Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

A Lightweight Log-Mel CNN for Audio Deepfake Detection: An Empirical Study of Speaker-Disjoint and Cross-Dataset Generalization

Gaurav Pratap Singh
Indian Institute of Technology Roorkee
Mayank Goswami
mayankfph@iitr.ac.in

Indian Institute of Technology Roorkee https://orcid.org/0000-0001-8829-9980

Method Article

Keywords: audio deepfake detection, spoofed speech, log-Mel spectrogram, convolutional neural network, speaker-disjoint evaluation, domain shift, WaveFake

Posted Date: November 17th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11729499/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: The authors declare no competing interests.

Abstract

Synthetic speech generated by text-to-speech and voice-conversion systems can be difficult to distinguish from bonafide speech, creating a need for practical audio-spoofing detectors. This report describes a compact convolutional neural network (CNN) that classifies audio as real or fake using log-Mel spectrogram features. The supplied implementation combines the ASVspoof 2019 Logical Access training partition with the training partition of the In-the-Wild dataset. Audio is converted to mono, resampled to 16 kHz, cropped or zero-padded to four seconds, transformed using a short-time Fourier transform, projected onto 64 Mel bands, and normalized in the log-decibel domain. A four-block 2-D CNN is optimized for five epochs using binary cross-entropy with logits and AdamW. Model selection is performed using validation ROC-AUC, with the epoch-three checkpoint selected at 0.9890. On a reported speaker-disjoint In-the-Wild test set of 8,966 recordings, the checkpoint achieves ROC-AUC 0.99278, accuracy 98.32%, F1 98.79%, and an estimated equal error rate (EER) of 2.40%. However, an external WaveFake sample containing 700 real and 700 generated clips yields ROC-AUC 0.6630 and 50% accuracy at the unchanged 0.5 threshold; all clips are predicted as real at that threshold. This discrepancy highlights the difference between performance on a held-out split from a familiar dataset and transfer to a different synthesis domain. The system is therefore a promising baseline for the evaluated split, but the external result does not support claims of robust general-purpose audio deepfake detection.

1. Introduction

Neural text-to-speech (TTS) and voice-conversion systems can synthesize speech that resembles a target speaker. This capability supports accessibility, content production, and human-computer interaction, but can also be misused for impersonation, social engineering, and the fabrication of spoken statements. A listening-only judgement is not a reliable authentication method, particularly when generated speech is fluent and the listener has limited contextual information.

Audio anti-spoofing research studies whether a recording is bona fide or produced by a synthesis, conversion, or replay attack. ASVspoof introduced standardized datasets and protocols to compare countermeasures, including a Logical Access (LA) condition for TTS and voice-conversion attacks [1]. Realistic evaluation remains challenging: the In-the-Wild dataset was introduced specifically to study whether detectors generalize to recordings collected from online sources [2]. External resources such as WaveFake provide samples from several neural waveform-generation architectures [3]. Recent surveys identify generalization, cross-dataset evaluation, and robustness as central open problems in speech deepfake detection [4].

This work documents an implementation built around a small 2-D CNN operating on log-Mel spectrograms. Its goal is not to claim a new state of the art. Instead, it reports the complete data-processing and training pipeline, evaluates a speaker-disjoint test split, and contrasts that result with a

small external WaveFake experiment. The contrast is important: strong performance under one evaluation protocol does not establish reliable detection for previously unseen generators or domains.

2. Related work

2.1 ASVspoof and logical-access spoofing

The ASVspoof 2019 database includes bona fide and spoofed examples under Logical Access and Physical Access conditions. The LA condition includes speech synthesis and voice conversion, making it relevant to the binary task studied here. The database paper describes the construction, protocols, attack implementations, and baseline countermeasures; it also notes that some spoofed signals can be difficult for human listeners to distinguish from genuine speech [1].

