Analog-to-Feature (A2F) Conversion for Audio-Event Classification

Xinming Liu, Emre Gönültaş, and Christoph Studer

Cornell University, Ithaca, NY, USA; email: {xl379, eg566, studer}@cornell.edu

***Abstract*—Always-on sensors continuously monitor the environment for certain events. Such sensors are often integrated on battery-powered devices, e.g., home automation devices or virtual assistants, which require power-efficient classification pipelines. However, conventional classification pipelines that digitize the analog signals at Nyquist rate followed by digital feature extraction and classification are wasteful in a sense that the "feature rate" is generally much smaller than the Nyquist rate. In this paper, we propose a novel classification pipeline called analog-to-feature (A2F) conversion that directly acquires features in the analog domain using non-uniform wavelet sampling (NUWS). Our approach effectively combines Nyquist-rate sampling and digital feature extraction, which has the potential to significantly reduce the power and costs of signal classification. We demonstrate the efficacy of our approach for the detection of audio events and show that NUWS-based A2F conversion is able to outperform existing methods that use compressive sensing.**

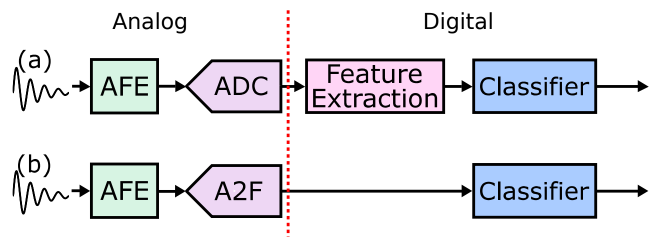


Fig. 1: (a) Illustration of a conventional signal classification pipeline that performs Nyquist sampling, extracts features in the digital domain, and performs classification. (b) The proposed analog-to-feature (A2F) conversion pipeline directly acquires features in the analog domain which effectively combines Nyquist sampling and feature extraction.

I. Introduction

In a growing number of applications and devices, always-on sensors continuously monitor the environment to detect certain events even if the rest of the device is in sleep mode. A prominent task of such sensors is audio-event classification (or detection), which finds use in, for example, smart phones or virtual assistants to wake-up the device by a voice command or to detect audio events (e.g., a certain home appliance is used) [1]. Audio-event classification is traditionally implemented using a pipeline shown in Fig. 1(a), in which an analog front-end (AFE) filters and amplifies the analog signal that is sampled using a Nyquist-rate analog-to-digital converter (ADC) followed by a digital feature extractor and a classifier [2]. As it has been noted in [3], [4], however, accurate signal classification does not require sampling at the Nyquist rate. In fact, digital feature extractors often generate far fewer features than the number of Nyquist samples, which indicates that traditional signal classification pipelines are wasteful in terms of power, cost, and the acquired amount of data.

A prominent way of reducing the sampling power, costs, and data rates is to use compressive sensing (CS) [5]. CS acquires fewer measurements than the Nyquist-rate suggests while still enabling faithful signal recovery as long as the signals are sparse in a given transform basis. To implement CS in practice, numerous acquisition schemes have been proposed in the past, including non-uniform sampling (NUS) [6] and random modulation (RM) [7], [8]. While one can directly perform signal classification from CS measurements [3], CS is designed for signal recovery and not for classification tasks, and suffers from noise folding and high dynamic range [9].

The work of XL, EG, and CS was supported in part by Xilinx Inc., and by the US NSF under grants ECCS-1408006, CCF-1535897, CAREER CCF-1652065, and CNS-1717559. The work of XL was supported in part by Cornell's Engineering Learning Initiatives (ELI). The work of EG research was supported in part by a fellowship from the Turkish Ministry of National Education. The Quadro P6000 GPU used for this research was donated by the NVIDIA Corporation. The authors would like to thank M. Pelissier, P. Chollet, and O. Castañeda for fruitful discussions on NUWS based signal classification.

