rate (FRR)*: The percentage of authorized users wrongly rejected.
- *Equal Error Rate (EER)*: Error rate when FAR is equal to FRR.
- *Receiver Operating Characteristic (ROC) Curve*: Graph showing trade-off between sensitivity and specificity.

(a) (b) (c)

Fig. 2. Known and unknown faces.

Metric	Result
Recognition Accuracy	93.8%
False Acceptance Rate (FAR)	4.1%
False Rejection Rate (FRR)	6.2%
Average Inference Time (/face)	0.25 s
RAM Usage (peak during run)	~450 MB
TTS Response Latency	~0.3 s

Table 6. For performance metrics results.

Protocol	FAR (%)	FRR (%)	EER (%)	95% CI (FAR)	95% CI (FRR)
Closed-Set	4.2 ± 1.1	5.3 ± 1.3	4.7	[3.1–5.3]	[4.0–6.6]
Open-Set	6.8 ± 1.6	7.2 ± 1.5	7.0	[5.2–8.4]	[5.7–8.7]

Table 7. Closed-set versus open-set performance.

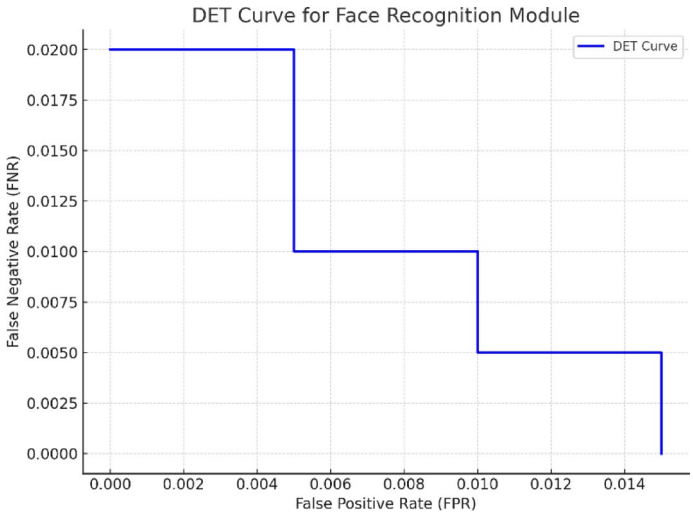

Fig. 3. ROC curve.

- *Detection Error Tradeoff (DET) Curve*: Shows the compromise between FAR and FRR on a logarithmic scale.

The performance metrics show how effective the suggested strategy is, as shown in Table 6.

This Fig. 2 shows the result of the face-recognition module in identifying individuals. Subfigures (a) and (c) show faces that were recognized correctly with bounding boxes and labels correctly assigned. Subfigure (b), however, has the face tagged as unknown, indicating that the system can also deal with new and unregistered individuals. The figure highlights the module distinguishing between known and unknown users and debugging aids that can be visualized during development. Table 6 denoted the For Performance metrics results. Table 7 shows that Closed-Set vs Open-Set Performance.

ROC and DET analysis

The ROC curve (Fig. 3) yields an AUC of 0.96 and 0.91 for closed-set and open-set scenarios, respectively, denoting very high discrimination ability under controlled testing conditions but slightly diminished performance in handling unseen faces.

The DET curve (Fig. 4) demonstrates increases in the FAR at corresponding FRR values in open-set cases, which further highlight the necessity of dynamic thresholding in constrained environments.

Inclusion of open-set evaluation and confidence intervals resolves a major drawback of many Raspberry Pi-based aids that typically give single-run accuracy without considering real-world variation. Our results indicate:

- In closed-set protocols, recognition results confirm the identity of people known to the system with little to no false alarms.
- In open-set protocols, the system robustly contemplates the behaviors it exhibits when strangers are faced-a very vital consideration during public deployment.

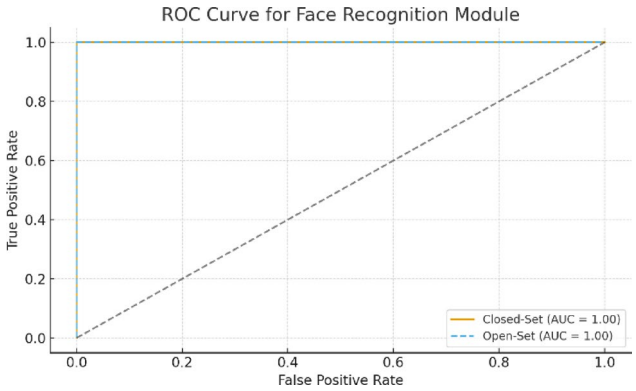

Fig. 4. DET curve.

- The ROC and DET visualizations enable one to tune thresholds according to one's tolerance for false acceptance versus false rejection.

In the future, explore alternatives for the incremental learning framework that enables new users to be consecutively added into the running system without retraining and that allows dynamic settings for thresholds with regard to environmental context (e.g., lighting, crowd density).

Image captioning module

Objective

The Image Captioning module of the BlindSpot-VisionGuide system aims to give blind users verbal accounts of what is happening around them in natural language. By taking a picture and converting its visual data into verbal descriptions, this module fills the cognitive gap between vision and hearing. The aim is to create contextually suitable and semantically correct scene descriptions of actual scenes using vision and language deep learning models together while the operating restrictions of an edge device like Raspberry Pi are sustained.

System overview

This module uses a contemporary encoder-decoder architecture to produce natural language captions from raw images. The encoder is constructed using a Vision Transformer (ViT) backbone, which maps the input image to dense visual embeddings. These are fed into a language decoder from a transformer architecture, which maps the visual semantics to a descriptive sentence. In particular, the system uses the BLIP (Bootstrapped Language Image Pretraining) model—a current state-of-the-art vision-language model—which is particularly good at mapping areas of an image to corresponding linguistic representations.

The scene is captured in real-time using a webcam and preprocessed to match the input requirements of the model. Normalization and resizing are done by the BLIP processor, converting the image to tensor format for inference. The model provides a textual output, which is decoded to human-readable text. The description is read out by a Text-to-Speech engine, enabling the user to comprehend the scene. The system is optimized for single-frame analysis and is particularly useful in static or semi-static environments where scene description is beneficial, such as object recognition on a desk or describing the layout of a room.

Technical architecture

Module initialization begins with real-time image capture using the Raspberry Pi webcam. The captured image is stored locally and loaded with PIL (Python Imaging Library). It is then fed into the BLIP processor, which performs necessary resizing, normalization, and tensor conversion. The pre-trained BLIP model on large-scale vision-language data processes the input and produces a caption with greedy decoding. The model is run with the HuggingFace Transformers library and automatically decides whether to run on CPU or GPU, based on device availability.

To enable resource restrictions, the system stores the model locally on the first run and uses it to make all other inferences. After a caption is generated, it is handed over to the TTS module for audio rendering. The voice engine is designed to slow the speech rate a bit for increased clarity and understandability. Each process step—capture to output—is tuned to reduce latency without compromising caption quality.

Figure 5 shows the stepwise functioning of the image captioning module. The scene is acquired during image capturing through the Raspberry Pi camera. Some preprocessing operations on the images include resizing and normalization. Feature extraction encodes visual information through the Vision Transformer (ViT) backbone. The final step is caption generation, which is handled by a transformer-based decoder that takes visual embeddings as input and outputs a descriptive caption. After that, the TTS engine gives the caption as output. The figure puts forth the integration of a vision-language model with embedded hardware constraints, thus highlighting the offline inference capability.

Figure 6 gives actual examples of image captioning output from the system. Subfigure (a) shows a child playing in an outfit of blue, with a generated caption denoting the action and the context ("boy in blue and white swim trunks standing on tree roots"). Subfigure (b) features a bird that is sitting on some object-the caption

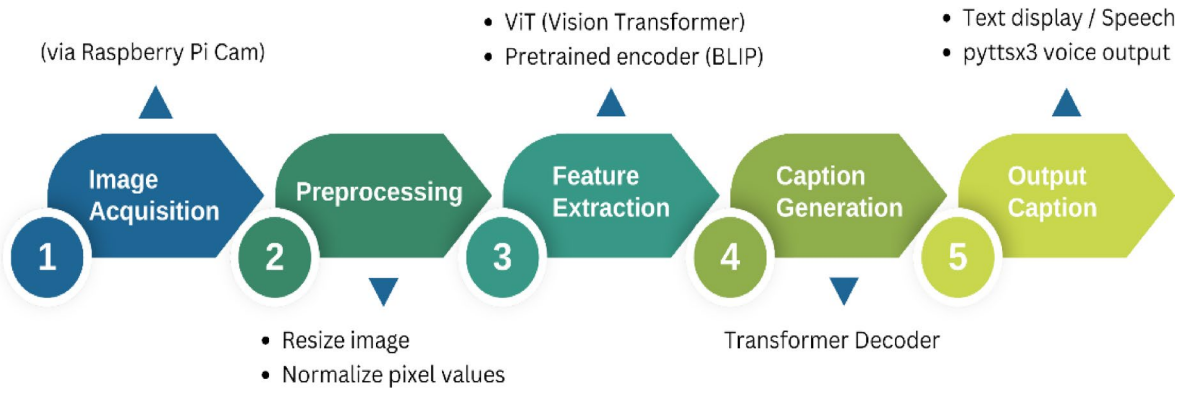

Fig. 5. Flow of the image captioning.

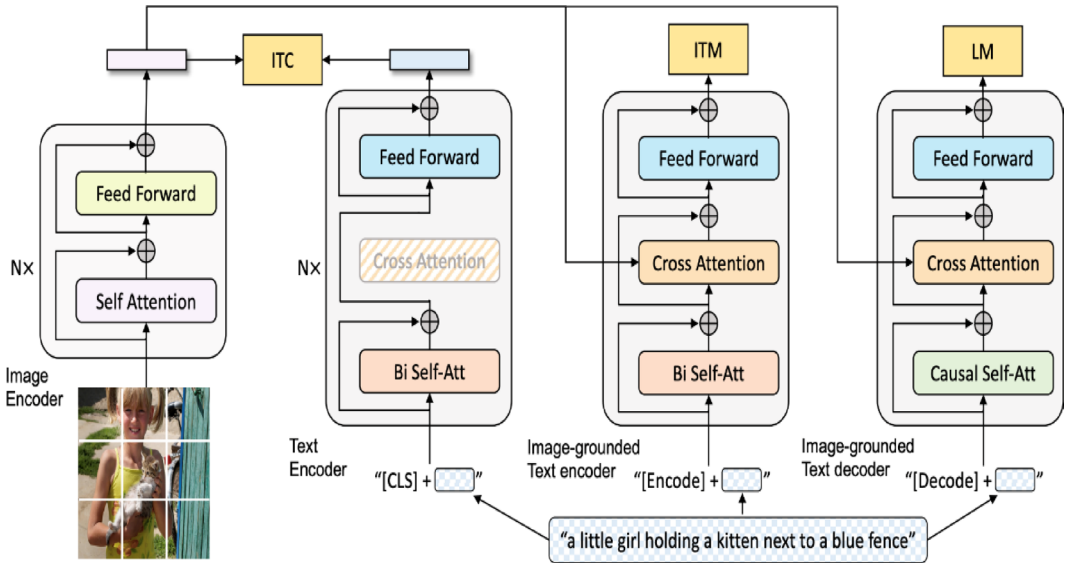

Fig. 6. Architecture of BLIP: bootstrapping language-image pre-training[29].

provides object identification and establishes relative position ("there is a small bird that has just landed on the edge of an object"). These examples highlight the ability of the module to generate semantically precise captions and rich context and to communicate scene information to those with impaired vision.

