iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** V **Month of publication:** May 2026

DOI: https://doi.org/10.22214/ijraset.2026.80312

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue V May 2026- Available at www.ijraset.com

A Survey on Assistive Vision Technologies for Visually Impaired Individuals: Approaches, Techniques, and Future Directions

Sahil Pramod Warade[1], Tabish Ibrahim Ansari[2], Swapnil Bhanudas Shingne[3], Sharayu Dilip Patil[4], Sachin Shelke[5]
Department of Information Technology, SCTR'S Pune Institute of Computer Technology Pune, Maharashtra, India

Abstract: ***This survey reviews advances and deployment strategies for assistive vision systems that combine continuous on-device object detection with selective, high-fidelity scene segmentation and succinct audio narration. We synthesize evidence concerning mobile-optimized detectors (including YOLO family and recent YOLOv8 variants), promptable segmentation foundation models (SAM and SAM2), vision-language approaches for narration, and hybrid on-device/edge/cloud architectures that trade latency, privacy and capability. We discuss datasets captured by visually impaired users (VizWiz family, ORBIT), propose evaluation metrics beyond classical mAP (latency, answerability, safety- critical misses), and identify open problems and near-term re- search directions for making hybrid detection–segmentation–TTS pipelines practical.***
Index Terms: Assistive vision, YOLOv8, SAM2, segmentation, VizWiz, real-time detection, text-to-speech

I. INTRODUCTION

Assistive vision systems aim to convert visual scenes into concise, actionable non-visual cues to increase autonomy for visually impaired users. Practical systems must reconcile two competing demands: (1) continuous, low-latency hazard detection sufficient for immediate safety, and (2) richer, high- fidelity scene understanding that supports complex queries but is computationally intensive.

Contemporary engineering favors hybrid pipelines: a compact detector runs continuously on-device for immediate alerts while larger foundation models (e.g., SAM2) are invoked selectively on edge or cloud resources for pixel-level analysis and richer narration [10], [13], [14]. This survey synthesizes the technical literature and implementation patterns most relevant to building such a hybrid YOLOv8 + SAM2 + TTS pipeline, focusing on models, datasets, metrics and deployment paradigms that directly inform engineering and evaluation choices. [1], [2], [20]

II. ASSISTIVE VISION PROBLEM STATEMENT AND USER REQUIREMENTS

Designing assistive vision systems requires careful attention to user needs: latency thresholds for hazard warnings, acceptable verbosity, privacy expectations, and social factors such as conspicuousness of the device. Low-vision and blind users have different tolerances and preferences (e.g., audio-first vs. augmentative visual interfaces).

Systems must minimize cognitive load (repeat-suppression, configurable verbosity) and support user customization for prioritized object classes (people, doors, stairs) [20], [21]. These human-centered constraints directly motivate a hybrid architecture in which a lightweight, quantized detector supplies immediate, predictable audio while heavier segmentation and VLM components are invoked only when beneficial.

III. DATASETS AND EVALUATION BENCHMARKS

A. Blind-captured datasets and their characteristics

Datasets captured by visually impaired users differ systematically from curated datasets: images are often off-center, blurred, occluded, or contain partial objects, and they frequently exhibit 'unanswerable' conditions for VQA tasks. The VizWiz dataset family documents these phenomena and has been widely used to measure practical answerability and caption quality in assistive scenarios [1], [2]. ORBIT focuses on few-shot recognition of personal objects, which is directly relevant for personalized 'find my item' capabilities [3]. Designers should therefore evaluate models on such specialized datasets in addition to general benchmarks (COCO) to ensure real-world applicability [4].

B. Evaluation metrics beyond mAP

Standard metrics (mAP, IoU) remain useful, but assistive systems require additional measures: end-to-end detection- to-speech latency (ms), answerability rate on blind-captured images, safety-critical missed-detection rate (e.g., missed stairs or immediate obstacles), per-frame energy cost, and human- subject outcomes (task success rates, NASA-TLX). Combining algorithmic and user-centered metrics yields a comprehensive assessment of operational readiness [7], [20], [23].

