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.

Risk-Aware Adaptive Video Processing with Freshness Control for Low-Latency Assistive Vision

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, adaptive video processing, edge intelligence, dynamic inference, frame freshness, resource-aware computing

Posted Date: September 3rd, 2026

DOI: https://doi.org/10.21203/rs.3.rs-10902455/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.

Risk-Aware Adaptive Video Processing with Freshness Control for Low-Latency Assistive Vision

Md Shahanur Islam Shagor

Independent Researcher

Corresponding author: Md Shahanur Islam Shagor

Corresponding author Email: shagormd.s.i@iunp.edu.rs

Abstract

Latency-sensitive assistive vision requires more than fast neural-network inference: frames can become obsolete during wireless transport, decoding, queueing, and feedback generation. This paper presents a risk-aware adaptive video-processing controller for the Wireless Vision-Aid for the Blind (WVAB) research direction. The controller monitors normalized network delay, compute pressure, queue pressure, frame age, and navigation risk, and then separates frame admission from inference-mode selection. Fresh low-risk frames may use a lightweight path when resources are constrained, stale non-critical frames may be shed before inference, and critical-risk observations always receive full-fidelity processing. A deterministic controller-level simulation with 960 steps evaluates nominal operation, network congestion, compute overload, and mixed stress. Relative to a fixed full-inference policy, the adaptive controller reduces modeled 95th-percentile end-to-end latency by 16.4% under network congestion, 18.6% under compute overload, and 31.5% under mixed stress. In the mixed condition, 12.5% of stale non-critical frames are shed, while the critical-risk override prevents critical frames from entering the shedding path by construction. The study deliberately distinguishes simulation from hardware or clinical evidence. Its contribution is a reproducible freshness-aware scheduling method, together with measurable criteria for later physical validation, including p95/p99 latency, stale-frame rate, critical-event recall, mode occupancy, energy use, and ablation of each adaptation mechanism.

Keywords: assistive vision; adaptive video processing; edge intelligence; dynamic inference; frame freshness; resource-aware computing

1 Introduction

Computer vision is increasingly used in accessibility tools that read text, identify objects, answer visual questions, and describe scenes. User studies, however, show that practical value depends on more than nominal recognition accuracy. People with visual impairment report task-dependent preferences, usability concerns, and differences in the usefulness of assistive applications [1, 2]. The visual data themselves are also challenging: images captured by blind users may contain blur, poor framing, occlusion, unusual viewpoints, or difficult illumination, and models trained on conventional benchmark imagery can degrade substantially on this domain [3, 4].

For mobility-oriented assistance, timeliness is part of correctness. A correct description delivered after the relevant scene has changed can be misleading even when the detector itself is accurate. The useful latency budget therefore spans camera acquisition, wireless transport, decoding, inference, planning, queueing, and feedback. Edge computing reduces dependence on distant cloud infrastructure and can shorten communication paths, but it does not make compute or network resources constant [5-7]. Wireless delay fluctuates, processors contend with other tasks, thermal throttling can reduce throughput, and speech output can become backlogged.

Dynamic inference provides a complementary mechanism. Early-exit networks, dynamic neural networks, mobile-optimized architectures, and once-for-all model families expose accuracy-computation trade-offs that can be used at runtime [8-12]. Yet most of these techniques optimize the inference model itself. In a live video pipeline, model adaptation alone cannot recover freshness after a frame has already waited in a transport or decode queue. The system also needs a policy for deciding which frame should be processed, at what fidelity, and whether the resulting decision is still temporally valid.

This paper develops that policy for the WVAB assistive-vision research direction. The work is intentionally scoped as an image/video-processing systems study rather than a clinical evaluation. The controller coordinates frame freshness, resource pressure, inference cost, and navigation risk. The results reported here come only from a new controller-level simulation; no hardware measurements, user-study outcomes, or results from an earlier WVAB manuscript are reused.

The main contributions are:

- a normalized runtime state and resource-pressure model that combines transport, computation, queue state, and frame freshness without mixing incompatible units;
- a two-stage policy that separates stale-frame admission from FULL/LIGHT inference selection and applies a hard full-fidelity override to critical-risk observations;
- a deterministic stress simulation with explicit thresholds, seed, component distributions, and tail-latency metrics; and
- a physical-validation protocol that links systems performance to stale-frame rate and critical-event recall rather than frame throughput alone.

