ASCDOMAIN: DOMAIN INVARIANT DEVICE-SELF-CHALLENGING ISOTROPIC CONVOLUTIONAL NEURAL ARCHITECTURE

Technical Report

Hubert Truchan[1]*, Tien Hung Ngo*[1]*, Zahra Ahmadi*[1,2]*,*

[1] Leibniz University Hannover, L3S Research Center, Hannover, Germany
[2] Peter L. Reichertz Medical Informatics Institute, Hannover Medical School, Hannover, Germany
{truchan, tien.ngo}@l3s.de, Ahmadi.Zahra@mh-hannover.de

ABSTRACT

The ongoing advancement of deep learning approaches for acoustic scene classification (ASC) has enabled robust performance in complex auditory environments. However, practical applications demand models that are computationally efficient for real-time decision making while remaining adaptive to diverse conditions. To address these constraints, we propose three innovative architectures: 1) a scalable isotropic convolutional network, 2) a recursive columnar architecture, and 3) a self-challenging representation method. These architectures are designed for low computational complexity and data-efficient ASC, leveraging advanced techniques such as device impulse response enhancement and two-dimensional signal embedding to enhance robustness against device mismatch. We validate our approaches in the DCASE 2024 Task 1 challenge using the TAU Urban Acoustic Scenes 2022 Mobile dataset, achieving state-of-the-art performance and significantly improving domain generalization capabilities. Our architectures offer computationally lean, yet highly effective solutions for real-world ASC applications across diverse auditory domains and recording devices.

***Index Terms*—** Isotropic Architecture, Domain Adaptation, Representation Learning

1. INTRODUCTION

Acoustic Scene Classification (ASC) plays a crucial role in our interaction with environments, enabling a wide range of applications, from urban planning and surveillance to multimedia retrieval and assistive technologies. The field of machine learning often draws inspiration from human cognitive abilities, particularly in synthesizing experience and knowledge to predict complex patterns. ASC represents a domain where such capabilities are crucial for interpreting diverse and dynamically changing environments.

The emphasis in the DCASE 2024 Challenge on low-complexity, data-efficient ASC underscores the need for innovative models that maintain high performance despite constraints like limited computational resources and minimal training data. These constraints are ubiquitous in real-world applications that require the deployment of lightweight models on edge devices. Such models must adapt rapidly to varied acoustic scenes, reflecting broader challenges in machine learning, where models must generalize across unseen domains [1].

Traditional approaches to ASC often involve extensive data manipulation strategies such as domain randomization [2], adversarial data enhancement [3], data generation [4], data preprocessing [5], domain-invariant representation learning [6], feature disentanglement [7], and various learning strategies including ensemble learning [8], meta-learning [9], and self-supervised learning [10]. Despite these efforts, achieving robust generalization remains challenging, as highlighted by recent studies advocating normalization perturbation to improve domain generalization [11].

To address these challenges, we introduce three distinct architectures: 1) an isotropic convolutional network that adjusts dynamically to computational resources, 2) a recursive columnar architecture that enhances information flow across the network, and 3) a representation self-challenging (RSC) method that iteratively deemphasizes dominant features to activate less prominent but relevant features. These models are enhanced by techniques such as adaptive instance normalization, device impulse response augmentation, and two-dimensional signal embedding, ensuring robustness across various acoustic environments.

Key Contributions:

- Introduction of an innovative isotropic architecture for low-complexity acoustic scene classification, enhancing adaptability and computational efficiency.
- Presentation of the recursiv architecture, providing an alternative to traditional bottleneck designs by employing a two-dimensional grid of isotropic convolutional blocks for effective information preservation and distillation.
- Application of the RSC method, enhancing model learning by focusing on less dominant but relevant features for classification.

These contributions, validated through rigorous testing in the DCASE 2024 Task 1 challenge, set new benchmarks in the ASC field, demonstrating our commitment to advancing technology that meets the demands of real-world applications in diverse acoustic environments.