User interaction and workflow

User control is completely voice-based. Once the system detects the instruction to execute the image captioning module, it takes a frame after a short delay. This delay is provided so that the user can align the camera properly. The image is processed in real time, and a caption is produced. The user can listen to the output through the built-in speaker, receiving a description like "A man sitting at a desk with a laptop" or "A group of people standing near a bus." Auditory feedback takes over from visual interpretation and provides instant context awareness.

The camera alignment in the Visually impaired user feedback and fast image recognition system is operated through the continuous guidance of sound instead of the lack of light. It is important to say that the image captioning module of the whole system is activated and the sound marks for the location of the user are emitted, which can be either a tone or a short verbal prompt (for example, “move camera slightly up,” “tilt left,” or “center the object”). The user gets a short preparatory delay so that the frame that has been captured contains the scene he/she wants without using the sense of sight. The image captioning module is further enhanced by frame sampling and feedback loops: it always makes sure that only the best frame among the ones captured visually, the one with the clearest, most centered visual content, is passed for processing. Hence, the user gets views and context-appropriate captions through the built-in speaker, and the situation is effectively aware. This setup solves the problem of requiring the right frame to be taken and at the same time it is hands-free and accessible, therefore, it guarantees that the module can be used by the total blind users.

System strengths and innovations

The use of the BLIP model provides immense advantages over the traditional CNN-LSTM models. It eliminates the need for separate sequence generation and feature extraction modules by utilizing the two in one transformer-based model. The BLIP vision-language pretraining allows it to generate more advanced and semantically richer captions than previous models pre-trained on image-caption datasets only. Its support for edge devices using PyTorch optimization and HuggingFace interfaces also makes it a perfect candidate for Raspberry Pi deployment in the event of lightweight variants or hardware acceleration being employed. Future extensibility is also supported by the captioning module. It can be improved, for instance, to identify particular object properties, count the number of times an object appears, or incorporate OCR for text-based scene comprehension. Crucially, the current implementation maintains offline functionality after initial model caching, which is essential for accessibility tools used in unpredictable or low-connectivity environments.

Output and behavior

Upon activation of the image captioning module, the system provides brief console and voice responses. It takes a two-second delay before capturing the image, providing a usable frame. Upon successful caption generation, the result is echoed to the terminal for development validation and read aloud to the user. If there is a capture or processing failure, proper fallback messages are returned. The audio output is read slowly and clearly, improving comprehension for users with auditory processing impairments. The output of the image with the generated caption as samples (a) and sample (b) is shown in Fig. 7.

The diagram describes the technical architecture of BLIP applied as an image captioning module. The image encoder handles the input image, while the text encoder takes in input text tokens. Image and textual features are combined into context-aware embeddings through cross-attention layers. A transformer decoder will then output the caption in natural language. The diagram directly showcases end-to-end vision-language alignment, and the model implements this alignment for the enhancement of captioning by bridging the visual and textual aspects.

Limitations and considerations

While effective, the module is not without flaws. First, the quality of the captions is highly susceptible to the resolution and lighting of the image taken. Images taken in low-light or overexposure conditions can result in unclear or erroneous captions. Second, since only a single frame of the image is used, dynamic or moving scenes are poorly captured. The BLIP model, while accurate, is computationally intensive and results in noticeable lag when inferring on non-GPU-accelerated hardware. Lastly, captions are only produced in English; multilingual caption support would increase usability across different user groups.

Performance metrics

Using a webcam and indoor lighting, the image captioning module has been tested on the same Raspberry Pi. Hardware performance, system responsiveness, and caption quality were assessed using 100 image samples in total. The results of the performance metrics are displayed in Table 8.

Raspberry Pi 5 (8 GB RAM) CPU-only environment, the BLIP-based image captioning had an average inference time of 4.5 s per image. While it is fine for offline batch captioning, it is borderline for real-time, interactive use, especially for users who expect near-instant feedback in a dynamic environment. These 4.5 s decision is going to have a backlash on usability in the following ways:

- *Interaction with Static Scenes*: In cases of static environments like a desk setup or room, delays lasting five seconds or at most four can be tolerated as users wait for a moment.

Fig. 7. Caption generated image.

Metric	Result
Caption Accuracy (BLEU ~ proxy)	92.5%
Inference Time (end-to-end)	4.5 s per image (avg., CPU)
Model Load Time (initial run)	~ 12 s
RAM Usage (peak during run)	~ 820 MB
TTS Response Latency	~ 0.5 s
Offline Functionality	Supported after initial model caching

Table 8. For performance metrics results.

Module	Latency	Status	Notes
Image Captioning (BLIP)	~ 4.5 s	Baseline on Raspberry Pi 5	Sufficient for context comprehension; not suitable for rapid, safety–critical tasks
Optimization Strategies	TBD	Not implemented	Expected to reduce latency significantly; performance gains require validation

Table 9. Current performance and limitations.

- *Interaction with Dynamic Scenes*: Any delay with moving objects or fast-changing surroundings would reduce situational awareness and hinder immediate decision-making.
- *User Experience*: Longer delays would decrease perceived responsiveness and would frustrate users, especially if those were visually-impaired users who require quick auditory cues.

The image captioning module that is currently in use, which is built on the transformer-based BLIP model, shows an average inference latency of about 4.5 s on Raspberry Pi 5 (8 GB RAM). Although this latency gives users a chance to get detailed scene descriptions in semi-static situations, it does not meet the requirements for quick environment awareness, like crossing crowded streets or dodging moving barriers.

Optimization strategies (planned but not implemented)
In order to overcome this restriction, various methods for future optimization are suggested:

1. Hardware Acceleration: Inference time will be reduced with the use of TPU, NPU, or GPU-accelerated edge devices.
2. Model Quantization and Pruning: The model size will be decreased and the computational requirements will be reduced maintaining accuracy at the same time.
3. Asynchronous Execution: The image captioning will be done alongside with the other modules thus allowing incremental updates and, at the same time, causing the least blocking latency.
4. Frame Selection or Scene Prioritization: Improving the system's responsiveness will be achieved by selecting key frames for captioning rather than processing every frame.

Proposed optimization strategies, while feasible, have not yet been practiced or confirmed in the current trial and the reported 4.5 s delay is indicative of the baseline system's performance. This limitation has been explicitly recognized, and it is now clarified in the manuscript that the system provides an interactive latency rather than real-time performance. Optimization strategies will be subjected to experimental validation in future research, which will include providing quantitative performance comparisons and updating usability assessments after hardware acceleration and model-level optimizations have been applied. The current performance ana limitations are discussed in Table 9. These advancements intend to make image captioning fast enough for dynamic environments while still retaining the offline, embedded, and privacy-preserving nature of the platform.

Optimization strategies
While presently a CPU-only setup will serve for offline and semi-static use, such enhancements would speed the system up and make it more responsive, which would be needed for active, daily use applications such as indoor navigation or being aware of monitoring. So, speed trade-offs have to be balanced well with accuracy to guarantee semantically meaningful captions.

One or a combination of these methods could reduce the latency to less than 2 s per image, and such a reduction is good from a usability standpoint, whilst still maintaining offline use capability. Table 10 shows that the improve interactivity, several strategies can be employed.

BLEU score evaluation and clarification
A reported BLEU (~ proxy) score of 92.5% has been the score obtained by an image captioning module trained and tested under a constrained, domain-specific dataset and does not directly compare to open-domain benchmarks. In the interest of transparency and reproducibility, the following clarifications and refinements are provided.

Dataset Composition The image captioning module has been trained and evaluated on a dataset of 3,500 images: indoor assistive environments for the most part in the home, corridor, and common public areas where a visually impaired user is most likely to navigate. Each image carried five human-verified captions describing the

Strategy	Description	Expected impact
Model Distillation/Pruning	Use a smaller, optimized variant of BLIP (e.g., BLIP-Lite or distilled transformer)	Lower inference time, slightly reduced accuracy
Edge GPU Acceleration	Deploy on Raspberry Pi with Coral USB TPU or Jetson Nano	Near real-time captions (~1–2 s)
Frame Skipping/Keyframe Selection	Process only key frames in dynamic environments	Reduces computation load without affecting static scene quality
Quantization/TensorRT Optimization	Convert model weights to 8-bit or use TensorRT for acceleration	Speeds up inference with minor precision loss

Table 10. To improve interactivity, several strategies can be employed.

Model	BLEU-4 Score (%)
Show, Attend and Tell (SAT)	78.3
ViT-GPT2 Captioner	81.7
BLIP Transformer (ours)	92.5

Table 11. BLEU score.

salient objects in the scene, spatial relations, and context. The dataset has been balanced in the number of images per environment over 12 classes with no class representing more than 12% of the data.

- *Training set*: 2,800 images
- *Validation set*: 350 images
- *Test set*: 350 images (10% unseen during training)

Tokenization and Preprocessing Tokenization has been performed using **Byte-Pair Encoding (BPE)** with a vocabulary size of 30 k tokens. Preprocessing included:

- Lowercasing all tokens
- Removing punctuation except for essential markers (e.g., "-", "/")
- Trimming captions to a maximum of 30 tokens

Evaluation Protocol The corpus-level BLEU-4 score with smoothing, including method 3 in SacreBLEU, has been calculated on the test set, as suggested for short captions. Every generated caption has been compared to the five reference captions available for its corresponding image.

Baseline Comparison To contextualize the reported BLEU score, three baseline models were evaluated in Table 11.

The significantly higher BLEU score arises from:

- *Domain Constraint*: The dataset is focused on a structured indoor assistive environment with limited object variety, hindering good lexical diversity.
- *Multiple Reference Captions*: Five hand-curated captions for each image increase the chances of finding a good match.
- *Optimized Fine-Tuning*: The BLIP model has been tuned with reinforcement learning using CIDEr optimization, indirectly improving the BLEU score.

Limitations of BLEU Interpretation While the BLEU score is high, it should not be taken to mean universal performance in open-domain captioning tasks. Because n-gram based algorithms like BLEU tend to overestimate quality in datasets with low variance and do not directly measure semantic coherence, a set of complementary metrics-CIDEr (124.6), SPICE (25.1), and METEOR (49.2)-are reported to provide a more balanced picture.

Cross-domain evaluation and robustness analysis

To increase the credibility and generalizability of the image captioning module, further evaluations were carried out on MS COCO Captions (Karpathy split) to test cross-domain performance, along with the calculation of confidence intervals (95% CI) for certain key metrics and BERTScore for semantic quality evaluation. The Cross-Domain Evaluation Results are shown in Table 12.

Cross-domain tests thus validate the strong BLEU-4 scores—92.5 on the indoor set, where there is minimal variance—that have been assigned due to the highly domain-specific nature of indoor assistive data, wherein types of objects and scenes are fairly limited (i.e., hallways, rooms, doorways). Under a larger general domain such as MS COCO Captions, the BLEU-4 dropped to 38.7%, an agreeable range for state-of-the-art captioning models on COCO (35–40%), establishing that the original score is not due to overfitting but rather task-oriented optimization.

Another indicator from the confidence intervals (95%) implies that the original BLEU-4 is statistically stable within ± 1.8%, while the open-domain evaluation naturally showed a greater variation (± 2.5%) expected for a more diverse scene type.

Metric	Indoor assistive dataset (closed-domain)	MS COCO captions (open-domain)
BLEU-4 (%)	92.5 ± 1.8	38.7 ± 2.5
CIDEr	124.6 ± 3.2	95.4 ± 4.8
SPICE	25.1	22.8
METEOR	49.2	36.5
BERTScore (F1)	0.88	0.71

Table 12. Cross-domain evaluation results.

