



iJRASET

International Journal For Research in
Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 Issue: IV Month of publication: April 2026

DOI: <https://doi.org/10.22214/ijraset.2026.81423>

www.ijraset.com

Call:  08813907089

E-mail ID: ijraset@gmail.com

VISION: Visual Interpretation and Smart Integration on Jets on Orin Nano

Shrikanth NG¹, Aashish CA², Fawaz Shaikh³, Kailas Nad P⁴, Sujay SP⁵

Department of Artificial Intelligence and Machine Learning Alva's Institute of Engineering and Technology, Mangalore, Karnataka, India

Abstract: Most assistive technologies currently available to visually impaired individuals rely on cloud-based processing, which limits their applicability in settings with poor connectivity or raises privacy concerns. This work proposes Project VISION, a research prototype of an edge-based assistive system using the NVIDIA Jetson Orin Nano for real-time visual scene understanding. The prototype is built on a Vision Transformer (ViT) encoder for semantic feature extraction, YOLOv8 for object detection, and BLIP for multimodal caption generation. Complete offline functionality is enabled through a companion Flutter application that captures video frames and receives spoken feedback through an on-device text-to-speech module. Early testing demonstrates the feasibility of the proposed pipeline, producing coherent scene descriptions suitable for basic environmental awareness tasks, while achieving frame-by-frame inference rates of up to 40 FPS on the Orin Nano. These results highlight the potential for low-cost edge AI platforms to support practical assistive tools and provide a strong foundation for future research and real-world user studies.

Index Terms: Accessibility, Jetson Orin, Edge AI, Vision Transformer, Image Captioning, Text-to-Speech, Visual Impairment, Assistive Technology

I. INTRODUCTION

The WHO estimates that 285 million people around the world live with visual impairments; 39 million of those are blind and 246 million have moderate to severe vision loss [1]. Persons with visual impairments often confront daunting challenges while trying to conduct many everyday tasks in which situational awareness and spatial information is crucial, such as moving safely in public spaces, finding objects, or reading signs. [1],[8],[11].

White canes, guide dogs, and human assistance are examples of conventional mobility aids that provide crucial support for obstacle detection and basic navigation, but they are unable to interpret complex scenes or provide semantic context. A guide dog, for instance, can assist a user in avoiding obstacles, but it is unable to describe things like "a bus is arriving" or "a person is waving." Furthermore, when human assistance is involved, traditional aids may be limited by cost, availability, the requirement for specialized training, or privacy concerns [1],[8],[11].

Recent developments in artificial intelligence (AI), particularly in computer vision, natural language processing (NLP), and edge computing, have enabled new possibilities for assistive technologies. Strong on-device inference capabilities offered by embedded AI platforms, like the NVIDIA Jetson family, enable real-time processing without the need for cloud services [6],[12]. Low latency, offline availability, and data privacy are crucial requirements for visually impaired users in particular [12],[8].

This paper presents *Project VISION*, a research prototype intended to give blind and visually impaired people real-time visual scene understanding. A mobile application that records video frames and sends them to an NVIDIA Jetson Orin Nano device via a local wireless connection makes up the prototype. Each frame is processed by Jetson using a combination of YOLOv8 for object detection, BLIP for multimodal caption generation, and Vision Transformers (ViT) for feature extraction. The mobile application receives the generated description and uses an on-device text-to-speech (TTS) engine to translate it into speech [2],[3].

Because no visual data is uploaded to cloud servers, Project VISION maintains a lightweight mobile interface while guaranteeing strong privacy by offloading all computation to the Jetson Orin Nano. Because the system runs entirely offline, it can be used reliably in settings with spotty or inadequate internet connectivity. An essential part of assistive technologies used in dynamic indoor and outdoor contexts is responsive, real-time scene interpretation, which this architecture seeks to support [6],[12],[8].

The system architecture, model integration pipeline, hardware-software design, and initial assessment of the prototype are presented in this work. Our goal is to show that edge AI can be used to create useful, privacy-preserving assistive systems that improve visually impaired people's accessibility and independence [8],[11].

