UDC 681.326

ADAPTIVE FILTERING AND MACHINE LEARNING METHODS IN NOISE SUPPRESSION SYSTEMS, IMPLEMENTED ON THE SoC

Shkil A. S. − PhD, Associate Professor, Associate Professor of Design Automation Department, Kharkiv National University of Radio Electronics, Kharkiv, Ukraine.

Filippenko O. I. – PhD, Associate Professor, Associate Professor of Infocommunication Engineering Department named by V.V. Popovsky, Kharkiv National University of Radio Electronics, Kharkiv, Ukraine.

Rakhlis D. Y. − PhD, Associate Professor, Associate Professor of Design Automation Department, Kharkiv National University of Radio Electronics, Kharkiv, Ukraine.

Filippenko I. V. – PhD, Associate Professor, Associate Professor of Design Automation Department, Kharkiv National University of Radio Electronics, Kharkiv, Ukraine.

Parkhomenko A. V. – PhD, Associate Professor, Associate Professor of Software Tools Department, National University Zaporizhzhia Polytechnic, Zaporizhzhia, Ukraine.

Korniienko V. R. – PhD student of Design Automation Department, Kharkiv National University of Radio Electronics, Kharkiv, Ukraine.

ABSTRACT

Context. Modern video conferencing systems work in different noise environments, so preservation of speech clarity and provision of quick adaptation to changes in this environment are relevant tasks. During the development of embedded systems, finding a balance between resource consumption, performance, and signal quality obtained after noise suppression is necessary. Systems on a chip allow us to use the power of both processor cores available on the hardware platform and FPGAs to perform complex calculations, which contributes to increasing the speed or reducing the load on the central SoC cores.

Objective. To conduct a comparative analysis of the noise suppression quality in audio signals by an adaptive filtering algorithm and a filtering algorithm using machine learning based on the RNNoise neural network in noise suppression devices on the technological platform SoC.

Method. Evaluation using objective metrics and spectrogram analysis using the Librosa library in Python. Neural network training and model design are performed on the basis of Python and Torch tools. The Vitis IDE package was used for the neural network implementation on the platform SoC.

Results. The analysis of two noise suppression methods using the adaptive Wiener filter and the RNNoise neural network was performed. In the considered scenarios, it was determined that the neural network shows better noise suppression results according to the analysis of spectrograms and objective metrics.

Conclusions. A comparative analysis of the effectiveness of noise suppression algorithms based on adaptive filters and a neural network was performed for scenarios with different noise environments. The results of objective SIGMOS metrics were obtained to evaluate the quality of the received audio signal. In addition, the possibility of running the RNNoise neural network on the technological platform SoC ZYNQ 7000 was verified.

KEYWORDS: embedded systems, system-on-a-chip, FPGA, adaptive filtering, digital signal processing algorithms, noise suppression algorithms, audio signals, machine learning, neural networks.

ABBREVIATIONS

ARM is an advanced RISC machine;
CNN is a convolutional neural network;
DNN is a deep neural network;
FPGA is a field programmable gate array;
FPS is frames per second;
GRU is a gated recurrent unit;
GTCRN is group temporal convolutional recurrent network;
GZIP is a GNU's not unix ZIP;
HLS is a high level synthesis;
ISP is an ITU;
ITU is an international telecommunication union;
LPF is a low pass filter;
MOS is a mean opinion score;
PESQ is a perceptual evaluation of speech quality;
POLQA is a perceptual objective listening quality analysis;
PSTN is a public switched telephone network;
RNN is a recurrent neural network;
SNR is a signal to noise ratio;
SoC is a system on a chip;
VAD is a voice activity detector;
VISQOL is a virtual speech quality objective listener;
VoIP is a Voice over Internet Protocol.

NOMENCLATURE

a,b are coefficients of the biquadratic filter;
D_{sig} is a set of signals to be investigated;
$h(n)$ is a set of the corresponding coefficients for the given filter;
i is an index of the input signal;
j is an index of the noise sample;
k is an index of the frequency domain bin;
$k(n)$ is a noisy audio signal with *n* samples in it;
L is a number of signals to be tested;
M is a length of the window function;
N is a length of the input signal buffer;
n is a index of count/sample;
S is a maximum number of counts in the signal;

DOI 10.15588/1607-3274-2024-4-16

$w(k)$ is a window function;

$X[k]$ is a frequency domain representation of the signal;

$x(n)$ is an input signal with speech and noise environment;

$x_c(n)$ is an useful signal with n samples in it;

$y(n)$ is an output signal.

INTRODUCTION

Currently, corporate video conferencing systems are actively evolving, where the determining factors are ensuring the reliability of information transmission, improving the quality of the transmitted far-end and near-end audio and video parts, and ensuring the reliability of the communication device.

In the field of audio signal processing, one of the improvement options of the quality and analysis is adaptive filtering usage, which solves a wide range of tasks. In particular, typical tasks are noise suppression with the selection of human speech, dereverberation and echo canceling, and signal separation. The different nature of the suppressed noises complicates the design of adaptive filters due to the need to constantly update the coefficients and adjust to a specific type of noise.

In particular, it is worth noting that different noise environments can have combined types of noise (stationary and non-stationary), which makes the task of adaptive filter design even more challenging.

Examples of such noise environments and their combinations are office or street noise, industrial rooms with ventilation. These types of noise are examples of stationary noise and its combination with momentary bursts in the spectrum, which can be from a keyboard (in the case of an office) or different sources that have a similar spectrum to human speech. Due to the different nature of noise, the use of classical adaptive filtering algorithms does not give a positive result for the user.