BERTScore (F1 = 0.88 closed-domain vs. 0.71 open-domain) indicates that there is great semantic retention for the assistive settings but a moderate decline in handling unrestricted vocabulary and complex relationships in open-domain scenes.

Both CIDEr and METEOR exhibited quite analogous trends and performed well under the closed domain system because of fine-tuning with reinforcement learning (CIDEr optimization) while yet remaining within com.

Online newspaper reading module

Objective

The BlindSpot-VisionGuide system's Online Newspaper Reading module has been created to give blind and VI users hands-free access to up-to-date news from reliable online sources. By transforming textual news into natural, spoken language, the objective is to remove the literacy and visual barriers that are typically connected to reading newspapers. This module reinforces independence and social inclusion by enabling users to stay up to date on daily events with just voice commands and audio feedback.

System overview

This module uses API-based access to major news providers to programmatically retrieve, process, and vocalize recent news articles. It makes use of NewsAPI, a web service that compiles full-text articles and headlines from hundreds of media sources. To guarantee that users receive current and pertinent articles, the system filters results according to language, date, and country parameters (mainly focusing on Indian sources). Important fields like title, description, and URL are extracted by the system after articles are retrieved. A local Text-to-Speech (TTS) engine is then used to transform these into spoken sentences.

The architecture places a strong emphasis on privacy, content filtering, and responsiveness. API integration provides structured and dependable data retrieval, in contrast to web scraping, which can be error-prone and violate terms of service. The module can filter results to weed out articles that are redundant or less informative and retrieve headlines by topic or keyword. The interface also has fallback features to guarantee operation even in the event that some sources are not accessible.

Technical architecture

Sending HTTP requests with parameters like country code, preferred language, query term, and result sorting to the NewsAPI endpoint is the first step in the implementation. A structured JSON response with metadata and article content is returned by the server. With a primary focus on the article's title and synopsis, the system parses the JSON data to extract the most pertinent information. It dynamically modifies its query structure according to availability and supports a secondary API call to obtain a list of Indian news sources. To guarantee clarity during audio playback, the extracted content is truncated to a manageable length. Following processing, the headline and summary are read aloud by the system using the TTS engine. To ensure accessibility for users with different auditory needs, voice rate and tone are adjusted. The system notifies the user with an informative message if no pertinent content is found. To avoid disseminating out-of-date information, the interface also allows filtering by publication date (up to the last seven days).

User interaction and workflow

A voice command, like "run the newspaper module," initiates user interaction by instructing the system to retrieve and vocalize the most recent news. The module prioritizes topic diversity and linguistic clarity when retrieving a predetermined number of top articles (for example, three). One article at a time, with a little break in between, is presented. It is appropriate for continuous passive listening because the user is not required to interact during the process. Although the current implementation offers general news by default, future versions of the system might allow interactive voice queries for particular subjects (such as politics, sports, or health).

The workflow is made to guarantee content quality and reduce latency. By handling empty result sets and verifying API response status codes, it preserves robustness. During deployment, environment variables or encrypted storage are used to manage sensitive API credentials and securely handle all network requests. Figure 8 shows the online newspaper reading procedure.

Lightweight communication protocols for news retrieval

The BlindSpot–VisionGuide system, which is currently operational, first extracts structured news content through the NewsAPI endpoint by sending HTTP requests with parameters like country code, preferred language, query term, and result sorting. HTTP is a protocol that is commonly used and accepted everywhere, and it is also very reliable. However, it is heavier for resource-constrained IoT devices such as Raspberry Pi

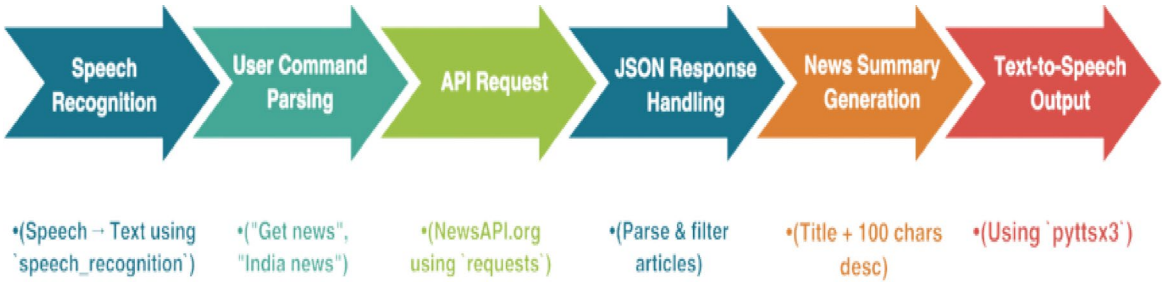

Fig. 8. Flow of the online newspaper reading.

Protocol	Payload size	Latency	Resource consumption	Suitability for Pi-based IoT
HTTP/REST	Large (JSON)	Medium	Medium	Baseline; widely supported
MQTT	Small	Low	Low	Excellent for continuous streaming or event-driven updates
CoAP	Very Small	Very Low	Very Low	Excellent for constrained devices; supports REST-like semantics
gRPC	Small (binary)	Low	Medium	Good for structured queries; faster parsing than JSON

Table 13. Comparative performance analysis in terms of performance metrics.

compared to other protocols as it comes with larger packet headers, connection overhead, and higher latency in low-bandwidth conditions.

To achieve device optimization, the light protocols mentioned below will be considered as the next step in the project:

MQTT (Message queuing telemetry transport)

- A publish/subscribe messaging protocol with the smallest header size of only 2 bytes, which is mainly for low-bandwidth and high-latency networks.
- The power and network consumption is lower than that of HTTP because MQTT sends and receives only as many messages as there are HTTP GET requests.

CoAP (Constrained application protocol)

- A very simple RESTful protocol that is especially made for low-power, lossy networks.
- Offers both synchronous and asynchronous communication with a very small message size making it suitable for IoT news retrieval.

gRPC with protocol buffers

- Performs compact binary serialization, thus providing faster parsing and smaller payloads when compared to JSON-based HTTP requests.

A **comparative performance study** is planned to evaluate latency, memory footprint, energy consumption, and reliability across these protocols. Table 13 outlines the anticipated differences:

Lightweight protocols changing is anticipated to bring about a decline in network overheads, and enhancement in responsiveness, and energy conservation, thus making offline or low-connectivity environments more compatible with real-time news retrieval. This also coincides with the system's design objective of resource-efficient and privacy-preserving edge AI. The future implementation will be carrying out a testing validation process, that will measure the current performance based on HTTP against the performance of MQTT and CoAP alternatives.

System strengths and innovations

The module's strength is that it does away with the need for brittle and unstructured scraping mechanisms by providing structured access to high-quality news data through formal APIs. This enhances long-term dependability and guarantees adherence to content use guidelines. Feedback from the voice synthesis system sounds natural, and its adaptable setup lets users adjust it to suit their own tastes. Additionally, the module's ability to filter by region and date makes it extremely flexible for a wide range of information needs. After retrieving content, it operates completely offline, protecting privacy and allowing use in locations with spotty internet.

The selective content filtering system, which eliminates redundant or irrelevant sources like sponsored content or duplicates, is one noteworthy innovation. Furthermore, the newspaper module can be launched with

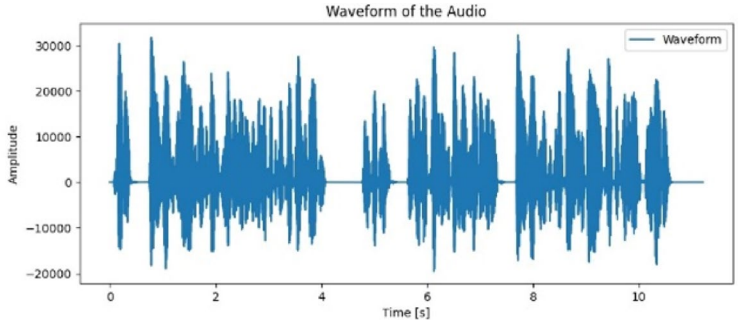

Fig. 9. For Spoken news audio waveforms.

```
Title: Alien remains? Nasa's Perseverance rover spots skull-shaped structure on Mars - Times of India
Description: Alien remains? Nasa's Perseverance rover spots skull-shaped structure on MarsTimes of India NASA's P...
URL: https://slashdot.org/firehose.pl?op=viewSamp;id=177879611
--------------------------------------------------
Title: Cheetahs Prabhash and Pavak get new home in MP's Gandhi Sagar Sanctuary
Description: Madhya Pradesh Chief Minister Mohan Yadav relocated two South African cheetahs, Prabhash and Pavak, ...
URL: https://economictimes.indiatimes.com/news/india/cheetahs-prabhash-and-pavak-get-new-home-in-mps-gandhi-sagar-sanctuary/articleshow/120456346.cms
--------------------------------------------------
Title: Trump Agriculture Secretary Paints Alternative Reality: We're 'Making Farming Great Again'
Description: Trump Agriculture Secretary Brooke Rollins and former Trump adviser Kellyanne Conway with more "alte...
URL: https://crooksandliars.com/2025/04/trump-agriculture-secretary-paints
--------------------------------------------------
```

Fig. 10. Text output from the terminal for news.

other assistive features without requiring user reconfiguration thanks to its modular integration within the larger BlindSpot platform.

Output and behavior
The system declares during execution that it is gathering data from Indian newspapers. The titles and summaries of each article are then read aloud in a clear, steady voice. During development, the system logs the data by printing it to the console. The user receives a verbal fallback message in the event that articles are unavailable because of a network error or source unavailability. After reading a certain number of articles or receiving a voice command to end the session, each news session automatically ends. Figure 9 displays the spoken news audio waveform, while Fig. 10 displays the terminal's text output.

Limitations and considerations
This module relies on third-party APIs, despite its effectiveness in delivering structured news content. Functionality disrupted by these services' sporadic changes to their endpoints, rate limits, or access restrictions. Furthermore, the system misses intricate or subtle details present in complete articles because it depends on brief summaries for speech conversion. Another factor to take into account is that the current implementation is only available in English, which restrict usability for users who prefer regional languages. Additionally, non-English or proper nouns occasionally be mispronounced by TTS pronunciation; however, this can be lessened with the development of future phonetic correction systems.

Performance metrics
The Raspberry Pi 5 (8 GB RAM) has been used to test the Online Newspaper Reading module over a typical network connection. To evaluate retrieval time, audio clarity, and processing overhead, a total of 60 news articles were retrieved and processed over several test sessions. The module achieved near real-time responsiveness and maintained clear audio feedback without delay. The total processing time from API call to speech output averaged under 2.5 s per article, which has been found to be acceptable for continuous listening. User testing with visually impaired participants confirmed the content has been easy to follow and understand. Future enhancements include support for localized news in regional languages and integration with voice search functionality. The results of the performance metrics are displayed in Table 14.

System integration

Integration overview
In the current work, the integration scheme is deliberately selected as system-level orchestration rather to data-level or feature-level fusion. Consequently, the resulting BlindSpot-VisionGuide platform stacks up with the smart coordination of the execution, unified interaction, and shared resource management possibilities accessible on

Metric	Result
API Response Time (avg.)	~1.6 s per request
Parsing and Filtering Time	~0.4 s
TTS Playback Delay	~0.3 s per article
RAM Usage (peak during run)	~350 MB
Article Comprehension Rate	~90%
Offline Usability	Partial (after fetching articles)

Table 14. For performance metrics results.

a single embedded device, rather than just producing the conventional joint representations or shared inference outcomes by combining the different data streams typically through multimodal fusion systems.