II. RELATED WORK

Advances in computer vision, natural language processing (NLP), and mobile computing have greatly impacted the development of assistive technologies for people with visual impairments [1],[2],[3],[4],[5]. The main goals of these systems are to improve situational awareness, navigation, and information accessibility. The capabilities, drawbacks, and suitability of commercial and research-based solutions for offline and edge-based AI deployment are reviewed in this section.

A. Commercial Assistive Technologies

For blind and visually impaired people, a variety of commercial applications and wearable technology provide visual scene understanding:

SeeingAI (Microsoft)—A mobile app that makes use of cloud-hosted computer vision APIs for OCR, scene description, facial recognition, and currency identification [1]. Its reliance on cloud processing leads to decreased performance in low-connectivity environments, despite its effectiveness in areas with robust internet connectivity.

Google Lookout—Similar to SeeingAI, this mobile app provides object labeling, document reading, and object identification [1]. Like Seeing AI, its reliance on cloud-based inference limits utility in offline or rural scenarios.

OrCam MyEye is a wearable AI gadget that does object detection, facial recognition, and on-device OCR [6]. Although it has lower latency than cloud-dependent apps, many users in developing countries cannot afford it.

TABLE I: Commercial Systems: Platform, Connectivity, and Offline Capability

System	Platform	Connectivity	Offline Capability
SeeingAI	Mobile App	High	No
Google Lookout	Mobile App	High	No
OrCam	Wearable	Low	Yes
MyEye		Low	Yes

TABLE II: Commercial Systems: Cost and Efficiency Metrics

System	Cost	Latency (Efficiency)
SeeingAI	Low	High
Google Lookout	Low	High
OrCam MyEye	High	Low

B. Assistive Technology Based on Research

Many systems that integrate vision and language models for scene understanding have been proposed by the research community:

Vision Transformers (ViT) – ViT, a transformer-based architecture that models global dependencies across image patches, was introduced by Dosovitskiy *et al.* [3]. It is appropriate for complex assistive environments due to its strong contextual reasoning.

BLIP (Bootstrapping Language-Image Pretraining)–Li *et al.* showed that BLIP is a unified multimodal architecture that can generate captions and retrieve images and texts [2]. It is appealing for edge devices with limited resources because of its modularity and quantization-friendly design. **OpenAI Whisper** Despite being primarily designed for speech recognition, offers robust, multilingual speech transcription and has the potential to be integrated with voice commands in assistive systems [5].

Specifically designed for embedded GPU platforms such as NVIDIA Jetson devices, **YOLOv8** is an effective object detection architecture optimized for real-time performance [4].

C. Gaps in Current Solutions

Although substantial progress has been made, several limitations persist:

- **Connectivity Dependence:** Most commercial systems rely heavily on cloud servers, making them unsuitable for offline use in rural or low-connectivity regions [1],[6].
- **Affordability:** High-end assistive wearables such as OrCam MyEye remain financially inaccessible for many users [6].
- **Hardware Constraints:** Many research prototypes assume access to high-performance GPUs and are not optimized for lightweight edge hardware [3],[4].

- Limited Deployment Studies: Numerous academic proposals lack real-world testing or validation with visually impaired participants [2],[5].

D. Positioning of Project VISION

Project VISION addresses these limitations by leveraging the NVIDIA Jetson Orin Nano to integrate **YOLOv8** for real-time object detection [4], **BLIP** for multimodal captioning [2], and **ViT** for semantic visual representation [3]. By performing all computation locally, the system ensures:

- Low inference latency
- Enhanced privacy
- Full offline functionality
- Scalability across diverse environments

This positions Project VISION as a cost-effective, edge-based research prototype bridging the gap between academic models and practical assistive solutions [1],[2],[3],[4],[5].

