EchoGuard: Energy-Aware Multi-Stage On-Device Emergency Sound Detection
Audio & Music

Nguyen Minh Khoa [1] **Hoang Tuan Anh** [2]

Abstract

We present EchoGuard, an energy-aware, always-on emergency sound detection system designed for commodity smartphones. The core idea is a strict three-stage cascade: (S1) a lightweight DSP gate that opens a compute window only when the audio exhibits event-like changes; (S2) a tiny binary danger/safe classifier that semantically filters benign events; and (S3) a conditional 50-class environmental sound classifier whose outputs are mapped to an emergency decision. We evaluate on ESC-50 (Piczak, 2015) and on synthetic 10-minute continuous soundscapes spanning six scenarios (quiet home, sleep night, busy indoor, kitchen-like, outdoor park, commute-like). Across scenarios, the cascade exposes clear Pareto frontiers between Emergency-F1 (EF1) and average compute (MACs/s): for example, in `sleep_night` we achieve EF1 = 0.667 with a ∼56–61× reduction in compute compared to always-on S3; in challenging mixtures such as `kitchen_like` and `busy_indoor`, S2 gating substantially improves EF1 over always-on S3 while keeping S3 duty cycle low. We release code and all training/evaluation pipelines at `https://github.com/khoa288/echoguard`.

1. Introduction

Smartphones are natural platforms for passive safety monitoring (e.g., detecting alarms, glass breaking, sirens, crying) because they are always present and have microphones and on-device compute. However, continuous audio inference is expensive: running a large classifier on every time step wastes energy on the overwhelmingly common benign audio. EchoGuard targets the practical constraint: *listen always, think rarely* (Gruenstein et al., 2017; Piedrahita Giraldo et al., 2019).

We formalize this as an energy-constrained detection problem on a continuous audio stream. The system must maintain high emergency recall (missed detections are costly) while minimizing average compute and reducing false alarms that harm user trust. To this end, we design a strict cascade with three distinct roles:

- **S1 (DSP gate):** extremely cheap signal processing features trigger a 5-second “compute-open” window.
- **S2 (binary danger/safe):** filters safe events and determines whether S3 is needed.
- **S3 (50-class ESC):** invoked conditionally; provides class posteriors and the final emergency decision.

Contributions. This final project delivers:

- A fully implemented end-to-end cascade with interchangeable S1/S2 variants and explicit compute accounting.
- A reproducible soundscape simulator generating continuous 10-minute scenes with controlled SNR, event rates, overlaps, and acoustic effects.
- A complete evaluation grid (125 experiments) spanning: S3 only, S1+S3, S2+S3, and S1+S2+S3 across six scenarios.
- A clear accuracy–compute analysis via EF1, AUROC, Trigger Rate (TR), Stage3 Duty Rate (DER), Time-to-Detect (TTD), and Energy Utility Index (EUI = EF1 / Cavg).

2. Related Work

Environmental sound classification (ESC). ESC addresses non-speech acoustic events (alarms, impacts, animals, weather) and is commonly benchmarked on ESC-50 (Piczak, 2015). Most ESC work evaluates short, isolated clips and optimizes clip-level accuracy, whereas always-on deployment must handle streaming inputs with overlapping sources, shifting backgrounds, and strict energy constraints.

[1] 23khoa.nm@vinuni.edu.vn [2] 23anh.ht@vinuni.edu.vn. Correspondence to: Nguyen Minh Khoa <23khoa.nm@vinuni.edu.vn>.

COMP3020 Machine Learning, CECS, VinUniversity.

ESC-NAS (Liu et al., 2024) searches raw-waveform architectures for resource-constrained edge devices, offering efficient candidates for on-device classification. However, prior ESC studies generally do not evaluate multi-stage cascades or report cascade-aware metrics under continuous mixtures, which are central to EchoGuard.

