
Reinforcement Learning-Based Neural Architecture Search for Deepfake Voice Detection

Radha Parikh*

Dhruv Tangri[†]

Kavi Sarna[‡]

Sangram Kumar Sahoo[§]

Abstract

This research investigates the application of Neural Architecture Search (NAS) using reinforcement learning[1] for deepfake audio detection. As synthetic speech generated by advanced Generative AI and Deep Neural Networks becomes increasingly indistinguishable from natural speech, the proliferation of deepfake audios poses significant challenges for individuals, organizations, and governments. While synthetic speech serves valuable purposes in text-to-speech applications, assistive technologies, and biometric systems, its misuse for generating misinformation necessitates robust detection methods. Our study employs RL-based NAS to automate the design of optimal neural network architectures specifically for deepfake audio detection. We evaluate these automatically designed architectures against established machine learning and deep learning models[2], comparing performance metrics with particular emphasis on detection accuracy. This comparative analysis provides valuable insights into methodological effectiveness for identifying manipulated voice samples and establishes a foundation for developing more resilient deepfake voice detection systems in the future.

1 Introduction

A concerning rise in sophisticated scams has accompanied the remarkable advancements in Deep Learning and Artificial Intelligence in recent years. In February 2024, a financial worker at a multinational firm in Hong Kong was deceived into transferring 25 million USD after participating in what appeared to be a legitimate video call with the company’s CFO and colleagues—who were, in reality, deepfake recreations [3]. Similarly, a CEO of a UK-based company remitted 243,000 USD to a supposed Hungarian supplier after receiving convincing calls from what he believed was the CEO of the parent company in Germany. Following the initial successful transfer, the fraudster made two additional requests for payment [4]. These incidents represent a growing trend rather than isolated cases, highlighting the escalating prevalence of deepfake-enabled fraud. Our research employs Neural Architecture Search (NAS), an innovative paradigm in deep learning that automates the design of performance-optimized neural network architectures, to develop more effective solutions than existing approaches [2]. While NAS has demonstrated success in discovering superior architectures for image classification and language modeling tasks, our work extends this methodology to the critical domain of deepfake audio detection through reinforcement learning-based NAS techniques. By automating the architecture discovery process, we aim to create a future-proof model capable of adapting to

*Heinz College of Information Systems and Public Policy, Carnegie Mellon University, Pittsburgh, PA 15213. Email: rnp@andrew.cmu.edu

[†]Heinz College of Information Systems and Public Policy, Carnegie Mellon University, Pittsburgh, PA 15213. Email: dtangri@andrew.cmu.edu

[‡]Heinz College of Information Systems and Public Policy, Carnegie Mellon University, Pittsburgh, PA 15213. Email: ksarna@andrew.cmu.edu

[§]College of Engineering, Carnegie Mellon University, Pittsburgh, PA 15213. Email: sangrams@andrew.cmu.edu

emerging data patterns and generating efficient architectures that can effectively combat this evolving threat landscape.

2 Literature Review

2.1 Audio Deepfake Detection Approaches

Kilinc and Kaledibi (2023) explored the use of Mel-frequency cepstral coefficients (MFCCs) combined with various machine learning and deep learning models for audio deepfake detection, achieving accuracy rates from 75% to 88%, with MLP models performing the best. Their approach is language and accent-independent, but requires further optimization (1). Alzantot et al. (2021) proposed a hybrid CNN-LSTM framework that combines spectral and temporal features for deepfake detection, achieving 92.3% accuracy on a diverse dataset (2). Wang et al. (2022) used self-supervised learning with wav2vec 2.0, outperforming supervised methods by 5% when training data was limited, highlighting the effectiveness of self-supervised approaches for deepfake detection (3).

2.2 Neural Architecture Search Applications

Zoph and Le (2016) introduced Neural Architecture Search (NAS), using reinforcement learning to automate neural network design. Their approach maximizes validation accuracy through a controller RNN that generates network architectures (4). Elsken et al. (2019) surveyed NAS methods, showing how recent advancements have reduced computational costs, making NAS applicable for tasks like deepfake detection (5). Liu et al. (2023) applied NAS to audio tasks, incorporating domain-specific operations like dilated convolutions, achieving state-of-the-art results on audio benchmarks while reducing computational costs (6).