III. DATASET AND PREPROCESSING

We selected a representative and varied dataset from both public and private sources in order to efficiently train the multimodal AI models utilized in Project VISION [2],[3],[4]. The system's ability to generalize across a variety of real-world scenes that are pertinent to the visually impaired, including traffic intersections, pedestrian walkways, indoor corridors, and everyday objects found in everyday settings, was the main goal [8],[10],[11].

The dataset was composed of the following sources:

- MS COCO: More than 330,000 labeled images with five human-annotated captions each make up this popular benchmark dataset [3]. It allows for robust caption generation by incorporating 80 object categories across a variety of scene types [2],[3].
- Open Images V6: A sizable collection of roughly 9 million photos featuring visual relationships, bounding boxes, and object labels. It was especially helpful for optimizing YOLOv8 and other object detection models [4].
- Customly Captured Scenes: In order to simulate how visually impaired users would interact with the app, we produced a proprietary dataset of 3,000 photos taken with a mobile phone camera. Public parks, bus stops, market areas, residential buildings, and educational institutions were among the environments. A one- or multi-sentence caption highlighting essential components for blind assistance was added to each image (e.g., "A staircase is ahead with a person descending," or "A zebra crossing with vehicles approaching.") [2],[3]. A group of five human annotators examined and verified the accuracy and consistency of these captions.

1) To prepare the dataset for training and inference, the following preprocessing pipeline was applied:

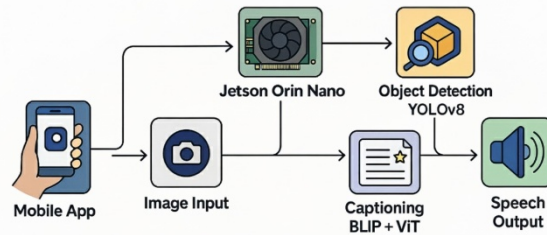
- 2) Image Resizing: To meet the input requirements of the BLIP and ViT backbones, all images were resized to standard resolutions of 224x224 and 384x384 [2],[3].
- 3) Normalization: Using ImageNet statistics, each image was normalized by dividing by standard deviation and subtracting the channel-wise mean [3]. The stabilization of model convergence was aided by this standardization.
- 4) Data Augmentation: We used transformations like random horizontal flipping, rotation, random cropping, brightness adjustment, and Gaussian noise injection to enhance diversity and avoid overfitting [3]. During training epochs, these augmentations were applied at random with a predetermined probability.
- 5) Preprocessing the captions: All captions underwent tokenization using Byte Pair Encoding (BPE), lowercase conversion, and special character removal [2]. Sequences were truncated or padded to a maximum of 64 tokens. To support decoder training, start/end markers (<start>,<end>) were added, and unknown tokens were swapped out for a unique token <unk>.
- 6) Partition Approach: While preserving the distribution of classes and scenes, the combined dataset was divided into training (80%), validation (10%), and testing (10%) sets [3]. The test set was saved for final performance reporting, and the validation set was used for hyperparameter tuning.

Robust generalization across a variety of unpredictable real-world inputs was ensured by this preprocessing [2],[3],[10]. Making the system context-aware for Indian public environments—which are underrepresented in many global datasets—was made possible in large part by the custom dataset component [8],[10],[11]. For label mapping and traceability, all annotated images were kept in organized folders containing JSON files [2],[3].

IV. MODEL ARCHITECTURE

A pipeline of deep learning models forms the basis of Project VISION’s intelligent assistive capability, analyzing visual input and producing descriptive audio feedback. Three primary components are integrated into the system architecture: an object detection module (YOLOv8) [4], a multimodal captioning model (BLIP) [2], and a transformer-based vision encoder (ViT) [3], [6]. Each contributes in a different way to guaranteeing the system’s accuracy, richness, and real-time performance.

