From-Scratch Audio Classification with Multi-Branch Feature Fusion: Comparing CNN, Transformer, and Mamba

Anderson H. Giacomini[1], **Andre M. S. Magalhães**[1], **Elaine P. M. Sousa**[1]

[1] Institute of Mathematics and Computer Sciences – University of São Paulo (USP)
São Carlos – SP – Brazil

{giacomini,andre.moreira.souza}@usp.br, parros@icmc.usp.br

Abstract. *From-scratch audio classification on weakly labeled data is essential for resource-constrained and domain-specific settings where large pretrained models are unavailable or prohibitively costly. We adopt multi-branch feature fusion as the core approach and compare CNN, Inception, Transformer, and Mamba architectures on the AudioSet balanced training split. Multi-branch fusion (per-feature encoders, merged before the head) improves all four architectures by 17–33% mAP. Five-fold CV shows Inception significantly outperforms the Transformer; Transformer and Mamba reach comparable multi-branch performance ($p = 0.19$), with preliminary results suggesting feature fusion may be at least as important as sequence model choice in this regime.*

1. Introduction

Weakly labeled audio classification (also called audio tagging), where only clip-level labels are available without precise temporal annotations and no temporal event boundaries are predicted, is a central challenge in audio machine learning [Zhang et al. 2025]. Large-scale AudioSet [Gemmeke et al. 2017] has become the standard benchmark, but top-performing methods rely on large-scale pretraining on the full AudioSet or other large corpora [Kong et al. 2020, Gong et al. 2021]. This paper asks what most drives from-scratch weakly supervised audio classification, i.e., the acoustic feature representation or the sequence model. We test the hypothesis that fusing complementary acoustic features through a multi-branch architecture consistently improves classification across architectural families, and that this feature-level effect is at least as influential as the sequence model; we also examine how frame length affects each family and which performs best from scratch. We use the AudioSet balanced training split [Gemmeke et al. 2017] (16,306 clips, 527 classes), a curated subset of the full 2M-clip set with approximately uniform class representation, enabling a controlled architectural comparison under consistent data conditions; note that results on this split are not directly comparable to published AudioSet benchmarks, which use a fixed $\sim$18k evaluation set.

Beyond its role as a benchmark task, weakly supervised audio classification underpins scalable multimedia data management: large-scale audio repositories, e.g., surveillance archives, biodiversity monitoring, and industrial streams, require automated annotation to enable indexing, retrieval, and domain-specific analytics

Acknowledgments: This study was financed in part by CAPES (Brazilian Coordination for Improvement of Higher Level Personnel). *Use of generative AI:* ChatGPT, Claude, and Grammarly were used to support experiment-code development and the writing and revision of this manuscript; all content was reviewed and validated by the authors, who take full responsibility for it.

[Moreira Souza et al. 2025]. Classifiers that operate without expensive pretraining are therefore a prerequisite for practical data pipelines in resource-constrained or domain-specific scenarios, motivating the from-scratch focus of this study.

Prior work shows that log-mel spectrograms, MFCC, chroma, ZCR, and RMS energy capture complementary aspects of audio, and no single representation covers all discriminative dimensions of sound events [Mohmmad and Sanampudi 2024, Mu et al. 2024, Turab et al. 2022]. Three architectural families have been applied to audio: convolutional (CNN and Inception-style [Szegedy et al. 2015]), self-attention (Transformer [Vaswani et al. 2017]), and selective state-space (Mamba [Gu and Dao 2023]), yet, to our knowledge, no controlled from-scratch comparison exists for weakly labeled data on the AudioSet balanced training set.

Our contributions are threefold: (i) multi-branch fusion improves all four architectures by 17–33% mAP, with the choice among feature subsets having marginal impact; (ii) five-fold CV shows Inception significantly outperforms Transformer ($p = 0.002$), indicating that convolutional inductive biases provide a consistent advantage over sequence models when training from scratch; and (iii) Transformer and Mamba reach comparable multi-branch performance ($p = 0.19$), suggesting that feature fusion may be at least as important as sequence model choice on the AudioSet balanced training set.

2. Related Work

Baselines are representative, widely cited works spanning the relevant architectural families and pretraining regimes; a formal systematic review is beyond the scope of a short paper, though our feature choices draw on a recent systematic review of the field [Mohmmad and Sanampudi 2024].