2.2 In-the-Wild evaluation

Müller et al. introduced the In-the-Wild dataset and investigated the extent to which audio deepfake detectors generalize beyond laboratory-style corpora [2]. The dataset contains bona fide and spoofed recordings collected from publicly accessible online sources and is intended to probe realistic recording and distribution conditions. The present project uses speaker partitions from this dataset for training, validation, and held-out testing, based on the supplied training logs.

2.3 WaveFake and cross-domain robustness

WaveFake was released to facilitate audio deepfake detection research and contains generated speech from multiple neural network architectures and reference corpora [3]. It is useful for examining detector behavior across synthesis methods. A broader survey by Li et al. reviews datasets, architectures, training strategies, metrics, and generalization challenges in speech deepfake detection [4]. The external evaluation in this project follows this motivation, although its sample is a small, matched subset rather than the complete WaveFake dataset.

3. Data and experimental protocol

3.1 Training sources

The final training log reports two sources. The ASVspoof 2019 LA training portion contributes 2,580 bona fide recordings and 22,800 spoof recordings (25,380 total). The In-the-Wild training partition contributes 12,401 real and 8,258 fake recordings (20,659 total). The combined training set therefore contains 46,039 recordings: 14,981 real and 31,058 fake. Labels are encoded as 1 for real/bona fide and 0 for fake/spoof. These sample counts are transcribed from the supplied experiment log; the corresponding manifest CSV files and source audio are not present in the supplied ZIP.

Source	Real	Fake / spoof	Total
ASVspoof 2019 LA train	2,580	22,800	25,380
In-the-Wild train	12,401	8,258	20,659
Combined train	14,981	31,058	46,039

3.2 Speaker-disjoint validation and test

The final training script loads separate In-the-Wild manifests for training, validation, and test. It checks that speaker identifiers are present and asserts that the speaker sets do not overlap. The training log reports 38 training speakers, eight validation speakers, and eight held-out test speakers. The validation set contains 2,154 recordings. The held-out test set contains 8,966 recordings, comprising 6,184 real and 2,782 fake/spoof clips. The test split is not used for gradient updates or checkpoint selection. The split check applies to the In-the-Wild partitions; the ASVspoof training set is added to the training examples.

Speaker-disjoint evaluation reduces the risk that a model can exploit speaker identity shared between training and test. It does not, by itself, remove all dataset-specific correlations or guarantee transfer to new codecs, synthesis systems, languages, or recording environments.

4. Method

4.1 Waveform standardization

Audio is read as float32 samples. Stereo files are converted to mono by averaging the channels. Each waveform is resampled to 16,000 samples per second when necessary. The target duration is four seconds, corresponding to 64,000 samples. During training, a longer recording is randomly cropped to four seconds; during validation and the final speaker-disjoint test pass, a center crop is used. A shorter recording is zero-padded at the end. Random cropping provides temporal variation across training presentations, while deterministic center cropping makes validation and test evaluation repeatable for a fixed waveform.

4.2 Short-time Fourier transform and log-Mel features

The short-time Fourier transform (STFT) uses an FFT size and window length of 400 samples, equivalent to 25 ms at 16 kHz, and a hop length of 160 samples, equivalent to 10 ms. A Hann window is applied, with centered frames and reflective boundary padding. The magnitude spectrum is squared to obtain power. A 64-band Mel filter bank covering 20 Hz to 8 kHz is applied to the power spectrogram. In matrix form, the Mel representation is M = W_Mel P, where P is the power spectrogram and W_Mel is the Mel filter-bank matrix.

The Mel power is converted to decibels using 10 log10(max(M, 10^-10)). For each example, the maximum value is subtracted, the result is clipped to [-80, 0] dB, and the values are mapped to approximately [-1, 1] using (L + 40) / 40. With a four-second waveform, the resulting feature tensor has shape (1, 64, 401): one input channel, 64 Mel bands, and 401 time frames. The implementation uses librosa to construct the Mel filter bank and PyTorch for per-example STFT computation.