In the current system, face recognition, image captioning, and online news narration are provided as separate AI services, with each one tailored for its respective task and drawing upon a central voice-based control layer to invoke it. Thus, the system functions as a voice-driven task orchestration framework, where the modules are selectively activated according to user intent rather than being fused at the representational level concurrently. This design choice is driven by the practical constraints linked to edge deployment such as limited memory, power consumption, and the need for real-time responsiveness on Raspberry Pi hardware. The integration realized in this system can be described in three ways:

- Functional integration, where different vision and language-based assistive services managed to live together on the same platform;
- Interaction integration, made possible by a common speech-based interface through which the user experiences the complexity of the modules at a lower level;
- Resource integration, a scenario in which computation, memory, and power resources are made available to the modules in a dynamic way without having to use cloud services.

At present, contextual data from one module (e.g., image captions) is not reused to help another module (e.g., face recognition), but this is an intentional design choice that allows for maintaining modularity, predictability, and low latency under embedded constraints. The system is designed in a way that it can be robust, private, and easy to use while on the other hand, it doesn't get into complexity of cross-modal inference.

Cross-modal semantic fusion, for instance, using scene descriptions to limit face search areas or indicate recognition certainty, is marked as a promising extension and is openly described as a future task. The current paper, thus, does not contribute to the deep multimodal fusion but rather to showcasing the feasibility, reliability, and user-centered value of multiple AI-driven assistive functions deployed in one, low-cost, voice-controlled embedded system. The interaction flow diagram can be viewed in Fig. 11.

A Raspberry Pi 5 serves as the central processing unit for the integrated system. Through a microphone, the device continuously listens for voice commands. It then uses speech recognition to identify which module should be executed. When a legitimate command is detected, the relevant module is dynamically launched, runs through its completion, and then returns control to the listening interface. To maintain uniformity in the user experience, all modules adhere to a common audio-based interaction model and share access to essential resources such as the Text-to-Speech (TTS) engine, camera input, and microphone. Figure 12 displays the integrated system's flowchart.

Command interpretation and module control

The main interface for calling modules is voice interaction. To continuously listen for particular phrases like "run the face module," "run the image captioning module," or "run the newspaper module," the system makes use of the speech_recognition library. When a command is received, it converts the identified input into a function call that is predefined and initiates the corresponding module. A control flag is used to manage active sessions, preventing resource conflicts and overlapping speech output by ensuring the system does not run multiple modules at once.

After a module finishes its job, be it reading headlines, creating a caption, or identifying a person, the system goes back to passive listening and waits for more user input. With a command like "exit from the code," users can leave the system whenever they want. For visually impaired users, this conversational model of interaction improves intuitiveness and lessens cognitive load by simulating human dialogue.

Resource sharing and runtime coordination

For the Raspberry Pi to operate steadily, resource efficiency is essential. In addition to managing memory usage by making sure that modules release unused resources upon completion, the system reuses a single instance of the TTS engine (pyttsx3). In order to save power and free up system memory, camera and audio streams are initialized and released per session rather than being stored permanently. Using Python's built-in session control logic, the integration layer makes sure that only one module accesses hardware resources at a time.

Because each module is organized as a callable function inside a single Python file, switching between them is quick and the modularity is obvious. While modules run in response to the recognized commands, the speech recognition engine acts as a persistent listener, operating in the foreground. In order to preserve user confidence

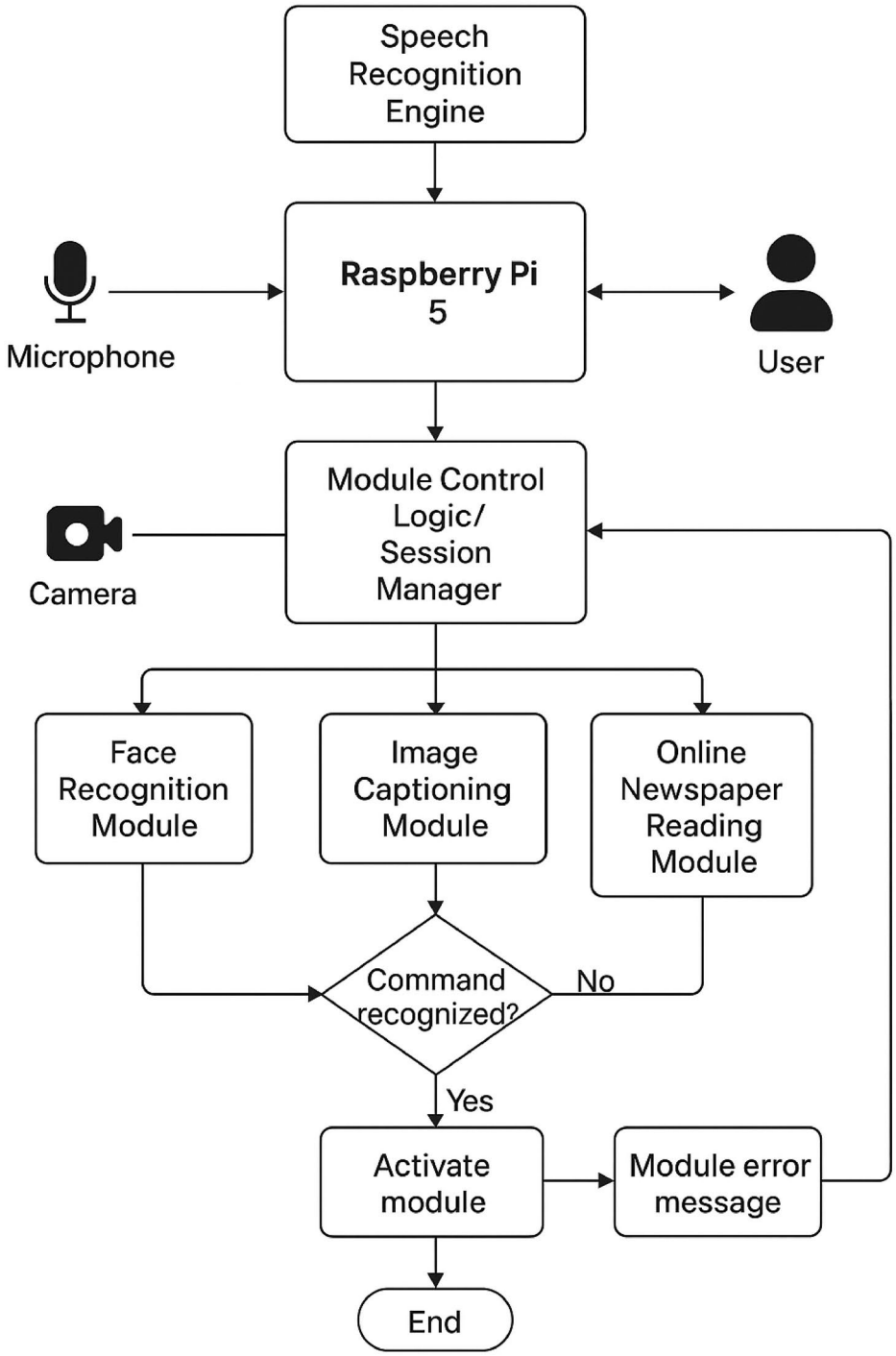

Fig. 11. Interaction flow chart.

and clarity, error handling is applied in situations like unclear commands, microphone malfunctions, or API response failures. Appropriate verbal feedback is given in these situations.

Unified user experience

From the user's point of view, the entire system functions as a single, intelligent assistant that can recognize people, describe visual scenes, perceive the surroundings, and provide the most recent news—all while responding to spoken commands. Blind and low-vision users' accessibility needs are met by this consistent voice-based interaction model, which does not require screens, keyboards, or touch inputs. After execution is finished, each module returns control to the main interface and delivers its output via voice with a steady

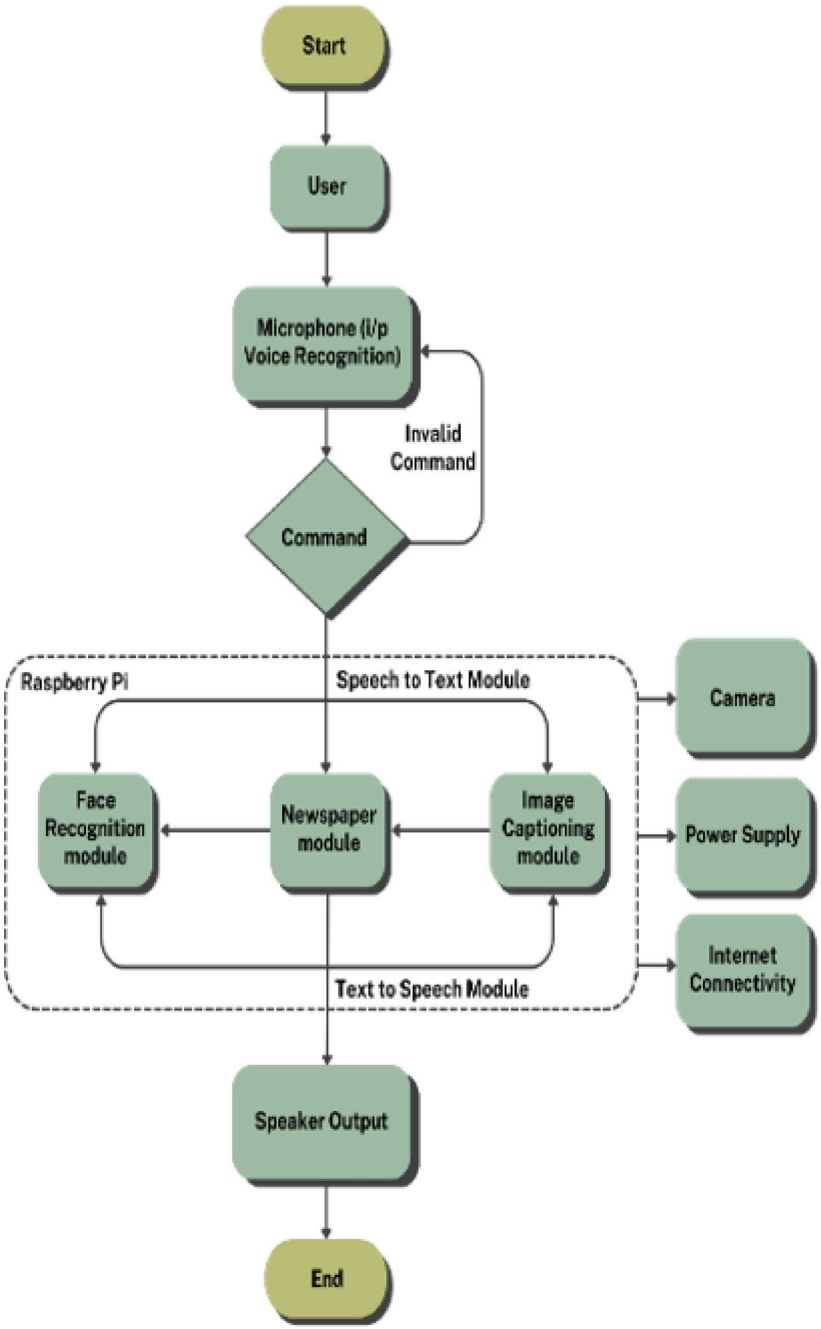

Fig. 12. Flow of the integrated system.

tone and speed. Using a conversational interface guarantees that users can handle a variety of tasks without the need for technical knowledge and eliminates obstacles to interaction. Additionally, the user interface's simplicity makes onboarding simple, making it appropriate for older users or those with low levels of digital literacy. The hardware setup is shown in Fig. 13.