Efficient audio tagging backbones for mobile inference. A practical path to strong on-device audio models is to reuse efficient CNN families and transfer them to audio via large-scale pretraining and distillation. MobileNetV3 (Howard et al., 2019) is a widely adopted efficiency-oriented design built around depthwise-separable convolutions and NAS, and has favorable FLOPs/latency characteristics on mobile hardware. EfficientAT-style training further shows that compact CNN backbones can be highly competitive for audio tagging when distilled from larger teachers (Schmid et al., 2023b). We therefore evaluate MobileNet (MN) and Dynamic MobileNet (DyMN) variants (Schmid et al., 2023a) as Stage 3 backbones within our cascade.

Tiny audio models and keyword spotting (KWS). Always-on KWS has produced a suite of tiny models optimized for low-power inference. DS-CNN/Hello Edge (Zhang et al., 2017b) popularized depthwise-separable CNNs (often paired with quantization) for microcontroller-scale keyword spotting, while BC-ResNet (Kim et al., 2023) improves efficiency/accuracy through broadcasted residual learning. Although KWS targets speech triggers, these architectures are well-suited as lightweight detectors or binary filters. We adopt this line of work for Stage 2 (danger/safe gating), where the goal is fast semantic rejection of benign audio rather than fine-grained classification.

ESC robustness and data augmentation. ESC-50 is small (2,000 clips) (Piczak, 2015), making models sensitive to overfitting and domain shift. Modern audio pipelines often rely on augmentation to improve robustness. Mixup (Zhang et al., 2017a) regularizes by training on convex combinations of examples and labels, and SpecAugment (Park et al., 2019) perturbs time–frequency regions to reduce reliance on narrow spectral cues. We incorporate these augmentations when training our gating and classification models to improve robustness under background noise and mixtures.

Cascaded and conditional audio inference. Cascaded KWS systems show that cheap stages can gate expensive models to reduce average compute and power (Gruenstein et al., 2017; Piedrahita Giraldo et al., 2019; Yang et al., 2022). These works are primarily designed for speech wake words and are commonly two-stage (e.g., DSP+NN or small+large NN). EchoGuard instead targets non-speech emergency acoustics and composes a strict three-stage pipeline (DSP event gate → binary danger/safe → conditional ESC), enabling cascade-aware evaluation (e.g., duty rate and compute/accuracy Pareto frontiers) under continuous soundscape mixtures.

Dynamic neural networks. More broadly, dynamic networks allocate computation per input via conditional execution or early exits (Han et al., 2021). EchoGuard follows this principle at the system level, but uses heterogeneous modules with explicit roles (DSP + tiny NN + ESC model), which simplifies control and analysis for deployment.

Prior on-device emergency detection. SafeCastor (Nguyen, 2023) explores single-stage on-device emergency audio detection and motivates feasibility; EchoGuard advances this direction with a principled cascade, systematic baselines, and cascade-aware metrics under continuous mixtures.

3. Method

3.1. Problem setup

Let $x(t)$ be a continuous audio stream. At decision times t_i (every 0.5 s), the system processes the trailing 5-second window $w_i = x[t_i - 5, t_i]$ and outputs an emergency decision $\hat{y}_i \in \{0, 1\}$. We measure:

$$\text{EF1} = \frac{2 \cdot \text{Prec} \cdot \text{Rec}}{\text{Prec} + \text{Rec}}, \qquad \text{EUI} = \frac{\text{EF1}}{C_{\text{avg}}},$$

where C_{avg} is average compute in MACs/s.

3.2. Dataset labeling: danger vs. safe

We use ESC-50 (32 kHz, 5 s clips, 50 classes). For binary danger/safe labeling (S2 and the emergency mapping in S3), we treat the following 8 categories as **danger/emergency**:

$$\left\{\begin{array}{l}\texttt{siren, car\_horn, glass\_breaking,}\\ \texttt{thunderstorm, crying\_baby, dog,}\\ \texttt{door\_wood\_knock, clock\_alarm}\end{array}\right\}$$

All other classes are labeled safe.