Machine learning algorithms based on neural networks, which can determine the type of noise and perform both Voice Activity determination (the presence of a voice in an audio fragment) and noise suppression in the resulting audio stream, have gained popularity in the tasks of noise classification and separation. The use of neural networks is a complex computational task that cannot be solved on stationary hardware due to the need for both a specialized software environment and compliance with the time requirements set by the real time audio streaming industry.

The main feature of using neural networks for adaptive noise suppression is the ability to perform training on a specific sample of noises, which improves performance in a specific environment. Typical noise suppression libraries for audio processing are RNNoise and Speexdsp.

Deep learning is used in noise suppression tasks for several key reasons, mainly because it allows significantly improve results compared to traditional signal processing methods. Traditional noise suppression methods are often based on linear models or statistical assumptions about noise characteristics. Deep neural networks are able to model complex non-linear dependencies between pure signal and noise, which allows to more efficiently separate the useful signal even in difficult conditions. Deep learning models can adapt to different types of noise and signals.

They can be trained on large data sets containing examples of different recording conditions and noise types, which makes them universal in use. Unlike traditional methods, which often require the manual design of features for each specific application, deep neural networks can automatically extract relevant features from raw data. It dramatically reduces the need for feature engineering and allows the creation of more general models.

Thus, **the object of the study** is adaptive filtering procedures in digital signal processing systems.

The subject of the study are models, methods and procedures for designing adaptive digital filters on the technological platform SoC using machine learning methods.

The objective of the study is to perform a comparative analysis of the noise suppression quality in audio signals using an adaptive filtering algorithm and a filtering algorithm using machine learning based on the RNNoise neural network in noise suppression devices on the technological platform SoC.

1 PROBLEM STATEMENT

Due to the above features and requirements, one of the options to solve the problem of adaptive noise suppression is the implementation of the specified algorithms on a specialized hardware platform from the SoC family. This provides an opportunity both to implement the business-logic of the firmware, executed on the device, and to design and use specialized hardware accelerators to ensure the optimal distribution of the data processing path in the SoC, taking into account limited resources both for processing time, and for maximum system power consumption as a whole.

Let's use a set of audio signals containing different types of noise and human speech. Each input signal from the set includes a noise environment in a given ratio to human speech. Thus, the signal-to-noise ratio and the present noise environment type characterize each element of the set.

Let's denote a set of audio signals as D_{sig} where each element is characterized by a noise environment and its SNR.

The data set can be expressed as $D_{sig} = \{x_i(n)\}$, where $i = 1,2,\ldots,L$ – the number of signals, $n = 0,1,2,\ldots,S$ – the number of counts of the corresponding signal. Each i-th signal is a combination of the useful signal $x_{ci}(n)$ and the j-th noise environment $k_j(n)$ in a given signal-to-noise ratio.

Thus, the input sample of signals makes a set of combinations $x_i(n) = \langle x_{ci}(n), k_j(n) \rangle$, where i

DOI 10.15588/1607-3274-2024-4-16

corresponds to the index of the speech signal, j corresponds to the index of the noise environment signal, $x_i(n)$ corresponds to their combination with the given SNR. In every noise suppression scenario, it is necessary to suppress $k_j(n)$ as much as possible and obtain a clean output signal $y_i(n)$. According to the general filtering equation $y_i(n) = h_i(n) \cdot x_i(n)$, the final result of using both a neural network and an adaptive filter is to find the coefficients $h_i(n)$ for each of the combinations of noise environment and speech during the processing of the input audio signal.

The main task of adaptive filtering for noise suppression is the quick response of the filter to changes in the noise environment, preservation of the dynamic range of the input signal, and suppression of signal fragments where there is no human speech, i.e., quick selection of coefficients. Several filtering algorithms are designed for noise suppression, among which the adaptive Wiener filter, LMS filter, and RNNoise neural network are known.

At the same time, the output signal after filtering is characterized by a set of objective metrics of audio signal quality, such as discontinuity, noiseiness, reverberation, coloration, and overall quality assessment, which can be used for the quality characteristics of the noise suppression algorithm.

When using the RNNoise neural network, a 10-millisecond-long input signal buffer at a sampling frequency of 48 KHz must be provided to its input; that is, the input buffer should contain 480 counts of the input signal.

The main criterion for comparing different noise suppression algorithms is the analysis of the obtained objective metrics of the output signal and visual observation of audio signal spectrograms before and after noise suppression.

Thus, the tasks of the study are the selection of an adaptive filtering algorithm and a machine learning method based on a neural network for noise suppression, the determination of metrics that can be used to evaluate the result of noise suppression, and the performance of a comparative analysis of the selected methods.

2 REVIEW OF THE LITERATURE

Paper [1] considers the possibility of using primitives of modern C++ standards for high-level synthesis. Features of the latest language standards and their features during synthesis are given and considered. A comparative analysis of Catapult HLS with Vitis HLS in terms of support for modern C++ primitives during synthesis was conducted. The hardware-accelerated matrix determinant calculator was implemented using C++03 and C++17.

Paper [2] considered the features of using high-level synthesis during the development of image processing algorithms. The main points related to synthesized and non-synthesized constructions of different variants of C-code to get maximum performance are given. As an example, the implementation of the Sobel filter algorithm with the analysis of performance changes from 10 to 388 FPS was used. Analysis of changes of IP-core main code to obtain a necessary performance was done.

Paper [3] analyzes the process of developing a cluster to implement GZIP compression using the ZedBoard debugging board. A comparative analysis of the energy efficiency of the obtained implementations and the productivity of the obtained cluster was conducted based on the Wordcount and Terasort benchmarks. According to the conclusions, a two-fold improvement in the energy consumption reduction of the cluster was obtained.