4.3 CNN architecture

The classifier is a 2-D convolutional neural network. It contains four blocks with channel transitions 1→16, 16→32, 32→64, and 64→96. Each block comprises a 3×3 convolution with padding 1 and no bias, Batch Normalization, ReLU, and 2×2 max pooling. The feature-map sizes reported by the architecture are 16×32×200, 32×16×100, 64×8×50, and 96×4×25. Adaptive average pooling reduces the final map to 96 scalar activations. The classifier flattens these values, applies dropout with probability 0.30, and uses a linear layer from 96 features to one logit. A sigmoid converts this logit to a real-class score for reporting and threshold-based prediction.

4.4 Training objective and optimization

The model is trained using binary cross-entropy with logits (BCEWithLogitsLoss). The loss combines sigmoid and binary cross-entropy in a numerically stable implementation. The targets are floating-point labels 0 and 1, and the network emits an unrestricted logit. AdamW is used with learning rate 10^-3 and weight decay 10^-4. Batch size is 32, training runs for five epochs, and random seeds are set to 42 for Python, NumPy, and PyTorch. The supplied log reports an NVIDIA GeForce RTX 3050 Laptop GPU with 6 GB memory and four data-loader workers.

The training sampler assigns each example a weight equal to the inverse frequency of its source-label group. The four groups are ASVspoof-real, ASVspoof-spoof, In-the-Wild-real, and In-the-Wild-fake. Sampling is with replacement and draws 46,039 examples per epoch. This gives each of the four groups approximately equal total selection probability, rather than allowing the largest group to dominate each epoch. Consequently, one epoch is a weighted sampled pass and is not guaranteed to visit each original training file exactly once.

4.5 Checkpoint selection and inference

After every epoch, the model is evaluated on the validation set. The checkpoint with the highest validation ROC-AUC is retained. The reported best validation AUC is 0.9890 at epoch three, and the saved checkpoint is best_cnn_speaker_disjoint.pt. For the final speaker-disjoint evaluation script, each test recording is passed through the dataset preprocessing path, which uses a deterministic center crop for recordings longer than four seconds. The separate prediction utilities use overlapping four-second windows with a one-second stride and aggregate window scores by their mean and median. A score greater than or equal to 0.5 is interpreted as real in the supplied scripts. The sigmoid output is a model score; it has not been shown to be a calibrated probability.

5. Evaluation metrics

Accuracy measures the fraction of correct labels at a chosen threshold. Precision for the real class is TP/(TP + FP), and recall for the real class is TP/(TP + FN). The F1 score is the harmonic mean of precision and recall. ROC-AUC evaluates the ranking of real versus fake examples across thresholds. EER is estimated from the ROC curve as the operating point at which false-positive rate and false-negative rate are closest; the implementation averages the two rates at the closest sampled ROC point. Because the EER is estimated from a discrete ROC curve, it is an approximation. The confusion matrix is reported with rows ordered as actual [fake, real] and columns as predicted [fake, real].

6. Results

6.1 Validation and checkpoint selection

Validation ROC-AUC was 0.9786, 0.9823, 0.9890, 0.9814, and 0.9822 for epochs one through five. The highest validation AUC occurred at epoch three. Training AUC continued to increase across all five epochs, from 0.9691 to 0.9984, while validation AUC declined after epoch three. This divergence is consistent with the need to select checkpoints using validation performance rather than training performance alone.

Epoch	Train AUC	Validation AUC	Validation accuracy
1	0.9691	0.9786	94.61%
2	0.9934	0.9823	94.71%
3	0.9962	0.9890	96.38%
4	0.9976	0.9814	96.52%
5	0.9984	0.9822	96.56%

6.2 Speaker-disjoint held-out test

The evaluation log reports ROC-AUC 0.992780 and EER 0.024008 (2.40%). At the fixed threshold of 0.5, accuracy is 0.9832, precision is 0.9792, recall is 0.9968, and F1 is 0.9879. The confusion matrix is: 2,651 fake recordings correctly predicted as fake; 131 fake recordings predicted as real; 20 real recordings predicted as fake; and 6,164 real recordings correctly predicted as real.