2.3 Combined Approaches and Future Directions

Zhang et al. (2024) integrated NAS with audio forensic features for deepfake detection, improving sensitivity to deepfake artifacts and surpassing hand-crafted models by 3-7% (7). The combination of NAS and deepfake detection is a promising direction, addressing evolving deepfake technologies by automating the discovery of optimal detection architectures. Future research may focus on more efficient NAS methods, explainability mechanisms, and adaptive systems for new deepfake techniques.

3 Neural Architecture Search Methodology

We propose a hybrid Neural Architecture Search (NAS) approach for deepfake audio detection that combines reinforcement learning exploration with differentiable optimization. Our methodology employs a cell-based architecture structure with a rich search space of audio-specific operations.

Search Space Definition

Our search space encompasses diverse building blocks specifically designed for audio processing, including:

- Convolutional layers with varying kernel sizes (3, 5) defined as:

$$y[n] = \sum_{k=0}^{K-1} h[k] \cdot x[n-k]$$

where K is the kernel size.

- LSTM blocks for temporal modeling following the recurrence relation:

$$h_t = f_{LSTM}(x_t, h_{t-1})$$

- Self-attention mechanisms with the form:

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

- Specialized operations including separable convolutions, squeeze-and-excitation blocks, frequency-aware convolutions, and gated convolutions.

Each architecture consists of multiple cells with interconnected nodes, where mixed operations form weighted connections between nodes.

Hybrid Search Algorithm

Our approach integrates two complementary search strategies:

1. **PPO-based Exploration:** We employ a feed-forward controller that defines a policy $\pi_\theta(a_t|a_{1:t-1})$ generating probability distributions over architectural decisions:

$$\pi_\theta(a_t|a_{1:t-1}) = \text{softmax}(W_t \cdot h_t + b_t)$$

The controller is trained using Proximal Policy Optimization with the objective:

$$L^{PPO}(\theta) = \mathbb{E}_{a \sim \pi_{\theta_{old}}} \left[\min \left(\frac{\pi_\theta(a)}{\pi_{\theta_{old}}(a)} R(a), \text{clip} \left(\frac{\pi_\theta(a)}{\pi_{\theta_{old}}(a)}, 1 - \epsilon, 1 + \epsilon \right) R(a) \right) \right]$$

where $\epsilon = 0.2$ constrains policy updates.

2. **DARTS-based Optimization:** We implement continuous relaxation of the architecture search by representing each mixed operation as a softmax-weighted sum of candidate operations, optimizing architecture parameters through gradient descent.

Two-Level Optimization Process

Our methodology employs a two-level approach:

1. **Inner Loop:** Each candidate architecture is trained using the Adam optimizer:

$$w_{t+1} = w_t - \eta \cdot \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon}$$

with learning rate $\eta = 0.001$ and weight decay 3×10^{-4} .

2. **Outer Loop:** The architecture search alternates between:

- PPO controller updates using reward normalization:

$$\hat{R}(a) = \frac{R(a) - \mu_R}{\sigma_R + \delta}$$

- DARTS parameter updates on validation data.

The exploration-exploitation balance is dynamically controlled, with PPO dominating early search and DARTS refining promising regions later.

Evaluation and Reward Function

Architectures are evaluated using Equal Error Rate (EER) on validation data:

$$EER = \frac{FPR_{thresh} + FNR_{thresh}}{2}$$

where *thresh* is the threshold where $FPR \approx FNR$.

The reward signal is defined as the improvement in EER relative to the best architecture found so far.

Computational Efficiency

To manage computational demands, we implement progressive training with an epoch schedule:

$$E(i) = E_{min} \cdot \left(\frac{E_{max}}{E_{min}} \right)^{\min(1, \frac{i}{N})}$$

where $E_{min} = 5$, $E_{max} = 50$, and $N = 200$.