In modern models the speech enhancement based on deep learning has made significant strides compared to traditional methods, but it often requires a large number of parameters and significant computing power, which makes them difficult to use on resource-constrained devices in real conditions. In paper [4], a GTCRN is considered, which uses group strategies to effectively simplify the competitive model of Dual-Path Convolution Recurrent Network. In addition, it applies subband feature extraction modules and temporal recurrent models to improve performance. The obtained model requires low computational resources, having only 23.7 thousand parameters and 39.6 million accumulation operations per second. Experimental results show that the proposed model not only outperforms RNNoise, a typical lightweight model with similar computational load, but also exhibits competitive performance compared to current baseline models that require significantly more computing resources.

Paper [5] proposes an accelerator for the Markov decision process, implemented in the AI-Toolbox public library using high-level synthesis tools using the "tiger-antelope" problem as an example. The proposed approach shows an acceleration of more than 7 times compared to the original version of the algorithm.

Hardware accelerators for deep learning algorithms is a relevant topic in the scientific and engineering environment. The use of FPGAs as embedded co-processors to accelerate algorithms has gained wide recognition.

Decision-making games are critical tasks where their real-time accuracy and efficiency directly affect the result. Traditional FPGA development using hardware description languages is associated with long development cycles, high complexity, slow iterations, and problems in fast responding to model algorithm updates. In comparison, HLS-based design of FPGA provides an appropriate technology path to eliminate these weaknesses.

Paper [6] considers the FPGA usage together with a high-level synthesis to solve problems in the field of decision-making algorithms design. A comparative analysis of the algorithm implementation on the FPGA with the software solution based on the x86 architecture is carried out. A dedicated deep learning algorithm related to decision making games was implemented, optimized and deployed on FPGA using HLS. The FPGA is

DOI 10.15588/1607-3274-2024-4-16

connected to the host PC via PCIe as a co-processor. Comparative testing with running the algorithm on an eight-core ftd2000 CPU and an Intel i9-9900k processor demonstrates that using the FPGA as a co-processor significantly reduces the execution time of the algorithm, resulting in a noticeable speedup effect.

Paper [7] presents an experimental study of the implementation of a face mask detection system based on the use of HLS and the concept of hardware and software co-design. The target platform is a Xilinx PYNQ-Z2 FPGA board that connects to the host computer and acts as a hardware accelerator for the face masks detection task. To simplify the hardware implementation complexity, the face mask detection algorithm uses the ISP approach instead of complex CNN models. The algorithm consists of color space transformation, skin color detection, morphological operations, connected component labeling and horizontal edge detection. Implementation results show that the FPS for detection can reach 18.

Deep neural networks are widely used to solve a variety of tasks: from speech recognition to image classification. Since DNNs require a lot of computing power, their hardware implementation on FPGA or ASIC has attracted significant attention. In turn HLS is widely used because it significantly increases performance and flexibility and requires minimal hardware knowledge. Paper [8] proposes DeepFlexiHLS, a two-stage design space exploration flow for searching a set of directives to achieve minimum latency. The results form a Pareto space from which the developer can choose whether his FPGA resources are limited or should not be fully utilized by the DNN module.

The growing trend of using deep learning techniques for noise suppression has led to the creation of hybrid noise suppression systems that combine classical signal processing with deep learning. For example, paper [9] focuses on expanding the RNNoise noise suppression system by including additional features during the training phase. The paper presents a detailed description of the configuration process of the modified system and the comparative results obtained from the performance analysis using the existing version of RNNoise as a benchmark.

Deep learning-based speech enhancement can provide near-best performance when processing non-stationary noise. Noise suppression methods, that combine classical signal processing with a RNN, can be implemented in real time due to their low complexity. However, in these methods long-term speech information is missed during features selection, which degrades the performance of noise suppression. This paper extends the RNN-based denoising method known as RNNoise by adding a long-term spectral difference feature. The amount of noise suppression is also limited to improve speech quality for a better compromise between noise removal level and speech distortion. The method proposed in [10] outperforms the RNNoise algorithm by 0.12 MOS points on average according to the results of the subjective audio test.

Background noise is a major source of quality degradation in VoIP and PSTN calls. Recent studies have shown the effectiveness of deep learning for noise suppression, but the datasets used were relatively small compared to other domains (e.g., ImageNet) and the corresponding evaluations were more concentrated. To better support deep learning speech enhancement research, paper [11] presents a noisy speech dataset that can scale to arbitrary sizes based on the number of speakers, noise types, and SNR levels. Increasing the dataset size shows the improvement of the noise suppression performance as expected. To demonstrate the data set and evaluation structure, subjective MOS with objective quality measures such as SNR, PESQ, POLQA, and VISQOL, was applied, and reasons of MOS relevance were demonstrated.

In teleconferencing scenarios, the speech is usually degraded by background noise, which reduces speech intelligibility and quality. Therefore, it is extremely important to improve speech in noisy environments. Paper [12] investigates a real-time speech enhancement method for a far-end signal based on an improved RNN with a unit of controlled GRU. The ideal amplitude masking values of the reverberated target speech are used as the target values for RNN training. Feature normalization is applied and subband normalization technology were proposed to reduce feature differences, which contributes to better RNN learning of long-term patterns. In addition, to suppress the residual interharmonic pseudo-steady noise due to subbanding, the work integrates RNN with the optimal modified log-spectral amplitude algorithm. Experimental results show that the proposed method improves speech quality and reduces distortion with low computational complexity for real-time operation.

Paper [13] considers the implementation of the Librosa software package for the analysis of audio data and operations on it. The Librosa library is widely used in research and development in digital signal processing and machine learning for audio data processing. The main capabilities and functions of Librosa are considered, such as loading and saving audio files, spectrogram calculation, mel-spectrograms and mel-frequency cepstral coefficients, as well as methods for tonal analysis, segmentation and extraction of sound features. Special attention is paid to functions for pre-processing of audio signals, including normalization, noise reduction and changes in playback speed.