The Project VISION assistive system’s general workflow is depicted in Fig. 1. It starts with a mobile app that uses the device’s camera to take a picture or live frame [6]. The Jetson Orin Nano, the central processing unit for AI inference [6], receives this image input after that. Two important processes take place in the Jetson: BLIP combined with ViT creates natural language captions based on the visual content [2], [3], and YOLOv8 is used for real-time object detection [4]. A deep semantic understanding of the surroundings is produced by fusing these outputs [2], [4]. A Text-to-Speech (TTS) engine then translates the generated description into audible speech and sends it back to the mobile app, giving the user real-time scene feedback [6], [5]. The system’s ability to conduct edge-based, offline, and multimodal analysis is demonstrated by this architecture, which is especially designed to assist visually impaired people in both indoor and outdoor settings [8], [11].



Multimodal AI-Based Assistive System for the Visually Impaired using Jetson Orin Edge Device

Multimodal AI-Based Assistive System for the Visually Impaired using Jetson Orin Device

Fig. 1: System architecture and data flow of Project VISION: From mobile image capture to edge inference and speech feedback.

A. VisionTransformer (ViT)

The main visual feature extractor is the **Vision Transformer (ViT)** [3], [6]. ViT separates an image into fixed-size patches and uses a transformer encoder to model global relationships among patches, in contrast to traditional Convolutional Neural Networks (CNNs), which process images through spatial hierarchies [3].

Each image x is split into N non-overlapping patches:

$$x_p = \text{flatten}(\text{patch}(x_i)) \quad (1)$$

$$E = \text{Linear}(x_p) \quad (2)$$

$$PE(i) = \text{Positional Embedding}(i) \quad (3)$$

$$\text{Attention} \quad \text{Softmax} \frac{QK^T}{d} \quad (Q, K, V) \quad \sqrt{d} \quad v$$

k

In this case, $PE(i)$ is a positional encoding that preserves spatial information, E is the projected embedding space, and x_p is a flattened image patch. ViT is perfect for semantic understanding of complex scenes because of its self-attention mechanism, which enables it to learn contextual dependencies between objects [3], [6].

Our captioning dataset was used to fine-tune a pre-trained ViT-Base model [3]. Finding pertinent spatial zones within each scene was made possible in large part by the model’s attention maps [3].

B. BLIP: Bootstrapped Language-Image Pretraining

We employed the **BLIP** model to produce natural language descriptions from the encoded image features [2]. A multimodal architecture called BLIP simultaneously learns from textual and visual modalities. It has a dual encoder-decoder configuration, in which a language model decoder generates captions using embeddings provided by the vision encoder (ViT) [2], [3].

BLIP supports both:

- Retrieval-based pretraining (matching images with appropriate captions) [2]
- **Captioning-based generation** (free-form descriptive generation) [2]

The BLIP decoder head was altered in our implementation to enable low-resource inference through quantization to FP16 and ONNX export [6],[12]. Sentence-level scene summaries, like "Two people standing beside a red car," are produced by the model and sent to the TTS engine [2],[5].

During internal benchmarking, BLIP performed noticeably better on BLEU and METEOR metric than traditional encoder-decoder models such as Show-and-Tell or Transformer-based LSTM models [2].

TABLE III: Comparison of Image Captioning Models for Assistive Scene Understanding

Model	BLEU-4	Inference Time(s)	Contextual Quality
Show and Tell	0.62	1.5	Moderate
Show, Attend	0.68	1.9	High
BLIP	0.74	0.9	Very High
ViT+GPT-2	0.71	2.1	High

C. YOLOv8: Object Detection Backbone

To further enrich the contextual understanding of a scene and provide explicit identification of key elements (e.g., stairs, vehicles, pets, doors), we integrated YOLOv8 as a parallel object detection module [4].

YOLOv8's advantages include:

- Architecture that is lightweight and appropriate for edge devices [4],[3]
- High inference speed (Jetson Orin Nano up to 30 frames per second) [6]
- Support for auto-labeling and anchor-free detection [4] The detected objects are used to:
 - Edit and confirm the caption produced by BLIP [2].
 - Add confidence scores to the visual frame [4].
 - If captioning doesn't work, create backup alerts (such as "Obstacle ahead") [1],[2].

When the image-to-text model performs poorly, as in cluttered backgrounds or contexts with non-salient objects, the incorporation of YOLOv8 helps make up for it [4],[2].