2 Related work

2.1 Assistive visual data and user requirements

VizWiz established a large visual-question-answering resource built from questions and images submitted by blind users [3]. The later VizWiz-Classification dataset showed that ImageNet-trained classifiers remain vulnerable to authentic image-quality problems in photographs taken by blind people [4]. These datasets are important for assistive navigation because detector confidence on conventional imagery should not be interpreted as a calibrated safety score. A systems policy should be evaluated under degraded image quality as well as under clean laboratory inputs.

User-facing studies reinforce this point. Kim [1] reported diverse needs and concerns when people with visual impairment used assistive applications, while Seiple et al. [2] found task-dependent differences across AI-based assistive implementations. These results motivate a design objective that includes timeliness, prioritization, and predictable behavior under uncertainty rather than maximizing recognition accuracy in isolation.

2.2 Edge and dynamic inference

Edge and mobile-edge computing place processing closer to the data source when wide-area latency, bandwidth, privacy, or connectivity are limiting factors [5-7]. Collaborative systems such as Neurosurgeon partition deep networks between mobile and edge resources according to measured latency or energy [13]. A second line of work changes the model itself. BranchyNet allows easy inputs to exit a network early [8], dynamic neural networks generalize conditional

computation across architectures [9], and MobileNet-family models target efficient mobile vision [10, 11, 15]. EfficientNet formalizes compound scaling across depth, width, and resolution [16], while Once-for-All supports runtime specialization from a shared trained network [12].

Recent work in Signal, Image and Video Processing also demonstrates that adaptive sampling can reduce sensing and inference cost while maintaining a controllable accuracy-resource trade-off [14]. That line of work adapts the sampled signal itself. The present study addresses a different but complementary level of the pipeline: temporal freshness and resource-aware scheduling of video frames that have already entered an assistive perception system.

2.3 Research gap

The unresolved problem is cross-layer coordination. A lighter model may reduce inference time while still producing a stale result. Conversely, aggressive frame dropping may improve throughput but erase safety-relevant evidence. The controller must therefore treat frame age as a first-class signal and preserve high-risk observations even under resource pressure. This requirement motivates the two-stage policy developed below.

3 Adaptive video-processing architecture

Figure 1 shows the proposed processing loop. Camera frames enter a freshness and telemetry monitor before inference. The runtime controller receives transport, compute, queue, and frame-age information, while the risk-aware planner feeds scene risk back to the controller. The controller can reduce inference cost, reject stale work, or retain full processing when the scene is potentially hazardous. Feedback scheduling is included because stale speech can remain harmful even after the visual inference stage has completed.

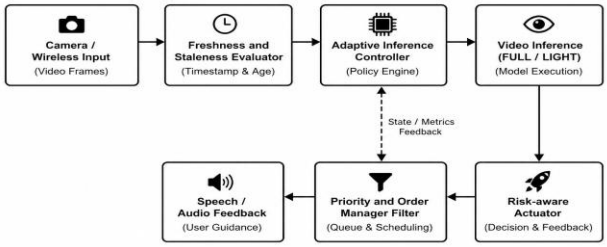


Fig. 1 Adaptive WVAB video-processing loop with runtime feedback from risk and queue state

3.1 Runtime state and resource-pressure estimation

At time t, the adaptive controller observes a state vector that summarizes the current operating condition of the video pipeline. The vector contains normalized network delay n_t, computational pressure c_t, audio-queue pressure q_t, frame-age pressure a_t, and estimated navigation risk r_t. Each variable is mapped to [0,1] using deployment-specific operating bounds. Normalization places heterogeneous measurements such as milliseconds, processor utilization, queue occupancy, and frame age on a common scale without treating them as physically interchangeable quantities.

$$x_t = [n_t, c_t, q_t, a_t, r_t] \quad (1)$$

The first four variables describe resource and latency conditions, whereas r_t is retained as a separate safety signal. The composite resource-pressure score P_t is therefore defined as

$$P_t = 0.35n_t + 0.35c_t + 0.15q_t + 0.15a_t \quad (2)$$

The coefficients sum to one, so P_t remains in [0,1] when all inputs 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. Queue and frame-age terms receive smaller weights because they mainly describe downstream congestion and temporal freshness. These values are controller design parameters used in the simulation, not physiological constants or empirically established human-response weights. Navigation risk is excluded from P_t so that high resource pressure cannot numerically dilute a safety-critical observation.