3.3. Stage 1: DSP gate (S1)

S1 runs continuously at 16 kHz on 0.5 s frames with 0.05 s hop (FFT size 512). Per frame, we compute: RMS energy (dB), spectral flux, and 4-band energies (0–300, 300–1200, 1200–3000, 3000–Nyquist). A noise floor is tracked via EMA ($\alpha = 0.02$), and the gate requires 2 consecutive hits plus a 0.5 s cooldown.

We implement three S1 variants:

- **S1_A_energy:** energy rise over the EMA noise floor by $\Delta = 12$ dB.

- **S1_B_energy_flux:** energy trigger OR (flux AND band-rise).
- **S1_C_sleep_aware:** switches to a more permissive mode when long-term RMS < -45 dB (sleep), reducing thresholds (e.g., $\Delta : 12 \to 8$ dB; flux threshold $0.35 \to 0.25$).

When S1 triggers, it opens a **5-second compute window** during which S2 and/or S3 are allowed to run.

3.4. Stage 2: binary danger/safe filter (S2)

S2 takes a 5-second 32 kHz window and outputs $p_d = p(\text{danger} \mid w)$. We evaluate two families:

Deep S2 models. We train BC-ResNet (broadcasted residual learning) with width/temperature scaling $\tau \in \{1, 3, 8\}$, and a MobileNet binary model (`mn04_binary`). All deep S2 models use 40-bin log-mel spectrograms (30 ms window, 10 ms hop, FFT 1024) with CMVN and SpecAugment (Park et al., 2019) during training. Optimization: SGD (lr 0.1, momentum 0.9, wd 10^{-3}), 200 epochs, warmup 5 epochs, cosine decay, batch size 128, AMP enabled. Imbalance handling: BCEWithLogitsLoss with `pos_weight` computed from the training split.

Classical ML baselines. We test Logistic Regression and Linear SVM (with probability calibration), using 80-D pooled features (mean+std over time of 40-bin log-mels). These baselines are included for completeness as extremely lightweight options.

3.5. Stage 3: 50-class classifier (S3)

S3 is an EfficientAT DyMN model fine-tuned for 50 ESC-50 classes following the EfficientAT repository instructions. It produces $p(k \mid w)$ over 50 classes. We define emergency probability:

$$p_e = \sum_{k \in \mathcal{E}} p(k \mid w),$$

where $\mathcal{E}$ is the same 8-class emergency set, and predict emergency if $p_e \geq 0.5$.

3.6. Escalation policy: when S3 runs

Given p_d from S2, S3 is invoked if:

$$p_d \geq 0.30 \quad \text{OR} \quad 0.25 \leq p_d \leq 0.65.$$

The uncertainty band explicitly escalates ambiguous segments to preserve recall.

3.7. Compute accounting (MACs/s)

We account for compute using:

$$C_{\text{avg}} = \frac{n_1 \cdot \text{MAC}_1 + n_2 \cdot \text{MAC}_2 + n_3 \cdot \text{MAC}_3}{T}, \quad \text{EUI} = \frac{\text{EF1}}{C_{\text{avg}}}.$$

Neural MACs are measured (as in the project notebooks) and S1 is approximated by an FFT proxy:

$$\text{MAC}_{\text{S1,hop}} = 8\, n_{\text{fft}} \log_2(n_{\text{fft}}) = 36{,}864 \text{ MACs/hop.}$$

Measured per-call MACs used in the final evaluation:

- S2 MobileNet binary: 65,430,656 MACs/call
- S2 BC-ResNet: 18,022,036 MACs/call
- S3 DyMN: 58,037,024 MACs/call

4. Experiments

4.1. Synthetic soundscape simulator (continuous 10-minute scenes)

To evaluate always-on behavior, we generate 10-minute soundscapes (600 s) with continuous background plus scheduled foreground events using only ESC-50 clips:

- **Ambient pool:** e.g., rain, sea waves, wind, engines, etc. tiled to fill the background.
- **Safe event pool:** non-emergency foreground events.
- **Emergency pool:** the 8 danger classes.

