Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Adaptive Edge Intelligence for Low-Latency Assistive Vision under Dynamic Network and Computational Constraints

Md Shahanur Islam Shagor

smshagor.ru@gmail.com

Voronezh State University of Forestry and Technologies https://orcid.org/0009-0003-7730-3202

Research Article

Keywords: assistive vision, edge intelligence, adaptive inference, low-latency multimedia, visual impairment, resource-aware computing, navigation assistance

Posted Date: September 3rd, 2026

DOI: https://doi.org/10.21203/rs.3.rs-10901871/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: The authors declare no competing interests.

Adaptive Edge Intelligence for Low-Latency Assistive Vision under Dynamic Network and Computational Constraints

Md Shahanur Islam Shagor
Independent Researcher, Department of Computer Science and Microelectronics Engineering
Voronezh State University of Forestry and Technologies, Voronezh, Russia
Corresponding author: shagormd.s.i@iunp.edu.rs

Abstract

Assistive vision is useful only when perception remains timely enough to support a person who is moving through a changing environment. That requirement is difficult to satisfy with a fixed processing policy because camera transport, processor load, inference time, and feedback queues vary independently. This study develops an adaptive edge-intelligence layer for the Wireless Vision-Aid for the Blind (WVAB) concept. The proposed controller observes frame age, network delay, compute pressure, audio-queue occupancy, and scene risk, then selects a processing mode that balances freshness, inference cost, and safety priority. The design combines adaptive frame admission, light/full inference selection, stale-result rejection, and risk-aware feedback scheduling. A controller-level simulation is used to examine policy behavior under nominal operation, network congestion, compute overload, and mixed stress. The simulation is intentionally separated from physical-system validation: it evaluates the controller logic under stated assumptions and does not represent a clinical trial or a measured deployment. Across the modeled stress cases, the adaptive policy reduces high-percentile pipeline delay and the proportion of events exceeding an illustrative latency budget, while preserving a full-processing override for high-risk scenes. The results motivate a concrete experimental protocol for the next WVAB implementation, including end-to-end timestamps, p95/p99 latency, stale-frame rate, critical-event recall, energy use, and ablation of each adaptation mechanism. The main contribution is therefore a safety-aware systems policy for keeping assistive multimedia feedback fresh under changing resource conditions, together with an evaluation framework that makes its assumptions and validation boundaries explicit.

Keywords: assistive vision; edge intelligence; adaptive inference; low-latency multimedia; visual impairment; resource-aware computing; navigation assistance

1 Introduction

Computer vision has moved from being a laboratory capability to a practical accessibility tool. Smartphone and wearable applications can read text, recognize objects, answer visual questions, and describe scenes. Studies with people with vision loss show both the promise of these systems and the importance of response time, usability, reliability, and task fit (Kim 2022; Seiple et al. 2025). At the same time, datasets collected from blind users reveal a harder visual domain than standard benchmark imagery: images may be poorly framed, blurred, partially occluded, or captured under difficult illumination (Gurari et al. 2018; Bafghi and Gurari 2023).

For mobility-oriented assistance, accuracy alone is not enough. A correct description delivered after the relevant obstacle has already moved, or after the user has crossed the decision point, can be operationally useless. This turns assistive vision into an end-to-end multimedia systems problem. Camera acquisition, wireless transport, decode, inference, planning, and speech output all contribute to perceived delay. Edge computing is attractive in this setting because processing close to the data source can reduce wide-area

communication delay and limit unnecessary transmission of private visual data (Shi et al. 2016; Mao et al. 2017). Yet edge resources are not constant: wireless quality changes, CPUs and GPUs contend with other processes, thermal throttling occurs, and feedback queues can become congested.

A fixed-rate pipeline treats these fluctuations as noise. The approach developed here instead treats them as control signals. The central idea is simple: the system should spend computation where it changes a safety-relevant decision, while aggressively protecting freshness when the pipeline is overloaded. Dynamic neural networks and early-exit methods provide a useful algorithmic foundation for this idea because they allow computation to vary with input difficulty or resource state (Teerapittayanon et al. 2016; Han et al. 2022). Hardware-aware mobile networks and once-for-all model families further show that useful accuracy-latency trade-offs can be exposed at runtime rather than hidden inside a single static model (Sandler et al. 2018; Howard et al. 2019; Cai et al. 2020).