AudioSet benchmarks. We report two representative baselines on the full AudioSet: [Kong et al. 2020] introduced PANNs, achieving mAP 0.431 with CNN14 trained from scratch; [Gong et al. 2021] introduced AST, reaching mAP 0.485 with ImageNet+AudioSet pretraining. These settings are not directly comparable to our balanced-only, from-scratch regime.

Transformer vs. CNN. [Schmid et al. 2023] show that Transformer-to-CNN knowledge distillation recovers most of the Transformer's quality at lower cost on the full AudioSet, consistent with our finding that CNNs outperform Transformers under limited data.

Mamba for audio. [Gu and Dao 2023] introduced Mamba, a selective state-space model that matches Transformer performance while avoiding quadratic attention; [Yadav and Tan 2024] applies it to audio with self-supervised pretraining on AudioSet. To our knowledge, no prior work compares Mamba against Transformers and CNNs in a from-scratch, weakly supervised setting.

Multi-branch fusion. MFF-EINV2 [Mu et al. 2024] achieves SOTA performance by fusing spectral/spatial/temporal scales of a multichannel spectrogram using EINV2 for sound event localization and detection on DCASE challenge data. In contrast, our approach fuses complementary acoustic feature types, includes Inception and Mamba in the comparison, and trains entirely from scratch on AudioSet balanced training set.

3. Experimental Setup

We evaluate four architectural families across four frame lengths, seven acoustic features, and single- vs. multi-branch configurations. The following subsections describe the dataset and features, the model architectures and training protocol, and the evaluation methodology.

3.1. Dataset and Features

We use the **AudioSet balanced** training split: 16,306 clips, each 10 s at 44,100 Hz, with 527 weakly labeled classes (multi-label, binary).

Seven acoustic features are extracted per clip, selected for their established and complementary roles in weakly labeled audio classification [Mohmmad and Sanampudi 2024]: *log-mel spectrogram* (40 bands, frequency content over time), *MFCC* (13 coefficients, spectral characteristics), *chroma* (12 bins, pitch-class content), *ZCR* (signal noisiness), *RMS energy* (amplitude dynamics), *statistical features* (mean/std/max/min/25th and 75th-percentile of audio frame amplitudes), and *spectral centroid* (brightness). Each feature is computed at four frame lengths (analysis window sizes, no overlap): w128, w256, w512, and w1024 (128–1024 ms; 5,644–45,158 samples per frame at 44,100 Hz), yielding $T \approx 80$ down to 10 frames per 10 s clip—longer frames give finer spectral but coarser-resolution time steps. Feature extraction uses non-overlapping frames (hop = frame length) to limit the computational and memory footprint of the per-feature multi-branch pipeline; standard overlap strategies (e.g., 50% hop) would increase the effective training-frame density and may improve performance, and are left for future work. Spectral centroid spans several orders of magnitude across clips, so it is per-clip standardized (zero mean, unit variance) before model input to prevent scale dominance over other features.

3.2. Architectures

All models are trained from scratch with binary cross-entropy loss, Adam optimizer (learning rate 10^{-3}, batch size 32, up to 200 epochs), and early stopping (patience = 5, monitor = validation loss)[1]. Transformer and Mamba additionally use linear warmup ($0 \rightarrow 10^{-3}$ over 400 steps) to prevent gradient instability during the early phase of training from random initialization; convolutional models converge reliably without it.

CNN. Lightweight baseline: spectral features ($F \geq 2$, e.g. log-mel, MFCC) use two Conv2D blocks (16 and 32 filters, 3×3, ReLU, 2×2 max-pooling); scalar features ($F = 1$, e.g. ZCR, RMS) use two Conv1D blocks (32 and 64 filters, kernel 3, max-pooling 2). GlobalAveragePooling collapses each branch before the shared head.

Inception-style CNN [Szegedy et al. 2015]. Two naive Inception modules stacked with max-pooling between them, each with four parallel branches (Conv 1×1, Conv 3×3, Conv 5×5, and MaxPool 3×3 followed by Conv 1×1) concatenated channel-wise. The naive variant (without the dimensionality-reduction 1×1 bottlenecks) limits parameter count on 16,306 clips; GlobalAveragePooling aggregates the frame sequence.