A. Contributions

In order to perform signal classification at sub-Nyquist rates, we propose a novel pipeline called *analog-to-feature (A2F) conversion* that directly acquires a small set of features in the analog domain followed by a digital classifier; see Fig. 1(b) for an illustration. In contrast to CS, A2F conversion is specifically designed for signal classification, which further reduces the sampling rates, costs, and power. Instead of acquiring a random subset of Nyquist samples, as it is the case for NUS [10], we extract a carefully-selected set of "wavelet features" directly from the analog signal using non-uniform wavelet sampling (NUWS) [11]. These features are then fed into a digital classifier (e.g., a neural network) that detects the events of interest. To identify a small set of features that maximizes the classification accuracy, we propose an algorithm that jointly optimizes the wavelet features and the classifier. To demonstrate the effectiveness of NUWS-based A2F conversion, we perform audio-event classification using a real-world dataset and show that our approach outperforms existing CS-based methods, such as NUS and RM, in terms of classification accuracy.

ISBN 978-90-827970-1-5

B. Notation

Lowercase boldface letters denote column vectors; uppercase boldface letters denote matrices. For a matrix $\mathbf{A}$, $A_{k,j}$ refers to the entry on the kth row and jth column, and $\mathbf{A}^H$ denotes the Hermitian conjugate. For a vector $\mathbf{a}$, a_k refers to the kth entry, $\|\mathbf{a}\|_2 = \sqrt{\sum_k |a_k|^2}$ to the ℓ_2-norm, and $\Re(\mathbf{a})$ and $\Im(\mathbf{a})$ to the real and imaginary part, respectively.

II. Analog-to-feature (A2F) Conversion via Non-Uniform Wavelet Sampling (NUWS)

We start by reviewing basics of CS, and briefly discuss NUS and RM. We then detail NUWS-based A2F conversion.

A. Basics of Compressive Sensing (CS)

CS samples signals at sub-Nyquist rates while still enabling faithful signal recovery [5]. Let $\mathbf{x} \in \mathbb{R}^N$ be the N-dimensional signal to be acquired with $\mathbf{x} = \mathbf{\Psi}\mathbf{s}$, where $\mathbf{s} \in \mathbb{R}^N$ the sparse representation (i.e., only a few entries carry most of the signal's energy) of $\mathbf{x}$ and $\mathbf{\Psi}$ is a unitary matrix that sparsifies $\mathbf{x}$. Then, CS samples the signal $\mathbf{x}$ as follows:

$$\mathbf{y} = \mathbf{\Phi}\mathbf{x} + \mathbf{n}. \tag{1}$$

Here, the vector $\mathbf{y} \in \mathbb{R}^M$ contains the CS measurements, $\mathbf{\Phi} \in \mathbb{R}^{M \times N}$ is a carefully designed sampling matrix with $M < N$, and $\mathbf{n} \in \mathbb{R}^M$ models measurement noise. If the signal $\mathbf{x}$ of interest has a sufficiently sparse representation $\mathbf{s}$ and if the effective sampling matrix $\mathbf{A} = \mathbf{\Phi}\mathbf{\Psi} \in \mathbb{R}^{M \times N}$ (so that $\mathbf{y} = \mathbf{A}\mathbf{s} + \mathbf{n}$) satisfies certain incoherence conditions, then one can faithfully recover the sparse representation $\mathbf{s}$ (and, hence, the signal of interest $\mathbf{x} = \mathbf{\Psi}\mathbf{s}$) from $\mathbf{y}$ with fewer measurements M than the signal's ambient dimension N [5].

B. Non-Uniform Sampling (NUS)

NUS is among the simplest CS methods and acquires a small subset of the Nyquist samples [10]. Mathematically, the sampling matrix is $\mathbf{\Phi} = \boldsymbol{R}_\Omega$, where $\boldsymbol{R}_\Omega = [\boldsymbol{I}_N]_\Omega$ is the $M \times N$-dimensional restriction operator which consists of the subset Ω with cardinality $|\Omega| = M$ of rows of the $N \times N$ identity matrix $\boldsymbol{I}_N$. NUS is conceptually simple and enables efficient hardware designs [12]. Furthermore, NUS is particularly well-suited for signals that are sparse in the frequency domain, i.e., where the sparsity transform basis $\mathbf{\Psi} = \mathbf{F}^H$ is the $N \times N$ inverse discrete Fourier transform (DFT) matrix [13].