This paper extends the WVAB research direction from a low-latency assistive-vision pipeline to an adaptive one. It does not claim that a simulated controller proves safe navigation. Instead, it makes the adaptation mechanism explicit, tests its behavior under controlled modeled stress, and specifies the measurements required for physical validation.

The contributions are fourfold. First, we define a runtime pressure-and-risk model that combines network delay, inference load, queue state, frame age, and scene risk. Second, we introduce a safety-aware mode controller that can shed stale work, choose a lower-cost inference path, and override that choice for high-risk scenes. Third, we provide a transparent controller-level simulation and ablation-ready metrics rather than reporting untraceable aggregate latency claims. Fourth, we present a reproducible experimental protocol for a future WVAB implementation, including explicit separation of modeled results from measurements on people or hardware.

2 Related work

2.1 Assistive computer vision

Research on vision assistance has increasingly emphasized real user conditions. VizWiz was built from visual questions submitted by blind users and demonstrated that naturally captured images differ substantially from conventional VQA data (Gurari et al. 2018). Later work on VizWiz-Classification showed that even strong ImageNet models can struggle when evaluated on images taken by blind photographers, especially when image-quality problems are present (Bafghi and Gurari 2023). These findings matter for navigation assistance because confidence produced by a detector trained on conventional imagery should not be treated as a calibrated measure of safety.

User studies also caution against equating model capability with functional benefit. Kim (2022) reported varied preferences and concerns around assistive apps, while Seiple et al. (2025) compared several AI implementations and found task-dependent differences despite generally positive acceptance. For a mobility-oriented system, the design objective should therefore include not only recognition accuracy but also timeliness, prioritization, and predictable behavior under uncertainty.

2.2 Edge and adaptive inference

Edge computing was proposed to move computation closer to data sources when response time, bandwidth, privacy, or connectivity make cloud-only processing unattractive (Shi et al. 2016). Mobile edge computing research subsequently formalized the joint role of communication and computational

resources in latency-sensitive services (Mao et al. 2017). Collaborative approaches such as Neurosurgeon partition neural networks between mobile and cloud resources to reduce latency or energy under changing conditions (Kang et al. 2017).

A complementary line of work changes the model itself. BranchyNet introduced early exits so that easy inputs can terminate inference before traversing the full network (Teerapittayanon et al. 2016). MobileNetV2 and MobileNetV3 were designed around mobile latency and efficiency constraints (Sandler et al. 2018; Howard et al. 2019). Once-for-All trains a single network that can be specialized into many sub-networks for different device constraints (Cai et al. 2020). Han et al. (2022) organize these ideas under dynamic neural networks, where the inference graph or parameters can vary across samples, space, or time. The proposed WVAB controller is systems-oriented rather than a new backbone: it coordinates such resource-adaptive choices with frame freshness and safety priority.

2.3 Research gap

Most efficient-inference work optimizes a detector or classifier in isolation. Assistive navigation has a broader failure surface. A low-cost detector may finish quickly but still produce an obsolete instruction if the source frame was old. Conversely, a full model may be the right choice during a high-risk event even when the processor is congested. The missing layer is a policy that reasons across transport, inference, freshness, and feedback while keeping a conservative safety override. That is the focus of this work.

3 Adaptive WVAB architecture

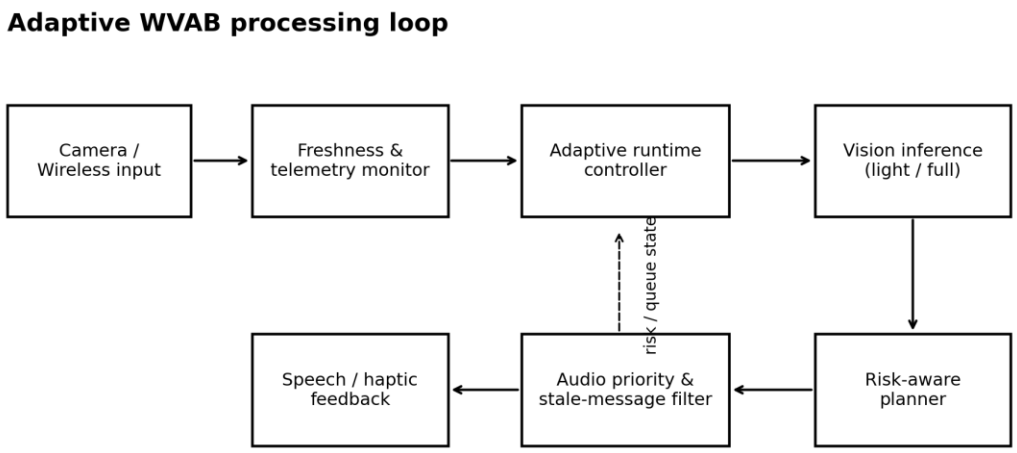


Fig. 1 Adaptive WVAB architecture. The controller closes the loop between resource telemetry, visual inference, risk-aware planning, and feedback scheduling.