Transformer [Vaswani et al. 2017]. Two-layer encoder with Multi-Head Self-Attention ($d_{model} = 64$, 4 heads, $d_{ff} = 128$), post-layer normalization, and residual

[1]Implemented in TensorFlow 2.15 with Keras.

connections. Fixed sinusoidal positional encoding [Vaswani et al. 2017] is used rather than learned embeddings because the from-scratch setting provides insufficient data to learn reliable position representations.

Mamba [Gu and Dao 2023]. Two MambaBlock layers with input-dependent parameters B, C, Δ, ZOH discretisation of A and B, causal depthwise Conv1D, and SiLU gating; pre-layer normalization with residual connections follows [Gu and Dao 2023]. Our TensorFlow implementation uses `tf.scan` in place of the CUDA-parallel scan; one additional SiLU activation is applied to the x branch before the causal convolution, a minor deviation from [Gu and Dao 2023] that was not ablated.

Multi-branch fusion. One encoder per feature type; branch outputs (after GlobalAveragePooling) are concatenated and fed through a shared Dense(128) head with Dropout(0.3). Single-branch experiments use the same builder with a single input feature, keeping the head identical so that gains reflect feature complementarity rather than capacity.

3.3. Evaluation

Primary metric: **mAP** (mean Average Precision), the mean across classes of the area under each class's precision–recall curve, which is the standard for AudioSet comparisons [Gong et al. 2021, Schmid et al. 2023]. Single-run results use the first fold of a 5-fold multilabel stratified split (80% train, 20% test); 10% of the training portion is held out for early-stopping validation. Cross-validation uses all five folds of the same split; statistical tests use paired t-tests, p is the two-tailed p-value. Class imbalance is handled only by the multilabel-stratified split (no resampling or reweighting). All experiments use a fixed seed (RANDOM_SEED = 42) for reproducibility[2].

Our 5-fold CV partitions only the balanced training split, so absolute mAP is not directly comparable to AudioSet's standard $\sim$18k evaluation set (evaluation on it is left for future work). Results are reported for the best configuration per architecture selected across frame lengths and feature subsets; this selection may introduce a degree of optimism bias.

4. Results

We organize the results around three questions: whether multi-branch fusion consistently helps, which architectural family performs best from scratch, and whether the choice of sequence model matters at this data scale.

Table 1 reports the best mAP for each architecture with one feature (1-feat.) and with multiple features (multi-feat.); both use the same multi-branch builder. Parameter counts (#P$_1$/#P$_M$) are reported for the exact configurations associated with the best single- and multi-feature results. CV results are reported as mean $\pm$ std where available.

Multi-branch Inception achieves the best result, with a single-run peak of mAP 0.141 (`logmel+MFCC+chroma_w128`).

4.1. Finding 1: Multi-Branch Fusion Improves All Architectures

Multi-branch feature fusion yields consistent gains across all architecture families: $+27\%$ for CNN (0.097 $\rightarrow$ 0.123), $+17\%$ for Inception (0.115 $\rightarrow$ 0.135), $+28\%$ for Transformer

[2]Code is publicly available at: `https://github.com/ander-hg/multibranch-sed`.

Table 1. Best mAP per architecture on AudioSet balanced. Single-feature uses log-mel; w_1 = best single-feature window; Features and w_M = best multi-feature configuration. † single-run; ‡ 5-fold CV mean ± std; [a] log-mel+MFCC+chroma.

Arch.	#P$_1$/#P$_M$	1-feat.	w_1	Multi-feat.	Features	w_M
CNN	77k/95k	$0.097^{\dagger}$	256	$0.123 \pm 0.006^{\ddagger}$	[a]	128
Inception	159k/482k	$0.115^{\dagger}$	128	$\mathbf{0.135 \pm 0.004}^{\ddagger}$	all-7	128
Transformer	146k/599k	$0.085^{\dagger}$	256	$0.109 \pm 0.005^{\ddagger}$	all-7	128
Mamba	137k/534k	$0.086^{\dagger}$	128	$0.114 \pm 0.003^{\ddagger}$	all-7	256

(0.085 → 0.109), and +33% for Mamba (0.086 → 0.114).