C. Random Modulation (RM)

RM enables the acquisition of more general classes of signals than NUS [7], [8]. Mathematically, the sampling matrix $\mathbf{\Phi}$ contains (pseudo-)random entries, e.g., i.i.d. Bernoulli $\{-1, +1\}$ or standard normal entries, and each compressive measurement $m = 1, \ldots, M$ corresponds to

$$y_m = \langle \boldsymbol{\phi}_m, \mathbf{x} \rangle + n_m, \tag{2}$$

where $\boldsymbol{\phi}_m$ denotes the mth row of $\mathbf{\Phi}$. Hence, RM acquires (pseudo-)random inner products of the signal vector $\mathbf{x}$ which can be implemented using a signal generator that produces the entries of $\boldsymbol{\phi}_m$, an analog multiplier, and an integrator [8], [11].

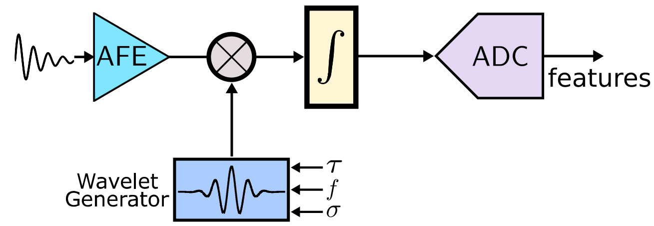


Fig. 2: High-level overview of A2F conversion using NUWS: An analog frontend (AFE) filters and amplifies the incoming signal. An analog multiplier mixes the input signal with a wavelet that can be tuned in time instant τ, center frequency f, and duration σ. An integrator computes the inner product and the resulting wavelet feature is sampled by a low-rate ADC.

The key drawbacks of RM are as follows: (i) the random sequence generator must still operate at Nyquist rate; (ii) one requires M parallel sampling branches (so-called fingers) to acquire M compressive measurements. Furthermore, RM suffers from noise folding due to the wideband nature of the (pseudo-)random sequences [9].

D. Non-Uniform Wavelet Sampling (NUWS)

NUS and RM, as well as other CS architectures (see [11] and the references therein), are specifically designed for signal recovery. Nevertheless, these methods can be used to directly perform signal classification [3] from the compressive measurements in $\mathbf{y}$. To avoid the practical limitations of NUS and RM while enabling further reductions in terms of the measurements to be acquired for signal classification tasks, we propose to use NUWS [11] for A2F conversion. In essence, NUWS combines the advantages of NUS and RM by acquiring inner products of the signal $\mathbf{x}$ as in (2) but with hardware-friendly wavelet functions rather than Dirac delta functions (for NUS) or (pseudo-)random sequences (for RM). The tunability of the wavelet functions enables one to adapt the compressive samples to the signal type and classification task, effectively extracting features directly in the analog domain.

NUWS, as illustrated in Fig. 2, computes inner products as in (2) between the analog signal and tunable wavelet functions that can be efficiently generated in hardware [14]. As in [11], we focus on Gabor-like wavelet functions defined as [15]

$$\phi_{\tau,f,\sigma}(t) = w_\sigma(t-\tau)e^{j2\pi f(t-\tau)}, \tag{3}$$

where t refers to continuous time, $w_\sigma(t)$ is a window function (e.g., a rectangular or Gaussian window) centered at zero and whose width is controlled by the parameter $\sigma > 0$, τ denotes the location of the wavelet, and $f > 0$ determines the center frequency. For simplicity, we discretize (3) and represent the wavelet functions as N-dimensional vectors $\boldsymbol{\phi}_{\tau,f,\sigma}$. In the discrete domain, one can collect a finite set of L wavelet vectors $\{\boldsymbol{\phi}_{\tau_\ell,f_\ell,\sigma_\ell}\}_{\ell=1}^L$ with given location τ_ℓ, frequency f_ℓ, and width σ_τ for $\ell = 1, \ldots, L$, and assign each vector to a column of the wavelet dictionary $\mathbf{W} \in \mathbb{C}^{N \times L}$. NUWS signal acquisition can then be represented in compact form as