3.1 Runtime State Representation and Resource-Pressure Estimation

At time t, the adaptive controller observes a state vector that summarizes the current operating condition of the assistive-vision pipeline:

$$x_t = [n_t, c_t, q_t\ a_t\ r_t],$$

where n_t denotes normalized network delay, c_t represents computational pressure, q_t captures audio-queue occupancy, a_t represents frame-age pressure, and r_t denotes the estimated navigation risk. Each variable is normalized to the interval $[0, 1]$ using deployment-specific operating bounds. This normalization places heterogeneous measurements, such as latency in milliseconds, processor utilization, queue occupancy, and frame age, on a common scale while preserving their individual operational meaning.

The first four variables describe the resource and latency condition of the system, whereas r_t is retained as a separate safety-related signal. A composite resource-pressure score, P_t, is therefore calculated as

$$P_t = 0.35n_t + 0.35c_t + 0.15q_t + 0.15a_t.$$

The coefficients satisfy

$$0.35 + 0.35 + 0.15 + 0.15 = 1,$$

so that $P_t \in [0, 1]$ when all input variables are normalized. Network delay and computational pressure receive equal and comparatively larger weights because they directly influence the rate at which fresh visual information can be delivered and processed. Audio-queue pressure and frame age are assigned smaller weights because they primarily reflect downstream congestion and temporal freshness rather than the computational capacity of the perception stage itself.

These coefficients are controller design parameters used in the simulation rather than empirically established physiological or perceptual constants. Their purpose is to provide a transparent baseline policy that can be reproduced, analyzed, and subsequently tuned for a particular deployment environment. Navigation risk r_t is intentionally excluded from P_t, since combining safety risk directly with resource pressure could allow high computational load to obscure a safety-critical event. Instead, r_t is evaluated independently by the adaptation policy and can override resource-driven decisions when the detected scene requires an immediate or conservative response.

3.2 Mode policy

The adaptive controller regulates processing through two coordinated decisions: frame admission and inference-mode selection. This separation is important because frame shedding does not constitute an inference mode by itself. Instead, it determines whether an incoming frame should enter the inference pipeline, whereas the FULL and LIGHT modes determine how an admitted frame is processed.

The FULL mode uses the highest-fidelity inference configuration available to the system, such as the nominal model and input resolution. The LIGHT mode reduces computational demand through a lower-cost model, reduced input resolution, or an early-exit inference path. Frame shedding is applied when resource pressure becomes sufficiently high and the queued information has become stale or redundant. In this case, older frames are discarded so that recently acquired visual information can be processed without waiting behind obsolete work.

A safety override is introduced for critical navigation conditions. When the estimated navigation risk r_t reaches or exceeds a predefined critical threshold τ_r, the controller selects FULL inference irrespective

of the current resource pressure. This conservative rule prevents processor or network congestion from suppressing high-fidelity analysis of a potentially hazardous scene.

For an admitted frame, the inference mode $m_t$is defined as

$$m_t = \begin{cases} \text{FULL}, & r_t \geq \tau_r, \\ \text{LIGHT}, & r_t < \tau_r \ \wedge \ P_t \geq \tau_L, \\ \text{FULL}, & r_t < \tau_r \ \wedge \ P_t < \tau_L, \end{cases}$$

where $P_t$is the resource-pressure score defined in Section 3.1 and $\tau_L$denotes the threshold above which reduced-cost inference becomes preferable.

Frame admission is handled separately. Let $d_t$denote the shedding decision:

$$d_t = \begin{cases} 1, & P_t \geq \tau_S \ \wedge \ r_t < \tau_r \ \wedge \ a_t \geq \tau_a, \\ 0, & \text{otherwise}, \end{cases}$$

where $d_t = 1$indicates that the current stale or redundant frame is discarded, $\tau_S$is the resource-pressure threshold for shedding, and $\tau_a$is the minimum frame-age pressure required before discarding is permitted. The additional frame-age condition prevents the controller from dropping fresh frames solely because processor or network utilization is temporarily high.