A feature sweep (CNN, Inception) shows three features (log-mel+MFCC+chroma) edge all-7 in single runs, but 5-fold CV ties them for CNN (0.123 ± 0.006 vs. 0.122 ± 0.005) and reverses for Inception (all-7: 0.135 ± 0.004 vs. 0.133 ± 0.003). Any multi-feature set substantially outperforms a single feature; the specific subset has a marginal, architecture-dependent impact.

4.2. Finding 2: Convolutional Architectures Outperform Sequence Models

Figure 1 (left) compares MB-Inception and MB-Transformer at both frame lengths using 5-fold CV. At w128, Inception (0.135 ± 0.004) significantly outperforms the Transformer (0.109 ± 0.005; paired t-test $t = 7.54$, $p = 0.002$). All five folds show Inception > Transformer (differences 0.016–0.037), confirming the effect is not an artifact of fold assignment. The same ordering holds at w256 ($p = 0.005$).

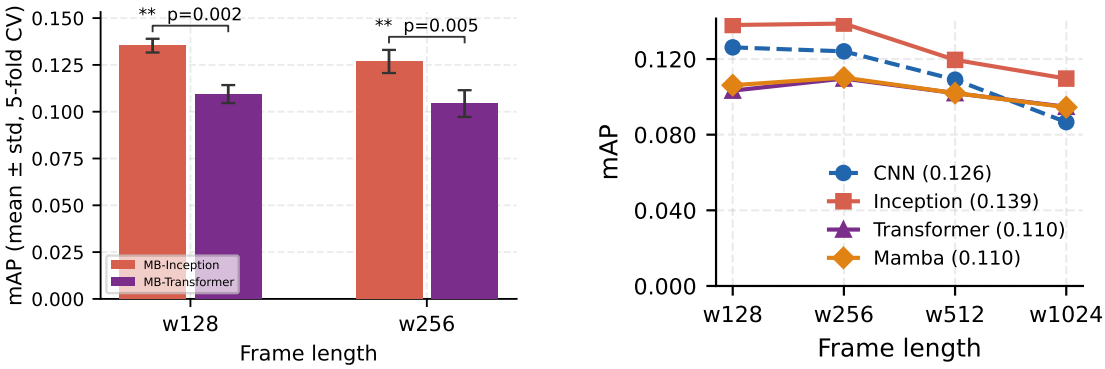

Figure 1. Left: 5-fold CV mAP (mean ± std) for MB-Inception and MB-Transformer at w128 and w256; Inception significantly outperforms the Transformer at both frame lengths ($^{**}p < 0.01$). Right: mAP vs. frame length per architecture (all 7 features); Inception peaks at w128, CNN at w256, and sequence models show no significant preference.

We interpret this gap via the task structure: AudioSet clips are 10 s, but the discriminative signal is typically localized to short events. Local convolutional receptive fields capture these directly, whereas global self-attention over $T \approx 10$–80 frames must learn to ignore much of the sequence without pretraining to guide it.

Frame-length analysis (Figure 1, right; all-7 features) reinforces this: Inception peaks at w128 (CV $p = 0.041$) and degrades monotonically at coarser frames, while CNN peaks at w256; both sequence models show no significant frame-length preference (Transformer CV $p = 0.207$).

4.3. Finding 3: Sequence Model Architecture Shows No Clear Advantage at This Scale

At the multi-branch level, Mamba (0.114 ± 0.003) and Transformer (0.109 ± 0.005) yield similar results ($p = 0.19$), a difference that is not statistically significant. This suggests that, at this data scale, feature fusion may be at least as important as sequence model choice.

5. Discussion

Within our from-scratch, balanced-split setting, the consistent superiority of convolutional architectures aligns with [Schmid et al. 2023]: with only 16,306 training clips and no pretraining, the Transformer may lack the data volume to learn effective attention patterns. This does not imply Transformers are inferior for audio in general—AST [Gong et al. 2021] and BEATs [Chen et al. 2022] reach state-of-the-art with pretraining—but convolutional inductive biases provide a stronger prior under data-limited conditions. Whether the gap to Inception would narrow with Mamba-specific optimizations (e.g. CUDA-parallel scan) or more data remains an open question [Gu and Dao 2023].