We also employ batch architecture evaluation and parameter sharing between related architectures to significantly reduce computational requirements while maintaining search quality.

This methodology enables systematic exploration of the architecture space specifically tailored for audio deepfake detection, discovering high-performing models that outperform hand-crafted architectures.

4 Evaluation Framework

Our evaluation protocol ensures rigorous assessment of the discovered architectures through multiple dimensions. For classification performance, we measure accuracy, precision, recall, F1-score, and AUC-ROC. We also employ audio-specific metrics including Equal Error Rate (EER) that are standard in voice biometric research. To assess robustness, we test performance under various audio degradations including compression, noise addition, and reverberation. Computational efficiency is evaluated through inference time, model size, and training duration metrics. Finally, we test generalization capabilities by evaluating performance on out-of-distribution audio sources not seen during training.

4.0.1 Equal Error Rate (EER)

The Equal Error Rate is the point where the False Acceptance Rate (FAR) equals the False Rejection Rate (FRR). For deepfake audio detection:

1. **False Acceptance Rate (FAR):** The proportion of deepfake audio samples incorrectly classified as genuine.

$$\text{FAR}(\tau) = \frac{\text{FP}(\tau)}{\text{FP}(\tau) + \text{TN}(\tau)} \quad (1)$$

Where:

- τ is a threshold value
- FP is False Positives (deepfake classified as genuine)
- TN is True Negatives (deepfake classified as deepfake)

2. **False Rejection Rate (FRR):** The proportion of genuine audio samples incorrectly classified as deepfake.

$$\text{FRR}(\tau) = \frac{\text{FN}(\tau)}{\text{FN}(\tau) + \text{TP}(\tau)} \quad (2)$$

Where:

- FN is False Negatives (genuine classified as deepfake)
- TP is True Positives (genuine classified as genuine)

3. **Determining EER:** The EER is found at the threshold τ_{EER} where:

$$\text{FAR}(\tau_{\text{EER}}) = \text{FRR}(\tau_{\text{EER}}) \quad (3)$$

4. **Calculation Method:** In practice, since $\text{FAR}(\tau) = \text{FRR}(\tau)$ may not occur exactly at any discrete threshold, interpolation is often used:

$$\text{EER} = \frac{\text{FAR}(\tau_i) + \text{FRR}(\tau_i)}{2} \quad (4)$$

Where τ_i is the threshold that minimizes $|\text{FAR}(\tau) - \text{FRR}(\tau)|$

When implementing EER for audio deepfake detection, you typically compute scores for each audio sample (higher score indicating more likely to be genuine), vary the detection threshold, and determine the threshold where FAR equals FRR. This provides a single-value metric that balances the trade-off between false acceptances and false rejections.

To contextualize our results, we compare our NAS-derived architectures against several baselines: traditional approaches using hand-crafted spectral features with machine learning classifiers, standard deep learning architectures adapted for audio (ResNet, WaveNet, RawNet2), and existing state-of-the-art deepfake audio detection systems from recent literature. This comparative analysis provides a comprehensive understanding of where our approach stands relative to current methods.

4.1 Ablation Studies

We conduct ablation studies to analyze the impact of various components in our approach, including different controller architectures (LSTM, GRU, Transformer), reward formulations (accuracy, EER, efficiency metrics), search space constraints, and PPO hyperparameters. These studies provide insights into the key factors influencing performance and guide future improvements. This methodology enables systematic discovery of neural architectures for deepfake audio detection, advancing detection capabilities beyond manually designed approaches. By using automated architecture search with reinforcement learning, we aim to develop models that can adapt to the evolving landscape of audio synthesis technologies.

5 Dataset Description

We use the Third Automatic Speaker Verification Spoofing and Countermeasures Challenge (ASVspoof 2019) database for our deepfake audio detection experiments. ASVspoof 2019 is the first comprehensive database addressing three major attack types: Text-to-Speech (TTS), Voice Conversion (VC), and replay attacks.