IV. REAL-TIME OBJECT DETECTION FOUNDATIONS

A. Two-stage detectors: accuracy vs latency

Two-stage detectors (R-CNN, Fast/Faster R-CNN) introduced region proposal mechanisms that improved accuracy but incurred substantial latency due to multi-stage processing. Mask R-CNN extended this family to instance segmentation by adding a parallel mask branch, but the increased computational cost makes such models unsuitable for continuous on-device hazard detection without substantial pruning or acceleration [5], [6].

B. Single-stage detectors and the YOLO lineage

Single-stage detectors (YOLO, SSD, RetinaNet) frame detection as dense regression from feature maps and empha- size throughput. RetinaNet introduced focal loss to correct foreground/background imbalance, narrowing the accuracy gap with two-stage methods while retaining one-stage speed. The YOLO family iteratively optimized backbone and head designs to improve both speed and accuracy across versions [8], [9].

C. YOLOv8: architecture, innovations, performance and trade- offs

YOLOv8 (Ultralytics) is a modern single-stage, anchor- free detector that combines a CSP-inspired backbone for efficient feature extraction, a PANet-like neck for multi-scale aggregation, and an anchor-free detection head to simplify post- processing. Key innovations include an emphasis on modular design for deployment, quantization-friendly operations, and

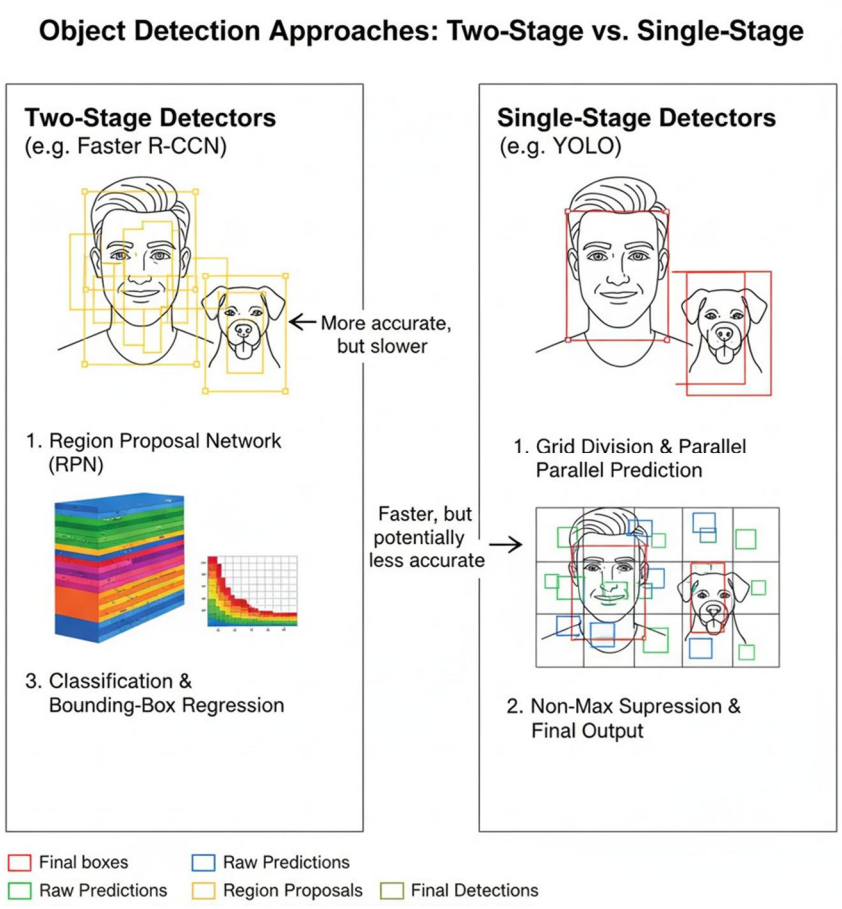


Fig. 1. Detection architecture comparison (two-stage vs single-stage).

a suite of model scales culminating in YOLOv8n (nano) for mobile deployment [**?**]. Performance characteristics reported in the literature show that YOLOv8n achieves real-time throughput on many smartphones with optimized runtimes and NPUs while retaining reasonable mAP on standard benchmarks (COCO- range figures vary by model scale and quantization). The primary trade-offs for the nano variant are reduced recall on tiny/occluded objects and a smaller context window; assistive deployments often address these limitations via small-object attention modules or by combining YOLO with selective segmentation for geometry [10]–[12].