Event times are sampled by Poisson processes (scenario-specific event/min ranges), with overlap/burstiness via probabilistic start-time shifts. Events undergo stochastic acoustic effects (mild reverb, low-pass filtering for occlusion), and are scaled to a target SNR relative to local background.

We evaluate six scenarios with distinct background levels, event densities, and emergency SNR ranges: `quiet_home_day`, `sleep_night`, `busy_indoor`, `kitchen_like`, `outdoor_park`, `commute_like`.

4.2. Evaluation protocol and metrics

Decision grid. For each 10-minute soundscape, we evaluate at 0.5 s steps (1200 decisions per scenario). At each decision time t_i, the model consumes the trailing 5 s window $[t_i - 5, t_i]$ (causal simulation).

Ground truth. A decision is positive if any emergency event overlaps the trailing window $[t_i - 5, t_i]$.

Reported metrics. We report EF1, Precision, Recall, and AUROC. Cascade efficiency metrics include:

- **TR:** Stage1 trigger rate (triggers/minute).
- **DER:** Stage3 duty rate $= \frac{\text{Stage3\_calls}}{1200}$.
- **TTD_sec:** time-to-detect (seconds).
- **Cavg_MACs_per_s** and **EUI**.

4.3. Systems compared (full grid)

We evaluate 125 total experiments (exactly):

- **Soundscapes (120 rows):** 6 scenarios × {S3 (1) + S1+S3 (3) + S2+S3 (4) + S1+S2+S3 (12)}.
- **Single-clip (5 rows):** S3 only (1) and S2+S3 (4) on ESC-50 clips.

Stage1 variants: {S1_A_energy, S1_B_energy_flux, S1_C_sleep_aware}. Stage2 variants: {bcresnet_tau1, bcresnet_tau3, bcresnet_tau8, mn04_binary}.

5. Results and Discussion

5.1. Per-scenario compute and EF1 (headline cascade vs. baseline)

Figures 1 and 2 compare always-on S3 to a representative full cascade configuration (S1_C + bcresnet_τ8 + S3). Overall, the cascade substantially reduces compute in most scenarios, with scenario-dependent EF1 changes that reflect the trade-off between aggressive gating (efficiency) and recall preservation.

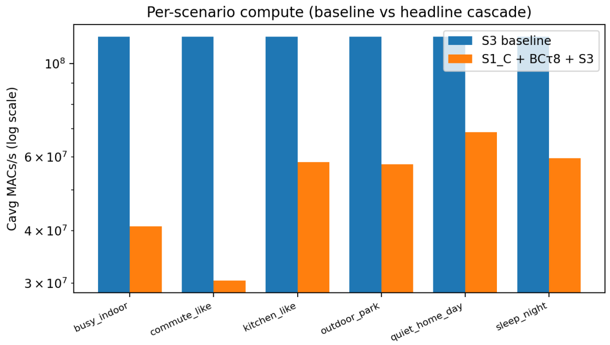

Figure 1. Per-scenario average compute (MACs/s, log scale): always-on S3 vs. headline cascade S1_C + bcresnet_τ8 + S3.

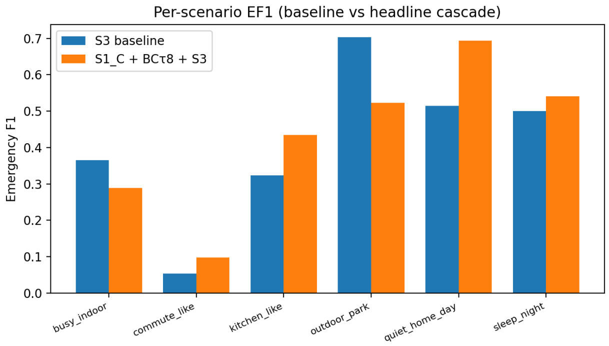

Figure 2. Per-scenario EF1: always-on S3 vs. headline cascade S1_C + bcresnet_τ8 + S3.