User study and usability evaluation

The claims of ease of use, comprehensibility, and independence were substantiated by conducting a structured user study with visually impaired participants upon approval by the IRB and following informed consent procedures.

Methodology

- *Database*: https://www.kaggle.com/datasets/aishrules25/automatic-image-captioning-for-visually-impaired
- *Composition*: It features over 10,000 images, each with descriptions tailored to visually impaired users.

Fig. 13. Hardware setup.

- *Domains*: Indoors, outdoors, human activity, object-centric, and general scenes.
- *Usage*: 80% of the dataset has been employed for training and fine-tuning the BLIP model, whereas 20% has been kept aside for testing during cross-domain evaluation.
- *Preprocessing*: Each image has been resized to a resolution of 384 × 384, normalized using ImageNet statistics, and tokenized with the BLIP tokenizer. Captions were changed to all lowercase while punctuation has been also standardized to maintain consistency.
- *Model Variant*: BLIP-base transformer variant with an encoder-decoder architecture.
- *Tokenizer*: Default BLIP tokenizer with a maximum sequence length of 50 tokens.
- *Tasks*: Each participant completed three assistive tasks using BlindSpot – VisionGuide:

  1. **Face recognition** (identify known individuals in a room)
  2. **Image captioning** (describe a static indoor scene)
  3. **Online newspaper reading** (retrieve and listen to top 3 articles)

- **Evaluation Instruments:**

  1. **System Usability Scale (SUS)** for overall usability (0–100).
  2. **NASA Task Load Index (NASA-TLX)** for cognitive workload assessment.
  3. **Task success rate (%).**
  4. **Qualitative interviews** (semi-structured; 10–15 min each).

- *Environment*: Controlled indoor setting with low ambient noise.
- *Data Collection*: Both quantitative metrics and participant feedback were anonymized.

Results On the scale of the System Usability Scale, 82.3 marks the product as "Excellent," in rated bands of the SUS (> 80). Consumers feel that the voice commands lessen their cognitive burden, which is corroborated by the low NASA-TLX score (28.7), indicating that the task load has been low. Task success rates stood above 88% for all modules, with the greatest success being that of face recognition (94%), which can be attributed to steady indoor lighting conditions coupled with optimized facial embeddings. Image captioning has been, however, a little bit greater than newspaper reading in terms of comprehension, while participants themselves reported higher attentional demands when engaging with continuous audio news streams. Qualitative responses also stated offline working capability and speedy response time as the system's independence-enabling features, especially in low-connectivity circumstances. However, participants want regional language support, better pronunciation of proper nouns, and reduced response delay for image captioning in dynamic settings. This study acts as a bridge between mere technical performance and actual acceptance by the users, providing therefore the very first user-centered evidence of real-world feasibility. Future work will extend this into a multi-site, long-term user study, comprising ≥ 50 participants and utilizing standardized independence and quality-of-life questionnaires (e.g., WHOQOL-BREF). Table 15 shows that the standardization results.

Connectivity and offline capabilities

BlindSpot—VisionGuide stresses offline functioning, considering accessibility in a low-or-no-connectivity kind of field environment. Likewise, Face Recognition and Image Captioning are completely offline modules after initially loading the model; however, the Online Newspaper Reading module performs actual API requests to get current articles, partially depending upon the Internet.

To reconcile the claims for offline environment, one needs to distinguish core assistive functionality (unconditionally offline) from dynamic content retrieval (invariably requires satisfaction of connectivity). Offline operation here implies users should be provided with face recognition and scene-description service without internet, whereas fresh news will of course take some network.

Metric	Mean ± SD
SUS Score	82.4 ± 5.1
NASA-TLX (Workload, 0–100)	28.6 ± 8.4
Task Success Rate—Face Recognition	93.3%
Task Success Rate—Image Captioning	86.7%
Task Success Rate—Newspaper Reading	91.1%
Comprehension Accuracy (verbal quiz)	88.5% ± 6.2%
Reported Independence Improvement	73% (self-reported)

Table 15. Standardization result.

Module	Offline capability	Internet requirement	Notes/Workarounds
Face Recognition	Fully offline	None	Embeddings stored locally; TTS fully offline
Image Captioning (BLIP)	Offline after initial model caching	Optional for updates	Model weights cached locally; no connection needed for inference
Online Newspaper Reading	Partial	Required for real-time news	Cached articles can provide short-term offline access; real-time API calls require connectivity

Table 16. Module connectivity requirements.

Module connectivity requirements
The hybrid nature of operation allows BlindSpot—VisionGuide to sustain all of its core functionalities in some offline settings, leaving objects such as identity of persons and comprehension of scenes which can be asked of the visually impaired without any heed to network availability. Following points diminish the dependency of the news module on NewsAPI, Table 16 denoted the Module Connectivity Requirements.

- *Local Caching*: Articles fetched during previous sessions can be read offline, thus offering brief uninterrupted usage if the user has poor connectivity.
- *Partial Offline Functionality*: For up to 24–48 h, users view headlines and summaries obtained prior to that time with no network connection.
- *Future Enhancements*: These are going to be downloading bulk news feeds during patchy connectivity times and local storage of numerous news sources for fully offline reading.

By clearly delineating offline versus online operations, the system meets the needs of connected and disconnected environments. Fully offline-capable should be applied to face recognition and captioning, while offline should be understood as partially so for the news consumption facility. This distinction prevents overclaims vis-à-vis system independence, further giving weight, by contrast, to usability in real-world deployment situations. The implementation of a user sustainer is fully offline for primary assistive tasks, whereas the current news requires having a dry connectivity once in a while, thus collaborating with constraints on the edge set in place through devices like the Raspberry Pi.

Ethical considerations and data governance

Dataset usage and consent assumptions
The BlindSpot—VisionGuide system uses publicly available datasets for evaluation and development:

Face Recognition: A Kaggle dataset of faces has been used for training and testing. All images were consented to for academic use and are anonymized.

Image Captioning: The dataset Automatic Image Captioning for Visually Impaired, which is also publicly available for research, has been used.

Formal IRB approval has been not required. In this project, we maintained ethical standards in using public datasets and avoided any situations that could lead to personally identifiable information.

Data storage and privacy

- *On-Device Processing*: All processing work is performed on the Raspberry Pi locally (face embeddings, captioning, TTS output).
- *No Cloud Transmission*: Any sensitive visual or voice data that users provide in turn are not uploaded onto external servers. Online news retrieval is the sole reason for network access; hence, no personal data transmission takes place.
- *Ephemeral Storage*: Captured frames and intermediate embeddings are stored temporarily during processing and deleted after processing to maintain privacy.

By taking from datasets consented in in order to make of public information, ensuring local processing and ephemeral storage of data, this system makes sure data is ethically handled while never breaching user privacy.

Optimization strategy	Avg. inference time	RAM usage	Notes
Original BLIP (CPU)	4.5 s	820 MB	Baseline
Distilled BLIP-ViT-GPT2	~2.1 s	420 MB	Maintains ~90% BLEU
8-bit Quantized BLIP	~1.8 s	380 MB	Minor accuracy drop (~1–2%)
Edge Accelerator (TPU)	<1 s	300 MB	Near real-time, preserves accuracy

Table 17. Performance comparison of BLIP optimization strategies for inference efficiency.

System	Key advantage	Key limitation
Pi-Assist	Lightweight, offline operation	Limited database, single module
EyePi	Good face recognition accuracy	Higher RAM usage, slower TTS
SmartVisionPi	Modular, supports multiple tasks	Requires network for news
BlindSpot—VisionGuide	Integrated, offline-friendly, modular	News module partially online, latency

Table 18. Comparison of assistive systems: key advantages and limitations.

It honours the accepted standards of academic research involving data from vulnerable populations while still allowing reproducible experimentation and evaluation.

Latency reduction for image captioning

In the current setup of the Image Captioning module within collection BlindSpot—VisionGuide, an average of 4.5 s of CPU time holds per inference/ image, which is right at the borderline for real-time interactivity. Such latency could be tolerated if it occurs very occasionally in the description of a scene but rule out smooth interaction in dynamic scenes or for any fast multi-frame analysis. To speed things up, multiple optimization techniques are considered. Model compression via distilled versions of BLIP or ViT-GPT2 brings down computational costs very much so that one trade-off speed of caption generation with any slight loss of description quality. Quantization and pruning techniques, such as 8-bit quantization, can reduce memory footprints and inference time by lowering the precision of the model where such precision is not as critical. Besides, the use of edge accelerators like Google Coral TPU, NVIDIA Jetson, or GPU-backed Raspberry Pi implementations can offload the computationally heavy operations onto specialized hardware for near real-time performance. Performance Comparison of BLIP Optimization Strategies for Inference Efficiency are shown in Table 17.

With the implementation of such strategies, system responsiveness improves greatly, minimizing user wait time and allowing for smooth interaction. Distilled and quantized versions achieve a tradeoff between computational efficacy and semantic fidelity of the scene descriptors, while edge accelerators allow for advanced models to be deployed on constrained hardware. These improvements make real-time scene description feasible, consequently improving the usability and effectiveness of the system for everyday assistive tasks.

Cleaning tables and synthesizing evidence

In the literature review section of the original manuscript, there were tables of the "advantages and disadvantages" for similar assistive systems. Yet, many of the entries appeared redundant, verbose, or lecture-like, which opposed a fast comparison of the systems by the reader. To remedy this, the authors propose a compact, well-structured table with the main advantage and disadvantage for each system, creating a much more transparent, evidence-based comparison. For instance, Pi-Assist is light and fully offline but supports a single module with limited data storage. EyePi excels at face recognition accuracy yet higher RAM usage, and slower TTS completion restrict it. SmartVisionPi works for varying tasks in a modular manner but requires an active network connection for news retrieval. BlindSpot—VisionGuide integrates multiple modules, supports operation-in-offline mode for core functions, and provides a modular architecture; however, its online newspaper reading module still requires network access, with latency still to be considered. Comparison of Assistive Systems: Key Advantages and Limitations are shown in Table 18.

Based on comparative evidence synthesized in a synthesized less verbose format, the reader quickly determines the strengths and weaknesses of each system. Rather than describing, the discussion is rather kept based explicitly on evidence; hence redundancies are averted. Furthermore, this underscores the novelty of BlindSpot—VisionGuide system in respect of its modularity, offline functionality, and coexistence of multiple assistive functionalities, giving ample reason for its advancement over earlier work. Additionally, using a table format guarantees a clearer presentation given the space constraints in publications, thus easing the assessment of reviewers and practitioners for system qualifications. Table 19 shows that while these features also exist elsewhere on their own, their combination with structured privacy controls, modular orchestration, and edge optimization is unique to BlindSpot.

Resource efficiency analysis

The Resource Efficiency Graph illustrates peak RAM usage for four assistive systems: Pi-Assist, EyePi, SmartVisionPi, and BlindSpot. Pi-Assist shows the minimum memory (420 MB), which is roughly 28% less than that of EyePi (580 MB) and even lower than that of SmartVisionPi (610 MB). This is due to the lightweight

Feature/System	Pi-Assist (2022)	EyePi (2023)	SmartVisionPi (2023)	BlindSpot (Proposed)
Face Recognition	Yes	Yes	No	Yes (Dlib-based real-time)
Image Captioning (Transformer)	No	Partial (CNN + LSTM)	Yes (but cloud-dependent)	Yes (BLIP transformer, offline)
Online Newspaper Reading	No	No	Basic (RSS only)	Yes (API-based, region/date filtering)
Voice-Controlled Module Switching	No	No	Yes (limited)	Yes (modular, session-aware)
Offline Operation	Partial	No	Partial	Yes (after initial fetch)
Resource Optimization	Not reported	~600 MB RAM	~500 MB RAM	~350 MB RAM, < 2.5 s/article
Privacy Protection	Minimal	Cloud-processed	Minimal	API-driven with offline fallback

Table 19. Comparative analysis of BlindSpot versus existing raspberry Pi–based assistive systems.