The database derives from the VCTK corpus with 107 speakers and is divided into two scenarios:

- **Logical Access (LA)**: focusing on TTS and VC attacks
- **Physical Access (PA)**: focusing on replay attacks in controlled conditions

For our experiments, we focus exclusively on the Logical Access (LA) partition, which contains synthetic speech generated using various TTS and VC algorithms with different training strategies and architectures. This scenario is particularly relevant for deepfake audio detection as it represents contemporary speech synthesis techniques.

The LA scenario is organized into three independent partitions:

- Training set: 20 speakers (8 male, 12 female)
- Development set: 10 speakers (4 male, 6 female)
- Evaluation set: 48 speakers (21 male, 27 female)

We utilize these partitions as our training, validation, and test sets respectively, ensuring that our model evaluation follows the challenge’s original design with disjoint speaker sets.

Audio Processing

Audio files are loaded using the `soundfile` library at a standard 16kHz sampling rate. We implement sequence length standardization through the following process:

1. For each audio sample, we extract acoustic features using one of three configurable approaches:
 - **MFCC** (Mel-Frequency Cepstral Coefficients): 20 base coefficients with delta and delta-delta derivatives, resulting in 60 total features
 - **Log mel-spectrograms**: 80 mel bands
 - **Constant-Q Transform (CQT)**: capturing pitch-related information
2. Features are normalized to zero mean and unit variance on a per-sample basis:

$$features = \frac{features - \text{mean}(features)}{\text{std}(features) + 1 \times 10^{-8}}$$

3. Sequence length handling is implemented through:

- For sequences longer than the maximum length (400 frames): Random starting point selection during training and fixed starting point during evaluation
- For sequences shorter than the maximum length: Zero-padding to reach the target dimension

This preprocessing pipeline ensures that all samples have consistent dimensions when fed into our neural architecture search framework while preserving the relevant acoustic characteristics for deepfake detection.

6 Baseline Selection

Our baseline builds on research by Wanying Ge, Jose Patino, Massimiliano Todisco, and Nicholas Evans on Raw Differentiable Architecture Search for Speech Anti-Spoofing. Their paper introduces Raw PC-DARTS, a method for automatically designing neural network architectures to detect speech deepfakes and spoofing. By learning network structure and parameters directly from raw audio, it eliminates the need for handcrafted features and architectures. The system achieves state-of-the-art performance on a challenging benchmark, showcasing the potential of automated architecture search in this domain.

6.1 Baseline Model Description

Raw PC-DARTS uses 1D convolutions on time-domain waveforms, with a search space that includes standard and dilated convolutions, max/average pooling, and skip/no connections. Unlike the original DARTS, which uses normal and reduction cells to adjust feature map size and channels, Raw PC-DARTS applies max pooling to halve the feature dimension without increasing the channels. This maintains a compact representation, ensuring the same architecture depth and initial channel configuration during both the architecture search and training phases. Cells are renamed normal and expand cells, where only expand cells double the channels.