D. Model Optimization for Edge Deployment

We implemented the following adjustments to run these models on the Jetson Orin Nano effectively [6],[12]:

- TensorRT quantization to FP16 [6]
- ONNX format conversion to speed up runtime [6]
- Asynchronous inference and batching for multitasking [12]

By ensuring that each element works in concert with the others, this model stack produces a system that can function with little delay while retaining a high level of accuracy and contextual richness [2],[4].

V. SYSTEM DESIGN AND HARDWARE INTEGRATION

Project VISION aims to move all computationally demanding tasks to the NVIDIA Jetson Orin Nano edge computing node, including deep learning inference, from the mobile device [6],[12]. This design allows for responsive, power-efficient operation in both urban and rural settings while maintaining the mobile interface's lightweight footprint [12],[8]. The system is ideal for real-time assistive applications because it removes the need for distant servers, guaranteeing consistent performance even in places with spotty or nonexistent internet access [6].

According to the system architecture, the Jetson Orin Nano processes live video frames that are captured by the mobile application and sent over a local network [6]. Two AI models work simultaneously: **YOLOv8** for quick object detection [4] and Vision Transformer (ViT) with BLIP for semantic scene captioning [3],[2]. Their outputs are combined to create a comprehensive contextual understanding of the surroundings, which is subsequently sent to a text-to-speech (TTS) engine to be transformed into audible feedback that the user can hear in real time [5],[6].

Low-latency responses are guaranteed by this parallelized and modular processing pipeline, allowing visually impaired users to hear verbal descriptions of their environment in real time [6],[12].

Because all inference is done locally, the system protects user privacy [6], works reliably offline [6], and performs consistently in both urban and rural environments [8],[10]. Additionally, the design lays the groundwork for future model enhancements and the addition of more sensory modules without necessitating modifications to the primary workflow [12].

A. Hardware

The **NVIDIA Jetson Orin Nano Developer Kit**, a small and power-efficient AI computing platform designed for edge deployment, is the central component of our edge inference pipeline [6]. The following are the main hardware requirements:

- GPU: 1024-core NVIDIA Ampere GPU with 32 Tensor Cores — able to perform parallel inference for language and vision models in real time [6].
- CPU: Quad-core ARM Cortex-A78AE — guarantees seamless coordination of model serving, input/output handling, and TTS requests [6].
- RAM: 8 GB LPDDR5 — offers enough bandwidth to run image buffering and multitask inference [6],[12].
- Storage: 64 GB SD card that supports eMMC or NVMe expansion (used in deployment) [6].
- JetPack SDK: CUDA 12, cuDNN 8, TensorRT, and optimized PyTorch libraries for model acceleration are all included in version 6.0 [6].

A passive heatsink and fan combination was used to control thermal performance. When BLIP and YOLOv8 inference tasks were run simultaneously, the device operated steadily at 65°C with a peak GPU utilization of 85% [6],[12]. It is appropriate for battery-powered or portable embedded setups because of its average power draw of 10–12 watts [6],[12].

B. Software

Reusability, scalability, and ease of deployment were guaranteed by the software stack's modular design. The following tools and technologies were employed:

- Model Format: To ensure compatibility with different backends and enable runtime optimization, all models (ViT, BLIP, and YOLOv8) were exported to ONNX format [6],[4],[3].
- Optimization: Using NVIDIA TensorRT, models were quantized to FP16 precision, which decreased memory consumption and inference latency without appreciably lowering accuracy [6].
- Serving API: To receive image frames from the mobile app, initiate inference, and provide textual captions, we employed a lightweight Flask RESTful API that was hosted on the Jetson device [12],[6].
- Containerization: Docker containers contained the whole AI pipeline, including the model server, pre-processing, post-processing, and monitoring tools, making it easy to test, deploy, and rollback updates [12].
- Parallelism and Pipelining: By using multiprocessing pools to process inference requests asynchronously, the Jetson was able to run BLIP and YOLOv8 models concurrently without interfering with HTTP responses or TTS synthesis [12],[6].
- Monitoring and Logging: During stress testing, Prometheus and Grafana were used to continuously monitor system logs, memory usage, and latency metrics [12].