V. IMPROVED YOLO VARIANTS AND SMALL-OBJECT TECHNIQUES

Assistive scenarios require reliable detection of small per- sonal items and environmental affordances (handles, buttons, signage). Architectural modifications—context-attention blocks (CAB), improved neck designs (LDGConv), and multi-scale context modules—have been proposed to boost small-object recall while preserving throughput. Empirical studies from the uploaded literature indicate modest mAP gains and limited increases in latency, which may be acceptable in mobile assistive systems when paired with selective segmentation for verification [11], [12].

VI. SCENE SEGMENTATION: SEMANTIC, INSTANCE AND PANOPTIC

A. Segmentation taxonomy and role for assistance

Segmentation provides pixel-level geometry required to reason about occlusion, blockage and affordances—information bounding boxes cannot supply reliably. Semantic segmentation (DeepLab/FCN/U-Net) labels 'stuff' regions (road, sidewalk), instance segmentation (Mask R-CNN, YOLACT) isolates distinct objects, and panoptic segmentation unifies both views, though at higher compute cost [16]. For assistive narration, instance masks inform phrases such as "door partially blocked by chair," enabling decisions beyond simple object lists.

B. Segment Anything Model (SAM): concept and limitations

SAM introduced a promptable segmentation paradigm: a universal image encoder plus a prompt encoder and mask de- coder allow the system to produce zero-shot masks when given sparse cues (points, boxes, text). Trained on an exceptionally large mask corpus, SAM generalizes to many domains without per-task retraining, which is useful when the field environment contains novel objects [13]. However, SAM's full-resolution inference is computationally demanding, limiting continuous on-device use.

C. SAM2: video extension, memory and temporal consistency

SAM2 extends promptable segmentation to video by adding a temporal-memory mechanism that stores and propagates object identities across frames, enabling consistent instance tracking through occlusions and motion. Architectureally, SAM2 combines a backbone image encoder (often ViT-based), a prompt encoder that fuses sparse cues, a mask decoder, and a memory attention module that aligns temporal features for identity persistence. Reported performance shows SAM2 can produce temporally consistent masks suitable for short video streams when run on GPUs at moderate frame rates; the model's computational footprint necessitates selective, asynchronous offload in mobile settings rather than continuous on-device execution [14], [15]. For a hybrid pipeline, SAM2 is valuable as an on-demand, edge-hosted module that refines YOLO detection outputs into pixel-accurate, temporally coherent masks.

VII. VISION–LANGUAGE MODELS AND NARRATION STRATEGIES

A. Template vs learning-based narration

Safety-critical alerts demand deterministic, low-latency phras- ing; template-based TTS ("Person ahead, two meters") fulfills that need. Learning-based captioners and VLMs (BLIP, CLIP- augmented captioners) can produce richer descriptions and support VQA, but they increase latency and risk hallucination— therefore they are best reserved for on-demand queries that tolerate higher latency [17], [18], [23].

B. Practical TTS design considerations

TTS choices must balance naturalness, latency and privacy. On-device TTS engines provide immediate, private speech but may have lower naturalness; cloud neural TTS yields higher quality at the cost of connectivity and privacy. Design features critical for assistive use include adjustable speaking rate, verbosity presets, repeat-suppression windows, and the ability to prioritize safety-related messages over verbose descriptions [20], [23].

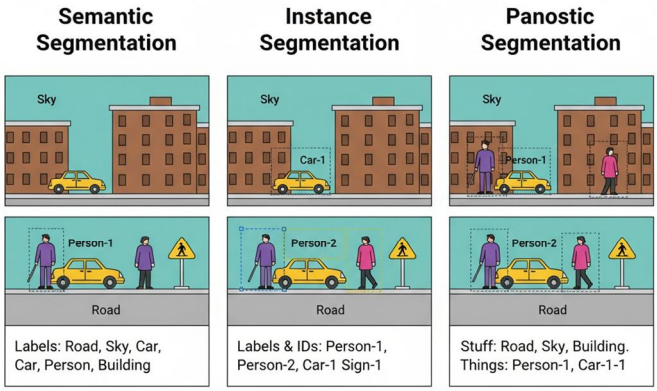


Fig. 2. Semantic vs instance vs panoptic segmentation.