$$\mathbf{y} = \mathbf{R}_\Omega \mathbf{W}^H \mathbf{x} + \mathbf{n},$$

ISBN 978-90-827970-1-5 © EURASIP 2018

where the restriction operator $\mathbf{R}_\Omega = [\mathbf{I}_L]_\Omega$ selects the subset Ω with cardinality $|\Omega| = M$ of wavelets from the wavelet dictionary $\mathbf{W}$. Intuitively, NUWS first expands the signal $\mathbf{x}$ in the (overcomplete) wavelet frame $\mathbf{W}^H\mathbf{x}$ followed by selecting a subset of the coefficients indexed by Ω. The acquired (noisy) wavelet coefficients collected in $\mathbf{y}$, which we call *wavelet features*, are then fed directly to a digital classifier.

It is important to realize that each (noiseless) wavelet feature $\tilde{y}_\ell = \langle \boldsymbol{\phi}_{\tau_\ell, f_\ell, \sigma_\ell}, \mathbf{x}\rangle$ represents a certain portion of the frequency spectrum of the signal $\mathbf{x}$ to be classified, where τ determines the phase, f the center frequency, and $1/\sigma$ the bandwidth to be extracted. Evidently, by carefully selecting the set of wavelet vectors $\boldsymbol{\phi}_{\tau_\ell, f_\ell, \sigma_\ell}$, $\ell = 1, \ldots, L$, one can extract the spectral components that are relevant for signal classification.

III. Joint Feature Selection and Classifier Design

The classification performance of A2F conversion is determined by the features *and* the classifier. Acquiring a large number of features will improve classification performance but also negatively affect the power consumption and feature rates. In general, the key goal of A2F conversion is to identify the smallest set of relevant features that yields acceptable classification performance. For NUWS-based A2F conversion, this problem boils down to (i) constructing a suitable wavelet dictionary $\mathbf{W}$, (ii) selecting a small set Ω of wavelet functions, and (iii) designing a digital classifier. In what follows, we separate step (i) from (ii) and (iii), i.e., we first construct $\mathbf{W}$ and then, jointly select features and design the classifier.

A. Construction of Wavelet Dictionary

We consider rectangular window functions of width N, $N/2$, $N/4$, and $N/8$ samples, and we construct the wavelet dictionary by taking inverse DFTs of dimension N, $N/2$, $N/4$, and $N/8$ with non-overlapping windows on each scale. Concretely, our base wavelet dictionary is compactly described as

$$\overline{\mathbf{W}} = \left[\mathbf{I}_1 \otimes \mathbf{F}_N^H \,,\, \mathbf{I}_2 \otimes \mathbf{F}_{N/2}^H \,,\, \mathbf{I}_4 \otimes \mathbf{F}_{N/4}^H \,,\, \mathbf{I}_8 \otimes \mathbf{F}_{N/8}^H \right], \quad (4)$$

where $\otimes$ denotes the Kronecker product. For the audio-event classification task considered in Sec. IV, we have observed that one does not require continuous wavelet functions. In particular, NUWS-based A2F conversion can be carried out using simpler functions with entries in $\{-1, 0, +1\}$, which enables one to replace the analog multiplier in Fig. 2 with simple mixing (or chopping) circuitry. Furthermore, we are interested in acquiring real-valued wavelet features (the functions in (3) are complex-valued). To this end, we first convert (4) into the real domain followed by quantization of the entries to the set $\{-1, 0, +1\}$. Specifically, we design our wavelet dictionary as

$$\mathbf{W} = \left[\operatorname{sgn}\big(\Re(\overline{\boldsymbol{W}})\big) \,,\, \operatorname{sgn}\big(\Im(\overline{\boldsymbol{W}})\big) \right], \quad (5)$$

where the sign operator $\operatorname{sgn}(A)$ is applied element-wise to matrices, and $\operatorname{sgn}(A) = +1$ if $A > 0$, $\operatorname{sgn}(A) = 0$ if $A = 0$, and $\operatorname{sgn}(A) = -1$ if $A < 0$. Note that the wavelet dictionary construction in (5) contains ternary-valued sequences that are localized in time and roughly localized in frequency, corresponding to the frequencies given by the inverse DFTs of varying dimensions. Hence, the wavelet dictionary contains vectors that span a broad range of frequencies, bandwidths, and phases, from which a suitable subset can be selected.