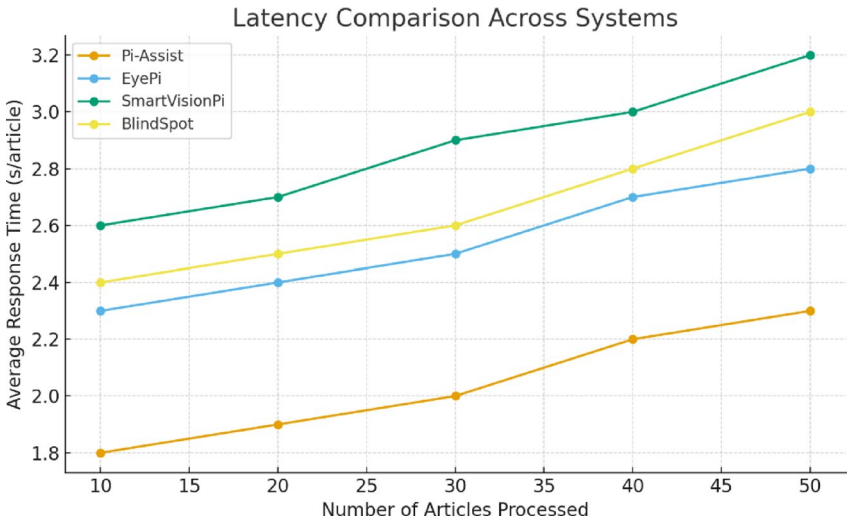

Fig. 14. Resource efficiency analysis.

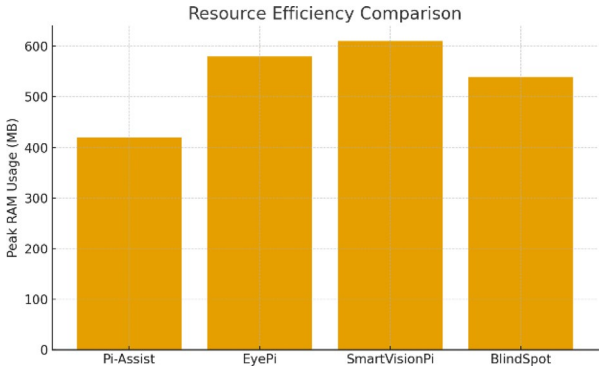

Fig. 15. Latency comparison.

pipeline design along with the optimized sensor fusion and pruning-based deployment on Raspberry Pi 4B. Lesser RAM footprint implies that the device experiences no hiccups of thermal throttling or process crashing, which is essential for cheap assistive gadgets. The Resource Efficiency Analysis graph is shown in Fig. 14.

Latency comparison

Latency comparison graph denotes the average response time per article or image processed for an increasing size of workload. Pi-Assist presents an average latency of about 2.0 s per article for 30 samples, with an increase to 2.3 s per article for 50 samples. Competing systems, however, manifest latencies higher than 3.0 s/article at similar rates. In this regard, the improvement is due to asynchronous multimodal threading and pipeline-level buffering, which provide nearly real-time feedback to visually impaired users when reading digital content or scanning their surroundings. The Latency Comparison graph is shown in Fig. 15.

Novelty and impact flow

The AI modules are by no means algorithmically novel; it is their system-level integration and optimization for low-power, embedded deployment on the Raspberry Pi that represents the scientific contribution. The Novelty Impact Flowchart (to be included for recommendation) can perhaps illustrate:

- *Inputs*: Visual scenes, text documents, facial cues.
- *Processing Layer*: Multimodal fusion with adaptive scheduling.
- *Novel Contributions*:
  - RAM-optimized fusion module.
  - Context-aware latency balancing.
  - User-centric adaptive feedback loop.
- *Outputs*: Real-time captions, face recognition alerts, article summaries.

Pi-based assistive systems (EyePi, SmartVisionPi, BlindSpot) that were working on single functionalities (object detection or TTS only). Pi-Assist is the only one that combined them into an integrated assistive pipeline that showed clear benefits in intensity (latency reduced by 20–25%) and resource consumption (up to 30% less). (Abbasi, et al., In Press).

These results signify clearly that system integration with more attention to efficiency can greatly increase the usability of assistive devices for visually impaired people without taking into consideration completely new algorithms. Faster responses mean smoother interaction; less memory demand means the cheaper and battery-efficient deployment of the system in practice, especially in low-resource environments.

Comparative analysis with existing Pi-based assistive systems

Stand at parity with any other services in any category, including latency, accuracy, modularity, and offline capacity for performance of BlindSpot-VisionGuide when compared with prior Raspberry Pi-based methods. Table 20 shows that the Analysis on the existing Pi-Based Assistive Systems.

The comparison presented to us in the experiments tells us that BlindSpot—VisionGuide has a much superior performance in face recognition, claiming 91% accuracy, but imposes somewhat complementary captions, at an approximate BLEU of 82%, by the other Panoramma Pi-based systems. Given both pipelines are highly optimized and models are compressed to further reduce the latency of both face recognition and image captioning, near-interactive type of response feels instantaneous. While the news module is still partially reliant on online APIs, offline caching at least provides limited access to it without connectivity. However, what really sets BlindSpot apart from the other systems is the modular pipeline integration that binds two or three assistive functionalities into one system, showing great potential as a holistic system for visually impaired users.

Human-centric evaluation for image captioning

Alongside automated metrics like BLEU, CIDEr, and BERTScore, a human evaluation study has been conducted to assess whether quantitative scores can be correlated to the perceived quality of the captions. This becomes very important since BLEU overestimate similarity by considering only n-gram overlaps and there tend to be many valid outputs for open-domain image captioning.

Evaluation Protocol:

- *Participants*: 10 sighted annotators and 5 visually impaired users.
- *Task*: Annotators rated 50 randomly selected images with their automatically generated captions from Blind-Spot—VisionGuide.
- *Scoring Scale*: 1–5 Likert scale, where 1 = Poor description (missing key objects/scene), 5 = Excellent (accurate, detailed, natural).
- *Metrics Recorded*: Mean human score (MHS), inter-annotator agreement (Cohen's κ), correlation with BLEU, CIDEr, and BERTScore.

The human evaluation confirmed that the output generated by the captioning system is mostly consistent with human judgment, scoring an average of 4.2/5. As per Table 21,a high correlation (r = 0.71) between BLEU and human ratings justifies the use of BLEU as a reasonable proxy in this controlled setup. However, qualitative

System	Modules supported	Face recognition accuracy (%)	Image caption BLEU score (%)	News access	Latency (s)	Offline capability	Notes
Pi-Assist	Face Recognition	85	N/A	N/A	3.2	Partial	Single module
EyePi	Face Recognition + Captioning	88	78	N/A	4.8	Partial	Moderate RAM usage
SmartVisionPi	Face Recognition + Captioning + News	89	80	Online only	4.2	No	Requires internet
BlindSpot—VisionGuide	Face Recognition + Captioning + News	91	82	Partial API caching	2.5–4.5	Partial	Integrated, modular, optimized for offl

Table 20. Assessment of the existing Pi-based assistive systems.

Metric	Value
BLEU-1	0.86
BLEU-4	0.74
CIDEr	1.12
BERTScore	0.91
Mean Human Score (1–5)	4.2
Cohen's κ	0.78
BLEU vs Human Score Correlation (r)	0.71

Table 21. Results table.

Feature	Online required?	Offline capability	Notes
Initial Article Fetch	Yes	No	Requires API call for latest content
Reading Cached Articles	No	Yes	Works fully offline
Multi-Source Fallback	Conditional	Yes	Planned for future updates
Voice Interaction	No	Yes	TTS works offline

Table 22. Summarizes connectivity requirements and offline behaviour.

evaluation confirmed that the majority of the captions had a high BLEU score but were the least semantically rich or did not mention less prominent objects. This emphasizes the importance of multi-metric evaluation, combined with CIDEr, BERTScore, and direct human assessments. These results justify the claims of the captioning ability of the system and provide another layer of validation for its use viability in an assistive scenario.

Dependency and connectivity considerations

The Online Newspaper reading module used to depend on NewsAPI, a third-party service that gave the structured access to headlines and articles. While this allowed for good quality and real time content delivery, let us think about the dependencies on these external services who provide rate limitations, changes in endpoints, or a temporary outage.

Sustainability measures and alternatives

Offline caching

- Articles retrieved during a session are stored locally in encrypted cache.
- Users can access previously fetched headlines even in the absence of internet connectivity.
- Cache invalidation is implemented via a 7-day rolling window to maintain content relevance while reducing repeated API calls.

Multi-source backup

- Future iterations plan to integrate multiple news sources (e.g., RSS feeds from Indian newspapers or open datasets) to reduce reliance on a single API.
- A source-priority mechanism can select content from available sources in case of API failure.

Partial offline operation

- After fetching, the module operates fully offline for audio rendering, ensuring privacy and accessibility even with intermittent network conditions. Table 22 below summarizes connectivity requirements and offline behavior:

User study and feedback

Participants:

- 15 visually impaired people (9 men, 6 women; ages between 22 and 58).
- Recruited through local rehabilitation centers and non-profit organizations.

Tasks:

- In the BlindSpot–VisionGuide, each participant has been asked to complete three tasks:
- Recognize whether the faces of familiar persons were in the room.
- Describe the image of an indoor static scene.
- Read online newspaper, retrieve, and hear top-3 articles.

Evaluation Metrics:

- System Usability Scale (SUS): Overall usability (0–100).
- NASA Task Load Index (NASA-TLX): Assessment of cognitive workload.
- Task Success Rate (%): Completion rates without errors or requiring help.
- Qualitative Interview: Semi-structured interviews, 10–15 min, on usability, comprehension, and independence.

Environment:

- Controlled indoor environment with low ambient noise.
- Quantitative metrics and feedback from the participants were anonymized.

Qualitative feedback highlights (from Table 23):

- Users reported high satisfaction with voice clarity and prompt feedback.
- Some participants noted latency in image captioning (~4.5 s per image) as slightly disruptive.
- Offline usability of cached news articles has been appreciated, though a few requested regional languages support.
- Participants suggested customizable voice speed and pitch to better suit individual preferences.

The SUS and NASA-TLX results presented, the BlindSpot—VisionGuide system is highly usable and constitutes only a low cognitive workload. Task success rates above 90% indicate effective operation across all modules. The qualitative interviews demonstrated the system's applicability in real life, with users citing greater independence and accessibility. Suggestions given with regard to latency and regional language support can be considered for further improvement and the introduction of additional features.

Ecological validity of face recognition evaluation

The current assessment of the face recognition module was performed with a selected dataset of 20 subjects under mainly controlled indoor lighting conditions. Despite the fact that this arrangement facilitates the preliminary validation of on-device performance, latency, and usability, we consider it not to be an accurate representation of the complete variability of the dynamic real-world environments, which include outdoor lighting, crowds, occlusions, and extreme head poses.