Actual class	Predicted fake	Predicted real	Total
Fake / spoof	2,651	131	2,782
Real / bona fide	20	6,164	6,184
Total	2,671	6,295	8,966

The mean real-class score was 0.9875 for actual real recordings and 0.0541 for actual fake/spoof recordings in this test run. These values indicate strong separation on the reported test split. They do not imply that the model is equally accurate on arbitrary audio from unknown sources.

6.3 External WaveFake benchmark

The preparation script selects 100 LJSpeech utterance identifiers common to seven generator folders: HiFi-GAN, Parallel WaveGAN, Full-Band MelGAN, MelGAN, WaveGlow, Multi-Band MelGAN, and MelGAN Large. For every selected utterance and generator, it pairs the original LJSpeech clip with the corresponding generated clip, giving 700 real-fake pairs and 1,400 scored rows. The same 100 original LJSpeech recordings are reused across the seven generator pairings; therefore, the 700 real observations are repeated instances of only 100 unique source utterances, and the overall micro-average is not a fully independent-sample estimate.

External metric	Reported result
Number of real / fake rows	700 / 700
ROC-AUC	0.66299
Estimated EER	37.79%
Accuracy at threshold 0.5	50.00%
Real-class precision at 0.5	50.00%
Real-class recall at 0.5	100.00%

At the unchanged threshold of 0.5, all 700 generated clips and all 700 original clips were predicted as real. Thus, the thresholded classifier did not reject any fake sample in this external set. The ROC-AUC above 0.5 indicates some ranking information remains in the scores, but the discrimination is weak. The estimated EER of approximately 37.79% is also much worse than the speaker-disjoint test EER. This benchmark therefore reveals a substantial cross-dataset generalization gap.

7. Discussion

The speaker-disjoint test result is encouraging for the particular In-the-Wild split used in the experiment. Separating speakers reduces one prominent source of leakage: the model cannot rely only on recognizing speakers shared between the In-the-Wild training and test partitions. The high validation AUC and high held-out AUC suggest that the learned log-Mel patterns transfer to the held-out speakers in this data distribution.

The WaveFake result materially changes the interpretation. The classifier assigns high real-class scores to both real and generated clips, resulting in a failure at the 0.5 threshold. This could reflect a mismatch in vocoder artifacts, source-corpus characteristics, recording/synthesis pipelines, or score calibration. The external experiment is small and reuses original clips across generators, so it should be treated as a

diagnostic rather than a definitive benchmark ranking. Nonetheless, its failure is strong evidence that the reported in-domain accuracy cannot be generalized to arbitrary synthetic speech.

A threshold chosen on WaveFake itself would not constitute a fair deployment fix unless a separate WaveFake development subset were used and a disjoint external test were preserved. Threshold selection must be separated from final evaluation. Likewise, any future hyperparameter tuning based on the held-out In-the-Wild test set would invalidate its role as an untouched test set.

8. Limitations

The supplied archive does not include raw ASVspoof, In-the-Wild, or WaveFake audio, nor the final manifest CSV files. The reported experiment cannot be reproduced from the archive alone.

The included create_manifests.py script creates generic In-the-Wild manifests without a speaker column and with names different from those expected by the final speaker-disjoint training script. The final run therefore depended on speaker-aware manifests that are absent from the ZIP.

The external WaveFake subset uses 100 LJSpeech originals repeatedly across seven generator pairings. This induces dependence among rows and limits the statistical interpretation of the micro-averaged metrics.

The model uses a small CNN and four-second crops. It may miss partial manipulation, short utterances, noisy channels, compression artifacts, multilingual conditions, and generators outside the training/evaluation data.

The fixed threshold of 0.5 is not calibrated across domains. The external result demonstrates severe threshold and domain mismatch.