VIII. WEARABLES AND MOBILE PROTOTYPES: LESSONS LEARNED

Prototypes such as AIris and Sight Guide demonstrate practical trade-offs: form factor constraints, the utility of audio- first UIs, and the value of asynchronous heavy processing for richer tasks. NavCog's indoor navigation work highlights the importance of reliable localization and concise turn-by- turn audio for wayfinding. These systems support key design choices for hybrid pipelines: minimal interaction, logging for iterative personalization, multimodal alerts, and user controls for privacy and verbosity [25].

A. Accessibility and User Experience Considerations

Accessibility extends beyond perception accuracy: consistent timing, predictable phrasing, and ergonomic interaction patterns are crucial for adoption among visually impaired users. Work on wearable prototypes and navigation aids stresses audio- first designs, minimal explicit interaction, spatialized cues for directionality, and configurable verbosity to avoid cognitive overload during mobility tasks [20], [21], [25]. Personalization features—such as priority object lists, ad- justable repeat-suppression windows, and adaptive verbosity based on user motion—have been shown to increase task success and user satisfaction in field studies. Logging of detection histories and confidence scores enables iterative personalization and offline analysis while also providing data for safety audits and debugging [1], [22].

IX. SYSTEM-LEVEL INTEGRATION AND APPLICATION DESIGN

An assistive vision system attains practical value when its perception modules integrate coherently into an application pipeline that respects usability and privacy constraints. Prior wearable and smartphone prototypes demonstrate the value of a continuous lightweight detector combined with selective heavy analysis, and motivate our hybrid design where YOLOv8n performs continuous detection and SAM2 provides on-demand pixel-level segmentation. Such hybrid pipelines can be imple- mented with on-device quantized inference for immediate alerts and edge/cloud-hosted SAM2 for selective, compute-intensive tasks [10], [13], [14], [21].

To minimize redundant processing and conserve energy, the system routes YOLO bounding boxes and simple user cues to an asynchronous orchestration layer that controls offload, logging, and narration. The orchestration layer applies short-term memory buffers for object history (to suppress repeats), confidence thresholds to gate offload, and a lightweight scheduler for batching segmentation requests to the edge. These design patterns reflect architectures used in recent assistive prototypes and operational systems that prioritize low-latency hazard alerts while allowing deeper analysis when the user or environment requires it [20], [21].

X. DEPLOYMENT PARADIGMS AND TRIGGER POLICIES

A. On-device inference

On-device inference offers minimal latency and strong privacy guarantees. Practical deployment requires model com- pression (pruning, quantization), optimized runtimes (TFLite, PyTorch Mobile, TensorRT) and careful thermal/energy man- agement to maintain acceptable battery life [10], [24].

B. Edge/cloud offload

Offloading enables large models (SAM2, VLMs) but intro- duces latency and privacy concerns. Edge (MEC) can reduce round-trip time compared to public cloud, making it a sensible compromise for selective segmentation and VLM queries [20], [21].

C. Trigger Mechanisms and Intelligent Context Switching

Effective offload policies determine when to incur the cost of high-fidelity segmentation. Simple triggers include explicit user requests, time-based sampling, and scene-change detection via optical flow or sudden class changes reported by the on-device detector. More advanced policies exploit temporal coherence (object permanence), short-term detection stability, and contextual cues (e.g., near/stationary status) to reduce unnecessary offload while maintaining safety-critical responsiveness [14], [20].

Recent work has explored adaptive trigger policies that learn when to offload using reinforcement learning or confidence- driven heuristics; these approaches optimize an objective combining latency, energy, and task utility. Incorporating lightweight motion estimation and object-tracking signals from the on-device loop reduces false-positive triggers and concentrates SAM2 invocations on frames where pixel-level masks yield maximal user benefit (for example, disambiguating occluded obstacles or verifying small-object affordances) [19], [20].

