The Automatic Speaker Verification Spoofing Countermeasures Workshop (ASVspoof 2024)
31 August 2024, Kos, Greece

# IDVoice team System Description for ASVSpoof5 Challenge

*Alexandr Alenin, Andrei Balykin, Esteban Gómez, Rostislav Makarov*
*Pavel Malov, Anton Okhotnikov, Nikita Torgashov, Ivan Yakovlev*

ID R&D Inc.
New York, USA
{alenin, andrew.balykin, esteban.gomez, makarov, pavel.malov,
ohotnikov, torgashov, yakovlev}@idrnd.net

## Abstract

ASVSpoof is a series of community-led challenges aimed at advancing the development of robust automatic speaker verification (ASV) systems and anti-spoofing countermeasures (CM). The fifth edition of the challenge focuses on speech deepfakes and features two tracks: Track 1: Robust Speech Deepfake Detection (DF) and Track 2: Spoofing-Robust Automatic Speaker Verification (SASV). In this report, we describe in detail the system submitted by the IDVoice team to the open condition of the SASV track (Track 2). Our solution is a score-level fusion of independently trained CM and ASV systems. The CM system is composed of six neural networks of four distinct architectures, while the ASV system is a ResNet-based model. Our final submission achieves a **0.1156** min a-DCF on the challenge evaluation set.

## 1. Introduction

Voice Cloning (VC) and Text-To-Speech (TTS) engines made significant progress in the last several years, making it very challenging for the human ear to detect synthetic speech. The ASVspoof5 [1] is an initiative driven by the research community that aims to evaluate and enhance the performance of modern CM and ASV systems against speech deepfakes.

This report presents the IDVoice team system description. We've been focused on the task of spoofing-robust automatic speaker verification (Track 2 SASV) with less-constrained data requirements (open condition). In our work, we utilized modern approaches: subnetworks [2] trained on top of SSL pre-trains and on top of discriminatively pre-trained conventional CNN backbones and new architectures, such as ReDimNet [3]. To combine the scores of independently trained CM systems, we used a linear ensemble with the following score calibration. To obtain a SASV joint score, we combined the CM system fused score with a standardized output of a cosine backend of the ASV system.

The paper is organized as follows: Section 2 describes the architectures of pre-trains we used for CM and ASV models. In Section 3 we present information on training datasets. Section 4 represents our experiment setup and implementation details. And finally, Section 5 and Section 6 highlight our results and findings.

## 2. System Description

As mentioned before, our submission is based on a combination of independently optimized ASV and CM systems, which were trained on various large training datasets with extensive data augmentation strategies applied. Below, we discuss the key components of the architectures we utilized.

### 2.1. Architectures

To enhance the robustness and accuracy of our system, we used several deep neural network architectures, successfully adopted for the ASV and CM problems.

#### *2.1.1. ASV Systems*

Our ASV system is built upon Convolutional Neural Network (CNN) ResNet100 architecture, which was trained on a large YouTube-based dataset. For the implementation details please refer to [4].

#### *2.1.2. CM Systems*

All of our CM systems are small neural networks built on top of a backbone with a subnetwork approach [2], where the backbone is either pre-trained in a supervised manner (e.g. for speaker recognition task) or in a self-supervised (SSL) fashion.

**SSL Transformers**: We utilized multiple SSL models that were pre-trained on the LibriSpeech 960h dataset [5] from the HuggingFace [6] platform:

- `wav2vec2-conformer-rel-pos-large` [7]
- `wav2vec2-large-960h` [8]
- `data2vec-audio-large-960h` [9]

**CNNs**: We also employed 2D CNN models pre-trained for the ASV task, because according to our experiment results they provide orthogonality to SSL models based on raw audio input and show great performance when fused together:

- `ResNet100` [4]
- `ReDimNet-B2` [3]

Both ASV and CM systems were trained independently and then fused to combine the strengths of each approach, ultimately enhancing the overall system performance in detecting spoofed audio samples.

## 3. Datasets

We utilized various datasets for training and development purposes. The datasets are categorized as follows:

### 3.1. CM system

To train our CM systems, we used the following data:

10.21437/ASVspoof.2024-7

*3.1.1. Training Datasets*

- ASVspoof5-Train: Training subset of the ASVSpoof5 challenge.
- ASVSpoof21-Eval [10]: Extensive dataset of various TTS and VC engines, evaluation part from the challenges of the last years. Includes more than 100 different engines released before 2021, and 14 different types of codecs and compressions.
- ASVSpoof15 [11] / ASVSpoof19 [12]: Train part from the challenges of the previous years. Includes old TTS engines from 2015 and 2019.
- Blizzard [13]: The Blizzard dataset contains synthetic speech materials submitted by participants in various Blizzard Challenges held annually since 2008. We utilized the editions of a challenge with non-LibriVox source data such as 2014, 2019-2021.
- DECRO [14]: The DECRO dataset consists of audio samples designed to evaluate the impact of language differences on deepfake detection, featuring both real and fake speech in English and Chinese. The spoofed samples are generated using a mix of commercial and open-source algorithms, including TTS and VC techniques.
- Internal train data: Internally collected TTS/VC data based on source audio from VoxTube [15], VCTK [16], and LibriSpeech data. The dataset is created using a combination of commercial and open-source algorithms, incorporating both TTS and VC techniques, as well as various Vocoder and neural codec systems including EnCodec* [17].

* We applied EnCodec to bonafide audios only and considered them as spoofs.

*3.1.2. Development Datasets*

As a development set, we decided to expand the development set issued by the organizers using such datasets as In The Wild [18], Chinese Fake Audio Detection [19] dataset, and internally collected data.

Afterward, we merged all the dev sets into one and used the resulting dataset to assess the quality of the models and also to optimize the hyperparameters of the models during training.

### 3.2. ASV system

We used the ResNet100 model trained on VoxTube-Large dataset [4] with the only difference in augmentation strategy, where we removed MUSAN Music augmentation and replaced MUSAN Speech corpus as a source of babble noise augmentation with speakers taken from VoxTube-Large dataset.

## 4. Experiments

### 4.1. Data setup

All CM models were trained using the same set of datasets described in 3.1.1. To make the training process and validation metrics stable, we uniformly sampled all VC/TTS/Vocoders engines from each dataset, resulting in distinct 750k training files within a training protocol, which includes 23% of bonafide and 77% of spoofs. All audios were sampled with a uniform probability to form train batches.

### 4.2. Features

As input into our neural networks, we used either spectral features for 2D models or raw signals for SSL-based models. For 2D CNN models, we used mean-normalized Mel filter bank log-energies with a 25 ms frame length, and an FFT size of 512 over 20-7600 Hz frequency limits. For ResNet100 we utilized 96 frequency bins and 10 ms step, while for ReDimNet we used 72 frequency bins and 12.5 ms step.

### 4.3. Augmentations setup

Extensive data augmentation techniques were employed to enhance the robustness of our ASV and CM models:

- **Room Impulse Response (RIR)** [20]: Artificially reverberated a signal via convolution with real RIRs.
- **Noise Addition**: Incorporating noise from Musan [21] (music and noise subsets), DEMAND [22], and DCASE [23] corpus with 0-15 db SNR.
- **Babble**: We randomly picked 3-7 bonafide speakers from the training dataset, summed them together, and then added them to the original signal with 13-20 db SNR to model the babble noise.
- **Mixed sample data augmentation**: CutMix [24] & MixUp [25]
- **RawBoost**: We used RawBoost algorithm #5 [26]
- **Low-pass Filtering**: We applied real-time low-pass filtering (up to 4kHz in frequency range) based on the FFT, with a probability of 30%.
- **SpecAug** [27]: We masked from 0 to 5 frames in the temporal axis and from 0 to 10 frames in the frequency axis using the SpecAug.
- **Codecs**: We applied diverse codec transformations to both bonafide and spoof samples. The transformations used include G722, A-law, Mu-law, Opus, MP3, and OGG. Codecs were applied using ffmpeg [1] library.

### 4.4. Hyperparameters

We had two distinct hyperparameter sets for training 2D CNN-based models and SSL-based models, these hyperparameters were derived from the hyper-parameter optimization algorithm: HPO. Also, we had some hyperparameters fixed across all models training:

- Optimizer: we used AdamW [28] optimizer with default betas & epsilon parameters
- Loss function: AMSoftmax [29] with scale set to 20.0
- Training segment duration: 3.0 seconds
- Num epochs: 1. Model sees each data sample during training only once.

We applied HEBO [30] hyper-parameter tuning algorithm from `ray.tune` [31] library to search in following hyperparameter spaces:

- **Learning rate**: loguniform in $[10^{-4}, 10^{-3}]$
- **Weigth decay**: loguniform in $[10^{-5}, 10^{-3}]$
- Prob of **RawBoost-5** algorithm: uniform in $[0.0, 1.0]$
- Prob of **Noise augmentation**: uniform in $[0.0, 1.0]$

[1] `https://github.com/FFmpeg/FFmpeg`

- Prob of **Reverb augmentation**: uniform in $[0.0, 1.0]$
- Prob of **CutMix** applied to batch: uniform in $[0.0, 1.0]$
- **Margin** of AMSoftmax loss: uniform in $[0.0, 0.2]$

After applying HPO, we got the following values for hyperparameters for 2 model groups presented in Table 1.

| | 2D-CNN subnets | SSL subnets |
|---|---|---|
| batch size * | 80 | 256 |
| learning rate | 0.0006 | 0.001 |
| RawBoost prob | 0.0 | 0.5 |
| weight decay | 0.0001 | 0.0001 |
| noise aug prob | 0.9 | 0.45 |
| reverb aug prob | 0.1 | 0.0 |
| CutMix prob | 0.5 | 0.017 |
| margin | 0.12 | 0.07 |

Table 1: Comparison of Hyperparameters for CNN and SSL models training

* Batch size was defined by GPU memory capacity: we picked the maximum batch size, that fits a single GPU for each model.

### 4.5. Fusion description

Firstly, a linear combination of 6 CM models was used to obtain a single CM system output. To determine the optimal weight of each model in fusion act-DCF optimization on an extended development set was used. Then, joint ASV and CM systems output was obtained as a minimum value of CM fusion system output and a standardized output of a cosine similarity backend of ASV system: $min(Score_{CM}, Score_{ASV})$. For optimization, we used a COBYLA optimizer [32] toolkit.

## 5. Results

### 5.1. Codecs impact on CM system

For clarity reasons, we provide our results analysis in a form where we split the impact of traditional codecs and neural codecs. From the perspective of performance on conventional codecs, our CM system exhibited the highest error on the 8k and 16k Speex codecs, as these codecs were not included in the training augmentations strategy. Similarly, the Opus 8k codec also demonstrated higher error, as only the 16k version of this codec was used during training. Complete results of our CM system on the 16k and 8k codecs are shown in Table 2.

Table 2: MinDCF and EER values for the CM system

| | MinDCF | EER, % |
|---|---|---|
| **No codec** | 0.0023 | 0.11 |
| **All codecs** | 0.1770 | 6.26 |
| **All codecs ex 4,7** | 0.0776 | 2.76 |
| **Codecs 4,7** | 0.6242 | 22.03 |
| **16k codecs** | 0.2170 | 7.66 |
| **16k codecs ex 4,7** | 0.0541 | 1.92 |
| **8k codecs** | 0.1070 | 3.82 |

Considering neural codec performance, it is important to note that EnCodec utilizes a GAN-like decoder with losses similar to HiFi-GAN-style vocoders. Currently, the primary application of these neural codecs is as a frontend system for LLM-based TTS and VC synthesizers, which, in the context of this challenge, are categorized as spoofs. Given this context, we observe a significant degradation in the CM system's performance on EnCodec (codec-4) and MP3+EnCodec (codec-7). This degradation can be explained by the fact that the challenge evaluation dataset includes the use of EnCodec encoding and decoding without changing the label of the audio sample. In contrast, during our training, we changed the labels of bonafide samples to a spoof-only label after applying such processing.

### 5.2. SASV fusion metrics

Table 3 illustrates the minimum a-DCF [33] values for the fused system under different codec conditions. The system performed optimally without the influence of any codec. The inclusion of all codecs resulted in reduced performance, with 16k codecs causing a minor degradation and 8k codecs having the most significant impact.

Table 3: Min a-DCF values for the Fusion System

| | Min a-DCF |
|---|---|
| **No codecs** | 0.0504 |
| **All codecs** | 0.1184 |
| **16k codecs** | 0.1030 |
| **8k codecs** | 0.1300 |

These results highlight the influence of codec conditions on the system's performance and its robustness in varying audio quality scenarios. Our system achieved the following performance metrics in the evaluation phase:

- **Minimum a-DCF**: 0.1156
- **t-EER**: 4.32%
- **Minimum t-DCF**: 0.4584

## 6. Conclusions

In this report, we presented our solution for the open condition of the SASV track of the ASVSpoof5 challenge. We showed again the effectiveness of the subnetwork approach for the detection of spoofed audio and analyzed the contribution of each codec to system accuracy. Besides, we can see ReDimNet applicability for the CM task. We found out that 8kHz codecs led to noticeable accuracy degradation, while the neural codec resulted in an even greater decrease in performance. On top of that, the detailed results show us stable errors across different types of attacks (engines A17-A32) of our SASV system in a no-codecs scenario with MaryTTS being the hardest type of attack when codecs are applied.

## 7. References

[1] Xin Wang, Héctor Delgado, Hemlata Tak, Jee-weon Jung, Hye-jin Shim, Massimiliano Todisco, Ivan Kukanov, Xuechen Liu, Md Sahidullah, Tomi Kinnunen, Nicholas Evans, Kong Aik Lee, and Junichi Yamagishi, "ASVspoof 5: Crowdsourced speech data, deepfakes, and adversarial attacks at scale," in *ASVspoof Workshop 2024 (accepted)*, 2024.

[2] Alexander Alenin, Nikita Torgashov, Anton Okhotnikov, Rostislav Makarov, and Ivan Yakovlev, "A Subnetwork Approach for Spoofing Aware Speaker Verification," in *Proc. Interspeech 2022*, 2022, pp. 2888–2892.

[3] Ivan Yakovlev, Rostislav Makarov, Andrei Balykin, Pavel Malov, Anton Okhotnikov, and Nikita Torgashov, "Reshape dimensions network for speaker recognition," *arXiv preprint arXiv:2407.18223*, 2024.

[4] Nikita Torgashov, Rostislav Makarov, Ivan Yakovlev, Pavel Malov, Andrei Balykin, and Anton Okhotnikov, "The id r&d voxceleb speaker recognition challenge 2023 system description," *arXiv preprint arXiv:2308.08294*, 2023.

[5] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2015, pp. 5206–5210.

[6] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al., "Huggingface's transformers: State-of-the-art natural language processing," *arXiv preprint arXiv:1910.03771*, 2019.

[7] Changhan Wang, Yun Tang, Xutai Ma, Anne Wu, Sravya Popuri, Dmytro Okhonko, and Juan Pino, "Fairseq s2t: Fast speech-to-text modeling with fairseq," *arXiv preprint arXiv:2010.05171*, 2020.

[8] Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," *Advances in neural information processing systems*, vol. 33, pp. 12449–12460, 2020.

[9] Alexei Baevski, Wei-Ning Hsu, Qiantong Xu, Arun Babu, Jiatao Gu, and Michael Auli, "Data2vec: A general framework for self-supervised learning in speech, vision and language," in *International Conference on Machine Learning*. PMLR, 2022, pp. 1298–1312.

[10] Junichi Yamagishi, Xin Wang, Massimiliano Todisco, Md Sahidullah, Jose Patino, Andreas Nautsch, Xuechen Liu, Kong Aik Lee, Tomi Kinnunen, Nicholas Evans, et al., "Asvspoof 2021: accelerating progress in spoofed and deepfake speech detection," in *ASVspoof 2021 Workshop-Automatic Speaker Verification and Spoofing Coutermeasures Challenge*, 2021.

[11] Zhizheng Wu, Tomi Kinnunen, Nicholas Evans, Junichi Yamagishi, Cemal Hanilçi, Md. Sahidullah, and Aleksandr Sizov, "ASVspoof 2015: the first automatic speaker verification spoofing and countermeasures challenge," in *Proc. Interspeech 2015*, 2015, pp. 2037–2041.

[12] Xin Wang, Junichi Yamagishi, Massimiliano Todisco, Héctor Delgado, Andreas Nautsch, Nicholas Evans, Md Sahidullah, Ville Vestman, Tomi Kinnunen, Kong Aik Lee, et al., "Asvspoof 2019: A large-scale public database of synthesized, converted and replayed speech," *Computer Speech & Language*, vol. 64, pp. 101114, 2020.

[13] Zhen-Hua Ling, Xiao Zhou, and Simon King, "The blizzard challenge 2021," in *Proc. Blizzard Challenge Workshop*, 2021.

[14] Zhongjie Ba, Qing Wen, Peng Cheng, Yuwei Wang, Feng Lin, Li Lu, and Zhenguang Liu, "Transferring audio deepfake detection capability across languages," in *Proceedings of the ACM Web Conference 2023*, 2023, pp. 2033–2044.