5.2. Stage3 duty cycle and energy savings

Figure 3 shows how frequently S3 is invoked under different gating strategies. The key energy mechanism is **reducing S3 duty**: S1 reduces the number of candidate windows, while S2 reduces S3 calls further by filtering safe events and escalating only dangerous/uncertain segments.

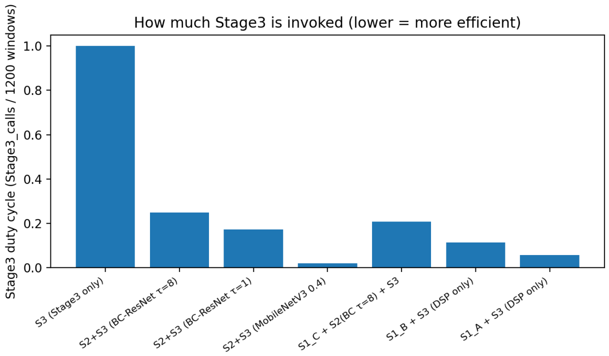

Figure 3. Stage3 duty cycle (fraction of 1200 decisions invoking S3) for different pipelines (lower is more efficient).

5.3. Accuracy–compute trade-off and Pareto structure

Figure 4 summarizes the accuracy–compute trade-off (macro EF1 over six soundscapes vs. compute, log scale). We observe distinct Pareto frontiers per scenario (non-dominated configurations), which explains why "best EF1" and "best EUI" often select different stage variants.

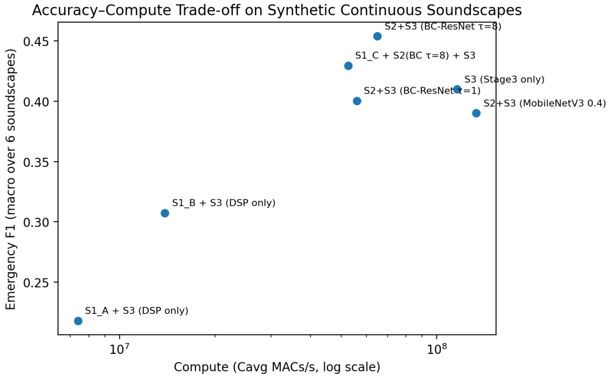

Figure 4. Accuracy–compute trade-off over six synthetic soundscapes (macro EF1 vs. compute, log scale).

5.4. Best-per-scenario summary (soundscapes)

Table 1 reports (i) always-on S3 EF1, (ii) the best EF1 configuration found in the full grid per scenario, and (iii) the corresponding average compute and S3 duty rate. These are direct summaries from the complete 125-row results table.

Scenario	**Baseline**	**Best-EF1 Config**	**EF1**	**Cavg (MAC/s)**	**DER**
busy_indoor	S3	S2+S3 (mn04_binary)	0.5287	1.350206e+08	0.0358
commute_like	S3	S1_C + S2(mn04_binary) + S3	0.1667	5.728125e+07	0.0183
kitchen_like	S3	S1_C + S2(mn04_binary) + S3	0.6452	1.078810e+08	0.0183
outdoor_park	S3	S1_C + S3	0.7081	1.105234e+08	0.9458
quiet_home_day	S3	S1_C + S2(bcresnet_τ8) + S3	0.6939	6.871440e+07	0.2767
sleep_night	S3	S1_A + S2(bcresnet_τ1) + S3	0.6667	2.064451e+06	0.0083

Table 1. Best EF1 per scenario on soundscapes (from the full experimental grid). Baseline S3 EF1 values: quiet_home_day 0.5147, sleep_night 0.5000, busy_indoor 0.3650, kitchen_like 0.3235, outdoor_park 0.7030, commute_like 0.0533.