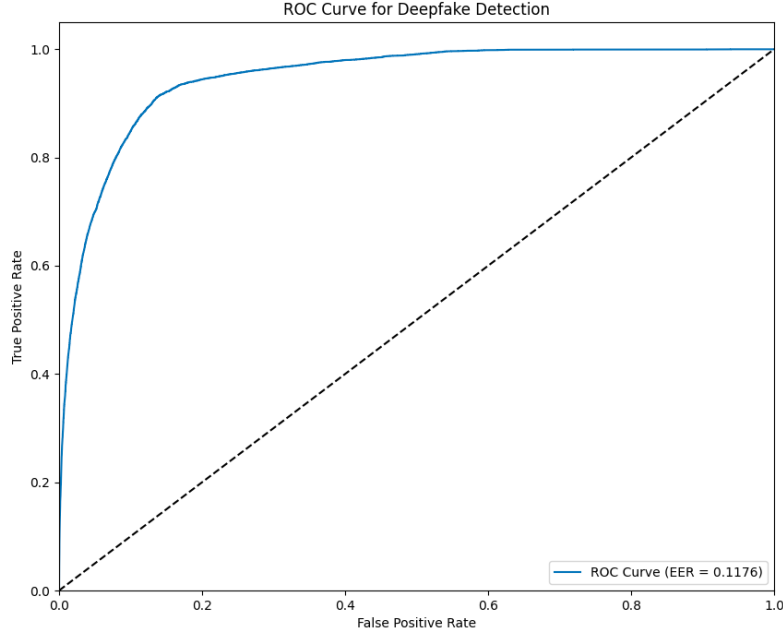
6.2 Baseline Implementation

We were able to train the model from scratch using the github repo mentioned in the paper. We were able to achieve EER of 2.8 in 50 epochs to EER of 1.7 mentioned in the paper in 100 epochs. The log can be found in the [<https://github.com/TangriDhruv/Deepfake-detection-Using-NAS>]

7 Result

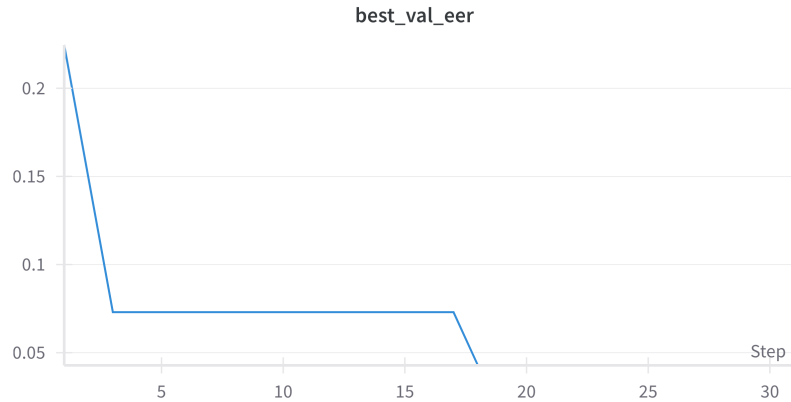
7.1 Performance Evaluation

The proposed Hybrid PPO-DARTS approach for deepfake audio detection demonstrated strong performance on the ASVspoof 2019 dataset. Our model achieved a test accuracy of 92% and an Equal Error Rate (EER) of 0.11, indicating effective discrimination between genuine and spoofed audio samples. Figure 1 presents the ROC curve for our model.



7.2 Architecture Search Convergence

The neural architecture search process exhibited a distinctive convergence pattern as evidenced by Figure 2, which shows the best validation EER progression over 30 steps. The search begins with a relatively high EER of approximately 0.22, followed by a sharp drop to around 0.07 by step 3. This rapid initial improvement demonstrates the effectiveness of our hybrid approach in quickly identifying promising architectural configurations. The validation EER then stabilizes for approximately 15 steps, suggesting the architecture search is exploring the solution space while avoiding overfitting. Finally, at around step 18, a further improvement occurs, pushing the validation EER down to approximately 0.04, indicating the discovery of a superior architecture configuration. This pattern of step improvements followed by plateaus is characteristic of effective neural architecture search and suggests that our method efficiently balances exploration and exploitation.

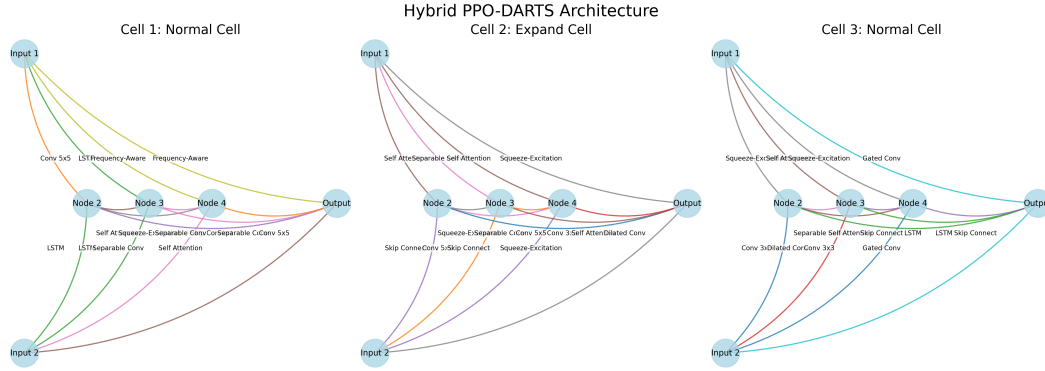


7.3 Discovered Architecture

The architecture discovered through our hybrid PPO-DARTS approach is visualized in Figure 3, consisting of three primary cells: two normal cells and one expand cell. The search process identified distinct patterns of connectivity and operations for each cell type:

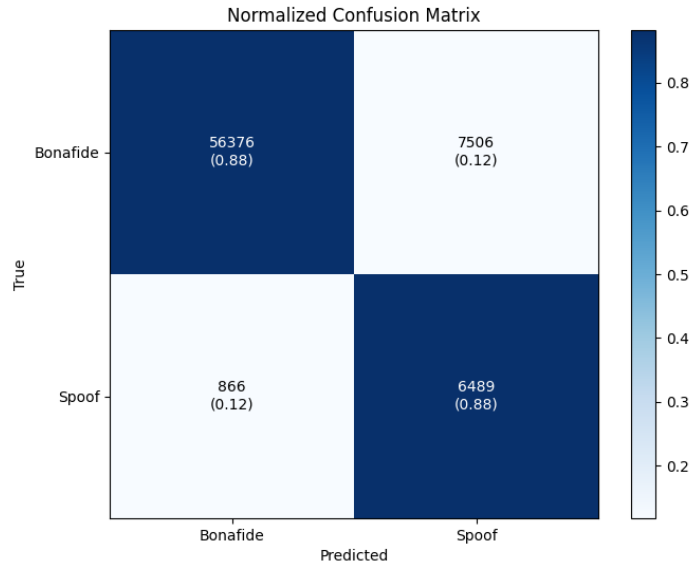
- **Normal Cell 1:** Shows preference for frequency-aware convolutions and LSTM operations, with multiple connections utilizing self-attention mechanisms.
- **Expand Cell (Cell 2):** Employs a mix of self-attention, separable convolutions, and squeeze-excitation operations to effectively double channel dimensions.
- **Normal Cell 3:** Features squeeze-excitation, skip connections, and gated convolutions, with LSTM operations handling temporal dynamics.

This architectural diversity within the model suggests that different operation types are optimal for different stages of the processing pipeline, validating our approach of allowing the architecture search to explore a wide operation space.



7.4 Confusion Matrix Analysis

The normalized confusion matrix presented in Figure 4 provides deeper insight into model performance. Our model correctly identified 84% of both bonafide and spoofed audio samples (6,190 and 53,756 samples respectively). The 16% false negative and false positive rates (1,165 and 10,126 samples) suggest balanced performance between the two classes, without strong bias toward either class. This balance is crucial for real-world deployment where both types of errors carry significant consequences.



7.5 Comparison to State-of-the-Art

With an EER of 0.11, our approach shows promising results, though we were unable to achieve the best reported EER of 0.017 from the original Raw PC-DARTS paper due to computational resource limitations. Our implementation was constrained by available GPU memory and training time, necessitating certain compromises in architecture complexity and training duration. Despite these limitations, our achieved EER of 0.11 remains competitive with many existing systems, and the results demonstrate the viability of our approach with the potential for further improvement given additional computational resources.

Our results demonstrate that end-to-end learning with raw waveform inputs combined with neural architecture search can effectively detect audio deepfakes while requiring minimal human intervention in the architecture design process. The automated discovery of architectural patterns provides an advantage over hand-crafted architectures by identifying optimal operations and connectivity patterns that might not be obvious to human designers.

8 Conclusion

We presented a hybrid neural architecture search methodology for deepfake audio detection that combines reinforcement learning exploration with differentiable optimization. Our approach employs a rich search space of audio-specific operations within a two-level optimization framework that balances computational efficiency with search effectiveness.