3.2 Two-stage mode policy

The adaptive controller makes two coordinated decisions: frame admission and inference-mode selection. This separation is important because shedding is not itself an inference mode. Frame admission decides whether a frame should enter the neural pipeline, whereas FULL and LIGHT specify how an admitted frame is processed. FULL uses the nominal highest-fidelity configuration. LIGHT uses a lower-cost model, reduced input resolution, or an early-exit path, depending on the deployed implementation.

For an admitted frame, the inference mode m_t is selected using the critical-risk threshold τ_r and the lightweight threshold τ^L:

$$m_t = \begin{cases} FULL, & \text{if } r_t \geq \tau_r \\ LIGHT, & \text{if } r_t < \tau_r \text{ and } P_t \geq \tau^L \\ FULL, & \text{if } r_t < \tau_r \text{ and } P_t < \tau^L \end{cases} \quad (3)$$

A critical-risk observation therefore forces FULL inference irrespective of resource pressure. This conservative override prevents compute or network congestion from suppressing high-fidelity analysis solely to preserve throughput.

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

$$d_t = \begin{cases} 1, & \text{if } P_t \geq \tau^s \text{ and } a_t \geq \tau_A \text{ and } r_t < \tau_r \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

Here $d_t = 1$ means that the stale or redundant frame is discarded before inference, τ^s is the pressure threshold for shedding, and τ_A is the minimum normalized frame-age pressure required before shedding is permitted. The frame-age condition prevents the controller from dropping fresh frames solely because processor or network utilization is briefly high. The thresholds satisfy $0 \leq \tau^L < \tau^s \leq 1$. In the simulation, $\tau_r = 0.80$, $\tau^L = 0.50$, $\tau^s = 0.72$, and $\tau_A = 0.70$.

In implementation, frame admission is checked before model selection. If a frame is rejected, the newest eligible observation can overtake it. Otherwise the admitted frame is processed in FULL or LIGHT mode. Hysteresis should be applied around τ^L and τ^s, together with a minimum dwell interval between non-critical mode changes, to prevent oscillation when pressure varies near a boundary. Critical-risk transitions are exempt from the dwell rule.

3.3 Freshness and stale-result rejection

Each frame is assigned a capture timestamp as close to the sensing stage as practical. The timestamp is carried through transmission, buffering, decoding, and inference so that temporal validity can be checked at the decision point. The age of a perception result is

$$A_t = t_decision - t_capture \quad (5)$$

where t_capture is the frame acquisition time and t_decision is the time at which the associated result becomes available to the planner. A result is stale when its age exceeds the maximum decision horizon τ^D:

$$s_t = 1, \text{ if } A_t > \tau^D \quad (6)$$

$$0, \text{ if } A_t \le \tau^D$$

When $s_t = 1$, the result is not allowed to generate a navigation instruction. This check is necessary because inference-only latency can look excellent even when the source frame has already spent substantial time in transport or a queue. The policy converts backlog into controlled information loss instead of delayed guidance based on obsolete scene content.

The corresponding evaluation metric is not raw frame-drop rate. A useful system may intentionally discard redundant frames while preserving safety-relevant observations. The stale-frame rate is

$$R_stale = N_stale / N_received \quad (7)$$

and should be interpreted together with critical-event recall measured on physical validation data:

$$Recall_critical = TP_critical / (TP_critical + FN_critical) \quad (8)$$

A high stale-frame rejection rate is acceptable only when discarded information is redundant and critical-event recall remains high. Conversely, preserving every frame is not a success if old frames delay recognition of a hazard. The design objective is therefore decision freshness rather than maximum processed-frame count.