Paper [14] considers the implementation of the Pyroomacoustics package, designed for rapid development and testing of audio signal processing algorithms using microphone arrays. The package contains three main components: an object-oriented interface on Python that allows quickly create various simulation scenarios involving multiple sound sources and microphones in 2D and 3D rooms; a fast C-implementation of a source image model for general

DOI 10.15588/1607-3274-2024-4-16

multifaceted rooms that efficiently generates room impulse characteristics and models sound propagation between sources and receivers; reference implementations of popular algorithms for beamforming, direction determination and adaptive filtering. Together, these components form a package that significantly reduces the time to implement new algorithms, reducing overhead at the performance evaluation stage.

Paper [15] considers the problem of single-channel speech enhancement in stationary conditions and proposes the use of a Wiener filter with a recursive noise estimation algorithm. The Wiener filter is a linear estimator and minimizes the root mean square error between the original and enhanced speech. The algorithm is implemented in the frequency domain and it depends on the transfer function of the filter, which varies from sample to sample based on the statistics of the speech signal, in particular the local mean and local dispersion. A recursive noise estimation approach is used for noise estimation. In this approach, noise estimation is performed based on past and present spectral power values using a smoothing parameter. The value of the smoothing parameter is chosen in the range from 0 to 1. To evaluate the effectiveness of the proposed speech enhancement algorithm, objective evaluations and informal listening of sentences from the NOIZEUS corpus, spoken by male and female voices, distorted by white and pink noise at different levels of the signal/noise, are conducted. SNR, segmental SNR and perceptual assessment of speech quality are used for objective measurements. The measurement results prove that speech, enhanced by the proposed algorithm, is more pleasing to the human ear under both noise conditions, compared to traditional speech enhancement methods.

3 MATERIALS AND METHODS

In the tasks of digital signal processing, the classic solution for noise suppression is the use of adaptive filters, which mostly work on the principle of minimizing the root mean square error between the signal with a noise component and the restored signal. Variations of Wiener adaptive filter [15] can include the analysis of harmonics in the signal, the use of VAD to determine the presence of speech. The main disadvantages of using adaptive filters are the need to determine the exact noise model and its stationarity. In many real-world situations, these characteristics can be unknown or difficult to estimate. In real conditions, signals and noise are often non-stationary, which can significantly reduce the effectiveness of the adaptive filter.

This study uses the RNNoise neural network, which is a RNN designed to work in environments with limited computing power. Neural network training and model design is performed on the basis of Python and Torch tools on a PC with x86 architecture, after which the interference (launch) of the model is performed in the environment, developed on C. The interference part of the trained model is performed on the ARM part of the SoC ZYNQ 7000 using the appropriate set of tools for bare-metal launch. The architecture of the RNNoise neural network is shown in Fig. 1.

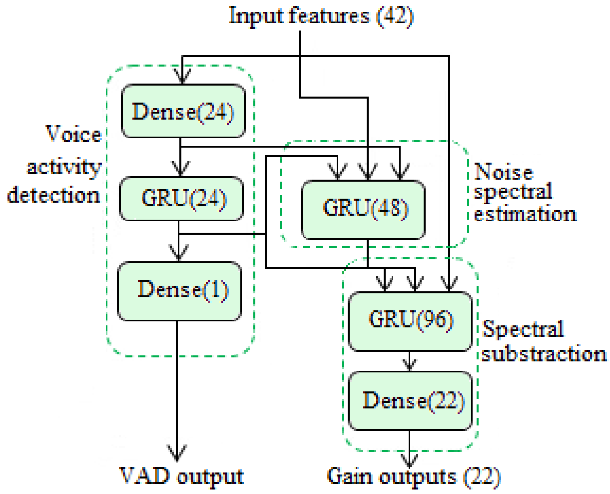


Figure 1 – General architecture of RNNoise neural network

Each block represents a layer of neurons, the number of which is indicated in parentheses. Dense layers: dense (24) and dense(1), are fully connected layers which are not repeated. One of the network outputs is a set of gains that are applied at different frequencies (gain outputs (22)). Another result is the probability of the voice activity that is not used for noise suppression, but is a useful by-product of the network, which can be further used to implement automatic signal level control after noise suppression. According to this strategy, the level equalization is performed taking into account the VAD data, which allows for additional noise suppression between speech fragments.

Note that, in general, filtering due to the use of an adaptive filter can be represented by the equation (1):

$$y(n) = h(n) \cdot x(n). \tag{1}$$

At the same time, in the case of using the RNNoise neural network, stages related to the preparation of data for processing by the neural network and data conversion from the frequency range to the time range are added to the signal processing path. RNNoise performs processing on audio frames equivalent to 10 milliseconds of audio stream with a sampling frequency of 48KHz. Signal processing includes a biquad filter, convolution with a window function and fast Fourier transform computation.

In general, the process of preprocessing of the input signal can be expressed as follows. Let’s consider an input signal buffer that contains 10 milliseconds of input data stream. In general, the input signal buffer can be represented as a one-dimensional vector with a given number of elements: X=[x(0), x(1), x(2),…, x(N–1)].

If the input signal is represented as a vector $x(t)$ and the biquadratic filter has coefficients b = [b_0, b_1, b_2] and a = [1, $-a_1$, $-a_2$], where a_0 is usually equal to 1, then

DOI 10.15588/1607-3274-2024-4-16

signal filtering using a biquadratic filter can be defined as (2):

$$x_b[n] = b_0 x[n] + b_1 x[n-1] + b_2 x[n-2] - a_1 y[n-1] - a_2 y[n-2], \quad (2)$$

where $y[n]$ – the output value of the filter, $b(0\ldots2)$ and $a(1\ldots2)$ – filter coefficients, $x[n]$ – the value of the input signal at the corresponding time.