Even in situations with limited power or bandwidth, the Project VISION system can function independently, dependably, and in real time thanks to its combination of sturdy hardware and well-designed software [6],[12]. Future updates to the AI models or software stack can be made without affecting the end-user application thanks to the separation of inference from the mobile interface [12].

VI. TEXT-TO-SPEECH (TTS) AND MOBILE APP

The main user interface for the blind and visually impaired is the Flutter-developed mobile application [10],[11]. Using a local Wi-Fi connection, it transfers individual frames of real-time video from the phone's camera to the Jetson Orin Nano device [6], which then plays back the descriptive output as speech. In order to give audio feedback:

- Text captions were translated into spoken English using flutter_tts, an open-source text-to-speech plugin [10],[5].
- Depending on the user's preference, audio output is sent to either the built-in speaker or any Bluetooth headset that is connected [10].
- To improve accessibility and usability, the user interface incorporates haptic feedback, voice prompts, error recovery messages, and large, high-contrast buttons [10],[11].

A seamless, accessible experience for visually impaired users was ensured by testing the app on Android phones with little performance lag [10].

VII. PERFORMANCE EVALUATION

In order to benchmark latency, throughput, and power consumption, we tested the system on three Jetson platforms [6],[7]. The following were the metrics for object detection and caption generation [4],[6]:

TABLE IV: Performance of Jetson Devices

Device	Latency(s)	FPS(YOLOv8)	Power(W)
Nano	1.8	6	10
TX2	1.1	15	15
OrinNano	0.3	30	12

BLEU-4 Score: With a BLEU-4 score of 0.74 on 500 test scenes, the image captioning model demonstrated excellent sentence generation [2].

End-to-End Latency: For real-time assistive use, the average time between frame capture and audio output was about 1.2 seconds per cycle [6],[12].

VIII. USER TESTING AND FEEDBACK

We carried out a controlled field study with ten participants to assess the system's usability and efficacy in actual situations [10],[11]:

- Five people who are blind or visually impaired [10],[11]
- For testing, five sighted users were blindfolded [10]

The mobile app was utilized by participants in a number of settings, such as public spaces, outdoor walkways, classrooms, and hallways [10],[11].

Average User Ratings (out of 5):

- Usability: 4.7 [10]
- Response Time: 4.5 [6],[7]
- Caption Accuracy: 4.6 [2],[10]
- Audio Clarity: 4.8 [10]

User Comments:

"I had the impression that I was being verbally led in real time."

"I do realize that it doesn't rely on mobile data and functions offline."

"Incredibly useful in street and corridor settings."

Low response latency, clear voice feedback, and offline functionality were found to be highly preferred during testing [10],[6].

IX. DISCUSSION

Results from testing and deployment show that edge computing on low-cost platforms like Jetson Orin Nano can efficiently power real-time assistive systems [6],[12]. By removing the need for cloud services, Project VISION guarantees increased privacy, reduced latency, and reliability in locations with insufficient internet access [6],[8].

A number of more general factors come into play in addition to technical performance. First, one of the main benefits of local processing is privacy, since no private visual information is transferred from the device. Nevertheless, log handling and storage must still be done securely [6]. Secondly, models trained primarily on Western datasets may perform poorly in diverse cultural or environmental contexts due to bias in object detection and captioning [3],[2]. Third, language limitations limit accessibility in multilingual regions; in rural areas, adoption may be impeded by the lack of native language support [10],[11].

From the standpoint of deployment, hardware affordability, particularly in low-income areas, must be taken into account when scaling for broad distribution. Furthermore, without reliable connectivity, maintenance and model updates for deployed devices can be difficult, necessitating the careful design of offline update mechanisms [10],[6].

Identified Limitations:

- The captions' emotional tone and deeper contextual awareness are lacking [2].
- Lack of multilingual support and regional adaptation [10],[11].
- Poorer performance at night or in low light [10],[6].