2. DATA PREPROCESSING AND AUGMENTATION

2.1. Audio Processing

All audio files are initially downsampled from a sampling rate of 44.1 kHz to 32 kHz. Feature extraction involves computing MEL-spectrograms with 256 frequency bins, using a window length of 3072 samples, a hop length of 500 samples, and 4096 FFT points. The audio processing steps adhere to the protocols established in the

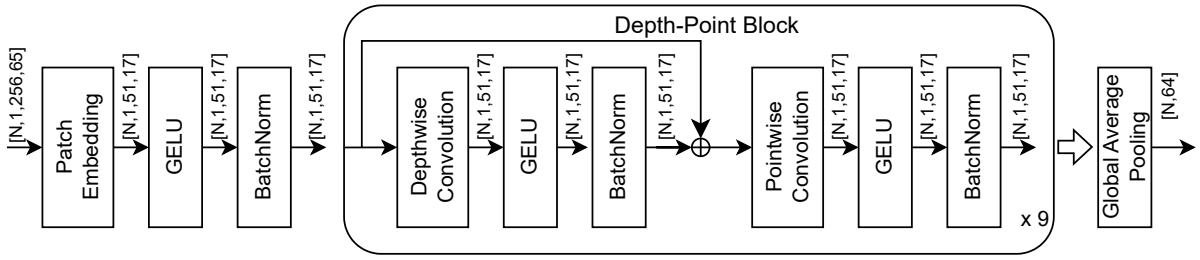

Figure 1: Isotropic architecture with the depth-point blocks.

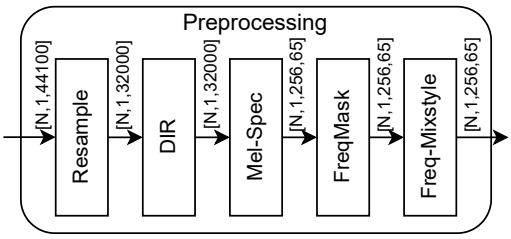

Figure 2: Data preprocessing steps.

baseline [12][1]. Additionally, frequency masking is implemented to randomly mask 48 frequency bins, resulting in a transformed shape of 1x256x65.

2.2. Freq-Mixstyle

In this study, normalization is applied exclusively to frequency bands instead of the traditional channel-wise approach. Subsequently, the frequency statistics from a random sample within the batch are integrated. This technique adapts the Mixstyle Domain Generalization method [4], which aims to improve generalization between unseen domains. The Freq-MixStyle, according to the baseline implementation, retains the default parameters with a probability of $p_{fms} = 0.4$ to apply the Freq-MixStyle and a mixing coefficient of $\alpha = 0.3$.

2.3. Device Impulse Response Augmentation

To further improve domain generalization, device impulse response (DIR) augmentation is employed. This process utilizes 66 freely available vintage microphone impulse responses from MicIRP[2]. DIR augmentation involves convolving a randomly selected impulse response with the resampled audio file, which is then truncated to the predefined sample size of 32k. This augmentation is applied with a probability of $p_{dir} = 0.6$ and is specifically targeted at the dominant device, identified as device A. By exposing the model to a wide range of microphone characteristics through different impulse responses, the system's ability to generalize across various domains is significantly enhanced. Inspired by DIR implementation details provided by Morocutti *et al.* [13], we propose DIR_domain augmentation implementation[3]. The comprehensive pre-processing pipeline for the audio snippets is illustrated in Figure 2.

3. NETWORK ARCHITECTURE

We present three distinct systems based on variations in an isotropic architecture to address the challenges of the DCASE 2024 competition. These systems include an isotropic architecture (System 1), the recursive multicolumn version (System 2) for enhanced feature extraction, and a Representation Self-Challenging (RSC) model (System 3). Each system is designed to be simple and lightweight, operating on patch embedding within a scalable isotropic convolutional architecture throughout the network. System 3 further processes the features outputted by the isotropic architecture post-average pooling layer, excluding the classification layer. The foundational model incorporates normalization and augmentation techniques to achieve domain generalization through data manipulation. The RSC versions build on this by adopting a representation-learning approach, ensuring that the models produce domain-invariant feature embeddings.