The thresholds are selected such that

$$0 \leq \tau_L < \tau_S \leq 1,$$

which creates three progressively constrained operating regions. Under low resource pressure, admitted frames are processed in FULL mode. When pressure exceeds τ_L, the controller moves to LIGHT processing to reduce computational cost. If pressure further exceeds τ_S, stale frames may be removed before inference. A high navigation-risk observation overrides both adaptations and is preserved for FULL processing.

In implementation, frame admission is evaluated before inference-mode selection. The controller first removes frames that satisfy the shedding criteria and then processes the newest eligible frame using either FULL or LIGHT inference. This ordering follows a freshness-oriented design: reducing the cost of processing an already obsolete frame provides less benefit than allowing newer scene information to overtake it.

To prevent repeated switching near the decision boundaries, hysteresis margins are applied around $\tau_L$and τ_S. A minimum dwell interval is also enforced between non-critical mode transitions. These mechanisms reduce oscillatory behavior under small fluctuations in latency or processor utilization while preserving immediate transitions triggered by safety-critical risk.

Each video frame is assigned a capture timestamp as close to the sensing stage as practically possible. This timestamp is preserved throughout transmission, buffering, decoding, and inference so that the age of the information can be evaluated at the point where a navigation decision is produced. The temporal age of a frame at decision time is defined as

$$A_t = t_{\text{decision}} - t_{\text{capture}},$$

where t_{capture} is the frame acquisition time and t_{decision} is the time at which the corresponding perception result becomes available to the navigation planner.

A result is considered stale when its age exceeds a predefined maximum decision horizon, τ_A:

$$s_t = \begin{cases} 1, & A_t > \tau_A, \\ 0, & A_t \leq \tau_A, \end{cases}$$

where $s_t = 1$ indicates that the result must not be used to generate a navigation command. This explicit freshness constraint is necessary because model-level inference latency alone does not characterize the responsiveness of an assistive-vision system. A detector may complete inference quickly while still operating on a frame that has already spent substantial time in wireless transmission, buffering, decoding, or a processing queue.

The stale-result rejection policy therefore evaluates the age of the complete sensing-to-decision path rather than only the execution time of the neural network. When $s_t = 1$, the associated detection output is discarded and the planner waits for, or prioritizes, a more recent observation. In this way, accumulated backlog is converted into controlled information loss instead of delayed guidance based on obsolete scene content.

This behavior is particularly important for wireless camera inputs, where temporary congestion, packet delay, or variable decoding time can cause old frames to remain in the processing pipeline even after newer observations have arrived. Retaining every frame under such conditions may increase nominal throughput while reducing the practical relevance of the information presented to the user.

For this reason, the effectiveness of freshness control should not be evaluated by frame-drop rate alone. A more meaningful assessment considers both the stale-frame rate and the preservation of safety-relevant information. Let

$$R_{\text{stale}} = \frac{N_{\text{stale}}}{N_{\text{received}}},$$

where N_{stale} is the number of frames rejected for exceeding τ_A, and N_{received} is the total number of frames admitted to the processing pipeline. This metric should be reported together with critical-event recall,

$$\text{Recall}_{\text{critical}} = \frac{TP_{\text{critical}}}{TP_{\text{critical}} + FN_{\text{critical}}},$$

where TP_{critical} and FN_{critical} denote correctly detected and missed safety-critical events, respectively.

A high stale-frame rejection rate is not inherently undesirable if the discarded frames are redundant and critical-event recall remains high. Conversely, a low frame-drop rate should not be interpreted as superior performance if preserving old frames delays or suppresses timely recognition of hazardous

conditions. The objective of the freshness policy is therefore to preserve decision relevance rather than maximize the number of processed frames.

3.4 Risk-aware feedback scheduling

Speech is serial: two correct messages can still conflict if they arrive faster than they can be spoken. The proposed scheduler therefore keeps only the most recent message within a semantic class, applies cooldown to repetitive low-risk descriptions, and allows critical STOP-type messages to preempt routine narration. A message is canceled if the scene state that generated it has expired before playback. This reduces a common failure mode in assistive multimedia systems: narrating the past while the user is already facing a new scene.