3.4 Risk-aware feedback scheduling

Speech is serial and can become a second source of staleness. The scheduler keeps only the most recent message within a semantic class, applies cooldown to repetitive low-risk descriptions, and gives STOP-type messages priority over routine narration. A queued message is canceled when the scene state that generated it has expired before playback. This prevents the system from narrating an earlier scene while the user is already facing a new one.

4 Controller-level simulation

4.1 Reproducible setup

The controller was evaluated with a deterministic Python simulation using NumPy and a fixed random seed of 42. Four conditions were simulated for 240 steps each, producing 960 controller observations. Component delays were sampled from non-negative normal distributions. Network and full-inference components varied by condition, while decode and planning delays used means of 18 ms and 8 ms, respectively. Pre-inference frame age was defined as network delay plus decode delay plus queue waiting time. The normalized variables were computed from explicit upper operating scales: 160 ms for network delay, 190 ms for full-compute time, 90 ms for queue delay, and 220 ms for pre-inference age.

The fixed baseline processes every frame with full compute cost and the original queue delay. The adaptive policy applies the thresholds defined in Section 3.2. LIGHT mode multiplies the sampled full-compute cost by 0.55, while non-critical feedback queue time is multiplied by 0.50. Critical-risk observations retain the full compute path and cap the modeled feedback queue at 18 ms. SHED frames are excluded from processed-frame latency because no navigation result is produced for those observations. The 220 ms line used in the figures is an illustrative engineering budget for comparing policy behavior; it is not presented as a clinically validated response-time threshold.

Table 1 Simulation conditions and component distributions (mean ± standard deviation, ms)

Condition	Operating parameters	Purpose
Nominal	Network 24 ± 6 ms; full compute 48 ± 8 ms; queue 18 ± 5 ms; critical risk 4%	Reference operation
Network congestion	Network 95 ± 20 ms; full compute 52 ± 9 ms; queue 34 ± 10 ms; critical risk 4%	Transport stress
Compute overload	Network 32 ± 8 ms; full compute 135 ± 25 ms; queue 45 ± 12 ms; critical risk 4%	Inference stress
Mixed stress	Network 90 ± 25 ms; full compute 120 ± 28 ms; queue 55 ± 15 ms; critical risk 8%	Joint transport/compute stress

4.2 Tail-latency results

Table 2 Controller-level simulation results

Condition	Fixed policy	Adaptive policy	Shed
Nominal	mean 115.4; p95 133.4; >220 ms 0.0%	mean 107.0; p95 123.0; >220 ms 0.0%	0.0%
Network congestion	mean 207.9; p95 249.8; >220 ms 29.6%	mean 184.3; p95 208.9; >220 ms 1.2%	0.0%
Compute overload	mean 234.3; p95 284.4; >220 ms 68.8%	mean 195.9; p95 231.5; >220 ms 14.6%	0.0%
Mixed stress	mean 290.0; p95 354.5; >220 ms 97.1%	mean 208.5; p95 242.8; >220 ms 31.0%	12.5%

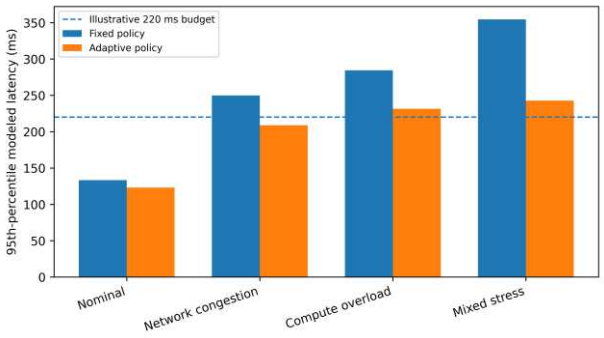


Fig. 2 Modeled 95th-percentile latency for fixed and adaptive policies under four stress conditions

The nominal condition produced the smallest change, with p95 latency decreasing from 133.4 ms to 123.0 ms. This is expected because the controller remains in FULL mode when resources are healthy. Under network congestion, p95 latency fell from 249.8 ms to 208.9 ms, a 16.4% reduction. Under compute overload, p95 latency decreased from 284.4 ms to 231.5 ms, an 18.6% reduction. The largest change occurred under mixed stress, where p95 latency declined from 354.5 ms to 242.8 ms, a 31.5% reduction. The proportion of processed adaptive observations above the illustrative 220 ms budget also fell sharply in the network and compute stress cases.

