Evolutive Network Architecture for Speech Deepfake Detection

Wanying Ge

Digital Security Department, EURECOM, France

Sept. 2nd, 2021

Introduction

- **Speech anti-spoofing**

  - To distinguish between human speech and replayed/synthetic speech

- **Problem**

  - Hand-crafted model architectures require lots of human effort

- **We try to**

  - Explore automatic approaches to learn the network architecture

Our works

- **PC-DARTS for anti-spoofing (INTERSPEECH2021)**

  - Architecture search with LFCC feature

- **Raw PC-DARTS (ASVspoof 2021 Workshop)**

  - Architecture search with Raw waveform

Related Works

- **NEAT - NeuroEvolution of Augmenting Topologies [1]**

  - We tried performing NEAT on raw audio waveform [2], to generate the network architecture automatically. But it's slow & no good results.

- **DARTS - Differentiable ARchiTecture Search [3]**

  - Inspired by the successful DARTS applications to speech task[4, 5], we turn our focus to Neural Architecture Search (NAS) algorithms, that instead of completely building the network & connections, the structure and inside connections are selected from a fix set of convolutional operations, which are proved relatively faster & good learning ability

[1] Stanley, Kenneth O., and Risto Miikkulainen. "Evolving neural networks through augmenting topologies." Evolutionary computation 10.2 (2002): 99-127.
[2] Valenti, Giacomo, et al. "An end-to-end spoofing countermeasure for automatic speaker verification using evolving recurrent neural networks." Odyssey. 2018.
[3] Liu, Hanxiao, et al. "Darts: Differentiable architecture search." in Proc. ICML 2019.
[4] Mo, Tong, et al. "Neural architecture search for keyword spotting." Proc. Interspeech 2020.
[5] Ding, Shaojin, et al. "Autospeech: Neural architecture search for speaker recognition." Proc. Interspeech 2020.

Differentiable Architecture Search[3]

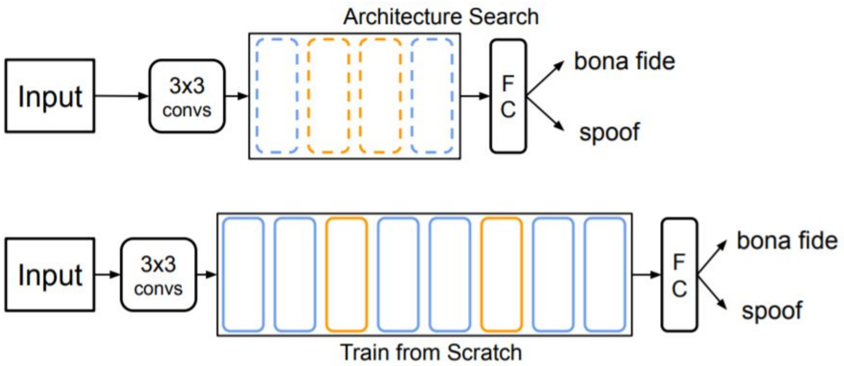


Figure 1. An illustration of architecture search stage and train from scratch stage

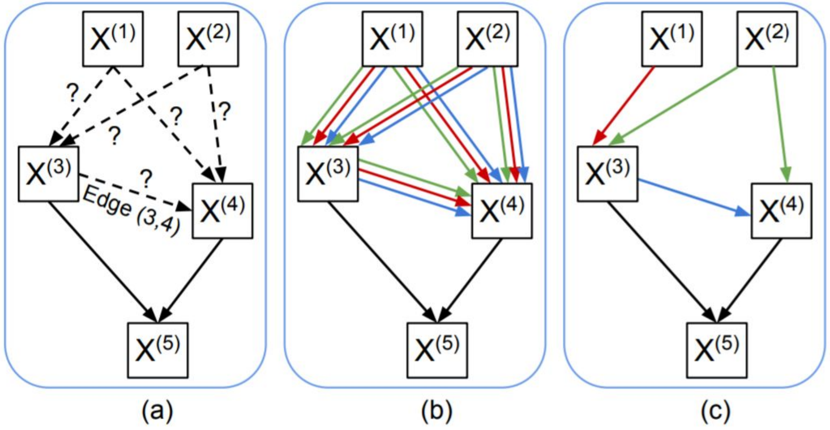


Figure 2. An illustration of cell during architecture search

- **Cells** are stacked in both stages
- **Node X^n** (feature map) are connected with **operations** in the search space

- Operations are assigned with **learnable** weights
- Weights are **optimised** during searching
- But searching is **computationally demanding**

[3] H. Liu, et al., "DARTS: Differentiable Architecture Search," in Proc. ICML 2019.

Partial channel connections[6]

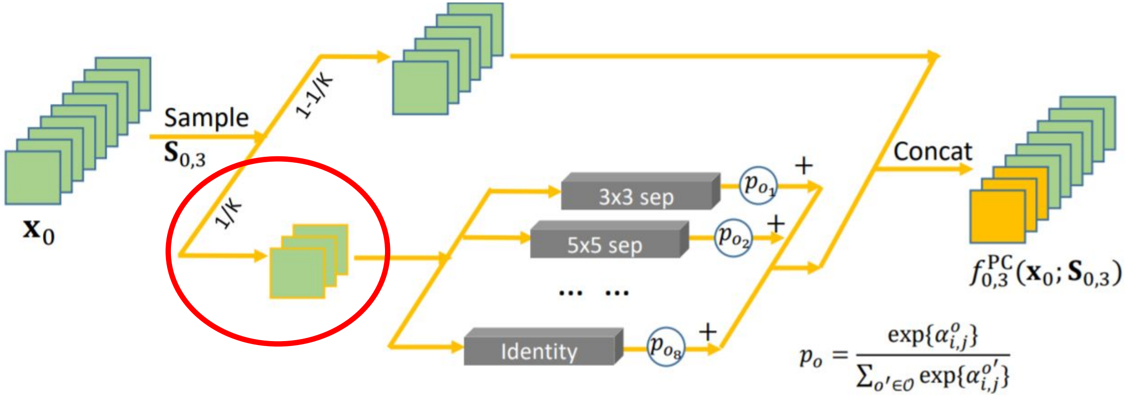


Figure 3. 1/k of the channels are selected and the others are left and stay unchanged

[6] Y. Xu, et al., "PC-DARTS: Partial channel connections for memory-efficient architecture search," in ICLR 2020.

Results

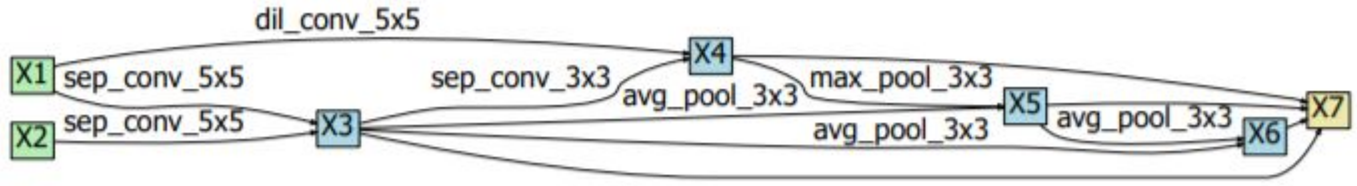
Table 1. Comparison between original DARTS and PC-DARTS

Model size	Systems	Search Cost GPU-days	Best Architecture Train Acc	Dev Acc
$(L=4,$	DARTS	0.29	98.80	97.21
$C=16)$	PC-DARTS	0.15	99.97	100

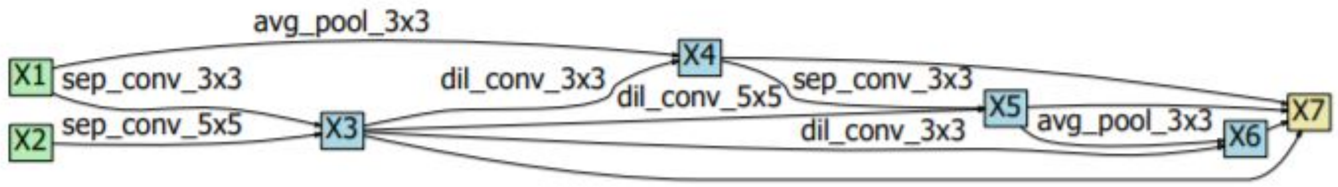
Table 2. Results on ASVspoof2019 LA database

Systems	Features	min-tDCF	EER	Params
Res2Net [26]	CQT	0.0743	2.50	0.96M
Res2Net [26]	LFCC	0.0786	2.87	0.96M
PC-DARTS $(16, 64)$	LFCC	0.0914	4.96	7.51M
PC-DARTS $(4, 16)$	LFCC	0.0992	5.53	0.14M
LCNN [27] [28]	LFCC	0.1000	5.06	10M
LCNN [27] [28]	LPS	0.1028	4.53	10M
LFCC-GMM [25]	LFCC	0.2116	8.09	-
Res2Net [26]	LPS	0.2237	8.78	0.96M
CQCC-GMM [25]	CQCC	0.2366	9.57	-
Deep Res-Net [29]	LPS	0.2741	9.68	0.31M

Searched cells



(a) *Normal cell*



(b) *Reduction cell*

Summary

- **Automatically** searching for the network architecture for speech spoofing detection
- Partial channel connection helps to reduce **memory cost** and improve **efficiency**
- Achieved **competitive** performance against other **hand-crafted** deep neural networks

From LFCC to Waveform

- **Input features**

  - Mostly, time-frequency (T-F) representations, like CQCC, LFCC.

- **Problem**

  - T-F calculations will lose part of the input information

  - Same model architecture trained on different features obtain different result

- **We try to**

  - Directly fed audio waveform to the network

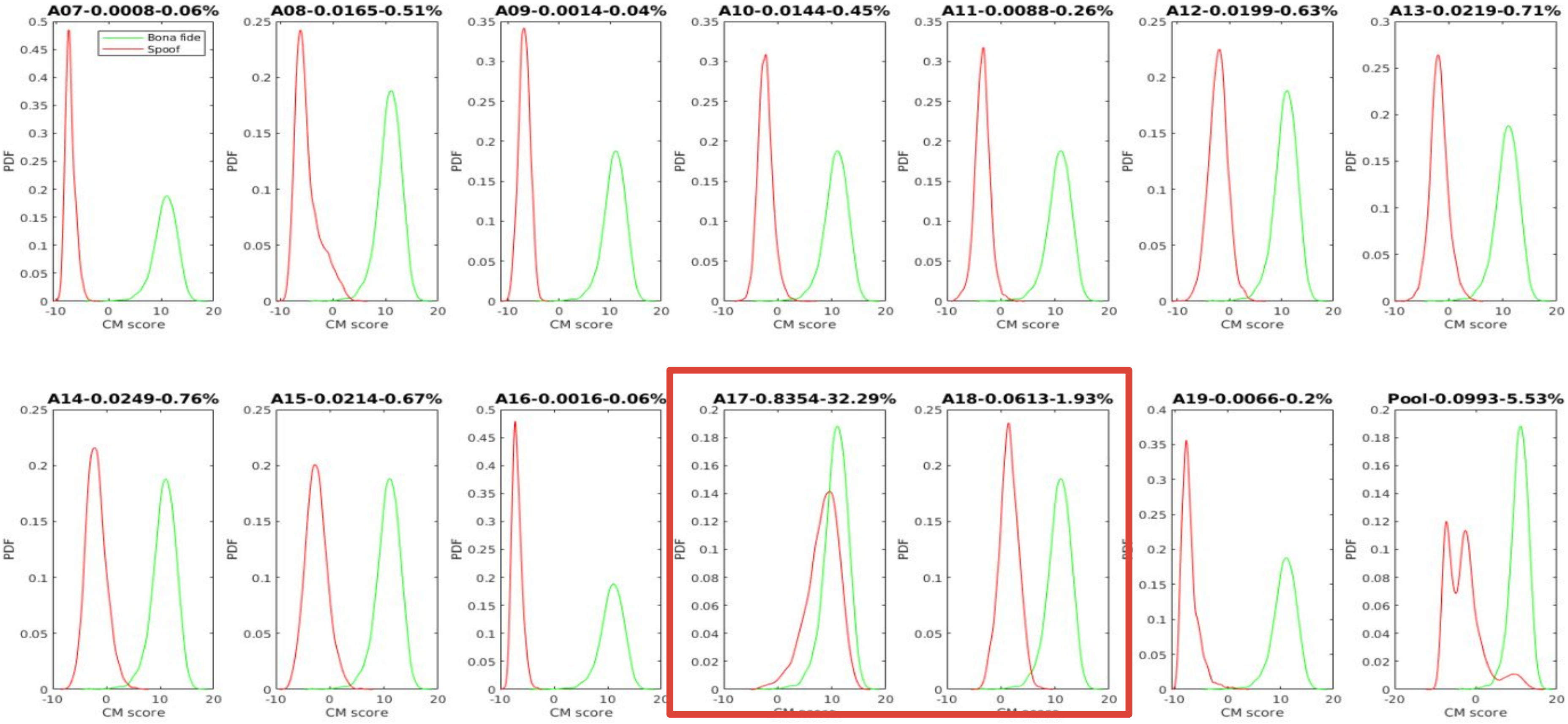
Modifications

	Input	Operations	Pre-processing	Classifier
T-F Feature	**2D matrix**	**Conv2D**	**2 Conv layers**	**FC**
Raw signal	**1D vector**	**Conv1D**	**Sinc layer**	**GRU + FC**

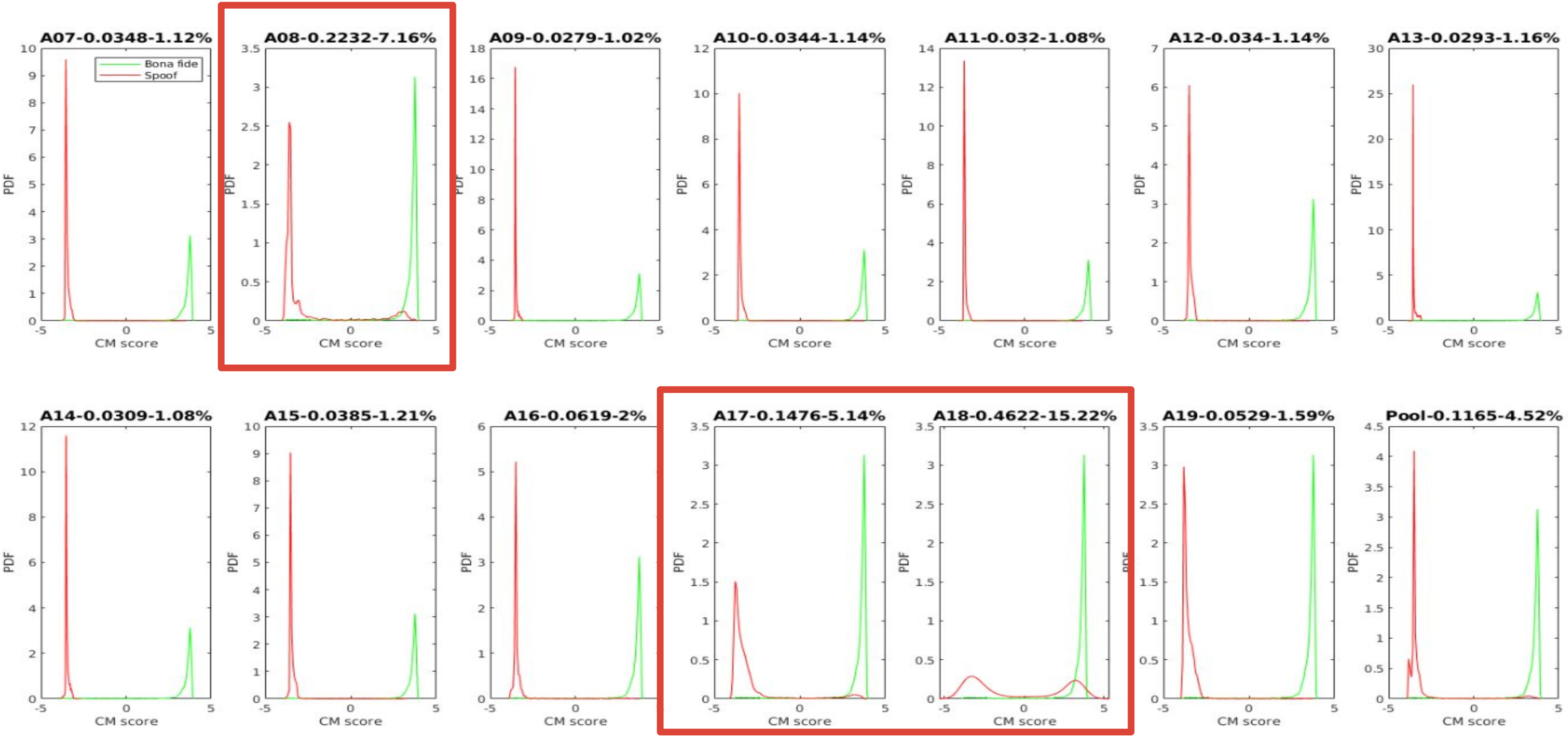
Results

	Fixed		Learnable	
Type	**min-tDCF**	**EER**	**min-tDCF**	**EER**
Mel	0.0517	1.77	0.0899	3.62
Inverse-Mel	0.0700	3.25	0.0655	2.80
Linear	0.0926	3.29	0.0583	2.10
Conv_0	×	×	0.0733	2.49

Score distribution - LFCC



Score distribution - Raw waveform



Thanks