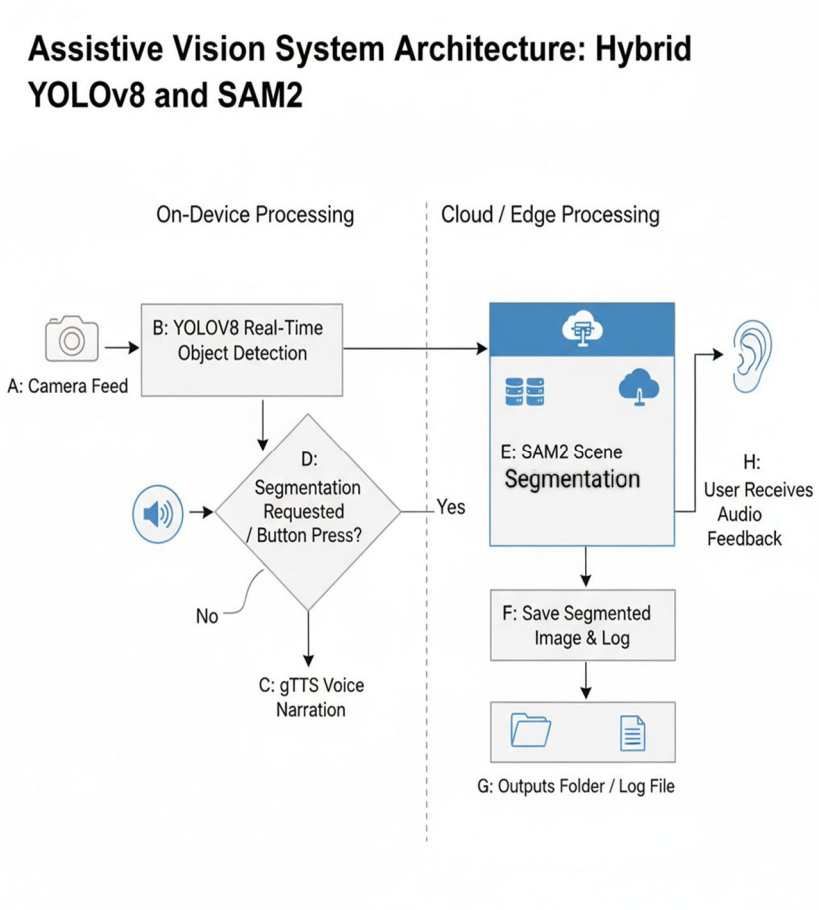


Fig. 3. The system architecture of the hybrid assistive vision system, combining on-device YOLOv8 for real-time detection and on-demand SAM2 for detailed scene segmentation.

TABLE I
COMPACT METHOD COMPARISON (SUITABILITY FOR ASSISTIVE HYBRID PIPELINES).

Method	Latency	On- device	Suitable role
Faster R-CNN / Mask R-CNN [5], [6]	High	No	Offline segmenta- tion/benchmarks
RetinaNet [7]	Moderate	Limited	Near-real-time on powerful devices
YOLOv8n [10]	Low	Yes	Continuous detec- tion on-device
YOLOv8-CAB / LDG-Conv [11], [12]	Low	Yes	Small-object improved detection
SAM2 [14]	High (GPU)	No	Selective, on-demand segmentation (edge/cloud)
VLMs (BLIP, CLIP) [17], [18]	Moderate-High	No	On-demand enriched narration/VQA

XI. COMPARATIVE ANALYSIS

Table I summarizes representative methods and their suit- ability for continuous assistive deployment: latency, accuracy, on-device readiness and segmentation capability.

XII. OPEN PROBLEMS AND FUTURE DIRECTIONS

Key open problems include robustness under adverse lighting and weather, prompt compression and privacy-preserving offload, energy-aware scheduling for selective segmentation, improved small-object recall in compact detectors, and a shortage of large-scale user studies measuring real-world efficacy. Promising directions are multimodal fusion (camera + IMU + LiDAR), semantic SLAM for persistent environment memory, distilled prompt representations for SAM-like models, and edge-hosted VLM/LLM instances to lower interactive- latency for VQA [15], [19], [20].

XIII. EVALUATION METHODOLOGIES FOR ASSISTIVE VISION SYSTEMS

Assessing assistive vision systems requires a multi- dimensional evaluation that couples technical metrics with user-centered outcomes. Technical measurements should in- clude frame-level detection accuracy (mAP/IoU), end-to-end detection-to-speech latency (ms), energy per frame or per offload event, network bandwidth consumption for offloads, and the safety-critical miss rate (missed stairs, obstacles). These metrics quantify the runtime characteristics that determine whether a system can deliver timely and reliable alerts in the field [4], [7].