Key takeaways.

- **S2 gating is a strong semantic filter.** In difficult mixtures (busy_indoor, kitchen_like, commute_like), the best-EF1 configurations include S2 and reduce S3 duty to ≈1.8–3.6% while improving EF1 over always-on S3.
- **S1 can dominate in ultra-quiet regimes.** In sleep_night, S1 triggers are rare (TR ≈0.2/min in the best setting), yielding ∼56× lower compute than always-on S3 while also improving EF1.
- **Scenario dependence is unavoidable.** Outdoor park already matches S3 well; the best-EF1 config is S1_C+S3 with near-full S3 duty, indicating that in some environments the gate chooses safety (high recall) over savings.

5.5. Single-clip results (sanity check)

On isolated ESC-50 clips, S3 achieves very high emergency EF1 (0.9859) with AUROC 0.9998. S2-gated variants remain strong but exhibit different EUI trade-offs due to reduced S3 calls (and added S2 compute). We report these clip results primarily as a controlled baseline; soundscapes remain the main deployment-oriented benchmark.

5.6. Rationale for model choices (what we tested and why)

Stage 3 (MN / DyMN from EfficientAT). We selected MN and DyMN as edge-oriented audio backbones because the EfficientAT line targets state-of-the-art accuracy/compute trade-offs for audio tagging via large-scale pre-training and distillation; DyMN extends MN with dynamic components for improved performance at comparable efficiency. For this project, DyMN served as the high-capacity conditional expert (S3).

Stage 2 (BC-ResNet, DS-CNN/MobileNet, LR/SVM). We evaluated: (i) **BC-ResNet** because KWS and binary danger detection both rely on compact time-frequency cues and onset-like patterns; empirically, BC-ResNet transferred well and offered strong compute efficiency for S2; (ii) **DS-CNN/MobileNet** as standard efficient CNN baselines for mobile inference; (iii) **LR/SVM** as extremely lightweight baselines to test whether simple pooled log-mel features suffice (they did not, compared to CNN filters).

6. Conclusion

EchoGuard demonstrates that always-on emergency sound detection can be made practical by *explicitly structuring computation* as a cascade. Across six continuous soundscape scenarios, we find that conditional inference yields strong accuracy–efficiency Pareto frontiers: S2 gating often improves EF1 in challenging mixtures while keeping S3 duty below 4%, and S1 gating can reduce compute by more than an order of magnitude in quiet regimes. The main implication is that mobile deployment should optimize *system-level policies* (when to escalate) in addition to model compression.

Future directions.

- **Policy learning for gating:** replace fixed thresholds with learned (but still cheap) policies that adapt to scenario statistics while preserving recall.
- **Domain realism:** validate the simulator against real background recordings (device microphones, room impulse responses) and quantify sim-to-real gaps.
- **Personalization and calibration:** per-user calibration

for acceptable false-alarm rates and context (sleep vs. commute) with privacy-preserving on-device adaptation.

Acknowledgements

This work was completed as the final project for COMP3020 at VinUniversity. We thank Dr. Khoa Doan and the course staff for guidance and feedback.

Team Contributions.

Nguyen Minh Khoa: Designed the cascade architecture and Stage-1 DSP gate variants; implemented Stage-2 training (BC-ResNet/DS-CNN and ML baselines) and compute accounting; contributed to evaluation design and report writing.
Hoang Tuan Anh: Implemented Stage-3 EfficientAT fine-tuning and the full cascade evaluation pipeline; built the soundscape simulator and implemented streaming metrics (EF1/AUROC/TR/DER/TTD); contributed to analysis and report writing.
Joint work: Integration, debugging, experimental sweeps, and final analysis/figures.
SafeCastor Note: SafeCastor (Nguyen, 2023) is acknowledged as prior motivational work by Nguyen Minh Khoa. All EchoGuard models, cascades, datasets, and experiments described in this proposal are developed specifically for this course project.