In turn, the next stage with the convolution of the signal with a window function can be defined as (3):

$$y[n] = \sum_{k=0}^{M-1} h(n-k)\cdot w(k). \quad (3)$$

And accordingly, the fast Fourier transform using the Kiss_fft library is generally defined as (4):

$$X[k] = \sum_{n=0}^{N-1} x[n]\cdot e^{-\frac{2\pi kn}{N}}. \quad (4)$$

After that, signal analysis in the frequency domain, band gain calculation for opus frequency bands and inverse Fourier transformation are performed.

Several examples of signal and noise combinations were chosen for the study, namely road noise, street noise and crowd noise. The SNR was chosen to be 5.10 dB and 15dB, respectively.

Processing with the help of the Wiener filter was performed using the Pyroomacoustics package and its module Denoise [14].

4 EXPERIMENTS

For effective study of noise suppression with the help of the selected algorithms, the test signals of crowd noise, road noise and street noise were selected in ratios with the useful signal SNR5/SNR10/SNR15.

Spectrograms of some of the received resulting signals and a comparative table of all conducted experiments are given in the paper. As an example for further consideration, we will use an input signal composed of fragments of male and female speech with intervals between replicas about 0.8 seconds. The spectrogram of the input signal without noise is shown in Fig. 2.

An experimental study of the obtained processing speed was performed on the hardware platform ZYNQ Zedoard [16]. The analysis of spectrograms and the obtained results of noise suppression in the study were performed with the help of developed utilities based on libraries Librosa, Libsndfile, Pyroomacoustics, Pyloudnorm, Numpy and Matplotlib for the development and analysis of DSP algorithms on an ARM-based PC.

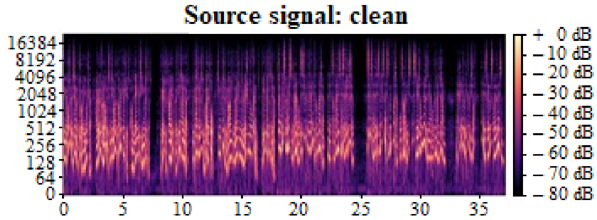


Figure 2 – Spectrogram of the input signal without noisy environment

To obtain objective signal metrics before and after processing, the SIGMOS package [17, 18] was used to determine discontinuity, noisenes, reverberation, coloration, and overall quality assessment. Objective metrics allow us to determine quickly the quality of the received signal processing results and compare algorithm implementations with each other.

During the research, objective metrics were obtained for the input signal, for the signal with noise suppression based on the adaptive Wiener filter, and for the signal with noise suppression based on the RNNoise neural network. SIGMOS metrics are calculated using a neural network that was developed to evaluate echo and noise suppression algorithms in telecommunications environments. The quality of the received signal is measured in the SIG according to ITU-T P.804 (subjective diagnostic test method for conversational speech quality analysis). SIGMOS evaluates sound quality parameters according to P.804. This model was trained using subjectively annotated data from P.804 to simulate human perception of sound quality.

5 RESULTS

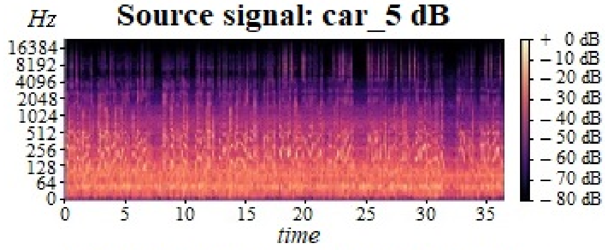
For each example of an input signal and noise combination, objective metrics were obtained and the resulting spectrogram of the signal was analyzed before and after noise suppression. Taking into account the characteristics of the subject field, it is appropriate to consider the ratio of SNR5 and SNR15 between the useful signal and the noisy environment.

In the following experiments, noise of a certain type (mixed/combined) will be added to this signal and it will be suppressed by different types of filters.

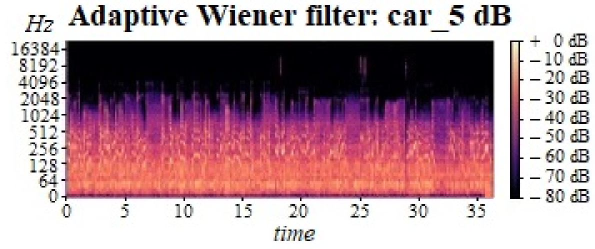
The following diagrams show the results of the obtained objective metrics for the input and cleaned signal, obtained using Librosa. The order of the spectrograms for each test scenario is as follows: the input signal, the signal after processing by the Wiener filter, and the signal processed by the neural network.

Fig. 3 shows the input speech signal with added road noise with the ratio SNR5 (Fig. 3a). Second spectrogram shows the result of noise suppression using the Wiener filter (Fig. 3b), and the third shows noise suppression using a neural network (Fig. 3c).

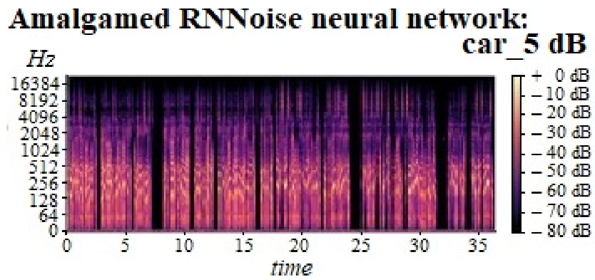
DOI 10.15588/1607-3274-2024-4-16



a



b



c

Figure 3 – Spectrograms of road noise suppression results with SNR5: a – input signal; b – noise suppression using the Wiener filter; c – noise suppression using neural network

The analysis of the second spectrogram (Fig. 3b) shows that the Wiener filter in this case worked as a LPF and slightly suppressed the fragments where there is no speech. Spectrum leaks between 15 and 30 seconds in the form of short-term high-frequency bursts are also visible.