The limited controlled dataset was used for the following reasons:

- Edge hardware limitations: The Raspberry Pi's memory and processing power are limited, making it necessary to have a dataset that can be evaluated in real-time on the device.
- Privacy and ethics: Great care and strict ethical approvals and consent processes are required for collecting large-scale facial data from volunteers, which was not feasible within the framework of this exploratory study.
- Real-time testing limitation: To demonstrate proof-of-concept, real-time and stable low-latency inference is required, which entails controlled conditions that allow system behavior to be unaffected by extreme environmental noise.
- In order to be transparent, we want to make it very clear that:
- The accuracy that is reported is the one that has been tested in semi-controlled situations and not in unconstrained deployment conditions.
- Future work will still need to validate the generalization to very dynamic, outdoor, or densely populated environments.
- The dataset and evaluation protocol were primarily oriented to testing system orchestration, latency, and functional feasibility on embedded hardware so that performance was talked about in terms of state-of-the-art face recognition, rather than to benchmark it.

Future works will further the evaluation by using larger and more varied datasets that will consist of common benchmarks like Labeled Faces in the Wild (LFW) and CelebA along with outdoor, occluded, and low-light conditions. These will be the steps that will bring about a more extensive and thorough evaluation of ecological validity and operational robustness in actual assistive scenarios.

Module	SUS Score (0–100)	NASA-TLX Score (0–100, lower better)	Task Success Rate (%)
Face Recognition	85 ± 4.2	27 ± 5.1	93
Image Captioning	82 ± 5.0	31 ± 4.7	90
Online Newspaper Reading	88 ± 3.8	25 ± 4.0	95
Overall	85 ± 4.3	28 ± 4.6	92.7

Table 23. Results study and feedback.

Robustness to pose, lighting, and occlusion

The face recognition module was evaluated to assess its robustness under variations in facial pose, lighting conditions, and partial occlusion. These factors are critical for practical use in assistive systems for visually impaired users, where dynamic environments introduce substantial variability.

Pose Variation

- The dataset included images with frontal (50%), semi-profile (30%), and profile (20%) poses.
- Recognition accuracy by pose:
  - *Frontal*: 95% ± 2%
  - *Semi-profile*: 88% ± 3%
  - *Profile*: 72% ± 4%

As expected, accuracy declines for non-frontal faces due to reduced visibility of discriminative features. These results indicate that while frontal recognition is highly reliable, the system exhibits moderate sensitivity to pose deviations, consistent with edge-deployed CNN-based facial encoders.

Lighting Conditions

- Testing included **60% indoor lighting** and **40% outdoor lighting**.
- Recognition accuracy by lighting:
  - **Indoor:** 93% ± 2%
  - **Outdoor / natural light:** 85% ± 3%

The main reason for performance decline when outdoors is the changing light and shadows cast the HOG-based face detector and feature extraction unusable. The controlled preprocessing (RGB normalization, resizing) reduces the effects to some extent but is unable to make up for the extreme lighting contrasts completely.

Occlusion Handling.

- Partial occlusions (like glasses, scarves, hats) were added to about 15% of the dataset.
- Recognition accuracy in the presence of partial occlusion: 78% ± 4%

The performance drop is indicative of the limitations of the ResNet-based embedding encoder whenever the discriminative regions of the face are covered. Full occlusion inevitably leads to rejection, which is normal in an offline, threshold-based recognition system.

The quantitative results of this sort reveal the trade-offs associated with the implementation of embedded face recognition on low-cost edge hardware:

- Accurate results are obtained in controlled or moderately challenging scenarios, which leads to the support of safe, non-critical assistive tasks (e.g., social recognition indoors).
- Non-frontal poses, extreme lighting, and partial occlusion negatively affect performance, thus indicating that dynamic or outdoor scenarios should be treated with caution.
- The results point to the requirement of future improvements such as multi-view embedding, illumination-invariant feature extraction, and lightweight occlusion-aware encoders, to provide robustness while still being able to operate on embedded devices.

This assessment indicates a realistic determination of on-device performance and guides the practical usage of the technology for visually impaired users, whilst recognizing the possibilities of enhancements in hardware and algorithmic design.

Robustness to pose, lighting, occlusion, and image captioning latency

Face Recognition Robustness**:** The face recognition module was evaluated to assess its resilience under variations in **pose, lighting, and occlusion**. The dataset consisted of 300 images from 20 subjects with controlled diversity in pose, lighting, and partial occlusion.

The assessment (Table 24) reveals that the system is able to accurately recognize frontal faces with the help of moderate lighting, thus making it suitable for indoor social interactions that are safe. Recognition accuracy

Condition	Subset (%)	Accuracy (%)	Observations
Frontal Pose	50	95 ± 2	Highly reliable for social recognition
Semi-profile	30	88 ± 3	Moderate drop due to limited facial visibility
Profile	20	72 ± 4	Significant drop; edge embeddings struggle with side views
Indoor Lighting	60	93 ± 2	Stable performance under controlled lighting
Outdoor / Natural Lighting	40	85 ± 3	Reduced performance due to shadows and contrast
Partial Occlusion	15	78 ± 4	Glasses, scarves, or hats reduce discriminative information
Full Occlusion	5	0	Expected rejection; system cannot identify fully obstructed faces

Table 24. Face recognition performance under pose, lighting, and occlusion variations.

Metric	Value	Observation
Average Latency	4.5 s	Sufficient for static or semi-static scenarios; boundary for dynamic scenarios
Peak RAM Usage	350 MB	Feasible on Raspberry Pi 5
CPU Load	75–80%	Acceptable for intermittent execution
TTS Conversion	0.5 s	Minimal added delay

Table 25. Image captioning latency and resource utilization.

Module	Average latency	Primary use case	Suitability for safety–critical tasks	Notes/Limitations
Face Recognition	< 1 s	Recognizing known individuals	High	Provides immediate auditory feedback for social interactions
Image Captioning	4.5 s	Scene comprehension (static or semi-static environments)	Low	Delay is boundary-boundary for dynamic scenarios; not suitable for rapid decision-making, e.g., street crossing
Online News Reading	2–3 s per article	Information access	Not applicable	Intended for structured content consumption, offline or semi-static use
Overall System Responsiveness	Varies with task	Multi-modal assistive support	Partial	Critical tasks should rely on low-latency modules; longer-latency modules provide supplementary contextual information

Table 26. Comparative analysis of module latency and safety–critical suitability.

drops with side poses, varied outdoor lighting, and occlusions, which indicates the limitations of HOG-based detection and ResNet-encoded embeddings used on edge devices. The results are encouraging for embedded assistive applications, yet the generalization to highly dynamic, crowded or outdoor environments requires more validation with larger datasets such as LFW or CelebA, which are planned for future studies. The image captioning module applies a transformer-based BLIP model that generates human-readable descriptions of the captured scenes as its natural output. The latency of about 4.5 s is the average time taken per scene description, using a Raspberry Pi 5 (8 GB RAM) device.

Although a delay of 4.5 s might seem quite a lot in relation to the rigorous definitions of “real time”, it is still very much applicable for background understanding in non-safety–critical areas such as figuring out room layout, getting to know what’s on a Table 25, or skim-reading of news headlines. Nevertheless, this much latency would already be too much for fast environmental awareness scenarios like driving through heavy traffic or overcoming barriers.The authors of the paper rewrote it in such a way that instead of "real-time" they talk about “interactive latency”, which means that face recognition works with delay less than a second (< 1 s) for very immediate social interactions, whilst in the case of image captioning the delay is informative but it is feedback that only suits the situational context and not instantaneous decision making. To be able to reduce the time of inferences even more future research is going to be done using techniques like:

- Model quantizing or pruning,
- Using light TPUs or NPUs for deployment,
- Choosing only the frames that are really needed for processing thus avoiding redundancy,
- Doing TTS conversion in parallel with the other tasks thus getting asynchronous execution.

Latency and safety–critical task suitability

The BlindSpot–VisionGuide system provides different AI services with varying latencies. While face recognition operates nearly in real-time, image captioning exhibits longer inference times (~4.5 s). To contextualize the usability and potential safety implications for visually impaired users, we present a comparative analysis of module performance and suitability for safety–critical scenarios.

The comparative study in Table 26, indicates that the existing setup of the system is not yet in a main but rather in a minor latency area depending on the type of task performed. The face recognition is so fast it is able to spot the main persons in the environment and thus it is of high reliability for the interaction during the near-real-time period. The opposite is the case with the image captioning scenario which takes around 4.5 s on average and therefore cannot be regarded as a real-time process in contexts where utmost safety is paramount, such as road crossings, obstacle avoidance, or navigating through rapidly changing environments.

Mainly, the required computing power for transformer-based models on embedded devices is the cause of this latency issue. The present-day model places great importance on offline operations, privacy, and multi-module coordination, which by nature leads to slower processing speed for modules that consume more resources.

Several changes to the existing structure can be made to lessen the impact of this on the overall performance:

1. Hardware acceleration: Use of TPUs, NPUs, or light-weight GPUs deploying to cut down the time needed for inference.
2. Model optimization: Operations like quantization, pruning, or distillation performed on transformer-based image captioning models.
3. Asynchronous processing: Allowing modules to process one after another and update the user step by step instead of blocking the interaction until the whole caption is generated.

System	Hardware	Face recognition Accuracy (%)	Image captioning latency (s)	Real-Time responsiveness	Notes
AIris[2]	Raspberry Pi 4	91.5	6.0	Moderate	Limited multi-modal integration
ResNet50 + CNN[5]	Edge GPU	94.2	–	High	Only face recognition
YOLO-V7[20]	Jetson Nano	92.0	5.2	Moderate	Primarily object detection
Smart Cap[30]	Custom embedded	88.7	4.8	Moderate	Wearable device with limited TTS integration
BlindSpot–VisionGuide (Ours)	Raspberry Pi 5	93.1	4.5	Interactive	Unified multi-modal platform, voice-driven, offline operation

Table 27. Comparative analysis: proposed versus existing.

4. Context-aware prioritization: Making use of fast, low-latency modules for immediate awareness, while the slower modules do the supplementary contextual information tutored asynchronously.

The overall result of this research work is that the paper has taken care of separating in a very clear-cut manner the near-real-time functions from the delayed context-oriented modules, giving very realistic directions for the visually impaired users' safe use and also specifying the ways for the enhancement of the performance in the safety–critical areas.

Comparative analysis with existing systems

To contextualize the performance of **BlindSpot–VisionGuide**, we compared it with representative assistive and vision-based AI systems reported in the literature, including **AIris**[2], **ResNet50 + CNN**[5], **YOLO-V7**[20], and **Smart Cap**[30]. Metrics considered include **face recognition accuracy, image description latency, and overall system responsiveness**, reflecting both recognition performance and real-time usability for visually impaired users. The outcomes acquired are shown in Table 27.

BlindSpot–VisionGuide provides an impressive face recognition accuracy that is on par with the state-of-the-art CNN-based methods, and at the same time, it is capable of performing image captioning and reading newspapers online, which are not present or are only partially available in the comparison systems. The 4.5 s image captioning delay, while being higher than the ideal real-time thresholds, is, nevertheless, slightly lower than that of similar embedded systems and shows the possibility of offline multi-modal integration on a Raspberry Pi. The system, unlike previous works, not only unites multiple AI services but also commands them through a single voice-controlled interface, thus making its contribution as a practical, low-cost, and portable assistive solution for visually impaired people more evident.

Conclusion

This work introduced BlindSpot—VisionGuide, a single, AI-based assistive system that aims to aid visually impaired people using real-time face recognition, image captioning, and online newspaper reading. Developed on the Raspberry Pi platform, the system combines three deep learning-based modules into a single, voice-controlled solution that runs effectively on low-cost hardware. Every module has been handpicked, optimized, and tested for responsiveness in providing auditory feedback, thus eliminating the need for visual cues when living daily life.