6. Limitations and Future Work

The low absolute mAP (0.110–0.141) relative to CNN14's 0.431 [Kong et al. 2020] reflects three compounding factors: (i) a $\sim$100$\times$ smaller training set (16,306 vs. $\sim$2M clips); (ii) non-overlapping frame extraction, reducing effective frame density; and (iii) lightweight model capacity chosen to limit overfitting. Parameter counts (Table 1) are our only efficiency proxy; training time, memory, and inference latency are unmeasured, so the resource-efficiency motivation is a design rationale, not a demonstrated result.

Future directions include: evaluation on the standard AudioSet test set; pretrained baselines (PANNs, VGGish); overlapping frame strategies; profiling of training time, memory, and inference latency to substantiate the efficiency motivation; interpretability analyses (e.g. attention and error breakdowns) to test the convolutional-bias explanation; and scaling to the full $\sim$2M-clip split.

7. Conclusion

We compared CNN, Inception, Transformer, and Mamba for from-scratch weakly labeled audio classification on AudioSet balanced. Multi-branch feature fusion consistently improves all architectures (17–33% mAP), robust to the specific feature subset chosen. Five-fold cross-validation confirms that Inception significantly outperforms the Transformer ($p = 0.002$), convolutional inductive biases providing a decisive advantage at this scale. Transformer and Mamba show no significant gap ($p = 0.19$); this preliminary comparison suggests feature fusion may be at least as important as sequence-model choice here.

References

Chen, S., Wu, Y., Wang, C., Liu, S., Tompkins, D., Chen, Z., and Wei, F. (2022). Beats: Audio pre-training with acoustic tokenizers. *arXiv preprint arXiv:2212.09058*.

Gemmeke, J. F., Ellis, D. P. W., Freedman, D., Jansen, A., Lawrence, W., Moore, R. C., Plakal, M., and Ritter, M. (2017). AudioSet: An ontology and human-labeled dataset for audio events. In *Proc. IEEE ICASSP*, pages 776–780.

Gong, Y., Chung, Y.-A., and Glass, J. (2021). AST: Audio spectrogram transformer. In *Proc. Interspeech*, pages 571–575.

Gu, A. and Dao, T. (2023). Mamba: Linear-time sequence modeling with selective state spaces. *arXiv:2312.00752*.

Kong, Q., Cao, Y., Iqbal, T., Wang, Y., Wang, W., and Plumbley, M. D. (2020). PANNs: Large-scale pretrained audio neural networks for audio pattern recognition. *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, 28:2880–2894.

Mohmmad, S. and Sanampudi, S. K. (2024). Exploring current research trends in sound event detection: a systematic literature review. *Multimedia Tools and Applications*, 83:84699–84741.

Moreira Souza, A., Moreira, G. A., and Pulcinelli, L. E. G. (2025). A Comparative Analysis of Denoising Methods for Deep Learning-Based Audio Event Detection in Noisy Agricultural Environments. In *Anais do XL Simpósio Brasileiro de Banco de Dados (SBBD 2025)*, pages 942–948, Brasil. Sociedade Brasileira de Computação - SBC.

Mu, D., Zhang, Z., and Yue, H. (2024). MFF-EINV2: Multi-scale feature fusion across spectral-spatial-temporal domains for sound event localization and detection. In *Proc. Interspeech*, pages 92–96.

Schmid, F., Koutini, K., and Widmer, G. (2023). Efficient large-scale audio tagging via transformer-to-CNN knowledge distillation. In *Proc. IEEE ICASSP*, pages 1–5.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2015). Going deeper with convolutions. In *Proc. IEEE CVPR*, pages 1–9.

Turab, M., Kumar, T., Bendechache, M., and Saber, T. (2022). Investigating multi-feature selection and ensembling for audio classification. *arXiv:2206.07511*.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, pages 5998–6008.

Yadav, S. and Tan, Z.-H. (2024). Audio mamba: Selective state spaces for self-supervised audio representations. In *Proc. Interspeech*, pages 552–556.

Zhang, Y., Huang, D., and Togneri, R. (2025). Pseudo strong labels from frame-level predictions for weakly supervised sound event detection. *arXiv:2501.03740*.