Perceptual and usability metrics complement algorithmic evaluation: speech intelligibility and naturalness (Mean Opin- ion Score), cognitive workload (NASA-TLX), task success rates in navigation or object-finding tasks, and standardized usability measures (SUS) provide evidence that algorithmic improvements translate to meaningful user benefits. Real-world datasets captured by visually impaired users (VizWiz, ORBIT) and field trials are essential to measure 'answerability' and to expose deployment artifacts not visible in curated benchmarks [1]–[3]. Combining these measurement classes in controlled lab studies and in-situ field trials yields comprehensive evidence of practical efficacy.

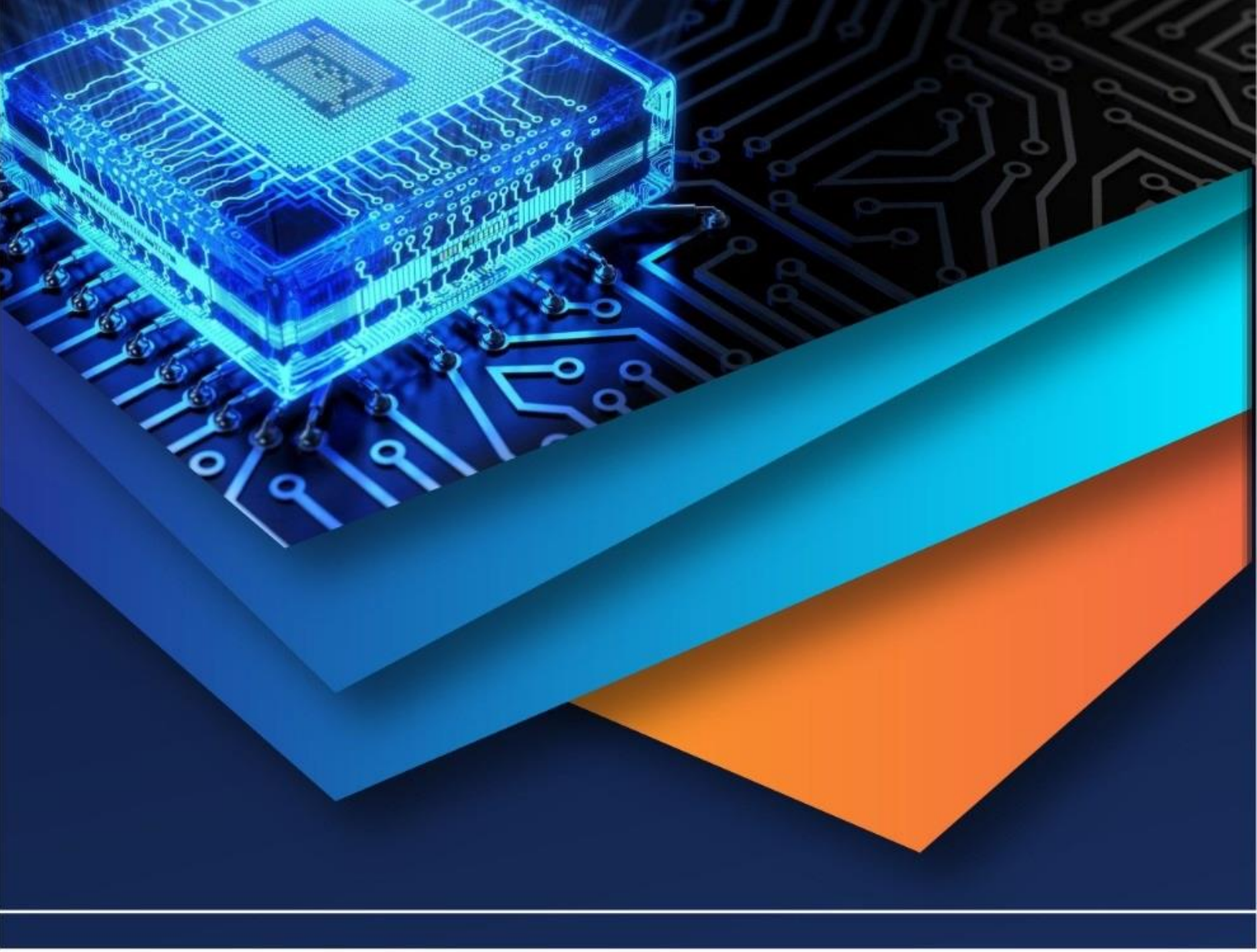
XIV. CONCLUSION

Hybrid architectures that combine continuous on-device detection (YOLOv8n or similar) with selective, promptable segmentation (SAM2) and a tiered narration strategy offer a pragmatic path toward deployable assistive vision systems. Achieving field readiness requires careful trigger policies, energy-efficient scheduling, privacy-preserving offload mech- anisms, and extensive user-centered evaluation to ensure practical utility.

This survey consolidates current progress in assistive vision and highlights how emerging detection, segmentation, and vision-language techniques can converge to build safer, more intelligent, and context-aware systems for visually impaired individuals.

REFERENCES

[1] J. P. Bigham *et al.*, "VizWiz: Nearly real-time answers to visual questions," in *Proc. UIST*, 2010.
[2] D. Gurari et al., "VizWiz-Captions: Captioning images taken by people who are blind," VizWiz Workshop / CVPR, 2020.
[3] D. B. Walker et al., "ORBIT: A dataset for few-shot personal object recognition," 2021.
[4] T.-Y. Lin et al., "Microsoft COCO: Common objects in context," in Proc. ECCV, 2014.
[5] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real- time object detection with region proposal networks," IEEE TPAMI, vol. 39, no. 6, pp. 1137–1149, 2017.
[6] K. He, G. Gkioxari, P. Dolla´r, and R. Girshick, "Mask R-CNN," in Proc. ICCV, 2017.
[7] T.-Y. Lin et al., "Focal loss for dense object detection," IEEE TPAMI, vol. 42, no. 2, pp. 318–327, 2020.