No confidence intervals, repeated-seed experiments, statistical significance tests, calibration curves, or comprehensive codec/language stress tests are provided in the supplied artifacts.

The results are transcribed from supplied logs and CSV summaries; the raw data were not available to independently re-run the model during preparation of this report.

9. Conclusion and future work

This report describes a compact audio deepfake detector that converts four-second audio segments into 64-band log-Mel spectrograms and classifies them with a four-block 2-D CNN. The supplied training logs report 46,039 training recordings, weighted sampling over four source-label groups, and selection of an epoch-three checkpoint using validation ROC-AUC. On an 8,966-recording speaker-disjoint In-the-Wild test set, the model reaches 98.32% accuracy and 0.99278 ROC-AUC. However, on a small external WaveFake sample, the model achieves only 0.663 ROC-AUC and predicts every recording as real at threshold 0.5. The broader problem is therefore not solved: the system is a useful baseline for one evaluated distribution, not a robust universal detector.

Future work should restore and version the raw data and speaker-aware manifests; evaluate a larger set of independent synthesis generators; use multiple seeds and confidence intervals; test compression, codec, noise, language, and partial-spoof conditions; evaluate self-supervised speech encoders and stronger anti-spoofing architectures; and calibrate decision thresholds using a development set that is separate from final test data. Per-generator and per-speaker analysis should remain diagnostic and must not leak held-out labels into model selection.

10. Reproducibility and availability statement

The supplied project archive includes Python source files, three checkpoint files, training/evaluation logs, and WaveFake per-file prediction summaries. It does not include the raw datasets or final manifest CSV files. The final training script expects data/manifests/asvspoof_train.csv and speaker-aware files named itw_speaker_train.csv, itw_speaker_val.csv, and itw_speaker_test.csv. The included create_manifests.py script does not generate the speaker-aware manifests required by that final script. A helper and manifest template are included in the accompanying cleaned package to make the expected schema explicit; the user must still supply the dataset and speaker/split annotations. Dataset access and redistribution must follow the applicable dataset licence and terms. The trained checkpoints are included as provided.

Declarations

Funding:

No external funding information was supplied for this project.

Ethics and privacy: The project uses third-party audio datasets. Before public redistribution, verify dataset terms, provenance, privacy conditions, and any restrictions related to public figures or voice recordings. The detector output should not be treated as conclusive forensic evidence.

AI assistance and authorship: some AI tools are used to write the code and texts.

Code availability:

[Insert reviewed GitHub repository URL after approval]. Data availability: original datasets are third-party resources; see the cited dataset pages for access and licence information. Project outputs and checkpoints: Video: https://youtu.be/smH-QzxcTPg

References

1. Todisco M, Wang X, Vestman V, Sahidullah M, Delgado H, Nautsch A et al (2020) ASVspoof 2019: A large-scale public database of synthesized, converted and replayed speech. Comput Speech Lang 64(101114). https://doi.org/10.1016/j.csl.2020.101114

2. Müller NM, Czempin P, Dieckmann F, Froghyar A, Böttinger K (2022) Does Audio Deepfake Detection Generalize? Proceedings of Interspeech 2022. https://arxiv.org/abs/2203.16263
3. Frank J, Schönherr L (2021) WaveFake: A Data Set to Facilitate Audio DeepFake Detection, NeurIPS Datasets and Benchmarks 2021. https://arxiv.org/abs/2111.02813
4. Li M, Ahmadiadli Y, Zhang X-P (2025) A Survey on Speech Deepfake Detection. ACM-CSUR 57. https://doi.org/10.1145/3714458
5. PyTorch BCEWithLogitsLoss, official documentation. https://docs.pytorch.org/docs/stable/generated/torch.nn.BCEWithLogitsLoss.html
6. librosa Mel spectrogram and Mel filter-bank implementation, official documentation/source. https://librosa.org/doc/and https://github.com/librosa/librosa