During the analysis of the results of the objective metrics calculation, it was found out that absence of noise (namely Noiseness), is the best in the case of using a neural network. It is advisable to display the results of the SIGMOS calculation in the form of a histogram.

The data will be grouped into groups of three elements. Each group contains the results of one of the calculated MOS parameters for the input signal, the signal after processing using the Wiener filter and for the signal after processing with the RNNoise neural network. The result of objective metrics evaluation SIGMOS for the selected scenario is shown in Fig. 4.

The next stage was the analysis of noise suppression with a signal-to-noise ratio of 15dB. In this case, the neural network also performed better, which is evident from the given comparative spectrograms and objective metrics, obtained for input and output audio streams. The spectrograms of the input signal, the signal after cleaning using a Wiener filter and a neural network are represented in Fig. 5.

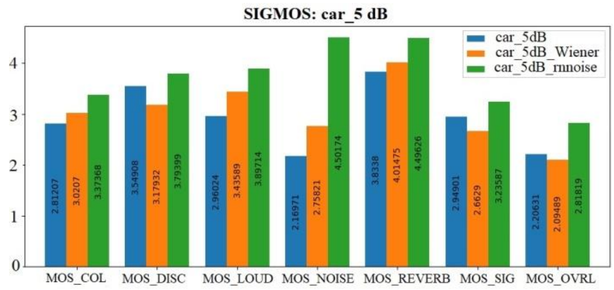
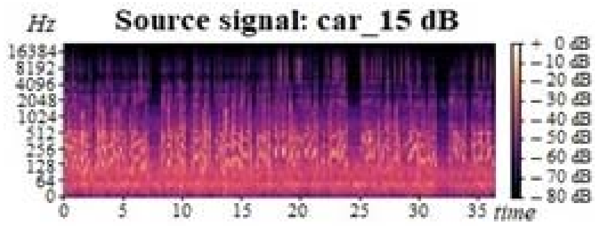
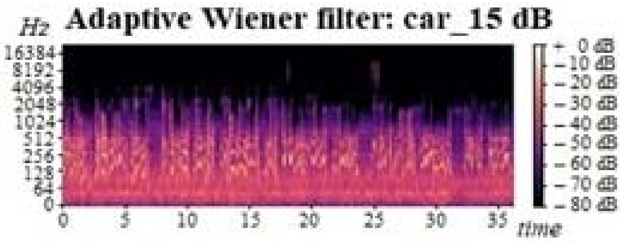


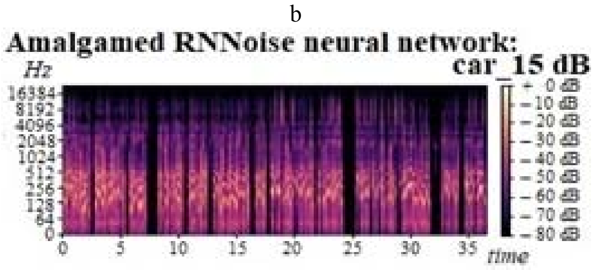
Figure 4 – SIGMOS objective metrics obtained for the corresponding human speech scenario with added road noise with SNR5



a



b



c

Figure 5 – Spectrograms of the the input signal for the scenario of human speech with added road noise with SNR15: a – input signal; b – noise suppression using the Wiener filter; c – noise suppression using neural network

It is appropriate to define that fragments are clearly visible, where there is no speech and only the noise component is present on the spectrogram after processing by the neural network. In the objective metrics for this recording, a trend with the best result of noise suppression using RNNoise, similar to the previous one, is observed. However, it should be noted that a small variation is present in the results for Loudness and Discontinuity between the input and output signals. The obtained SIGMOS objective metrics for the corresponding scenario with added road noise is shown in Fig. 6.

Next, a noise suppression analysis was performed for the scenario of added street noise to the speech with SNR5 and SNR15.

DOI 10.15588/1607-3274-2024-4-16

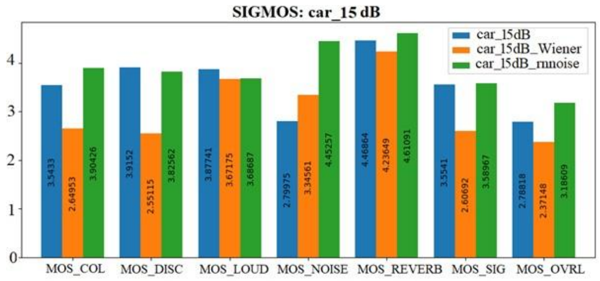
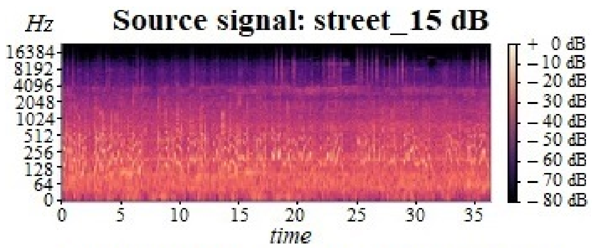
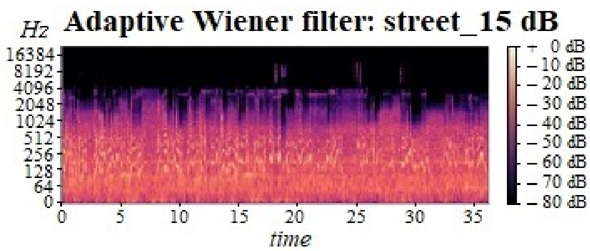