3.1. Isotropic architecture

The isotropic architecture is renowned for its flexibility to process inputs of varying sizes and aspect ratios effectively while requiring fewer labeled examples for training. As illustrated in Figure 1, this architecture initially divides the input into multiple patches, effectively reducing their internal resolution. Repeated depth-wise and point-wise convolutions are then applied, which decouple channel features from spatial features. This separation allows the architecture to manage distant spatial patterns with linear complexity. The efficiency of the isotropic architecture is dependent on several hyperparameters: patch size p, embedding dimension h, kernel size k, and depth d:

- $patch\_size$: Governs the size of the patches, determining how much of the original input's features are preserved during processing. A smaller p results in larger patches, retaining more original features. We adjust this parameter to maximize the allowed multiply-accumulate operations (MACs).
- $embedding\_dimension$: Dictates the number of filters that the architecture will learn, where a higher number is typically preferable.
- $kernel\_size$: Determines the size of the convolution filters, set relative to the patch size to optimize MAC usage.

[1] https://github.com/CPJKU/dcase2024_task1_baseline

[2] https://micirp.blogspot.com/?m=0

[3] https://github.com/hubtru/ASCDomain

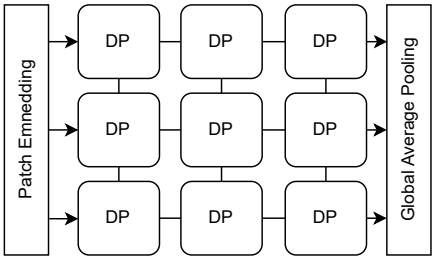

Figure 3: Recursive architecture with 3×3 array structure of depth-point blocks (DP).

- $depth$: Specifies the number of convolutional blocks adjusted to meet the maximum complexity allowed in the competition.

After experimenting with various settings to fine-tune the best setting, we determined that a patch size of $p = 5$, embedding dimension of $h = 64$, kernel size of $k = 3$, and depth of $d = 9$ provided the best performance, achieving high accuracy with 29,747,914 MACs and 47,946 parameters. This configuration ensures that the network maintains consistent resolution and size throughout, as detailed in Table 1.

Table 1: Details of isotropic network architecture.

Layer	N	Shape	Params	MACs
Input	-	$1 \times 256 \times 65$	-	-
Patch Embedding	-	$64 \times 51 \times 13$	1.664	1.103.232
Depth-Point Block	9	$64 \times 51 \times 13$	5.056	3.182.656
AvgPool	-	64	-	-
Classification	-	10	650	650

3.2. Recursiv

Recursiv extends the isotropic architecture by introducing an additional parameter, $columns$, which organizes the depth-point blocks into a two-dimensional grid structure. This reconfiguration improves connectivity between blocks, fostering a richer information flow throughout the network, and preserving total information without the typical loss or compression seen in conventional architectures. The simplified structure of the recursiv network, shown in Figure 3, features a 3×3 array of depth-point blocks. To maintain complexity requirements, we preserved the isotropic architecture's hyperparameters except for depth, adapting a 3×3 grid of depth-point blocks. Consequently, recursiv architecture retains the same number of depth-point blocks as the isotropic model, ensuring equal MACs and parameter counts.

3.3. RSC

The Representation Self-Challenging (RSC) method enhances learning by iteratively de-emphasizing dominant features, compelling the network to focus on less prominent yet relevant features for classification. This technique involves creating a mask that nullifies features in the p^{th} percentile of the top gradients, effectively challenging the network to adapt. The effectiveness of RSC extends to batch processing, where it similarly disregards dominant samples based on their cross-entropy loss values. We tested various settings for RSC and found that dropping 5% of features and samples ($drop_f = 0.05$ and $drop_b = 0.05$) consistently yielded the best results. The hyperparameters sensitivity studies of changing $drop_f$ and $drop_b$ values are presented in our repository. The visual representation of the RSC process is provided in Figure 4.