Impact Statement

EchoGuard aims to improve personal safety and accessibility by enabling always-on acoustic awareness on-device, potentially benefiting people living alone, older adults, and users with hearing impairments. Because the pipeline is designed for on-device inference, it can reduce privacy risks associated with streaming raw audio to cloud services. However, the system's societal impact depends on responsible deployment: false negatives may delay emergency response, while false positives can cause distress, alarm fatigue, or inappropriate escalation. There is also risk of misuse if such technology is repurposed for surveillance; deployments should include clear user consent, transparency, local processing by default, and safeguards against covert recording. Finally, performance may vary across acoustic environments and devices; careful calibration and real-world evaluation are necessary before high-stakes use.

References

Gruenstein, A., Alvarez, R., Thornton, C., and Ghodrat, M. A cascade architecture for keyword spotting on mobile devices. *arXiv preprint arXiv:1712.03603*, 2017.

Han, Y., Huang, G., Song, S., Yang, L., Wang, H., and Wang, Y. Dynamic neural networks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021. doi: 10.1109/TPAMI.2021.3054824.

Howard, A., Sandler, M., Chu, G., Chen, L.-C., Chen, B., Tan, M., Wang, W., Zhu, Y., Pang, R., Vasudevan, V., et al. Searching for mobilenetv3. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 1314–1324, 2019.

Kim, B., Chang, S., Lee, J., and Sung, D. Broadcasted residual learning for efficient keyword spotting, 2023. URL https://arxiv.org/abs/2106.04140.

Liu, H., Yang, B., Zhang, Y., Liu, G., and Yan, H. ESC-NAS: Environment sound classification using neural architecture search on raw audio waveform for resource-constrained edge devices. *Sensors*, 24(12):3749, 2024. ISSN 1424-8220. doi: 10.3390/s24123749.

Nguyen, M. K. Safecastor: Always-on privacy-preserving emergency audio detection on mobile devices. https://github.com/khoa288/safe-castor, 2023. Accessed: 2025-10-11.

Park, D. S., Chan, W., Zhang, Y., Chiu, C.-C., Zoph, B., Cubuk, E. D., and Le, Q. V. Specaugment: A simple data augmentation method for automatic speech recognition. *arXiv preprint arXiv:1904.08779*, 2019.

Piczak, K. J. ESC-50: Dataset for environmental sound classification. https://github.com/karoldvl/ESC-50, 2015. Accessed: 2025-10-11.

Piedrahita Giraldo, J. S., O'Connor, C., and Verhelst, M. Efficient keyword spotting through hardware-aware conditional execution of deep neural networks. In *Proceedings of the IEEE/ACS 16th International Conference on Computer Systems and Applications (AICCSA)*, 2019. doi: 10.1109/AICCSA47632.2019.9035289.

Schmid, F., Koutini, K., and Widmer, G. Dynamic convolutional neural networks as efficient pre-trained audio models, 2023a. URL https://arxiv.org/abs/2310.15648.

Schmid, F., Koutini, K., and Widmer, G. Efficient large-scale audio tagging via transformer-to-cnn knowledge distillation, 2023b. URL https://arxiv.org/abs/2211.04772.

Yang, Z., Sun, S., Li, J., Zhang, X., Wang, X., Ma, L., and Xie, L. Catt-kws: A multi-stage customized keyword spotting framework based on cascaded transducer-transformer. In *Proceedings of Interspeech*, 2022.

Zhang, H., Cisse, M., Dauphin, Y. N., and Lopez-Paz, D. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017a.

Zhang, Y., Suda, N., Lai, L., and Chandra, V. Hello edge: Keyword spotting on microcontrollers. *arXiv preprint arXiv:1711.07128*, 2017b.