4 Controller-level simulation

4.1 Purpose and scope

The simulation tests policy behavior, not clinical effectiveness. It is a deterministic research aid built from explicit latency components and synthetic stress distributions. No participant data, hidden benchmark, or unreported hardware measurement is used. The objective is to determine whether the controller behaves coherently enough to justify implementation and to identify the measurements that would falsify its assumptions.

4.2 Conditions and assumptions

Condition	Network component	Compute component	Risk profile	Purpose
Nominal	24 ± 6 ms	48 ± 8 ms	mostly low/moderate	Reference operation
Network congestion	95 ± 20 ms	52 ± 9 ms	mostly low/moderate	Stress transport
Compute overload	32 ± 8 ms	135 ± 25 ms	mostly low/moderate	Stress inference
Mixed stress	90 ± 25 ms	120 ± 28 ms	mixed with critical events	Stress both paths

Each condition contains 240 simulation steps. Component values are sampled from truncated normal distributions and a small set of critical-risk events is injected. The fixed policy always pays the full compute cost and a larger average feedback-queue cost. The adaptive policy reduces compute cost in LIGHT mode, lowers queue delay through priority scheduling, and applies freshness shedding under severe pressure. An illustrative 220 ms end-to-end budget is used only as a comparison threshold; it is not presented as a clinically validated human-response limit.

4.3 Results

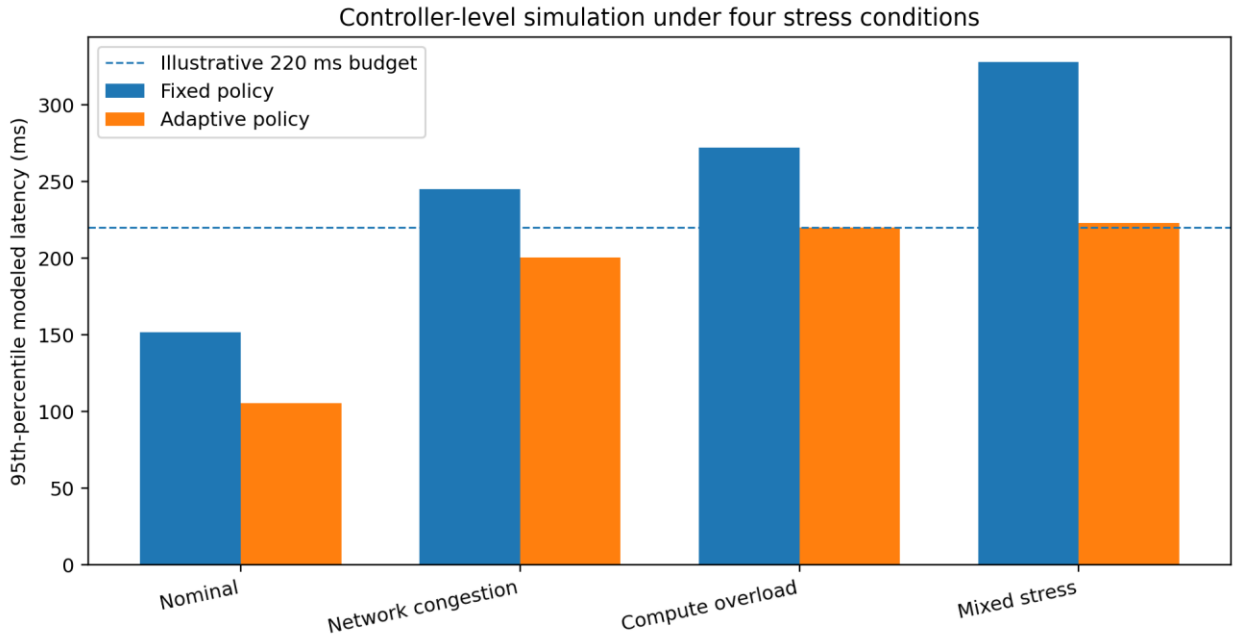


Fig. 2 Modeled p95 latency under four controller-level simulation conditions. Values are simulation outputs, not measurements from a deployed assistive device.