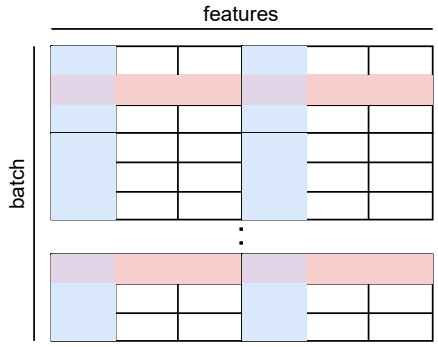

Figure 4: Intuition of RSC: Red marked rows are samples to be dropped, blue marked columns are features to be dropped.

4. EMPIRICAL EVALUATION

4.1. Evaluation method

Our evaluation adheres strictly to the guidelines outlined in the DCASE2024 Task 1 Challenge [14][4]. We use the official evaluation package provided by the organizers[5], evaluating our models on the TAU Urban Acoustic Scenes 2022 Mobile development dataset [15]. To report model parameters and MACs (million multiply-accumulate operations), we use the officially recommended NeSsi tool[6]. Model tuning involves first adjusting the parameters without DIR augmentation and subsequently incorporating them, aiming to optimize either the number of MACs or the parameters. This results in three distinct versions of each isotropic and recursiv architecture. Details on MACs and parameters for each model variant are provided in Table 2.

4.2. Training setup

All models are trained under similar conditions to ensure uniformity across evaluations. The training parameters are as follows: each model undergoes 150 epochs with a batch size of 256, and a maximum learning rate of 0.005, using the AdamW optimizer. The learning rate is managed by a cosine scheduler with an initial warm-up of 2000 steps and a weight decay set at 0.0001. These training settings are in line with those used by Baseline24, ensuring a fair comparison of performance enhancements across all systems and dataset splits. Model-specific hyperparameters are not varied between splits to maintain consistent training conditions throughout the study.

[4] https://dcase.community/challenge2024/

[5] https://github.com/toni-heittola/dcase2024_task1_submission_validator

[6] https://github.com/AlbertoAncilotto/NeSsi

Table 2: Comprehensive model macro average accuracy including training sizes, parameter counts, MACs, and hyperparameters: p, d, c, and k represent patch size, depth, columns, and kernel size, respectively. Freq-MixStyle and device impulse response (DIR) are applied as default. For submitted systems, the augment with aug_p = 6 is applied. Abbreviations HPO = Hyper Parameter Optimisation, N/A = Not Available.

Model	5%	10%	25%	50%	100%	HPO	**Params**	**%**	**MACs**	**%**	**dtype**	**p**	**d × c**	**k**
Baseline24	42.40%	45.29%	50.29%	53.29%	56.99%	-	61.148	72.06	29.419.156	97.57	16			
Isotropic	44.22%	47.66%	**55.51%**	**58.36%**	**60.97%**	-	47.946	74.92	29.747.914	99.16	16	5	9	3
Isotropicv2	42.28%	47.54%	53.11%	54.77%	58.18%	×	86.666	67.71	21.270.026	70.90	8	8	4×4	4
Isotropicv3	42.39%	47.87%	51.52%	56.12%	58.17%	-	119.690	93.51	29.463.050	98.21	8	8	6×4	4
Recursiv (3x3)	**45.76%**	**50.46%**	52.68%	55.41%	59.98%	-	47.946	74.92	29.747.914	99.16	16	5	3×3	3
Recursivv2 (2x2)	42.65%	47.57%	52.51%	55.53%	58.84%	×	86.666	67.71	21.270.026	70.90	8	8	2×2	4
Recursivv3 (2x3)	42.09%	46.78%	53.68%	55.74%	N/A	-	119.690	93.51	29.463.050	98.21	8	8	2×3	4
RSC (0.05,0.05)	44.75%	48.76%	54.86%	56.93%	58.85%	-	47.946	74.92	29.747.914	99.16	16	5	9	3

4.3. Results

Our proposed methods significantly outperform Baseline24, a modification of the top model from the DCASE23 challenge. The isotropic architecture achieves superior performance in three out of five data splits (25%, 50%, and 100%), recording macro-average accuracy of 55. 21%, 58. 36%, and 60. 97%, respectively. The recursiv architecture excels in the 5% and 10% splits, achieving macro-average accuracy of 45. 76% and 50. 46%. Remarkably, both architectures require only 47,946 parameters, which is 74.92% of the allowed limit. Detailed results from network component ablation studies, sensitivity analyzes, and visualizations are available in our repository[7]. The comprehensive performance data for all the systems submitted, their variants, and Baseline24 across the five data groups are presented in Table 2.