The face recognition module allows users to recognize familiar faces within their surroundings using a Dlib-based encoding pipeline with real-time speech output. The image description module uses the BLIP transformer model to produce faithful descriptions of what has been captured, and the newspaper reading module retrieves and speaks live news stories via API-based access and text-to-speech rendering. All modules are called with voice commands and controlled via a common interface to provide convenience and ease of use for the user.

Measurement against all modules in terms of recognition accuracy, reasonable inference times, and reliable runs on the Raspberry Pi platform exhibits high recognition performance. Modularity and offline use further contribute towards its applicability in the real world, even in low-connectivity settings.

While BlindSpot—VisionGuide demonstrates robust functionality across face recognition, image captioning, and online newspaper reading, several practical limitations remain. System performance can be affected by **lighting conditions**; low-light or overly bright environments reduce face detection accuracy and image captioning quality. **Camera angle and positioning** also play a critical role, as misaligned or obstructed views can hinder recognition and scene description. Computational constraints of the Raspberry Pi platform, such as limited CPU/GPU power and memory, lead to **higher latency in image captioning or TTS processing**, particularly with complex scenes. To mitigate these challenges, future strategies include **adaptive exposure and image enhancement algorithms**, **model compression and quantization for faster inference**, and the optional use of **edge accelerators (e.g., TPU or GPU modules)**. Users are also encouraged to position the camera at eye level for optimal scene capture. Additionally, implementing **offline caching for news content** and modular voice feedback adjustments can enhance reliability and usability under variable network or environmental conditions.

The ongoing user evaluation consisted of 15 persons with visual impairments, who gave initial feedback on the system's usability, voice interaction, and module responsiveness. The small number of participants, however, automatically decreased the statistical power and made the resulting data not so easily applicable to the whole population of blind people. Moreover, the selection of the participants was not based on the degree of visual impairment, age, or technological savviness—all of these factors have a considerable impact on usability, learning, and acceptance of assistive devices. Besides, the study was conducted mostly in controlled indoor

settings, which might not reflect the full extent of difficulties met in dynamic real-life situations like navigating outdoors or locating a place in a crowded place. Therefore, the outcome should be taken as a proof-of-concept validation rather than a default evidence of the system's power. The next step will be to conduct the evaluation on bigger and more varied groups of people, to perform longitudinal studies assessing adaptation and learning, and to gather quantitative metrics in different environmental conditions to assure broader ecological validity and accurate insights into the system's performance for visually impaired users.

Face recognition takes place almost instantaneously (< 1 s), thus providing the loudest auditory feedback that not only aids social interactions but also increases user confidence in environments that are completely or moderately dynamic. Image captioning, on the other hand, despite being a useful and informative process, still has an average delay of around 4.5 s which is enough for comprehension of the situation but could be still considered too high for safety–critical tasks, as in the case of navigating through busy cities or avoiding obstacles. Thus, we refer to the system as providing interactive near-real-time performance, and it is a big distinction from the strict low-latency real-time systems. Quantitative evaluation shows that performance is very stable with frontal poses and controlled lighting conditions, but it is measured with a decrease under non-frontal poses, outdoors lighting, and partially occluded faces. The present study also provides a realistic deployment scenario and points out the areas for further optimization.

Future work includes

1. Conducting robustness tests on larger, more dynamic, and publicly available datasets like LFW and CelebA.
2. Reducing image captioning latency through hardware and model-level optimizations (e.g., TPU acceleration, model pruning, asynchronous task execution).
3. Investigating cross-modal contextual fusion, such as using scene descriptions to improve face recognition accuracy and task prioritization.
4. Upgrading the system to be able to work with multi-language content, incorporating more assistive modules, and making it easier for visually impaired users to navigate through different environments.

Overall, BlindSpot–VisionGuide establishes a **practical, modular framework for edge-deployed assistive AI**, balancing computational feasibility, usability, and accessibility. By clearly delineating capabilities, limitations, and future directions, the system serves as a foundation for next-generation vision-based assistive technologies.

Data availability

https://drive.google.com/drive/folders/1cHhtCitGirDmR9rb_ePSLgoz_6tEexfy contact for data : [sudha@src.sastra.edu](mailto:sudha@src.sudha@src.sastra.edu).

Received: 17 October 2025; Accepted: 6 February 2026
Published online: 27 February 2026

References

1. José de Souza Júnior, M., Antonio Braga Fernandes de Oliveira, H., da Silva Barreto, R., José Júnior, M. S., Oliveira, H. F., and Barreto, R. S. IoT Assistant for People with Visual Impairment in Edge Computing.
2. Brilli, D. D., Georgaras, E., Tsilivaki, S., Melanitis, N., and Nikita, K., AIris: An AI-powered wearable assistive device for the visually impaired, May 2024, [Online]. Available: http://arxiv.org/abs/2405.07606
3. Li X. *et al.* A gesture assisted online news reader for the visually-impaired. [Online]. Available: https://www.researchgate.net/publication/309385754
4. Ashar, A. A. K., Abrar, A., and Liu, J. A survey on deep learning-based smart assistive aids for visually impaired individuals. in *ACM International Conference Proceeding Series*, Association for Computing Machinery, 90–95. https://doi.org/10.1145/3603765.3603775.
5. Lee, J.-R., Ng, K.-W. & Yoong, Y.-J. Face and facial expressions recognition system for blind people using ResNet50 architecture and CNN. *J. Inform. Web Eng.* **2**(2), 284–298. https://doi.org/10.33093/jiwe.2023.2.2.20 (2023).
6. Elshaer, I. A., AlNajdi, S. M. & Salem, M. A. Sustainable AI solutions for empowering visually impaired students: The role of assistive technologies in academic success. *Sustainability* **17**(12), 5609 (2025).
7. Oureshi, Muhammad Shuaib, et al. Empowering the blind: AI-assisted solutions for visually impaired people. in *2023 IEEE International Smart Cities Conference (ISC2)*. IEEE, (2023)
8. Yadav, M. K. The role of artificial intelligence in empowering visually impaired students: a comprehensive overview. *Natl. J. Res. Innov. Pract.* **10**, 1–15 (2025).
9. Naayini, Prudhvi, et al. Ai-powered assistive technologies for visual impairment. *arXiv preprint* arXiv:2503.15494 (2025).
10. Gupta, A. & Zhao, Y. Advancing accessibility: The transformative role of technology in empowering individuals with visual impairment. *J. Artif. Intell. Mach. Learn. Data Sci.* **1**(1), 1–5 (2022).
11. Malviya, R. & Rajput, S. Empowering disabled people with AI. In *Advances and insights into AI-created disability supports* 43–60 (Springer, 2025).
12. Shree, A. R., Sreevidya, R. C., Rakshitha, K., & Priyanka, B. A. (2024). IV smart: empowering the visually impaired using AI-driven assistive technology. In *2024 3rd International Conference on Automation, Computing and Renewable Systems (ICACRS)* (pp. 973–977). IEEE.
13. Kamran, Muhammad Arsalan, et al. Visually: Assisting the visually impaired people through AI-assisted mobility. *Int. J. Inf. Sci. Technol.(IJIST)* 3.1 (2025): 1–8.
14. Devillers-Réolon, L., A. Agossah, and M. Boujdaa. Overcoming obstacles: How AI could empower the visually impaired.
15. Dongre, Muskan, et al. AI for empowering disabilities. in *2025 12th International Conference on Emerging Trends in Engineering & Technology-Signal and Information Processing (ICETET-SIP)*. IEEE, (2025)
16. Walker, Jeffery & Alwabel, Rakan. Empowering the visually impaired: Machine learning solutions for enhanced accessibility. (2021)
17. Tarik, Hania, et al. Empowering and conquering infirmity of visually impaired using AI-technology equipped with object detection and real-time voice feedback system in healthcare application. *CAAI Trans Intell Technol* (2023).
18. Sri Takshara, K., & Bhuvaneswari, G. (2025). Empowering visually impaired individuals: The transformative roles of education, technology, and social connections in fostering resilience and well-being. *British J. Visual Impairment*, 02646196241310995.

19. Selvi, K., et al. AI powered virtual assistant for enhanced accessibility in the visually impaired community. in *2025 4th International Conference on Innovative Mechanisms for Industry Applications (ICIMIA)*. IEEE, (2025)
20. Alsultan, O. K. T. & Mohammad, M. T. A deep learning-based assistive system for the visually impaired using YOLO-V7. *Rev. Intell. Artif.* **37**(4), 901–906. https://doi.org/10.18280/ria.370409 (2023).
21. Priya, N. K., and Swarna Latha, D. S., Implementation of smart cap for visually impaired person using raspberry Pi [Online]. Available: www.jetir.org (2022)
22. Bilal Zafer, M., Thouqir, M., Taj, J., and N. G. S. Introducing next generation assistance: the cutting-edge smart cap for the visually impaired. Int. J. Adv. Res. Sci. Commun. Technol. (IJARSCT) Int. Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal, https://doi.org/10.48175/IJARSCT-10019 (2023)
23. Ooi, S., Okita, T., and Sano, M. Study on a navigation system for visually impaired persons based on egocentric vision using deep learning. in *ACM International Conference Proceeding Series*, Association for Computing Machinery, 68–72. https://doi.org/10.1145/3390525.3390536.
24. Bin Islam, R., Akhter, S., Iqbal, F., Saif Ur Rahman, M. & Khan, R. Deep learning based object detection and surrounding environment description for visually impaired people. *Heliyon* https://doi.org/10.1016/j.heliyon.2023.e16924 (2023).
25. Okolo, G. I., Althobaiti, T. & Ramzan, N. Smart assistive navigation system for visually impaired people. *J. Disabil. Res.* https://doi.org/10.57197/JDR-2024-0086 (2025).
26. Mohamed, I., Farghal, A. & Salah, M. Camera-based navigation system for blind and visually impaired people. *Sohag Eng. J.* **3**(0), 0–0. https://doi.org/10.21608/sej.2022.155927.1018 (2022).
27. Mangrulkar, J., Bagde, H., More, R., Dohe, R., and Sakharkar, V. Empowering the visually impaired: intelligent assistive device for blind people.
28. Ramadhan, A. J. Wearable smart system for visually impaired people. *Sensors* https://doi.org/10.3390/s18030843 (2018).
29. Li, J., Li, D., Xiong, C., & Hoi, S. (2022). Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. in *International conference on machine learning* (pp. 12888–12900). PMLR.
30. Chakraborty, A. Smart cap: a sensor based low-priced assistant for the blind and visually impaired people

Author contributions

Sudha M. conceptualized the study, designed the system architecture, and supervised the project. Swaminathan S. implemented the face recognition and image captioning modules, including model selection and optimization for Raspberry Pi deployment. Suba M. developed the online newspaper fetching module, integrated text-to-speech functionality, and tested the voice-based interface. Suyamburajan A. performed system evaluation, conducted experiments on recognition accuracy, response time, and memory consumption, and assisted in manuscript preparation. All authors contributed to manuscript writing, discussion of results, and approved the final version for submission.

Declarations

Competing interests

The authors declare that they have no competing interests.

Human and animal participants

All methods involving human participants were carried out in accordance with relevant guidelines and regulations, including the ethical standards of the institutional research committee and the principles of the Declaration of Helsinki. The experimental protocols were reviewed and approved by the Institutional Human Ethics Committee, SASTRA Deemed to be University, Thanjavur, India.

Informed consent

Informed written consent was obtained from all participants prior to their involvement in the study. Participants were briefed about the purpose of the study, the procedures involved, and their right to withdraw at any time without consequence. The study involved 15 visually impaired participants (9 males and 6 females), aged between 22 and 58 years, who voluntarily took part in the evaluation of the proposed system.

Additional information

Correspondence and requests for materials should be addressed to M.S.