[8] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," arXiv:1804.02767, 2018.
[9] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in
[10] Proc. CVPR, 2017.
[11] G. Jocher, A. Chaurasia, and J. Qiu, "Ultralytics YOLO," 2023. (Ultra- lytics YOLOv8 resources and documentation uploaded.)
[12] M. Talib et al., "YOLOv8-CAB: Improved YOLOv8 for real-time object detection," Karbala Int. J. Mod. Sci., 2024.
[13] T.-W. Sung et al., "Improvement of YOLOv8 object detection based on lightweight neck model for complex images," Image Anal. Stereol., 2025.
[14] A. Kirillov et al., "Segment Anything," in Proc. ICCV, 2023.
[15] N. Ravi et al., "SAM 2: Segment anything in images and videos," arXiv:2408.00714, 2024.
[16] Y. Yamagishi et al., "SAM2 for zero-shot 3D segmentation," JMIR AI, 2025.
[17] D. Bolya et al., "YOLACT: Real-time instance segmentation," in Proc. ICCV, 2019.
[18] J. Li et al., "BLIP: Bootstrapping language-image pre-training for unified vision-language understanding and generation," in Proc. ICML, 2022.
[19] A. Radford et al., "CLIP: Learning transferable visual models from natural language supervision," in Proc. ICML, 2021.
[20] WalkVLM and related video-VLM works (uploaded arXiv entries, 2024).
[21] G. I. Okolo et al., "Assistive systems for visually impaired persons: Challenges and opportunities for navigation assistance," Sensors, 2024.
[22] P. Pfreundschuh et al., "Sight Guide: A wearable assistive perception and navigation system," arXiv:2506.02676, 2025.
[23] M. Talib et al., "Leveraging assistive technology for visually impaired people through optimal deep transfer learning based object detection," Sci. Rep., 2025.
[24] B.-H. Le et al., "Leveraging large vision-language models for visual question answering in VizWiz Grand Challenge," CVPR Workshop, 2024.
[25] A. G. Howard et al., "MobileNets: Efficient convolutional neural networks for mobile vision applications," arXiv:1704.04861, 2017.
[26] D. Ahmetovic´, C. Gleason, C. Ruan, K. M. Kitani, H. Takagi, and C. Asakawa, "NavCog: A navigational cognitive assistant for the blind," in Proc. MobileHCI, Florence, Italy, 2016, pp. 1–10, doi:10.1145/2935334.2935361.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY