Subband modeling for spoofing detection in automatic speaker verification

Bhusan Chettri[1,2], Tomi Kinnunen[1], Emmanouil Benetos[2]

[1]School of Computing, University of Eastern Finland, FI-80101, Joensuu, Finland
[2]School of EECS, Queen Mary University of London, United Kingdom

Abstract

Spectrograms — time-frequency representations of audio signals — have found widespread use in neural network-based spoofing detection. While deep models are trained on the full-band spectrum of the signal, we argue that not all frequency bands are useful for these tasks. In this paper, we systematically investigate the impact of different subbands and their importance on replay spoofing detection on two benchmark datasets: ASVspoof 2017 v2.0 and ASVspoof 2019 PA. We propose a joint subband modelling framework that employs n different sub-networks to learn subband specific features. These are later combined and passed to a classifier and the whole network weights are updated during training. Our findings on the ASVspoof 2017 dataset suggest that the most discriminative information appears to be in the first and the last 1 kHz frequency bands, and the joint model trained on these two subbands shows the best performance outperforming the baselines by a large margin. However, these findings do not generalise on the ASVspoof 2019 PA dataset. This suggests that the datasets available for training these models do not reflect real world replay conditions suggesting a need for careful design of datasets for training replay spoofing countermeasures.

1. Introduction

Automatic speaker verification (ASV) [1] systems, similar to other biometric modalities, are prone to being intentionally fooled using *spoofing attacks* [2] (or *presentation attacks* [3]), such as replay, text-to-speech (TTS), and voice conversion (VC). High-stakes ASV applications demand trustworthy fail-safe mechanisms (countermeasures) against such attacks. Here, a *countermeasure* (CM) is defined as a binary classifier that aims at discriminating *bonafide* (human speech) utterances from spoofing attacks. To allow maximum re-usability across different applications, the ideal CM should generalize across environments, speakers, languages, channels, and attacks. In practice, this is not the case; CMs are prone to overfitting. This could be due to variations within the spoof class (*e.g.* speech synthesizers not present in the training set), within the bonafide class (*e.g.* due to content and speaker), or extrinsic nuisance factors (*e.g.* background noise).

We focus on **feature extraction** for audio spoofing attack detection. There is a vast body of prior research on developing and enhancing different low-level feature extractors (most relevant work is reviewed in Section 2), some of them obtaining very low spoof-bonafide detection error rates (even 0%) on specific datasets. Many of these techniques leverage domain
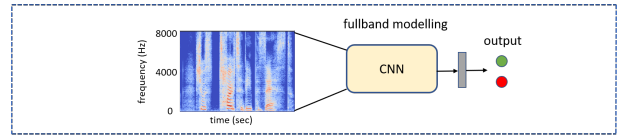
(a) Baseline. Traditional CNN model trained on the fullband spectrum.
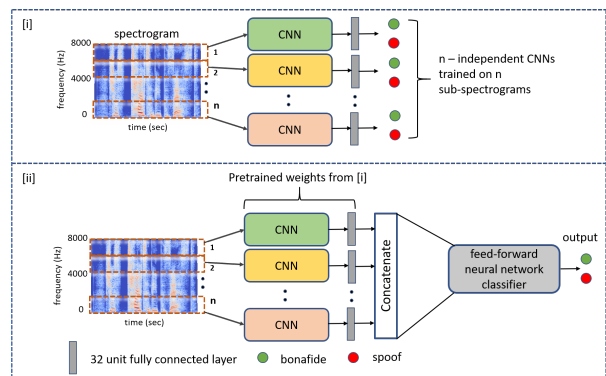
(b) Proposed framework. (i) The original spectrogram is split into n sub-spectrograms on which n independent CNNs are trained. (ii) uses pretrained weights from (i) to initialise n subband CNNs and the whole network weights are updated during training.

Figure 1: Proposed subband CNN modeling framework.

knowledge, whether speech science (speech production or perception), signal processing theory, or both. A potential benefit of such rationale is transparency and interpretability. At the same time, feature extractors crafted with the aid of domain knowledge might be too simplistic. As illustrated in Figure 1 (b), we aim at hitting a suitable balance between hand-crafted and data-driven feature extraction: we use **spectrograms** (a meaningful representation of audio), processed in disjoint **subbands** (to divide-and-conquer high-dimensional spectrogram modeling across several, frequency-localized models, each handling a lower-dimensional feature space), each modeled with a **convolutional neural network** to learn band-specific features. The subband-specific features are concatenated to form feature vectors that are then classified with a feedforward neural network.

The general idea of processing a power spectrogram in subbands, as such, is not new in the speech field. Mel-frequency cepstral coefficients (MFCCs) are extracted using a filterbank consisting of frequency-localized subbands and subband-based modeling of speaker traits dates at least two decades back [4]. In conventional, or *fullband* models, one trains a single model (with a large number of parameters) using a descriptor of the fullband spectrum. In *subband* based models, the rationale is to instead divide-and-conquer the task across independently mod-

eled subbands which are later recombined using feature or score fusion techniques. The potential benefits include the possibility to side-step the 'curse of dimensionality' by using a set of models trained on lower-dimensional inputs, to achieve robustness to *frequency-selective noise*, and the possibility to analyze the importance (contribution) of each subband to the classification results. To this end, we summarise the main contributions of our work below.

- We perform a systematic study on different subbands and their contribution in replay spoofing detection using convolutional neural networks (CNNs).

- We propose a CNN-based subband modelling framework (Figure 1 (b)) that offers substantial improvement over traditional fullband modelling (Section 5).

- We study the effect of late fusion of subband CNNs using linear and weighted linear sum fusion approaches.

- Finally, we study the generalisability of all our replay spoofing countermeasures on the ASVspoof 2019 "real" PA test dataset.

2. Relation to prior work

Since the release of publicly available benchmark datasets [5, 6, 7], there is a substantial body of prior research on various types of spoofing countermeasures; refer to [8] for a recent survey. The front-end processing part of spoofing countermeasures relies heavily on hand-crafted signal processing techniques. Besides the now-standard *Constant Q cepstral coefficients* (CQCCs) [9] (and extensions, e.g. [10]), there are methods leveraging from modulation domain processing (e.g. [11, 12, 13]), adaptive filterbanks [14], restricted Boltzmann machines [15], and envelopes of subband signals [16] to name a few. In this work, we focus on modeling spectral subbands obtained through a standard windowed discrete Fourier transform.

There are some research works investigating the impact of subbands on model performance in different tasks. For instance, the authors in [4] used a subband approach to extract relevant features, each modeled using Gaussians. Authors in [17] investigated dependencies of different frequency bands and speaker characteristics in a speech signal for speaker verification applications. Recently, authors in [18] demonstrated improved performance using different subbands for building acoustic scene classification models. They use a *sub-spectrogram*, obtained by cropping a mel-spectrogram at different bands, to train a convolutional neural network (CNN) for learning band-specific features. This is closely related to our work. One notable difference, however, is that we do not use overlapping bands. Furthermore, this approach has never been applied for spoofing detection tasks.

In the context of spoofing detection, the most relevant studies include [19, 20, 21, 22, 23, 24]. The authors of [19] investigated different subbands to find the most informative bands useful for spoofing detection tasks. Using the Kullback-Leibler divergence (at model-level) and classification-level analysis, they identified 0-1 kHz, 2.5-5.5 kHz and 7-8 kHz as the most informative subbands on their dataset (SAS corpus). Features were then extracted from these bands to train a classifier (*Gaussian mixture model — universal background model*), demonstrating improved performance over traditional fullband models. The authors of [20] investigated the importance of different subbands for spoofing detection on the ASVspoof 2017 dataset. They extracted five different types of features from these subbands to train a GMM back-end classifier. They found the

high frequency range of 6-8 kHz to be the most informative. The authors of [21] similarly performed subband analysis using CQCC and MFCC features for replay spoofing detection on the ASVspoof 2017 dataset, with similar findings to those in [20]. The high frequency band (6-8 kHz) provides particularly relevant information for replay attack detection on this dataset. The authors in [22] proposed *high-frequency cepstral coefficient* (HFCC) features extracted from a high-frequency spectrum (above 3.5 kHz). They combined HFCCs with CQCCs and trained a deep neural network as a feature extractor. A support vector machine classifier was trained on deep features, outperforming the baseline GMM models on the ASVspoof 2017 dataset. The authors of [24] trained a subband autoencoder (SBAE) for feature extraction by restricting connections between units in the input and the first hidden layer of the encoder. Doing so allows the model to learn band-specific features useful in spoofing detection, demonstrating substantial gain in detection performance on the ASVspoof 2015 dataset. Another line of study that investigates subbands for spoofing detection is [23]. Using the subbands that provide discriminative information, they design new filters for feature extraction. Experimental results on the ASVspoof 2017 v1.0 dataset indicate that the 0-1 and 7-8 kHz subbands offer the most discriminative information.

To sum up, previous studies indicate that certain frequency subbands are potentially more informative to the detection of spoofing attacks, even though no standardized approach how that unevenly distributed information across the frequency axis should be utilized. Our current work is different from prior works [19, 20, 21, 22, 23] because most of them aim at handcrafting or learning features [24] based on the relevance of specific subbands for spoofing detection. To the best of our knowledge, there is no work in spoofing detection aiming to learn band-specific features by discriminatively training CNNs on a spectrogram input. This is one of the objectives of this paper.

3. Proposed methodology

Convolutional neural network (CNN) based countermeasure models trained using spectrograms have shown state-of-the-art performance in spoofing detection tasks in the ASVspoof 2017 challenge. They are usually trained using the fullband spectrum of the input signal and use a fixed-duration input representation [25]. This *conventional approach* of building CNN-based countermeasures is illustrated in Figure 1 (a). As the CNN is trained discriminatively, it is forced to learn discriminative features using the entire frequency spectrum of the input signal, using a *single* worker to extract usable information across all the frequency subbands for spoofing attack detection.

But as the prior studies (Section 2) suggest, not all the subbands are necessarily equally informative. From a modeling perspective, the raw spectrogram patch (extracted by stacking multiple frames using all the frequency bands) is a high-dimensional vector, with strong correlations between any neighboring time (frame) or frequency (DFT bin) indices. As such low-level redundancy is common to both human and spoofed samples, it does not necessarily help in the discrimination (classification) task itself; instead, the model will have to learn both data compression (suppressing statistical redundancy to a useful intermediate representation) *and* classification tasks. This may also result in additional computational time during convolution operations.

Therefore, rather than having a single CNN that merges information across different frequency bands, we propose to in-

corporate a *bank* of n different CNNs, each operating on non-overlapping n frequency subbands. Our proposed methodology is illustrated in Figure 1 (b). Each of the *subband CNNs* now has to model a much lower-dimensional subspace, producing a less redundant and more relevant representation of its respective subband. Note that the subband representations are afterwards re-combined through concatenation. This new representation now contains again information across the full frequency band, allowing any subsequent model to exploit possibly useful band-level correlations. In our case, we use a simple feedforward neural network (FFNN) for the final classification. Natural questions that arise now are *how to perform the frequency-domain split* and *how to choose n*. We address three different forms of splits, $n = 2$, $n = 4$ and $n = 8$, and go for the easiest choice of uniform frequency division.

This choice is motivated from [23]. They divide the original spectrogram into n uniform subbands with bandwidth 1 kHz corresponding to $n = 8$ splits and 0.5 kHz bandwidth for $n = 16$ splits. They remove one subband at a time and handcraft features from the remaining subbands. GMMs are then trained on these features for spoofing detection and the performance is evaluated in terms of EER on the ASVspoof 2017 evaluation sets. As our dataset consists of 16 kHz audio (Nyquist range 8 kHz), our three choices correspond to subbands of bandwidths 4 kHz ($n = 2$), 2 kHz ($n = 4$) and 1 kHz ($n = 8$). It should be noted that the default case $n = 1$ corresponds to the baseline CNN (Figure 1 a), i.e the model trained on the fullband spectrogram. From hereon we use "CNN" to refer to the baseline CNN. And, we use "sub-CNN" to refer to models trained on the subband spectrograms.

We operate on power spectrograms instead of other alternative time-frequency representations, following findings in [25]. All our sub-CNNs use the architecture described in [26], which is an adapted version of the best performing model [25] in the ASVspoof 2017 challenge. It consists of 9 convolutional layers, 5 max-pooling layers and 2 fully connected (FC) layers. The key difference while training such sub-CNNs is in terms of the input they receive. The bandwidth of the input sub-spectrogram varies depending upon different values of n (number of splits). Section 4 provides more details regarding input representations, training and testing of these models.

The proposed joint sub-CNN model of Figure 1 (b) second row uses the same architecture as in sub-CNNs (first row of the same figure) with the following updates: (1) there is no output layer now, and (2) a concatenation layer is added that merges the fully-connected (FC) layer output from n sub-CNN models producing a $32 \times n$ dimensional vector. It should be noted that the choice of 32 units in the FC layer comes from the baseline CNN of Figure 1 (a). Furthermore, this architecture with 32 FC units has shown promising results as described in [26]. Next, the concatenated vector is fed to a feedforward neural network (FFNN) for class discrimination. The FFNN consists of two fully-connected layers with 256 and 128 units. This is followed by a single unit output layer with sigmoid non linearity for class discrimination. We apply batch normalisation before applying ReLU non-linearity to these layers. The architecture of the FFNN is optimised through model validation (on the development set). Training and optimisation of our proposed framework is done in two steps:

- First, the input spectrogram is split into n non-overlapping sub-spectrograms and n sub-CNNs are trained independently on them. The training dataset is used for model training and the development dataset is

used for model validation. This step is depicted in the top row of Figure 1(b).

- Second, the pretrained sub-CNNs (excluding the last layer) are used to initialise the weights of the sub-CNN modules of our joint sub-CNN framework shown in the bottom row of Figure 1 (b). The weights of the FFNN layers are initialised randomly using xavier initialization [27]. The biases are initialised to zero. Given an input spectrogram, the framework first splits it into n non-overlapping sub-spectrograms which are processed by n sub-CNNs and the whole network parameters are jointly updated during backpropagation. This step can be interpreted as fine-tuning of the subband CNNs and the classifier back-end jointly for best performance. As in the earlier step, model parameters are trained on the training dataset and the development set is used for model validation.

Our proposed work is different from prior works [19, 20, 21, 22, 23] because most of them aim at hand-crafting or learning features [24] based on the relevance of specific subbands for spoofing detection. To the best of our knowledge, there is no work in spoofing detection aiming to learn band-specific features by discriminatively training CNNs on a spectrogram input.

4. Experimental setup

Here we describe our experimental setup including the details of dataset used, model architecture, training and scoring procedures and also the evaluation measures.

4.1. Dataset

We use two publicly available spoofing datasets, **ASVspoof 2017 v2.0** [28] and **ASVspoof 2019 physical access** (PA), [7] for model training and testing. In addition, we also include results on the recently released **ASVspoof2019 real PA** dataset[1] for the challenging case of cross-database performance evaluation. All the datasets are representative of *replay attacks* and are complementary to each other.

The ASVspoof 2017 v2.0 dataset consists of bonafide and spoof class audio recordings grouped into three subsets: training, development and evaluation [6, 28]. It contains male speakers only: 10, 8 and 24 in the training, development and evaluation sets, respectively. The training set contains a balanced set of audio files, 1507 for both bonafide and spoof classes. The development set has 760 bonafide and 950 spoof files. The evaluation set has 1298 bonafide and 12008 spoof audio files [28].

The ASVspoof 2019 PA dataset, in turn, consists of 8 male and 12 female speakers in the training and development subsets. Both the training and development sets have 5,400 bonafide utterances, while 48,600 and 24,300 spoofed utterances are included in the training and development sets, respectively. The evaluation set has 135,000 test utterances [7, 29]. Following our prior findings [30] on the ASVspoof 2019 dataset, we adopt a custom, but publicly available protocol[2].

While the ASVspoof 2019 PA dataset was created using *simulated* replay attacks, the ASVspoof2019 real PA dataset consists of audio recordings developed under real replay conditions. We use this last data to evaluate our models (trained using ASVspoof 2017 and 2019 PA data) to gauge cross-dataset

---

[1] https://www.asvspoof.org/database
[2] https://github.com/BhusanChettri/ASVspoof2019

performance. The real PA dataset consists of 2700 audio files with 540 bonafide and 2160 spoof recordings [7].

4.2. Input representation and preprocessing

The input to the network is a mean-variance normalized log power spectrogram of 3 seconds. This normalisation, motivated from [25], is performed at the utterance-level to standardize the features (zero mean and unit variance for all frequency bins) within a given recording. We use a 512-point *fast Fourier transform* (FFT), and a 32 ms window with a hop of 10 ms. Therefore, the original input spectrogram has a shape of 300×257, where 300 is the number of frames and 257 the number of FFT bins. To obtain a consistent input representation we replicate the audio samples (in the time domain) if the duration is smaller, or truncate the samples to 3 seconds duration. When $n = 1$, the input shape to the baseline CNN remains the same (300×257) however it varies for sub-CNNs depending upon different splits we use. For example, when $n = 2$, the input shape becomes 300×128 and 300×129. We always include the leftover bin to the last split. Similarly, when $n = 4$ we have four sub-spectrograms where the first three will have a shape of 300×64, and a shape of 300×65 for the last split. Likewise, for $n = 8$ we have seven sub-spectrograms of shape 300×32, and a shape of 300×33 for the last split.

Following our prior findings [30] on the ASVspoof 2019 PA dataset, we remove zero-valued samples from the start and end of every audio recording in the dataset. Likewise, on the ASVspoof 2017 v2.0 dataset, we remove leading and trailing silence/nonspeech samples following our findings in [26]. For this, we use our publicly released speech endpoint annotations [31]. Applying such preprocessing helps the model to avoid exploiting cues that are actually not relevant to the problem, rather this forces the models to now learn relevant factors in replay spoofing detection.

4.3. Model training and testing

We train the network to optimize the binary cross entropy loss between a bonafide and a spoof class. We use a batch size of 32 and learning rate of $1e^{-4}$. We use the *ADAM* [32] optimizer with default parameters. We apply a dropout of 50% to the input of the fully connected layers. If the validation loss does not improve for 5 epochs we stop the training process to avoid overfitting. We train models for a maximum of 100 training epochs. Using this approach we train 5 models with random initialisation. We choose the model showing the best performance on the development set and use it to test the performance on the evaluation set. At test time, for each audio spectrogram we use the model output — the bonafide-class posterior probability — as our detection score. The approach described above is the same for all our models.

4.4. Performance evaluation

We assess the performance of different countermeasure models using two different metrics. The first one gauges the ability of the countermeasure to discriminate bonafide and spoofed utterances from each other. It is measured through the *equal error rate* (EER), which was the primary evaluation metric of the ASVspoof 2017 challenge, and a secondary metric of the ASVspoof 2019 challenge. EER is the error rate at an operating point where the false acceptance (*false alarm*) and false rejection (*miss*) rates are equal. A reference value of 50% indicates the chance level. Our second metric is *minimum normalized*

Table 1: Performance of the baselines.

Baseline	Subbands (kHz)	ASVspoof 2017		ASVspoof 2019	
		t-DCF	EER%	t-DCF	EER%
CNN	0-8	0.3873	13.02	0.2019	7.00
GMM	0-8	0.5054	18.33	0.9928	59.12

Table 2: Performance of sub-CNNs M_1, M_2 and joint model J_1. Bold indicates the best performance.

Model	Subbands (kHz)	ASVspoof 2017		ASVspoof 2019	
		t-DCF	EER%	t-DCF	EER%
M_1	0-4	0.7045	33.03	0.1925	6.97
M_2	4-8	0.4800	18.95	0.5201	19.69
J_1	-	**0.2893**	**10.63**	**0.1864**	**6.44**

tandem detection cost function (t-DCF) [33] metric that evaluates both countermeasure and ASV system as a whole. We use the evaluation scripts and the ASV scores (x-vector recognizer) released by the organisers. A reference value 1.00 of (normalized) t-DCF indicates an uninformative countermeasure.

5. Experiments and results

We now describe different experimental studies along with the results in this section.

Baselines. To assess the performance of our proposed framework we train a baseline CNN model (Figure 1(a)) using a full-band spectrogram. For completeness, we also train and test a CQCC-based GMM model so as to compare it with our proposed framework. It should be noted that due to the preprocessing applied on the ASVspoof 2017 and 2019 PA datasets during training and testing, our baselines are different from the official ones [28, 7]; therefore, the numbers reported here should not be directly compared with the results of ASVspoof challenges. Table 1 summarises the results.

On the ASVspoof 2017 v2.0 we use the same parameterisation as in [28] to train the GMM model. On the evaluation set, the GMM model reaches EER = 18.33% and t-DCF = 0.5054. On the ASVspoof 2019 PA[3] evaluation set EER = 59.12% and t-DCF = 0.9928. The results suggest that GMM models become less confident in making classification decisions when silence cues are removed during training and testing. Refer to [30] for more details. On both datasets, the CNN baseline outperforms the GMM on both metrics. This demonstrates its effectiveness on learning relevant features useful for discrimination despite the preprocessing applied on the audio signals in contrast to hand-crafted CQCC features used in the GMM.

5.1. Experiment 1: subband modeling

We design four experimental setups for different values of n using our proposed methodology. We use $n = 2$ in our first setup. Using the architecture and training methodology described earlier we train two independent sub-CNNs M_1 and M_2. M_1 operates on the first 4 kHz and M_2 on the last 4 kHz subband spectrograms. Next, we use them (except the last output layer) to initialise the respective sub-CNN module weights of our joint sub-CNN model framework as shown in the bottom row of Fig-

---

[3]We used the pretrained GMM model from our prior work [30] to test on the evaluation set.

Table 3: Performance of sub-CNNs M_3 through M_6 and joint model J_2. Bold indicates the best performance.

Model	Subbands (kHz)	ASVspoof 2017		ASVspoof 2019	
		t-DCF	EER%	t-DCF	EER%
M_3	0-2	0.6172	31.20	0.2265	7.97
M_4	2-4	0.9773	41.58	0.4568	17.72
M_5	4-6	0.9995	50.70	0.5903	23.16
M_6	6-8	0.5032	23.50	0.6019	23.59
J_2	-	**0.3343**	**11.78**	**0.1977**	**6.99**

ure 1(b). We call this joint model J_1. Our second setup uses $n = 4$. Therefore, we train four independent sub-CNNs M_3 through M_6 on 2 kHz subband spectrograms. We then use these pretrained models to initialise the weights of sub-CNN modules of our joint model J_2. Our final setup uses $n = 8$. We now train eight independent sub-CNNs M_7 through M_{14} operating on 1 kHz subband spectrograms. We then use them (except the last layer weights) to initialise our joint model J_3 shown in Figure 1(b) bottom row.

Finally, motivated from the results of sub-CNNs M_7 and M_{14} on the ASVspoof 2017 dataset (shown in Table 4) we design a joint model J_4 operating on the first and the last 1 kHz subband spectrograms. This setup is different from the previous setups as more than half of the information is being discarded here. Overall J_4 utilises only 2 kHz of information (the first and last 1 kHz bands). As in the earlier setups, the pretrained weights (here M_7 and M_{14} models) are used to initialise the weights of sub-CNN modules of our joint model J_4. It should be noted that the entire model parameters are jointly optimised while training J_1, J_2, J_3 and J_4. We perform these experiments on both the ASVspoof 2017 v2.0 and 2019 PA datasets.

Table 2 summarises the performance of our individual sub-CNNs M_1, M_2 and the joint model J_1. On the ASVspoof 2017 dataset, the higher frequency bands ($4 - 8$ kHz) seem to carry more discriminative information than the lower bands ($0 - 4$ kHz). However, we observe the opposite pattern on the ASVspoof 2019 dataset where M_1 trained on $0 - 4$ kHz shows better results than M_2. This might be due to differences in dataset design (real vs. simulated replay) and compilation (different speakers and audio qualities of the source corpora). Nonetheless, on both the datasets our proposed joint model J_1 outperforms M_1, M_2 and the baselines by a large margin on both performance metrics. This demonstrates the effectiveness of our proposed approach for spoofing attack detection.

Next we discuss the performance of sub-CNNs M_3 through M_6 and the joint model J_2. Table 3 summarises this. Overall, the individual sub-CNNs now show poor performance, which is expected as each of them now receives only half of the information as the previous case ($n = 2$). Nonetheless, the proposed model J_2 that merges information from all the bands again outperforms the fullband baselines (CNN and GMM) and sub-CNNs M_3 through M_6.

Table 4 summarises the performance of our individual sub-CNNs M_7 through M_{14} and joint models J_3 and J_4. These results provide deeper insights in understanding the influence of different subbands for spoofing detection. As in the previous two setups ($n = 2$ and $n = 4$), our joint model J_3 shows better results than training individual sub-CNNs indicating that joint training offers some form of complementary information across different frequency bands. We also find that on the ASVspoof 2017 dataset, M_{14} trained on the *last* $7 - 8$ kHz band outper-

Table 4: Performance of sub-CNNs M_7 through M_{14} and joint models J_3 and J_4. Bold indicates the best performance.

Model	Subbands (kHz)	ASVspoof 2017		ASVspoof 2019	
		t-DCF	EER%	t-DCF	EER%
M_7	0-1	0.6216	31.59	0.2354	8.48
M_8	1-2	0.9977	48.92	0.6557	25.15
M_9	2-3	0.9971	50.61	0.7327	28.60
M_{10}	3-4	0.9969	49.85	0.5390	21.60
M_{11}	4-5	0.9992	44.06	0.6605	25.69
M_{12}	5-6	0.9914	45.05	0.7118	28.34
M_{13}	6-7	0.9817	42.23	0.7123	28.17
M_{14}	7-8	0.4747	18.10	0.7231	28.01
J_3	-	**0.2734**	11.02	**0.1975**	**7.34**
J_4	-	0.2771	**10.40**	0.2238	8.30

Table 5: Score-level fusion results. LS: linear sum of scores. WLS: weighted sum of scores.

Model	Fusion (type)	ASVspoof 2017		ASVspoof 2019	
		t-DCF	EER%	t-DCF	EER%
F_1	LS	0.3208	11.86	0.2208	7.50
F_2	WLS	0.3189	11.72	0.2034	**6.78**
F_3	LS	0.6151	24.25	0.2197	8.00
F_4	WLS	**0.3079**	**11.55**	0.1898	6.95
F_5	LS	0.4932	18.25	0.2626	9.78
F_6	WLS	0.3548	12.33	**0.1848**	6.99

forms all other sub-CNNs by a large margin. This is followed by M_7 (operating on the $0 - 1$ kHz band) that also performs substantially better than the remaining sub-CNNs. Inspired by this finding, we further train another joint model J_4 trained *only* with the lowest and highest sub-CNNs. Interestingly, this highly reduced model yields the best performance on the ASVspoof 2017 dataset, matching with the findings reported on the version 1.0 dataset by [23]. However, on the ASVspoof 2019 dataset, the first 1 kHz subband appears to carry most relevant information as opposed to other subbands. Furthermore, the results found for the ASVspoof 2017 and 2019 dataset do not match completely with the main differences being in dataset design and collection — the two datasets ASVspoof 2017 and 2019 PA are designed and collected differently.

5.2. Experiment 2: score fusion

We perform fusion experiments to understand if combining information through score-level fusion helps improve detection performance. In this setting we use the scores from each of the pretrained sub-CNNs and combine their scores using two simple approaches: linear sum of scores (LS), and linear weighted sum of scores (WLS). Let $S_1, S_2, S_3, ..., S_n$ represent scores from n sub-CNNs for a test utterance X. The fused score using the two approaches is obtained as: linear sum $= S_1 + S_2 + S_3 + .. + S_n$ and weighted sum $= w_1 * S_1 + w_2 * S_2 + w_3 * S_3 + .. + w_n * S_n$ where $w_1, w_2, w_3, ..., w_n$ are weights corresponding to each sub-CNN score learned using logistic regression (LR). We use the Bosaris toolkit [34] for the LR implementation. We perform six different fusion experiments: F_1 - F_6. F_1 uses linear sum fusion on the M_1 and M_2 model scores, F_2 on the other hand uses weighted linear fusion on them. F_3 uses linear sum fusion on the scores of the models M_3 through M_6, F_4 on the other hand uses weighted linear fusion on them. F_5 uses linear sum fusion on the scores of the models M_7 through M_{14}, F_6 on

Table 6: Cross-dataset performance evaluation on the 2019 PA real testset. D1: ASVspoof 2017 v2.0, D2: ASVspoof 2019 PA.

Model	Subbands (kHz)	Trained on D1		Trained on D2	
		t-DCF	EER%	t-DCF	EER%
CNN	0-8	0.9986	41.29	0.6374	34.25
M_1	0-4	0.8205	44.81	0.6800	33.12
M_2	4-8	0.9687	42.59	0.7820	39.44
J_1	-	0.9939	44.07	0.6741	35.92
M_3	0-2	0.8984	38.33	0.6760	33.49
M_4	2-4	**0.8025**	46.08	0.7311	34.62
M_5	4-6	0.9995	59.44	0.8033	35.74
M_6	6-8	0.9976	51.85	0.6877	28.37
J_2	-	0.9944	40.55	0.6544	30.18
M_7	0-1	0.8508	**34.81**	0.6678	34.95
M_8	1-2	0.9875	47.63	1.0	51.29
M_9	2-3	0.9545	51.48	0.8014	32.22
M_{10}	3-4	0.9634	42.22	0.8553	35.41
M_{11}	4-5	0.9003	43.33	0.7524	32.80
M_{12}	5-6	0.9788	47.03	0.8322	38.65
M_{13}	6-7	0.9828	48.49	0.7038	28.70
M_{14}	7-8	0.9981	42.38	**0.6483**	**27.22**
J_3	-	0.9208	37.61	0.6439	30.97
J_4	-	0.9081	36.62	0.7541	36.71

the contrary applies weighted linear fusion on them.

Table 5 summarises the results. In general, WLS fusion shows better performance than LS fusion, outperforming the baselines. However they show poor (or similar) performance compared to our joint subband modeling framework. For example, on the ASVspoof 2017 evaluation set the joint model J_4 (Table 4) shows better results in comparison to all our fusion models. Similarly, on the ASVspoof 2019 PA evaluation set the joint model J_1 (Table 2) shows better EER (and slightly worse t-DCF). As expected, combining information through score-fusion techniques offers gains in detection performance. However, our proposed joint subband modelling framework shows better results over score-fusion approaches. This further confirms that the complementary information provided by individual sub-CNNs helps improve overall detection performance.

5.3. Experiment 3 - cross-database evaluation

The goal of our final experiment is to understand the generalisability of our subband specific models (M_1 through M_{11}) and joint subband models (J_1, J_2, J_3 and J_4) in *cross-database* evaluation. We take all our pretrained models trained on the ASVspoof 2017 and ASVspoof 2019 PA dataset and evaluate their performance on the ASVspoof 2019 real PA test set. Table 6 shows the results.

While our models show good performance on the respective evaluation sets (see Tables 2, 3, 4), they fail to generalise well on the unseen real replay attack conditions of the ASVspoof 2019 real PA test set. On both the t-DCF and EER we observe high error rates for our joint models (J_1, J_2, J_3 and J_4). This also indicates overfitting on the respective dataset and lack of generalisability in unseen attack conditions. One possible interpretation to this observation may be attributed to dataset design and collection for the ASVspoof 2017 and 2019 PA datasets. It is worth noting that the ASVspoof 2019 PA dataset was developed through controlled simulation while the ASVspoof 2017 dataset was collected in real world recording and replay conditions. Due to the dataset issues [26, 30] identified on both ASVspoof

Table 7: Summary of results showing the comparison of baselines with our proposed models. * results taken from Table 1.

Model	ASVspoof 2017		ASVspoof 2019	
	t-DCF	EER%	t-DCF	EER%
GMM*	0.5054	18.33	0.9928	59.12
CNN*	0.3873	13.02	0.2019	7.0
J_1	0.2893	10.63	**0.1864**	**6.44**
J_2	0.3343	11.78	0.1977	6.99
J_3	**0.2734**	11.02	0.1975	7.34
J_4	0.2771	**10.40**	0.2238	8.30

2017 and 2019 PA, the models trained on these datasets might not be able to capture real replay attack conditions and thus perform poorly on the real PA test set which has been designed carefully reflecting real replay attack conditions. This study further suggests that there is still need for a reliable replay training dataset that can be used to train models incorporating real world replay attack conditions.

6. Discussion and conclusion

In this paper we performed a detailed analysis on the impact of different subbands and their importance on replay spoofing detection tasks on the benchmark datasets ASVspoof 2017 v2.0 and ASVspoof 2019 PA. Our proposed subband CNN model outperformed the traditional fullband CNN model, and also the CQCC-GMM baselines by a large margin, demonstrating the significance of our approach. We also investigated how combining information at the score level from each subband CNNs compared with our joint subband modelling framework. The final set of studies we performed here is on cross-database evaluation to investigate the generalisability of replay spoofing countermeasures on the ASVspoof 2019 real PA dataset.

Table 7 provides a summary of our main findings on both the ASVspoof 2017 and 2019 PA datasets. Performance improvement obtained using our proposed approach over baslines are very promising, encouraging further research on subband modelling for spoofing detection. Furthermore, it should be noted that our main objective in this paper is not to beat the best performing models published on the two datasets, but to validate our hypothesis about subband modelling using CNNs trained on spectrogram inputs. An important take-home message from our study is that while the first and the last 1 kHz frequency bands offer substantial improvement in detection performance, the same does not hold true for models trained on ASVspoof 2019. This suggests that the datasets available for training these models do not reflect real world replay conditions, suggesting a need for careful design of datasets for training replay spoofing countermeasures.

As future work, we aim to extend this analysis on text-to-speech and voice-converted spoofing attacks using the ASVspoof 2015 and ASVspoof 2019 logical access spoofing datasets. Use of advanced backend classifiers (current work uses a simple feed forward neural network) in our joint subband modelling framework will also be investigated. Furthermore, we are also interested in investigating the design and implementation of a unified countermeasure model that can be used for any spoofing attack conditions.

7. References

[1] Douglas A Reynolds, "Speaker identification and verification using Gaussian mixture speaker models," *Speech communication*, vol. 17, no. 1, pp. 91–108, 1995.

[2] Zhizheng Wu, Nicholas Evans, Tomi Kinnunen, Junichi Yamagishi, Federico Alegre, and Haizhou Li, "Spoofing and countermeasures for speaker verification: A survey," *Speech Communication*, vol. 66, pp. 130 – 153, 2015.

[3] ISO/IEC 30107-1:2016, "Information technology — Biometric presentation attack detection — Part 1: Framework," https://www.iso.org/obp/ui/#iso:std:iso-iec:30107:-1:ed-1:v1:en, 2016, [Online; accessed 27-January-2020].

[4] Laurent Besacier and Jean-François Bonastre, "Subband architecture for automatic speaker recognition," *Signal Processing*, vol. 80, no. 7, pp. 1245 – 1259, 2000.

[5] Zhizheng Wu, Tomi Kinnunen, Nicholas Evans, Junichi Yamagishi, Cemal Hanilci, Md Sahidullah, and Aleksandr Sizov, "ASVspoof 2015: the First Automatic Speaker Verification Spoofing and Countermeasures Challenge," in *Proc. INTERSPEECH*, 2015.

[6] Tomi Kinnunen, Md Sahidullah, Hector Delgado, Massimiliano Todisco, Nicholas Evans, Junichi Yamagishi, and Kong Aik Lee, "The ASVspoof 2017 Challenge: Assessing the Limits of Replay Spoofing Attack Detection," in *Proc. INTERSPEECH*, 2017.

[7] Massimiliano Todisco, Xin Wang, Ville Vestman, Md Sahidullah, Héctor Delgado, Andreas Nautsh, Junichi Yamagishi, Nicholas Evans, Tomi Kinnunen, and Kong Aik Lee, "ASVspoof 2019: Future Horizons in Spoofed and Fake Audio Detection," in *Proc.INTERSPEECH*, 2019.

[8] Md Sahidullah, Hector Delgado, Massimiliano Todisco, Tomi Kinnunen, Nicholas Evans, Junichi Yamagishi, and Kong-Aik Lee, "Introduction to Voice Presentation Attack Detection and Recent Advances," 2019.

[9] Massimiliano Todisco, Hector Delgado, and Nicholas Evans, "Constant Q cepstral coefficients: A spoofing countermeasure for automatic speaker verification," *Computer Speech and Language,*, vol. Volume 45,, pp. Pages 516–535, 2017.

[10] J. Yang, R. K. Das, and H. Li, "Extended Constant-Q Cepstral Coefficients for Detection of Spoofing Attacks," in *2018 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, Nov 2018, pp. 1024–1029.

[11] Tharshini Gunendradasan, Buddhi Wickramasinghe, Phu Ngoc Le, Eliathamby Ambikairajah, and Julien Epps, "Detection of Replay-Spoofing Attacks using Frequency Modulation Features," in *Proc. INTERSPEECH*, September 2018.

[12] Hemant A. Patil, Madhu R. Kamble, Tanvina B. Patel, and Meet Soni, "Novel Variable Length Teager Energy Separation Based Instantaneous Frequency Features for Replay Detection," in *Proc. INTERSPEECH*, 2017.

[13] Madhu R. Kamble, Hemlata Tak, and Hemant A. Patil, "Effectiveness of Speech Demodulation-Based Features for Replay Detection," in *Proc. INTERSPEECH*, September 2018.

[14] Buddhi Wickramasinghe, Eliathamby Ambikairajah, and Julien Epps, "Biologically Inspired Adaptive-Q Filterbanks for Replay Spoofing Attack Detection," in *Proc. INTERSPEECH*, September 2019.

[15] Hardik B. Sailor, Madhu R. Kamble, and Hemant A. Patil, "Auditory Filterbank Learning for Temporal Modulation Features in Replay Spoof Speech Detection," in *Proc. INTERSPEECH*, September 2018.

[16] Buddhi Wickramasinghe, Saad Irtza, Eliathamby Ambikairajah, and Julien Epps, "Frequency domain linear prediction features for replay spoofing attack detection," in *Proc. INTERSPEECH*, September 2018.

[17] D. P. Kingma and J. Ba, "An investigation of dependencies between frequency components and speaker characteristics for text-independent speaker identification," *Speech Communication*, vol. 50, pp. 1820–1824, 2008.

[18] S. S. R. Phaye, E. Benetos, and Y. Wang, "Subspectralnet – using sub-spectrogram based convolutional neural networks for acoustic scene classification," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2019, pp. 825–829.

[19] Kaavya Sriskandaraja, Vidhyasaharan Sethu, Phu Ngoc Le, and Eliathamby Ambikairajah, "Investigation of sub-band discriminative information between spoofed and genuine speech," in *Interspeech 2016*, 2016, pp. 1710–1714.

[20] Marcin Witkowski, Stanisław Kacprzak, Piotr Żelasko, Konrad Kowalczyk, and Jakub Gałka, "Audio replay attack detection using high-frequency features," in *Proc. Interspeech 2017*, 2017, pp. 27–31.

[21] S. Garg, S. Bhilare, and V. Kanhangad, "Subband analysis for performance improvement of replay attack detection in speaker verification systems," in *2019 IEEE 5th International Conference on Identity, Security, and Behavior Analysis (ISBA)*, Jan 2019, pp. 1–7.

[22] Parav Nagarsheth, Elie Khoury, Kailash Patil, and Matt Garland, "Replay attack detection using dnn for channel discrimination," in *Proc. Interspeech 2017*, 2017, pp. 97–101.

[23] Lang Lin, Rangding Wang, and Diqun Yan, "A Replay Speech Detection Algorithm Based on Sub-band Analysis," in *10th IFIP TC 12 International Conference, IIP 2018, Nanning, China, October 19-22, 2018, Proceedings*, 01 2018, pp. 337–345.

[24] Meet H. Soni, Tanvina B. Patel, and Hemant A. Patil, "Novel subband autoencoder features for detection of spoofed speech," in *Interspeech 2016*, 2016, pp. 1820–1824.

[25] Galina Lavrentyeva, Sergey Novoselov, Egor Malykh, Alexander Kozlov, Kudashev Oleg, and Vadim Shchemelinin, "Audio Replay Attack Detection with Deep Learning Frameworks," in *Proc. INTERSPEECH*, 2017, pp. 82–86.

[26] B. Chettri, S. Mishra, B. L. Sturm, and E. Benetos, "Analysing the Predictions of a CNN-based Replay Spoofing Detection System," in *IEEE International Workshop on Spoken Language Technology (SLT)*, September 2018.

[27] Xavier Glorot and Yoshua Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*. 13–15 May 2010, vol. 9, pp. 249–256, PMLR.

[28] H. Delgado, M. Todisco, Md. Sahidullah, N. Evans, T. Kinnunen, K.A Lee, and J. Yamagishi, "ASVspoof 2017 Version 2.0: meta-data analysis and baseline enhancements," in *Proc. Speaker Odyssey*, 2018.

[29] ASVspoof 2019, "the Automatic Speaker Verification Spoofing and Countermeasures Challenge Evaluation Plan.," .

[30] Bhusan Chettri, Daniel Stoller, Veronica Morfi, Marco A. Martínez Ramírez, Emmanouil Benetos, and Bob L. Sturm, "Ensemble Models for Spoofing Detection in Automatic Speaker Verification," in *Proc. INTERSPEECH*, September 2019, pp. 1018–1022.

[31] Bhusan Chettri, Emmanouil Benetos, and Bob L. Sturm, "Speech endpoint annotations and artefact details for ASVspoof 2017 version 2.0 dataset," Jan. 2020.

[32] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014.

[33] T. Kinnunen, K.A. Lee, H. Delgado, N. Evans, M. Todisco, M. Sahidullah, J. Yamagishi, D.A, and Reynolds, "t-DCF: a Detection Cost Function for the Tandem Assessment of Spoofing Countermeasures and Automatic Speaker Verification," in *Proc. Speaker Odyssey*, 2018.

[34] N. Brümmer and E. D. Villiers, "The bosaris toolkit: Theory, algorithms and code for surviving the new dcf," *arXiv preprint arXiv:1304.2865*, 2013.