Why RDT ? Computational aspects in favor of the RDT transform

Radu and Ioana Dogaru, radu.dogaru@upb.ro ioana.dogaru@upb.ro oct. 2025

Approaching a signal classification problem may usually be tackled in 2 ways:

a) The most common approach, mostly applied in sound / speech classification is to employ MFCC transforms [1][2] to convert a large signal (*N* samples) into a feature vector with *m* elements (representing the spectral energy produced at the output of some *m* filter banks). Quite often the signal is split using some more analysis windows thus providing the output feature as a spectrogram. The classifier employed on this feature vector or spectrogram can be a simple one, ranging from a simple dense layer to a simple convolutional neural network (CNN). For a signal length *N* (i.e. there are *N* samples) and *m* filters the computational complexity of the MFCC feature extractor may be expressed as $k_1 Nm$. The algorithm requires multiplications and the computations of cosine functions giving thus a large value of k_1 multiplier. Moreover, when dealing with low power, low complexity MCU-based implementations such arithmetic operations would imply large latencies. In addition, MFCC-libraries for MCU units are not widely available and one notable exception is the one integrated in the EdgeImpulse [3] framework. Typically, in this approach, the overall latency of the signal recognizer is dominated by the computation of the MFCC spectrogram.

b) The second common approach is to employ a deep network directly on the signal with convolutional or LSTM layers as tunable feature extractors. In such cases, increasing the complexity of the deep neural network would imply large latencies associated mostly with the convolution layers operating on large inputs (of the size *N*).

A compact alternative to both previously presented approaches was first proposed in [4] and then applied for a variety of problems in either sound domain [5][6] or bio-medical signals [7] including the EEG Bonn dataset [8] with very good accuracy performance similar or better than state of the art values obtained using the most common approaches for feature extractors. This approach employs a low complexity feature extractor named RDT (detailed in the previous section) where no complicated arithmetic operations are employed and the overall complexity is also of the same type as in the case of MFCCs i.e. $k_2 Nm$. Here *m* is the number of "channels" or components of the delay vector (to be optimized for each signal classification problem). Unlike MFCC, in this case the multiplication coefficient k_2 proves to be tens of times smaller, thus making the latency associated with this signal processing stage negligible in comparison with the latency associated with the neural classifier employed to the resulting feature (spectrogram). Optimized code for the RDT called next iRDT (improved RDT) is provided in [9] including a fixed-point version (the function nrdt2025z()). The function is compiled with NUMBA JIT and represents the reference for producing several platform specific implementations. In its Python implementation, the iRDT is compared with the MFCC implementation from the LIBROSA library (where the NUMBA JIT compiler is also employed). For an EEG signal with *N=4097* samples while choosing *m* =6 for both iRDT and MFCC the running times (with the CPU support

offered in Google Collaboratory) are: 0.085 ms (iRDT) versus 6.99 ms (MFCC). Consequently, the iRDT method is almost $\frac{k1}{k2}$ =80 times faster! The reference code was next translated into C++ and using the Arduino IDE environment it was deployed into an Arduino Nano BLE33 Sense v2 (64 Mhz clock) where processing time for the same data as in the Python example was 4.7 ms. For comparison, the EdgeImpulse implementation of a comparable MFCC signal processor on the same platform gives around 150 ms. This result indicates that the ratio $k1/k2$ is almost unchanged (tens of times less computation for iRDT). The iRDT code was next transferred to the HLS tool from Vivado and following specific design flow an IP was developed (in sequential mode) with their associated overlay such that it can be evaluated using the same data on a PynqZ1 board [11]. The required resources are presented in Figure x indicating a relatively low occupancy of the FPGA array and no need for DSP multiplier. For comparison an MFCC processor for FPGA reported in [12] would require 4092 LUT (1.5 times more) and 14 DSP multipliers (compared to 0 in the iRDT). The intrinsic latency of the IP is 1.3 micro-seconds, although when using it as overlay some additional times for transferring the signal frame would increase the overall processing time to around 11ms.

Product family: zynq

Target device: xc7z020-clg400-1

Performance Estimates

⊞ **Timing**

⊞ **Latency**

Utilization Estimates

⊟ **Summary**

Name	BRAM_18K	DSP48E	FF	LUT	URAM
DSP	-	-	-	-	-
Expression	-	-	0	927	-
FIFO	-	-	-	-	-
Instance	2	-	1094	1159	-
Memory	0	-	519	65	0
Multiplexer	-	-	-	521	-
Register	-	-	917	-	-
Total	2	0	2530	2672	0
Available	280	220	106400	53200	0
Utilization (%)	~0	0	2	5	0

Fig. x Resources required on the FPGA array in PynqZ1 for the iRDT implementation

Concluding, there is strong evidence that replacing MFCC (or LSTM/convolutional layers) in a signal classification system with an optimized implementation of RDT would give significant benefits in terms of computational complexity, which is of particular interest when implementing the classifiers in MCU or SoC (MCU+ FPGA) platforms.

References

[1] A. Barovic, A. Moin, "TinyML for Speech Recognition", in https://arxiv.org/pdf/2504.16213?
[2] V. T. Tiwari, “Mfcc and its applications in speaker recognition,” Int. J. Emerg. Technol., vol. 1, 01 2010.
[3] S. Hymel et al., “Edge Impulse: An MLOps platform for tiny machine learning,” Proc. 6th MLSys Conf., vol. 5, 2023, pp. 1–15.
[4] R. Dogaru, “Fast and Efficient Speech Signal Classification with a Novel Nonlinear Transform,” in 2007 International Symposium on Information Technology Convergence (ISITC 2007), Jeonju, Korea: IEEE, Nov. 2007, pp. 43–47. doi: 10.1109/ISITC.2007.37.
[5] R. Dogaru and I. Dogaru, “RD-CNN: A Compact and Efficient Convolutional Neural Net for Sound Classification,” in 2020 International Symposium on Electronics and Telecommunications (ISETC), Timisoara, Romania: IEEE, Nov. 2020, pp. 1–4. doi: 10.1109/ISETC50328.2020.9301110.
[6] R. Dogaru and I. Dogaru, "State of the Art Recognition of Emotions from Speech, Using a Low Complexity Solution Based on Reaction-Diffusion Transform," 2022 International Symposium on Electronics and Telecommunications (ISETC), Timisoara, Romania, 2022, pp. 1-4, doi: 10.1109/ISETC56213.2022.10010234.
[7] R. Dogaru and I. Dogaru, "A low complexity solution for epilepsy detection using an improved version of the reaction-diffusion transform," 2017 5th International Symposium on Electrical and Electronics Engineering (ISEEE), Galati, Romania, 2017, pp. 1-6, doi: 10.1109/ISEEE.2017.8170678.
[8] R.G. Andrzejak RG, et.al., "Indications of nonlinear deterministic and finite dimensional structures in time series of brain electrical activity: Dependence on recording region and brain state", Phys. Rev. E, 64, 061907.
[9] https://github.com/radu-dogaru/rdt_transform_for_tiny_ml_signal_classifiers
[10] Xilinx, "Vivado design suite user guide: High-level synthesis", https://docs.amd.com/v/u/en-US/ug902-vivado-high-level-synthesis
[11] Digilent, "PYNQ-Z1 board", https://digilent.com/reference/programmable-logic/pynq-z1/start
[12] Glittas A X, Gopalakrishnan L. A low latency modular-level deeply integrated MFCC feature extraction architecture for speech recognition. Integration, 2021, 76: 69-75