B. Classifier Structure

For the audio event classification task considered in Sec. IV, we observed that (artificial) neural networks (NNs) work best with NUWS-based A2F conversion.[1] Furthermore, NNs can be implemented quite efficiently in hardware as they mainly consist of matrix-vector multiplications [16]. To minimize the complexity of the NN, we use a shallow network structure with only two hidden layers, each having 200 neurons. We use rectified linear units (ReLUs) as activation functions in the hidden layers and a soft-max function in the output layer. We use TensorFlow [17] and an NVIDIA Quadro P6000 GPU to learn the weights of the NN.

C. Joint Feature Selection and Classifier Design

We next describe an algorithm that jointly selects a small subset of wavelet features and learns the weights of the NN. We use a selection wrapper method, which extracts the set of features based on the performance of the classifier. More specifically, we build our approach on the forward selection wrapper method proposed in [18]. The basic steps are as follows. Start with an empty set of features $\Omega = \varnothing$. Train a classifier for each of the $\ell = 1, \ldots, L$ wavelet vectors in $\mathbf{W}$ and add the feature index $\hat{\ell}$ to the feature set $\Omega \leftarrow \{\Omega, \hat{\ell}\}$ that yields the highest classification accuracy. In subsequent steps, iterate over the remaining features, i.e., $\{1, \ldots, L\} \setminus \Omega$ and train a new classifier together with the previously selected feature set Ω. In words, sequentially add the next-best feature in every step by retraining the classifier. This greedy procedure is repeated until M features are selected. The generated set Ω contains an ordered list of indices, in which the features selected early are more important than those selected at later stages.

Unfortunately, if the number L of total features and the subset $|\Omega| = M$ are both large, then the method summarized above requires one to train $C = LM - \frac{1}{2}M(M-1)$ classifiers. For the situation considered in Sec. IV, where we use a wavelet dictionary with $L = 2048$ vectors and select up to $M = 32$ features, this requires training of $C = 65,040$ classifiers, resulting in approximately 180 hours feature selection time on the used GPU. To reduce the feature-selection time, we propose to keep only the best $\alpha \in [0,1]$ fraction features in each iteration, i.e., we remove $1-\alpha$ features with lowest classification accuracy. Note that we have to set $\alpha \geq L^{\frac{1}{1-M}}$ so that after M iterations we still have a sufficient number of features to select from. By setting $\alpha = L^{\frac{1}{1-M}}$, the number of times a classifier must be trained reduces to

$$\hat{C} = \left\lceil L\, \frac{1 - L^{\frac{M}{1-M}}}{1 - L^{\frac{1}{1-M}}} \right\rceil,$$

where $\lceil \cdot \rceil$ denotes rounding towards infinity. For the above example, the proposed strategy would reduce the number of

[1] We also evaluated random forests, which work well for binary classification tasks but not for multi-class situations as considered in this paper.

ISBN 978-90-827970-1-5 © EURASIP 2018

TABLE I: Ten audio events recorded with a Blue Yeti USB microphone in three different home environments.

# Audio event	# Audio event
1 Dryer, running	6 Sink, full stream center
2 Kettle, heating	7 Kitchen fan, low
3 Kitchen fan, high	8 Washing machine, running
4 Sink, fill large jug full stream	9 Shower, sprinkle
5 Washing machine, spin cycle	10 Washing machine, start

classifiers to be trained to $\hat{C} = 9,390$ or only 26 hours of feature selection time, which is almost $7\times$ faster. We note that by selecting α slightly higher than $L^{\frac{1}{1-M}}$, the proposed feature selection method only marginally degrades the classification accuracy in our application compared to a full search.

IV. Experiments

We now demonstrate the efficacy of NUWS-based A2F conversion for an audio-event classification task. Due to the lack of a publicly-available audio event dataset, we recorded ten different audio events as listed in Tbl. I using a Blue Yeti USB microphone (mono, omnidirectional mode, 48 kHz sampling rate) in three different home environments.