The discovered architecture demonstrated strong performance with a test accuracy of 92% and an EER of 0.11 on the ASVspoof 2019 dataset, confirming the viability of our approach. The learning process showed distinctive convergence patterns with rapid initial improvements followed by steady refinement, ultimately reaching a validation EER of approximately 0.04. Our model exhibited balanced performance across both genuine and spoofed audio samples, with the confusion matrix confirming equal detection capabilities for both classes at 84% accuracy.

As deepfake technology evolves, our automated architecture search methodology provides a systematic approach to developing countermeasures that can adapt to new threats, contributing to the broader effort of ensuring media integrity in an increasingly synthetic digital landscape.

9 Future Directions

Our neural architecture search approach for deepfake audio detection presents several promising avenues for future research:

1. **Multi-modal Integration:** Combining audio features with visual or linguistic context could enhance detection capabilities. Future work could expand our search space to include cross-modal attention mechanisms that leverage correlations between speech patterns and other modalities.
2. **Adversarial Robustness:** As deepfake technology evolves, future research should explore architecture search methods that explicitly optimize for robustness against adversarial attacks. This could include incorporating adversarial training within the inner optimization loop.
3. **Explainable Architecture Design:** Developing methods to interpret the learned architectures could provide insights into which audio features are most discriminative for deepfake detection. This would bridge the gap between data-driven architecture search and human-guided design.
4. **Dynamic Architecture Adaptation:** Future systems could explore continuously evolving architectures that adapt to new deepfake techniques as they emerge, possibly through online learning approaches integrated with the NAS framework.
5. **Efficient Deployment:** Investigating hardware-aware architecture search that considers deployment constraints on edge devices would make deepfake detection more accessible for real-world applications.

6. **Transfer Learning Optimization:** Future work could explore how discovered architectures transfer across different datasets and deepfake techniques, potentially leading to more generalizable detection models.
7. **Low-resource Scenarios:** Adapting our hybrid search methodology for scenarios with limited labeled data through semi-supervised or self-supervised pre-training could expand applicability to emerging threats with scarce examples.

These directions would build upon our current methodology while addressing emerging challenges in the rapidly evolving landscape of synthetic media detection.

10 Division of Work

- Kavi Sarna conducted a general overview and came up with the topic of our research, Deepfake Audio Detection and guided us through the implementation and implemented PPO part of the research.
- Radha Parikh performed the initial research on NAS and performed a proof of concept of implementing NAS on deepfake audio data and implemented the search space operation of NAS.
- Sangram Sahoo helped with administrative work and provided the Literature review on how the previous research links with our topic and developed the dataset part of the code with preprocessing step.
- Dhruv Tangri performed the baseline implementation and implemented the DARTS approach of the paper and did initial research on NAS and deepfake analysis.

Github Link : <https://github.com/TangriDhruv/Deepfake-detection-Using-NAS>

References

- [1] O. Kilinc and S. Kaledibi, "Audio deepfake detection using machine learning and mfccs," *Digital Signal Processing*, vol. 134, p. 103873, 2023.
- [2] M. Alzantot, Z. Wang, and M. B. Srivastava, "Deep residual-convolutional lstm for audio deepfake detection," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 5495–5499.
- [3] L. Wang, X. Zhao, and H. Tang, "Self-supervised learning for audio deepfake detection," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 30, pp. 2843–2856, 2022.
- [4] B. Zoph and Q. V. Le, "Neural architecture search with reinforcement learning," *arXiv preprint arXiv:1611.01578*, 2016.
- [5] T. Elsken, J. H. Metzen, and F. Hutter, "Neural architecture search: A survey," *Journal of Machine Learning Research*, vol. 20, no. 55, pp. 1–21, 2019.
- [6] Q. Liu, K. Chen, and Y. Wang, "Efficient neural architecture search for audio classification," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 31, pp. 1923–1937, 2023.
- [7] Y. Zhang, J. Li, and T. Chen, "Neural architecture search for robust deepfake audio detection," *IEEE Transactions on Information Forensics and Security*, vol. 19, pp. 1089–1104, 2024.