Figure 6 – SIGMOS objective metrics obtained for the corresponding human speech scenario with added road noise with SNR15

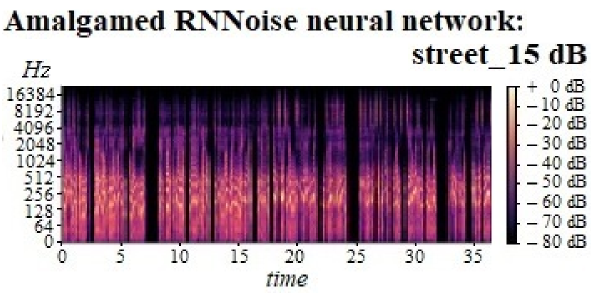
During the result analysis of obtained spectrograms, a similar trend of the Wiener filter in suppressing the high-frequency component is visible, but without detecting fragments where there is no speech. According to the obtained objective metrics, the neural network shows the best results in this scenario, compared to the Wiener filter. Fig. 7 shows the received spectrograms for the input signal, for the result of Wiener filter and RNNoise neural network processing. The results of objective metrics calculation for the scenario with added street noise with SNR5 is shown in Fig. 8.



a



b



c

Figure 7 – Spectrograms of the obtained results for the scenario with added street noise with SNR5: a – input signal; b – noise suppression using the Wiener filter; c – noise suppression using neural network

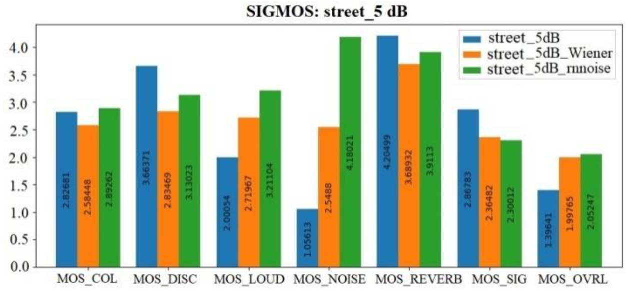


Figure 8 – SIGMOS objective metrics obtained for a human speech scenario with added street noise at SNR5

In the result of the analysis, tables, respectively, for the adaptive Wiener filter and the RNNoise neural network using Pandas tools were formed.

The obtained statistics of the objective metrics SIGMOS according to the experiment are shown in Table 1 and Table 2.

Metrics indicate the following characteristics of the received signal:

– MOS_COL (or MOS_COLORATION) is an indicator of audio coloration distortion. In the context of audio, coloration refers to a change in the sound timbre that can occur due to certain processing or recording/playback conditions;

– MOS_DISC (or MOS_DISCONTINUITY) is an indicator that characterizes the presence and degree of audio signal continuity;

– MOS_ LOUD (or MOS_LOUDNESS) is an indicator that characterizes the volume level of an audio signal. It evaluates as much as an audio volume matches the expected or desired level;

– MOS_NOISE is an indicator that evaluates the level of noise in an audio signal. It determines the degree of presence of background noise in the audio stream;

– MOS_REVERB (or MOS_REVERBERATION) is an indicator that characterizes the level of reverberation in an audio signal;

– MOS_SIG is an indicator that characterizes the quality of the audio signal itself without taking into account noise, reverberation or other distortions;

– MOS_OVRL (or MOS_OVERALL) is an overall indicator that reflects the overall quality of the audio, taking into account all the main aspects such as sound clarity, presence of noise, reverberation, signal continuity and other factors.

This is an integral indicator that combines all the other individual scores (MOS_COL, MOS_DISC, MOS_LOUD, MOS_NOISE, MOS_REVERB, MOS_SIG) to obtain a single value that reflects the overall quality of the audio signal.

SIGMOS/MOS has an absolute scale from 0 to 5, where a higher value indicates a better result for the selected metric, for example, for MOS_NOISENESS a higher value is a sign of the absence/suppression of noise.

DOI 10.15588/1607-3274-2024-4-16

Table 1 – Statistics of results for the adaptive Wiener filter

	MOS_COL	MOS_DISC	MOS_LOUD	MOS_NOISE	MOS_REVERB	MOS_SIG	MOS_OVRL
count	10	10	10	10	10	10	10
mean	2.648	2.974	3.178	2.939	3.764	2.687	2.214
std	0.246	0.407	0.332	0.603	0.326	0.185	0.195
min	2.178	2.436	2.720	2.178	3.224	2.365	1.998
25%	2.56	2.648	2.89	2.579	3.586	2.583	2.079
50%	2.606	2.891	3.182	2.842	3.668	2.687	2.155
75%	2.836	3.288	3.420	3.096	3.988	2.751	2.309
max	3.021	3.674	3.672	4.359	4.236	3.046	2.651

Table 2 – Statistics of results for the RNNoise neural network

	MOS_COL	MOS_DISC	MOS_LOUD	MOS_NOISE	MOS_REVERB	MOS_SIG	MOS_OVRL
count	10	10	10	10	10	10	10
mean	3.514	3.715	3.691	4.347	4.377	3.208	2.839
std	0.432	0.388	0.282	0.209	0.313	0.557	0.543
min	2.893	3.130	3.211	4.026	3.911	2.300	2.052
25%	3.219	3.552	3.526	4.199	4.076	2.920	2.561
50%	3.454	3.738	3.735	4.344	4.485	3.265	2.845
75%	3.766	3.883	3.905	4.453	4.589	3.383	2.965
max	4.350	4.478	4.077	4.757	4.800	4.348	4.049

The 25% and 75% values correspond to the first (Q1) and third (Q3) quartiles, respectively. These quartiles are part of descriptive statistics that show the distribution and central tendency of the data.