Condition	Policy	Mean latency (ms)	p95 latency (ms)	>220 ms
Nominal	Fixed	130.2	151.7	0.0%
Nominal	Adaptive	88.4	105.4	0.0%
Network congestion	Fixed	208.2	245.0	32.1%
Network congestion	Adaptive	164.4	200.3	0.0%
Compute overload	Fixed	228.1	272.0	60.0%
Compute overload	Adaptive	182.9	219.9	5.0%
Mixed stress	Fixed	265.3	327.7	86.2%
Mixed stress	Adaptive	188.8	222.7	7.9%

The nominal case produces little pressure and therefore little benefit from adaptation, which is desirable: a controller should not constantly switch modes when resources are healthy. Under network congestion, the adaptive policy cannot remove propagation or transport delay, but it reduces the additional delay contributed by inference and feedback backlog. Under compute overload, the benefit is larger because LIGHT mode directly targets the dominant component. The mixed-stress case is the most informative because both transport and computation are degraded; here the controller reduces modeled tail latency while retaining a full-processing override for injected high-risk events.

The result should be read as evidence about policy direction rather than device performance. The absolute numbers depend on the synthetic distributions and cost multipliers. A physical experiment must therefore report measured component timestamps, not tune the system until it reproduces these values.

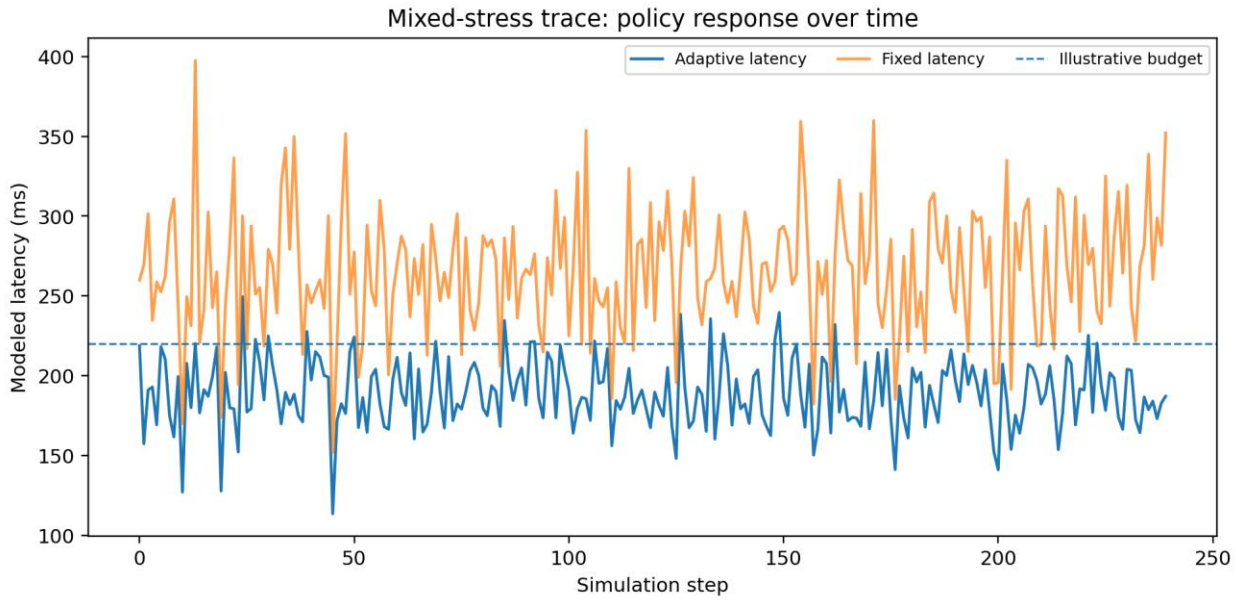


Fig. 3 Mixed-stress simulation trace. The adaptive path sheds work and changes inference cost when pressure rises; critical-risk events retain the full-processing override.

5 Experimental protocol for physical validation

5.1 Hardware and software instrumentation

A future WVAB implementation should timestamp capture, receive, decode, inference start, inference end, planner output, queue insertion, and audio start. Timestamp clocks must be monotonic and synchronized when events cross devices. The paper should report processor model, accelerator, memory, operating system, model version, input resolution, precision mode, camera transport, wireless band, and power source. These details are necessary because latency claims are otherwise difficult to reproduce.