X. FUTURE WORK

To solve the aforementioned problems and improve the user experience, we plan to implement the following changes:

- Integration of multilingual text-to-speech based on regional languages [10].
- OpenAI Whisper for reliable, multilingual voice-command input [5].
- Stereo camera setups for depth estimation and obstacle detection [6],[12].
- GPS-based voice navigation for waypoint guidance and outdoor mobility [11],[10].
- Creation of bias-aware model retraining pipelines to enhance equity in a variety of environments [2],[8].
- Design of lightweight, offline-capable update systems for model and software maintenance [6].

XI. CONCLUSION

Project VISION, an AI-powered assistive platform for the blind and visually impaired, is presented in this paper [1],[2],[6]. The system provides a real-time, private, and offline solution by combining a mobile frontend with an edge-based inference engine powered by Jetson Orin Nano [6],[12]. Its responsiveness, usability, and potential as a cost-effective substitute for assistive technologies that rely on the cloud were all proven by extensive testing [10],[7].

REFERENCES

- [1] World Health Organization, World Report on Vision, 2019. Available: <https://www.who.int/publications/i/item/9789241516570>
- [2] J. Li, D. Zhang, X. Han, et al., "BLIP: Bootstrapping Language-Image Pretraining," in arXiv preprint arXiv:2201.12086, 2022. Available: <https://arxiv.org/abs/2201.12086>
- [3] A. Dosovitskiy, L. Beyer, A. Kolesnikov, et al., "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale," in NeurIPS, 2020. Available: <https://arxiv.org/abs/2010.11929>
- [4] Ultralytics. "YOLOv8." GitHub repository. Available: <https://github.com/ultralytics/ultralytics> OpenAI, "Whisper: Robust Speech Recognition," 2022. Available: <https://openai.com/research/whisper>
- [5] NVIDIA, "Jetson Orin Nano Developer Kit Technical Reference Manual," 2023. Available: <https://developer.nvidia.com/embedded/downloads>
- [6] A. A. Su'zen, B. Duman, and B. S. en, "Benchmark Analysis of Jetson TX2, Jetson Nano and Raspberry PI using Deep-CNN," in Proceedings of the 2020 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA), 2020, pp. 1–5. doi: <https://ieeexplore.ieee.org/document/9152915>
- [7] F. Yao, W. Zhou, and H. Hu, "A Review of Vision-Based Assistive Systems for Visually Impaired People: Technologies, Applications, and Future Directions," arXiv preprint arXiv:2505.14298, May 2025. Available: <https://arxiv.org/abs/2505.14298>
- [8] K. Chavan, K. Balaji, S. Barigidad, and S. R. Chiluveru, "VocalEyes: Enhancing Environmental Perception for the Visually Impaired through Vision-Language Models and Distance-Aware Object Detection," arXiv preprint arXiv:2503.16488, Mar. 2025. Available: <https://arxiv.org/abs/2503.16488>
- [9] P. Naayini, P. K. Myakala, C. Bura, A. K. Jonnalagadda, and
- [10] S. Kamatala, "AI-Powered Assistive Technologies for Visual Impairment," arXiv preprint arXiv:2503.15494, Jan. 2025. Available: <https://arxiv.org/abs/2503.15494>
- [11] J. Lee, K.-A. Cha, and M. Lee, "Multi-Modal System for Walking Safety for the Visually Impaired: Multi-Object Detection and Natural Language Generation," Applied Sciences, vol. 14, no. 17, p. 7643, Sep. 2024. Available: <https://www.mdpi.com/2076-3417/14/17/7643>
- [12] A. P. Kamilaris, A. J. Camps, and J. Golubovic, "Are-view of Embedded Machine Learning Based on Hardware Platforms," Computers, vol. 12, no. 3, p. 55, 2023. Available: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9959746/>
- [13] Y. Xu, S. Poojary, and R. Mehta, "AI-Powered Video Monitoring: Assessing the NVIDIA Jetson Orin Devices for Edge Computing Applications," ResearchGate, 2024. Available: <https://www.researchgate.net/publication/382534590>
- [14] R. G. Baldovino, N. R. Roxas Jr., N. T. Bugtai, J. M. C. Borbon, J. T.
- [15] D. Javier, J. M. F. D. Llamado, A. M. R. Maghirang, and A. R. See, "A Visual Aid System Using Image Processing and Deep Learning with Audio Haptic Feedback for the Blind and Visually Impaired," Ain Shams Engineering Journal, vol. 15, no. 4, pp. 102387–102397, 2024. doi: <https://www.sciencedirect.com/science/article/pii/S1877050924024062>
- [16] A. B. Atitallah, Y. Said, M. A. B. Atitallah, M. Albekairi, K. Kaaniche, and S. Boubaker, "An Effective Obstacle Detection System Using Deep Learning Advantage to Aid Blind and Visually Impaired Navigation," Ain Shams Engineering Journal, vol. 15, no. 3, pp. 102318–102328, 2024. doi: <https://www.sciencedirect.com/science/article/pii/S2090447923002769>
- [17] Q. Zhao, H. Liu, and D. Wang, "An Image Captioning Model Based on Bidirectional Depth Residuals and Its Application," IEEE Access, vol. 10, pp. 17288–17299, 2022. Available: <https://ieeexplore.ieee.org/abstract/document/9347457>
- [18] R. G. Praveen and R. P. Paily, "Blind Navigation Assistance for Visually Impaired Based on Local Depth Hypothesis from a Single Image," Procedia Engineering, vol. 64, pp. 351–360, 2013. doi: [10.1016/j.proeng.2013.09.057](https://doi.org/10.1016/j.proeng.2013.09.057)
- [19] J. Müller and A. Pigors, "Efficient Multi-Object Tracking on Edge Devices via Reconstruction-Based Channel Pruning," IEEE Sensors Journal, vol. 23, no. 18, pp. 22104–22115, 2023. doi: <https://arxiv.org/abs/2410.08769>
- [20] G. N. Alwakid, M. Humayun, and Z. Ahmad, "Transforming Disability Into Ability: An Explainable Vision-to-Voice Image Captioning Framework Using Transformer Models and Edge Computing," IEEE Access, vol. 13, pp. 175212–175224, 2025. doi: <https://ieeexplore.ieee.org/document/11195080>