The mixed condition is deliberately severe. The adaptive p95 remains above 220 ms, so the simulation does not imply that software adaptation can eliminate high transport and compute delay. Instead, the controller sheds 12.5% of stale non-critical frames and moves 72.5% of observations to LIGHT mode. Only 15.0% of mixed-stress observations remain in FULL mode, consisting of low-pressure observations and the injected critical-risk events. Because the

risk override is a hard policy rule, every observation with $r_t \geq 0.80$ is routed to FULL and none is shed; this should be interpreted as a controller invariant, not a measured detection-recall result.

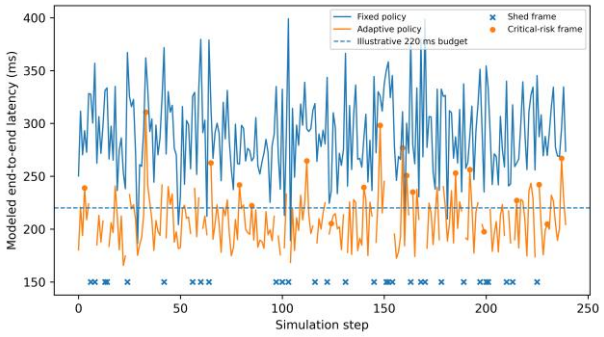


Fig. 3 Mixed-stress trace showing modeled fixed/adaptive latency, shed frames, and critical-risk observations

4.3 What the simulation establishes

The simulation establishes only the internal behavior of the scheduling policy under the stated assumptions. It shows that frame admission, lightweight inference, and priority feedback can reduce modeled tail latency when the corresponding cost terms dominate. It does not establish detector accuracy under resolution switching, battery life, thermal behavior, wireless robustness, user safety, or clinical benefit. Those questions require measured hardware and user-facing experiments.

5 Physical validation protocol

5.1 Instrumentation

A physical WVAB implementation should timestamp camera capture, receive, decode start/end, inference start/end, planner output, feedback-queue insertion, and audio start. Clocks must be monotonic and synchronized when events cross devices. The report should identify processor and accelerator models, memory, operating system, inference runtime, model version, input resolution, numerical precision, camera transport, wireless band, and power source. Without these details, end-to-end latency claims are difficult to reproduce.

Table 3 Recommended stress matrix for hardware validation

Factor	Levels	Primary outcomes
Network delay	baseline / +40 / +80 / +120 ms	p95/p99 latency; stale-frame rate
Packet loss	0 / 1 / 3 / 5%	decode failure; freshness; critical recall
Competing compute load	idle / 40 / 70 / 90%	mode occupancy; p95/p99 latency
Video input rate	5 / 10 / 15 / 20 fps	queue growth; shedding; freshness
Scene risk	low / moderate / critical	override correctness; decision latency
Feedback load	single / repetitive / burst	queue delay; stale-message cancellation

5.2 Metrics and baselines

The primary systems metric should be camera-to-feedback latency summarized by median, p95, and p99 rather than mean alone. Supporting measures should include transport delay, inference-only latency, stale-frame rate, processed frames per second, queue occupancy, FULL/LIGHT/SHED occupancy, memory use, and energy per minute. Safety-oriented measurements should include critical-event recall, STOP false-negative rate, critical-feedback latency, and the proportion of critical observations incorrectly handled by a reduced-fidelity path.

Accuracy must be stratified by image quality and scene difficulty. Results on VizWiz and VizWiz-Classification show that authentic images captured by blind users can differ sharply from conventional benchmark data [3, 4]. A validation set should therefore include blur, poor framing, partial occlusion, low illumination, glare, and motion. If depth or distance is later introduced, it must be calibrated and evaluated as a separate estimation problem rather than inferred from bounding-box area alone.