A. Data Preparation and Augmentation

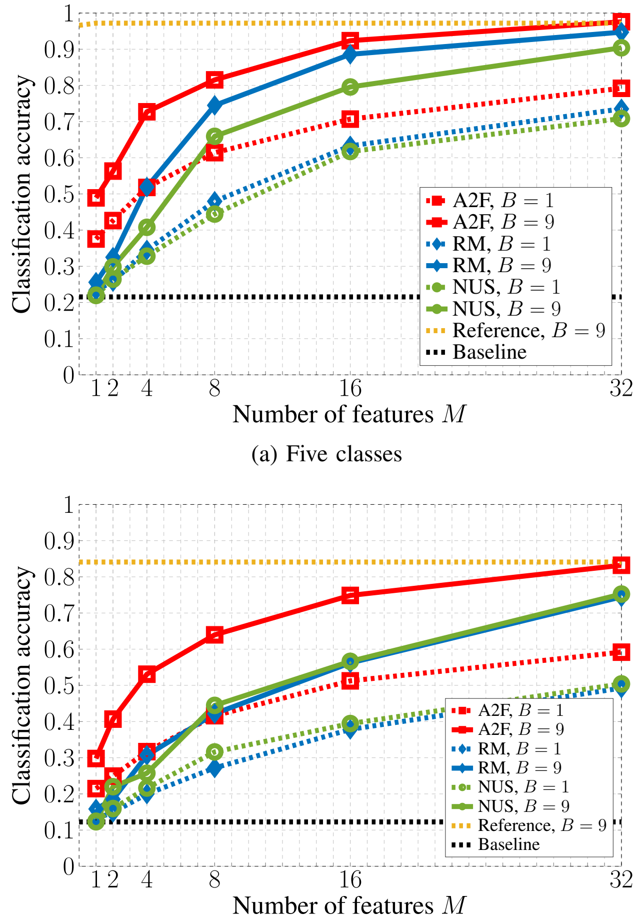
All of the following experiments are carried out in a Python simulator with real audio data. We perform block-wise processing of $N = 256$ audio samples and discard blocks whose ℓ_2-norm is below a threshold of $\|\mathbf{x}\|_2 < 0.25$. With the remaining blocks, we perform data augmentation [19] by randomly scaling (with a uniform distribution) the input signals $\mathbf{x}$ so that their ℓ_2-norm is between $0.25 \leq \|\mathbf{s}\|_2 \leq 1.25$, which emulates amplitude variations that may occur in practice.

B. Implementation Details

The remaining details of the classifier and feature selection algorithm are as follows. 70% of the available audio data is used for NN training; the rest is divided equally for validation during training and for testing to extract the classification accuracy. We select a subset of $M = 32$ features from a total number of $L = 2048$ wavelet features (see Sec. III-A). We set $\alpha = 0.9$, which results in excellent feature quality and requires us to train a total number of $19,854$ classifiers which takes roughly 55 hours to identify the set of wavelet features.

In realistic situations, a specific audio event often spans multiple signal blocks. This observation enables us to improve the classification accuracy by considering the classifier outputs over B consecutive blocks. Let $\mathbb{P}(X_b^j)$ denote the probability of class j in block b to be present. Assuming independence[2], the log-probability of class j being present after observing B blocks is as follows: $\log(\mathbb{P}(X^j)) = \sum_{b=1}^{B} \log(\mathbb{P}(X_b^j))$. We then select the most likely class as $\hat{j} = \arg\max_{j=1,\ldots,J} \log(\mathbb{P}(\hat{X}^j))$, where J represents the total number of classes.

[2]Neighboring blocks are clearly *not* independent. Nevertheless, we have observed substantial improvements by taking this leap of faith.



(a) Five classes

(b) Ten classes

Fig. 3: Accuracy versus the number of features for NUWS-based A2F conversion (red), RM (blue), and NUS (green). A2F conversion outperforms CS-based methods and approaches the Nyquist-rate reference performance for only $M = 16$ features.