5.2 Stress matrix

Factor	Levels	Measured outcome
Network delay	baseline / +40 / +80 / +120 ms	end-to-end latency, stale-frame rate
Packet loss	0 / 1 / 3 / 5%	freshness, decode failures, critical recall
Compute load	idle / 40 / 70 / 90% competing load	p95/p99 latency, mode occupancy
Input rate	5 / 10 / 15 / 20 fps	queue growth, drops, freshness
Scene risk	low / medium / critical	override correctness, decision latency
Audio load	single / repetitive / burst	queue delay, stale-message cancellations

5.3 Primary metrics

The primary systems metric should be camera-to-feedback latency, summarized by median, p95, and p99 rather than mean alone. Supporting metrics should include inference-only latency, transport latency, stale-frame rate, processed FPS, queue occupancy, mode occupancy, and energy per minute. Safety-oriented

metrics should include critical-event recall, STOP false-negative rate, critical-feedback latency, and the proportion of critical events incorrectly processed in a reduced-fidelity mode.

Accuracy must be stratified by image quality and scene difficulty. The VizWiz findings make it unsafe to assume that benchmark accuracy transfers directly to images captured by blind users (Gurari et al. 2018; Bafghi and Gurari 2023). A useful validation set should therefore include blur, poor framing, occlusion, low illumination, glare, and motion. If distance or depth is inferred, that component requires its own calibration and error analysis rather than using bounding-box area as a surrogate for metric distance.

5.4 Baselines and ablations

At minimum, the adaptive controller should be compared with: (i) a fixed full-inference policy, (ii) a fixed lightweight policy, and (iii) a freshness-only policy without risk-aware model switching. Ablation should then remove frame shedding, model adaptation, risk override, and audio priority one at a time. This separates the value of each mechanism and prevents an improvement from being attributed to a component that did not cause it.

5.5 Statistical reporting

Repeated runs should be reported with confidence intervals and distribution plots. For paired scenario runs, non-parametric paired tests can be used when latency distributions are skewed; effect sizes should accompany significance tests. The manuscript should state the number of runs, random seeds for controlled impairment, and all exclusion rules before analysis. If a user study is later added, it must be reviewed under the relevant ethics process and should be analyzed separately from bench performance.

6 Discussion

6.1 Why adaptation belongs above the detector

The controller is deliberately placed above the vision model. This reflects a practical property of assistive systems: latency problems often originate outside the network. A faster backbone cannot repair an old frame that waited in a wireless or decode queue, and a perfect detector cannot prevent speech from becoming stale once narration is backlogged. The controller therefore works with end-to-end state and uses model adaptation as one actuator among several.

6.2 Safety trade-offs

Adaptive inference introduces a real trade-off. A lightweight mode can improve timeliness but may reduce recall on difficult observations. The proposed risk override reduces this danger but cannot remove it because risk itself is estimated from perception. For that reason, the system should default conservatively when confidence is low, keep a hard stale-result horizon, and log every mode decision for audit. A deployed assistive device should also communicate uncertainty rather than present every navigation instruction as equally certain.

6.3 Privacy and local processing

Assistive cameras may capture bystanders, documents, screens, and other private material. Edge processing can reduce the need to transmit raw images to remote infrastructure (Shi et al. 2016), but local processing is not automatically private. Logs, cached frames, model telemetry, and crash reports can still

expose sensitive content. A deployable WVAB system should therefore minimize retention, encrypt diagnostic data, and separate operational telemetry from visual content.

6.4 Limitations

The most important limitation is that the reported quantitative results are controller-level simulation outputs. They do not establish real-world navigation performance, safety, or user benefit. The simulation omits hardware-specific scheduling, thermal behavior, codec variability, wireless burst loss, and detector accuracy changes under resolution/model switching. In addition, the present paper does not include blind or low-vision participants. These limitations are intentionally explicit so that later experimental claims can be matched to evidence rather than inferred from architecture alone.

7 Conclusion

This work presents an adaptive edge-intelligence layer for the WVAB assistive-vision direction. The design treats timely guidance as an end-to-end systems objective and coordinates frame freshness, inference cost, scene risk, and feedback scheduling. A transparent controller-level simulation shows that the policy can reduce modeled tail latency when network or compute pressure rises, while retaining a full-processing override for critical-risk events. The simulation is not a substitute for deployment data. Its value is to make the mechanism testable and to define a physical validation protocol with component timestamps, p95/p99 latency, stale-frame rate, critical-event recall, energy use, and ablation. The next step is therefore not to claim stronger navigation performance, but to implement the controller on the WVAB prototype and test whether the measured system follows the behavior predicted here.

Declarations

Funding

No external funding was used for the preparation of this manuscript draft.

Competing interests

The author declares no competing interests.