5. CONCLUSION

Our experimental findings substantiate that simple, lightweight isotropic architectures leveraging patch embedding, normalization, and audio data augmentation techniques deliver state-of-the-art results while maintaining low parameter complexity and a reduced number of MACs. Future research will aim to enhance the efficacy of adversarial and self-challenging representation techniques to further foster domain-invariant feature embeddings.

6. REFERENCES

[1] X. Li, W. Zhang, H. Ma, Z. Luo, and X. Li, "Domain generalization in rotating machinery fault diagnostics using deep neural networks," *Neurocomputing*, vol. 403, pp. 409–420, 2020.

[2] N. Honarvar Nazari and A. Kovashka, "Domain generalization using shape representation," in *European Conference on Computer Vision*. Springer, 2020, pp. 666–670.

[3] K. Zhou, Y. Yang, T. Hospedales, and T. Xiang, "Deep domain-adversarial image generation for domain generalisation," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, no. 07, 2020, pp. 13 025–13 032.

[4] K. Zhou, Y. Yang, Y. Qiao, and T. Xiang, "Domain generalization with mixstyle," *arXiv preprint arXiv:2104.02008*, 2021.

[5] H. Truchan, E. Naumov, R. Abedin, G. Palmer, and Z. Ahmadi, "Multimodal isotropic neural architecture with patch embedding," in *International Conference on Neural Information Processing*. Springer, 2023, pp. 173–187.

[6] T. Matsuura and T. Harada, "Domain generalization using a mixture of multiple latent domains," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 07, 2020, pp. 11 749–11 756.

[7] X. Zhang, P. Cui, R. Xu, L. Zhou, Y. He, and Z. Shen, "Deep stable learning for out-of-distribution generalization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 5372–5382.

[8] A. Dubey, V. Ramanathan, A. Pentland, and D. Mahajan, "Adaptive methods for real-world domain generalization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 14 340–14 349.

[9] Y. Shu, Z. Cao, C. Wang, J. Wang, and M. Long, "Open domain generalization with domain-augmented meta-learning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 9624–9633.

[10] D. Kim, Y. Yoo, S. Park, J. Kim, and J. Lee, "Selfreg: Self-supervised contrastive regularization for domain generalization," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 9619–9628.

[11] Q. Fan, M. Segu, Y.-W. Tai, F. Yu, C.-K. Tang, B. Schiele, and D. Dai, "Normalization perturbation: A simple domain generalization method for real-world domain shifts," *arXiv preprint arXiv:2211.04393*, 2022.

[12] F. Schmid, T. Morocutti, S. Masoudian, K. Koutini, and G. Widmer, "Distilling the knowledge of transformers and cnns with cp-mobile," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2023 Workshop (DCASE2023)*, 2023, pp. 161–165.

[13] T. Morocutti, F. Schmid, K. Koutini, and G. Widmer, "Device-robust acoustic scene classification via impulse response augmentation," in *2023 31st European Signal Processing Conference (EUSIPCO)*. IEEE, 2023, pp. 176–180.

[14] F. Schmid, P. Primus, T. Heittola, A. Mesaros, I. Martín-Morató, K. Koutini, and G. Widmer, "Data-efficient low-complexity acoustic scene classification in the dcase 2024 challenge," *arXiv preprint arXiv:2405.10018*, 2024.

[15] T. Heittola, A. Mesaros, and T. Virtanen, "Acoustic scene classification in dcase 2020 challenge: generalization across devices and low complexity solutions," in *Proceedings of the*

[7] https://github.com/hubtru/ASCDomain

Detection and Classification of Acoustic Scenes and Events 2020 Workshop (DCASE2020), 2020, pp. 56–60. [Online]. Available: https://arxiv.org/abs/2005.14623