C. Results

Figure 3 shows the classification accuracy versus the number of selected features M for different classification pipelines. Fig. 3(a) shows results for the first five events listed in Tbl. I and Fig. 3(b) for all ten events listed in Tbl. I. As a baseline, we include a classifier that always predicts the class with the largest number of data points (called "baseline"). We also include a reference classifier that applies a NN on all $N = 256$ Nyquist samples (called "reference"). The red curves show the accuracy of NUWS-based A2F conversion, where we used the feature selection algorithm proposed in Sec. III-C. The green curves show the accuracy of NUS, where we used the proposed feature selection algorithm to select the subset of Nyquist samples. The blue curves show the accuracy of RM where we used i.i.d. Bernoulli $\{+1, -1\}$ measurements.

We observe that NUWS-based A2F conversion outperforms both RM and NUS, especially for a small number of features and for the ten-class experiment. For five and ten classes, A2F conversion with $B = 9$ time slots is able to closely approach the reference performance (the Nyquist-based classifier with

ISBN 978-90-827970-1-5

Actual target \ Prediction	1	2	3	4	5
1	.139	.006	.004	.003	.016
2	.007	.185	.011	0	.009
3	0	.012	.201	0	.002
4	0	0	0	.203	0
5	.002	.002	.001	0	.195

(a) Five classes

Actual target \ Prediction	1	2	3	4	5	6	7	8	9	10
1	.087	.003	0	.007	.002	0	0	.005	.001	.002
2	.002	.110	.004	.005	0	0	0	.014	0	0
3	.002	.009	.108	.004	0	.003	0	.018	0	.02
4	.004	.002	.008	.035	0	0	0	.007	0	0
5	.001	0	0	0	.049	.002	.001	0	0	0
6	.001	0	0	0	.011	.079	.002	0	0	0
7	0	0	0	0	.03	.013	.116	0	0	.011
8	.001	0	.006	.009	0	0	0	.022	0	0
9	.003	.002	0	0	0	0	0	.004	.117	0
10	.001	0	.001	0	0	.002	.002	0	0	.049

(b) Ten classes

Fig. 4: Confusion matrices for NUWS-based A2F conversion with $M = 16$ wavelet features and processing $B = 9$ blocks.

$B = 9$) with only $M = 16$ features, which is equivalent to a compression of $16\times$. Furthermore, adding more features results in diminishing returns as we observe a saturation behavior.

Figure 4 shows the confusion matrices for NUWS-based A2F conversion with $M = 16$ wavelet features and joint processing $B = 9$ blocks. While the five-class experiment is not particularly challenging, the considered classifier struggles to identify audio-event number 8 for the ten-class experiment.

V. Conclusion

In this paper, we have proposed a novel signal classification pipeline which we call analog-to-feature (A2F) conversion. Our approach relies on non-uniform wavelet sampling (NUWS) which acquires spectral features directly in analog domain, effectively combining sub-Nyquist sampling and digital feature extraction. We have demonstrated our method on an audio-event classification task with real data which reveals that NUWS-based A2F conversion outperforms existing compressive sensing (CS)-based methods in terms of classification accuracy for the same number of measurements. Furthermore, we have shown that for the considered classification task, $16\times$ fewer measurements than Nyquist samples are sufficient to approach the performance of a traditional signal classification pipeline.

There are many avenues for future work. The design of a circuit-level prototype of NUWS-based A2F conversion that demonstrates the practical power savings is ongoing work. A theoretical analysis that studies the performance of our approach is an open research topic. Finally, applying A2F conversion to other signals and applications, such as communication systems and biomedical signals, is left for future work.

References

[1] G. Laput, Y. Zhang, and C. Harrison, "Synthetic sensors: Towards general-purpose sensing," in *Proceedings of the Conf. Human Fact. in Comput. Systems*, May 2017, pp. 3986–3999.

[2] P. K. Atrey, N. C. Maddage, and M. S. Kankanhalli, "Audio based event detection for multimedia surveillance," in *IEEE Intl. Conf. on Acoustics, Speech and Signal Proc. (ICASSP)*, May 2006, vol. 5, pp. 813–816.