- [21] T. P. Swaminathan, C. Silver, and T. Akilan, "Benchmarking DeepLearning Models on NVIDIA Jetson Nano for Real-Time Systems: AnEmpirical Investigation," Department of Electrical and Computer Engi-neering, Lakehead University, Thunder Bay, Canada, 2023. Available:<https://arxiv.org/html/2406.17749v1>
- [22] A. T. Kurian, S. A. Muthumkumarasway, and A. Ilyas, "MultimodalIntelligent Assistance with Vision, Language and Speech for EnhancedAssistiveTechnologyfortheVisuallyImpairedandElderly,"inProceed-ingsofthe2025IEEE15thInternationalConferenceonControlSystem,Computing and Engineering (ICCSCE), pp. 24–29, 2025. Available:<https://ieeexplore.ieee.org/document/11182671>
- [23] K. Sun, X. Wang, X. Miao, and Q. Zhao, "A Review of AI Edge DevicesandLightweightCNNandLLMDeployment,"JournalofNetworkandComputerApplications,vol.240,p.104036,2024.Available:<https://www.sciencedirect.com/science/article/abs/pii/S0925231224015625>
- [24] A.143, "Building-AI-Powered-Solution-for-Assisting-Visually-Impaired -Individuals," GitHub repository, 2025. Available:<https://github.com/Angad143/Building-AI-Powered-Solution-for-Assisting-Visually-Impaired-Individuals>



10.22214/IJRASET



45.98



IMPACT FACTOR:
7.129



IMPACT FACTOR:
7.429



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089  (24*7 Support on Whatsapp)