At minimum, the adaptive method should be compared with three baselines: fixed FULL inference, fixed LIGHT inference, and freshness-only control without risk-aware model switching. Ablations should remove frame shedding, model adaptation, the risk override, and feedback prioritization one at a time. This prevents system-level improvements from being attributed to a component that did not cause them.

5.3 Statistical reporting

Repeated hardware runs should report confidence intervals and distribution plots. Latency distributions are often skewed, so paired non-parametric tests may be preferable when assumptions for parametric testing are not met; effect sizes should accompany significance tests. The number of runs, random seeds for controlled network impairment, and exclusion criteria should be specified before analysis. Any later study involving blind or low-vision participants must undergo the appropriate ethics review and should be analyzed separately from bench performance.

6 Discussion

6.1 Why freshness must sit above the detector

The controller is deliberately placed above the vision model because many latency failures originate outside the network. A faster backbone cannot repair a frame that waited in a wireless or decode queue, and a perfect detector cannot prevent a spoken instruction from becoming stale after inference. The scheduler therefore treats model selection as one actuator among several. This system-level view complements adaptive sampling and dynamic-network research [9, 12, 14] by adding temporal validity and safety priority to the control objective.

6.2 Safety trade-offs

Adaptive inference creates a real accuracy-latency trade-off. LIGHT processing may improve freshness while reducing recognition performance on difficult observations. The critical-risk override reduces this exposure but cannot eliminate it because risk itself is estimated from perception. A deployable system should therefore default conservatively when confidence is low, retain a hard stale-result horizon, log mode decisions for audit, and communicate uncertainty rather than presenting every instruction as equally certain.

6.3 Privacy and local processing

Assistive cameras can capture bystanders, documents, screens, and other private material. Edge processing can reduce transmission to remote infrastructure [5], but local processing is not automatically private. Cached frames, logs, model telemetry, and crash reports can still expose sensitive content. A deployed implementation should minimize image retention, encrypt diagnostic data, and separate operational telemetry from visual content.

6.4 Limitations

The central limitation is that the quantitative results are controller-level simulation outputs. The model omits hardware scheduling, thermal throttling, codec variation, burst packet loss, and accuracy changes caused by model or resolution switching. The normalized thresholds are transparent design choices rather than values derived from clinical data. The present study also contains no blind or low-vision participants and therefore makes no claim about mobility benefit or safety in real environments. These boundaries are intentional: the simulation is used to test a scheduling mechanism before stronger empirical claims are attempted.

7 Conclusion

This paper presents a risk-aware adaptive video-processing policy for latency-sensitive assistive vision. The method separates frame admission from inference-mode selection, uses explicit frame age to reject stale information, and preserves full-fidelity processing for critical-risk observations. In a deterministic 960-step controller simulation, the adaptive policy reduced modeled p95 latency by 16.4% under network congestion, 18.6% under compute overload, and 31.5% under mixed stress relative to fixed full inference. These numbers describe the model, not a deployed device. The next empirical step is to implement the controller on the WVAB prototype and test whether measured camera-to-feedback latency, stale-frame rate, critical-event recall, and energy use follow the predicted trade-offs. The broader contribution is a reproducible way to treat temporal freshness as part of video-processing quality rather than as an afterthought to detector speed.

Statements and Declarations

Funding

No external funding was used for the controller-level simulation and preparation of this manuscript draft.

Competing interests

The author declares no competing interests.

Ethics approval

Not applicable. The study reports a controller-level simulation and does not involve human participants, identifiable human data, or animal subjects.

Data availability

The controller-level simulation script, generated trace data, and summary table supporting the reported numerical results are provided as electronic supplementary material accompanying this manuscript draft.

Code availability

The reproducible Python simulation used to generate the controller-level results is provided as electronic supplementary material.

Author contributions

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

References

[1] 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

[2] Seiple W, van der Aa HPA, Garcia-Piña F, Greco I, Roberts C, van Nispen R (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

[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] 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

[5] 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

[6] 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

[7] 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

[8] 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

[9] 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

[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] 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

[12] 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

[13] 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

[14] Machidon AL, Pejović V (2024) Adaptive data-driven subsampling for efficient neural network inference. Signal Image Video Process 18:5163-5171. https://doi.org/10.1007/s11760-024-03223-z

[15] 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

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