[15] Ivan Yakovlev, Anton Okhotnikov, Nikita Torgashov, Rostislav Makarov, Yuri Voevodin, and Konstantin Simonchik, "Voxtube: a multilingual speaker recognition dataset," in *Proc. Interspeech*, 2023, pp. 2238–2242.

[16] Christophe Veaux, Junichi Yamagishi, and Simon King, "The voice bank corpus: Design, collection and data analysis of a large regional accent speech database," in *2013 international conference oriental COCOSDA (O-COCOSDA/CASLRE)*. IEEE, 2013, pp. 1–4.

[17] Alexandre Défossez, Jade Copet, Gabriel Synnaeve, and Yossi Adi, "High fidelity neural audio compression," *arXiv preprint arXiv:2210.13438*, 2022.

[18] Nicolas M Müller, Pavel Czempin, Franziska Dieckmann, Adam Froghyar, and Konstantin Böttinger, "Does audio deepfake detection generalize?," *Interspeech*, 2022.

[19] Haoxin Ma, Jiangyan Yi, Chenglong Wang, Xinrui Yan, Jianhua Tao, Tao Wang, Shiming Wang, and Ruibo Fu, "Cfad: A chinese dataset for fake audio detection," *Speech Communication*, p. 103122, 2024.

[20] Igor Szöke, Miroslav Skácel, Ladislav Mošner, Jakub Paliesek, and Jan Černockỳ, "Building and evaluation of a real room impulse response dataset," *IEEE Journal of Selected Topics in Signal Processing*, vol. 13, no. 4, pp. 863–876, 2019.

[21] David Snyder, Guoguo Chen, and Daniel Povey, "Musan: A music, speech, and noise corpus," *arXiv preprint arXiv:1510.08484*, 2015.

[22] Joachim Thiemann, Nobutaka Ito, and Emmanuel Vincent, "The diverse environments multi-channel acoustic noise database (demand): A database of multichannel environmental noise recordings," in *Proceedings of Meetings on Acoustics*. AIP Publishing, 2013, vol. 19.

[23] Annamaria Mesaros, Toni Heittola, and Tuomas Virtanen, "A multi-device dataset for urban acoustic scene classification," *arXiv preprint arXiv:1807.09840*, 2018.

[24] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo, "Cutmix: Regularization strategy to train strong classifiers with localizable features," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 6023–6032.

[25] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz, "mixup: Beyond empirical risk minimization," *arXiv preprint arXiv:1710.09412*, 2017.

[26] Hemlata Tak, Madhu Kamble, Jose Patino, Massimiliano Todisco, and Nicholas Evans, "Rawboost: A raw data boosting and augmentation method applied to automatic speaker verification anti-spoofing," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 6382–6386.

[27] Daniel S Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D Cubuk, and Quoc V Le, "Specaugment: A simple data augmentation method for automatic speech recognition," *arXiv preprint arXiv:1904.08779*, 2019.

[28] Ilya Loshchilov and Frank Hutter, "Decoupled weight decay regularization," *arXiv preprint arXiv:1711.05101*, 2017.

[29] Feng Wang, Jian Cheng, Weiyang Liu, and Haijun Liu, “Additive margin softmax for face verification,” *IEEE Signal Processing Letters*, vol. 25, no. 7, pp. 926–930, July 2018.

[30] Alexander Cowen-Rivers, Wenlong Lyu, Rasul Tutunov, Zhi Wang, Antoine Grosnit, Ryan-Rhys Griffiths, Alexandre Maravel, Jianye Hao, Jun Wang, Jan Peters, and Haitham Bou Ammar, “Hebo: Pushing the limits of sample-efficient hyperparameter optimisation,” *Journal of Artificial Intelligence Research*, vol. 74, 07 2022.

[31] Richard Liaw, Eric Liang, Robert Nishihara, Philipp Moritz, Joseph E Gonzalez, and Ion Stoica, “Tune: A research platform for distributed model selection and training,” *arXiv preprint arXiv:1807.05118*, 2018.

[32] Michael JD Powell, “A view of algorithms for optimization without derivatives,” *Mathematics Today-Bulletin of the Institute of Mathematics and its Applications*, vol. 43, no. 5, pp. 170–174, 2007.

[33] Hye-jin Shim, Jee-weon Jung, Tomi Kinnunen, Nicholas Evans, Jean-Francois Bonastre, and Itshak Lapidot, “a-dcf: an architecture agnostic metric with application to spoofing-robust speaker verification,” *arXiv preprint arXiv:2403.01355*, 2024.