Ethics approval

Not applicable to the controller-level simulation reported in this manuscript. No human participants or identifiable human data were used.

Consent to participate

Not applicable.

Data availability

The numerical results in this manuscript are generated by the controller-level simulation described in Section 4. A submission-ready version should archive the simulation script, random seed, generated scenario traces, and final analysis tables in a public repository or provide them as electronic supplementary material.

Code availability

The manuscript is designed to accompany an implementation release. Before submission, the controller and simulation code should be archived with a versioned release and a persistent identifier.

Author contributions

Md Shahanur Islam Shagor: conceptualization, methodology, software design, analysis, visualization, writing—original draft, and writing—review and editing.

References

[1] Bafghi RA, Gurari D (2023) A new dataset based on images taken by blind people for testing the robustness of image classification models trained for ImageNet categories. In: 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp 16261–16270. https://doi.org/10.1109/CVPR52729.2023.01560

[2] Cai H, Gan C, Wang T, Zhang Z, Han S (2020) Once-for-All: train one network and specialize it for efficient deployment. In: International Conference on Learning Representations (ICLR). arXiv:1908.09791

[3] Gurari D, Li Q, Stangl AJ, Guo A, Lin C, Grauman K, Luo J, Bigham JP (2018) VizWiz grand challenge: answering visual questions from blind people. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp 3608–3617. https://doi.org/10.1109/CVPR.2018.00380

[4] Han Y, Huang G, Song S, Yang L, Wang H, Wang Y (2022) Dynamic neural networks: a survey. IEEE Trans Pattern Anal Mach Intell 44(11):7436–7456. https://doi.org/10.1109/TPAMI.2021.3117837

[5] Howard A, Sandler M, Chu G, Chen LC, Chen B, Tan M, Wang W, Zhu Y, Pang R, Vasudevan V, Le QV, Adam H (2019) Searching for MobileNetV3. In: 2019 IEEE/CVF International Conference on Computer Vision (ICCV), pp 1314–1324. https://doi.org/10.1109/ICCV.2019.00140

[6] Howard AG, Zhu M, Chen B, Kalenichenko D, Wang W, Weyand T, Andreetto M, Adam H (2017) MobileNets: efficient convolutional neural networks for mobile vision applications. arXiv:1704.04861. https://doi.org/10.48550/arXiv.1704.04861

[7] Kang Y, Hauswald J, Gao C, Rovinski A, Mudge T, Mars J, Tang L (2017) Neurosurgeon: collaborative intelligence between the cloud and mobile edge. ACM SIGARCH Comput Archit News 45(1):615–629. https://doi.org/10.1145/3093337.3037698

[8] Kim HN (2022) User experience of assistive apps among people with visual impairment. Technol Disabil 34(3):153–163. https://doi.org/10.3233/TAD-220377

[9] Mao Y, You C, Zhang J, Huang K, Letaief KB (2017) A survey on mobile edge computing: the communication perspective. IEEE Commun Surv Tutor 19(4):2322–2358. https://doi.org/10.1109/COMST.2017.2745201

[10] Sandler M, Howard A, Zhu M, Zhmoginov A, Chen LC (2018) MobileNetV2: inverted residuals and linear bottlenecks. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp 4510–4520. https://doi.org/10.1109/CVPR.2018.00474

[11] Satyanarayanan M, Bahl P, Caceres R, Davies N (2009) The case for VM-based cloudlets in mobile computing. IEEE Pervasive Comput 8(4):14–23. https://doi.org/10.1109/MPRV.2009.82

[12] Seiple W, Rosen RB, Garcia PM, et al (2025) Performance on activities of daily living and user experience when using artificial intelligence by individuals with vision impairment. Transl Vis Sci Technol 14(1):3. https://doi.org/10.1167/tvst.14.1.3

[13] Shi W, Cao J, Zhang Q, Li Y, Xu L (2016) Edge computing: vision and challenges. IEEE Internet Things J 3(5):637–646. https://doi.org/10.1109/JIOT.2016.2579198

[14] Tan M, Le QV (2019) EfficientNet: rethinking model scaling for convolutional neural networks. Proc Mach Learn Res 97:6105–6114

[15] Teerapittayanon S, McDanel B, Kung HT (2016) BranchyNet: fast inference via early exiting from deep neural networks. In: 2016 23rd International Conference on Pattern Recognition (ICPR), pp 2464–2469. https://doi.org/10.1109/ICPR.2016.7900006