[3] M. A. Davenport, P. T. Boufounos, M. B. Wakin, and R. G. Baraniuk, "Signal processing with compressive measurements," *IEEE J. Sel. Topics Signal Process.*, vol. 4, no. 2, pp. 445–460, Apr. 2010.

[4] M. Verhelst and A. Bahai, "Where analog meets digital: Analog-to-Information conversion and beyond," *IEEE Solid State Circuits Mag.*, vol. 7, no. 3, pp. 67–80, Sep. 2015.

[5] E. J. Candès and M. B. Wakin, "An introduction to compressive sampling," *IEEE Signal Process. Mag.*, vol. 25, no. 2, pp. 21–30, Mar. 2008.

[6] S. M. S. Zobly and Y. M. Kakah, "Compressed sensing: Doppler ultrasound signal recovery by using non-uniform sampling amp; random sampling," in *28th National Radio Science Conf. (NRSC)*, Apr. 2011, pp. 341–350.

[7] S. Kirolos, J. N. Laska, M. B. Wakin, M. F. Duarte, D. Baron, T. Ragheb, Y. Massoud, and R. G. Baraniuk, "Analog-to-information conversion via random demodulation," in *IEEE Dallas/CAS Workshop on Design, Applications, Integration and Software*, Oct. 2006, pp. 71–74.

[8] M. Mishali, Y. C. Eldar, and A. J. Elron, "Xampling: Signal acquisition and processing in union of subspaces," *IEEE Trans. Signal Proc.*, vol. 59, no. 10, pp. 4719–4734, Oct. 2011.

[9] M. A. Davenport, J. N. Laska, J. R. Treichler, and R. G. Baraniuk, "The pros and cons of compressive sensing for wideband signal acquisition: Noise folding versus dynamic range," *IEEE Trans. Signal Proc.*, vol. 60, no. 9, pp. 4628–4642, Sep. 2012.

[10] M. B. Wakin, S. Becker, E. Nakamura, M. Grant, E. Sovero, D. Ching, J. Yoo, J. Romberg, A. Emami-Neyestanak, and E. J. Candès, "A nonuniform sampler for wideband spectrally-sparse environments," *IEEE J. Emerg. Sel. Topics Circuits Syst.*, vol. 2, no. 3, pp. 516–529, Sep. 2012.

[11] M. Pelissier and C. Studer, "Non-uniform wavelet sampling for RF analog-to-information conversion," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 65, no. 99, pp. 471–484, Aug. 2017.

[12] D. E. Bellasi, L. Bettini, C. Benkeser, T. Burger, Q. Huang, and C. Studer, "VLSI design of a monolithic compressive-sensing wideband Analog-to-Information converter," *IEEE Trans. Emerg. Sel. Topics Circuits Syst.*, vol. 3, no. 4, pp. 552–565, Dec. 2013.

[13] S. Foucart and H. Rauhut, *A mathematical introduction to compressive sensing*, vol. 1, Birkhäuser Basel, 2013.

[14] M. Pelissier, G. Masson, L. Ouvry, L. F. F. Dias, and M. Marnat, "Hardware platform of Analog-to-Information converter using Non Uniform Wavelet Bandpass Sampling for RF signal activity detection," in *IEEE Int. Symp. Circuits and Systems (ISCAS)*, May. 2018.

[15] J. J. Benedetto, C. Heil, and D. F. Walnut, *Gabor systems and the Balian-Low Theorem*, Birkhäuser Boston, Boston, MA, 1998.

[16] Y. Liao, "Neural networks in hardware: A survey," Tech. Rep., 2001.

[17] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, et al., "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, Software available from tensorflow.org.

[18] P. Pudil, J. Novovičová, and J. Kittler, "Floating search methods in feature selection," *Pattern Recognition Letters*, vol. 15, no. 11, pp. 1119–1125, Nov. 1994.

[19] N. Takahashi, M. Gygli, B. Pfister, and L. Van Gool, "Deep convolutional neural networks and data augmentation for acoustic event detection," *ArXiv preprint: 1604.07160*, Apr. 2016.

ISBN 978-90-827970-1-5