25% (first quartile, Q1) represents the value below which 25% of the data lies. This is the median of the lower half of the data set. This metric is useful for understanding the lower bound of the central 50% of the data.

75% (third quartile, Q3) represents the value below which 75% of the data lies. This is the median of the upper half of the data set, which is useful for understanding the upper bound of the central 50% of the data.

According to the Tables 1, 2, it is noticeable that the average value for all metrics of the neural network is higher, as well as both minimum and maximum. It is also noticeable that the noise suppression metric MOS_NOISE is 1.5 times higher than the results after filtering with an adaptive Wiener filter, which proves the effectiveness of using a neural network to suppress noise in different noisy environments, as well as MOS_COL, which indicates the coloration of the signal and the absence of tonal distortions.

Also, as part of the study, the RNNoise neural network was launched on the ARM part of the hardware platform ZYNQ. In particular, the necessary software finalizations of the source code were determined for the compilation of the interference part for the bare-metal environment. With the use of cross-compilation tools and refinement of the source code, the model execution time on the hardware platform was obtained, which allows us to perform the next stage of its performance analysis in the case of the development of a separate IP block using the Vitis-HLS toolkit.

6 DISCUSSION

Within the framework of this study, two approaches to filter the noise in an audio signal based on SoC were considered: adaptive filters and neural networks. Models with different bursts at random moments of time (non-stationary noise models) were used as a noise environment. Implemented adaptive filter and neural network for filtering signals with non-stationary noise and human speech in a given ratio were verified. The simulation results showed that both methods have significant potential for solving filtering problems, but show different results in terms of evaluation criteria.

Adaptive filters (the Wiener filter was considered) are relatively easy to implement (C, VHDL/Verilog), are effective for uniform noise, echo suppression due to the ability to achieve stable signal filtering without significant degradation in a relatively small number of counts. But, if the signal contains a non-stationary component (which is usually the real conditions of use), then the efficiency of the adaptive filter is significantly reduced.

A neural network can be implemented programmatically using both built-in accelerators, which are often found in SoCs, and with partial transfer of complex computing components to FPGA. Simulations have shown that neural networks can handle complex nonlinear dependencies much better than adaptive filters. For example, in the tasks of road and street noise filtering, where the noises had a complex structure, the RNNoise network showed a significantly better filtering result according to the objective metrics and the obtained spectrograms. Neural networks are effective in processing real-time changing signals and can also extract important features of the signal, which can reduce the need for manual tuning and data pre-processing unlike an adaptive filter. However, their implementation requires more computing resources and training time.

DOI 10.15588/1607-3274-2024-4-16

The results of the study show that the choice between adaptive filters and neural networks depends on the nature of the task. In systems, where stationary noise types predominate, adaptive filters are an appropriate choice. Their simplicity, transparency and adaptability make them suitable for processing audio signals with a relatively simple noise structure. Neural networks are appropriate for tasks where there are complex, non-linear noises and a high quality of noise suppression with maximum preservation of the useful signal is necessary. Also, their implementation in the SoC is possible due to the potential distribution of calculations between system components and due to the presence of hardware calculations accelerators, specific for neural networks, in some SoCs.

Also, it is necessary to consider the possibility of using during real-time processing of the audio stream objective metrics calculation and parallel processing in the situation, where combined approaches are possible, that is using adaptive filters and neural networks together to achieve maximum efficiency. For example, hybrid systems can use adaptive filters for primary signal processing, and neural networks – for more accurate filtering of complex noises in the second stage. Further research can be aimed at neural networks optimization for real-time operation and reduction of their requirements for computing resources and data volume.

Thus, the final choice between the specified filtering methods should be based on a detailed analysis of the specific task requirements, available computing resources, and power consumption limitations.

CONCLUSIONS

During the study two methods of noise suppression using the adaptive Wiener filter and the RNNoise neural network, considered in different noisy environments, were analyzed. The spectrograms of the output signals and the results of the objective SIGMOS metrics for the received topologies of signal transmission were obtained. During the analysis of the received spectrograms it was discovered that in scenarios with different noise environments, the Wiener filter works in a similar way to a low-pass filter, which can be partly used to suppress the high-frequency component or for restriction of the frequency range of the signal.

Better results were found in the case of using the RNNoise neural network in noise suppression tasks for typical scenarios with a noisy environment due to a fast response time to its change. According to the received statistical data, formed in table 1 and table 2, it was found that on average RNNoise shows the best results of coloration, noisiness and discontinuity.

The scientific novelty is that the method of adaptive filtering and noise suppression of the audio signal due to the predetermination of adaptive filtering coefficients in the process of machine learning based on neural networks has been further developed, which made it possible to improve the quality of noise suppression by an average of 20% for specific types of noise and increase the quality of noise suppression in different noise environments.

The practical significance of the study consists of running the RNNoise neural network on the ARM part of the hardware platform ZYNQ Zedboard using Vitis-IDE tools and obtaining objective signal metrics before and after processing using the SIGMOS package. Also, the hardware implementation of various adaptive filtering algorithms made it possible to prove the advantages of a filter based on the neural network for noise suppression in audio signals with different parameters.

Prospects for further research are related to the analysis of the possibility of the weight coefficients database formation of trained neural networks, which implement adaptive filtering for the corresponding types of noise. The computationally complex part of the neural network can be implemented using high-level synthesis and adapted for loading into the SoC. This will significantly reduce the noise suppression devices' design time and allow novice designers to use optimized and verified technical solutions for their specific tasks.

ACKNOWLEDGEMENTS

The study was completed with the consultation support from the Logitech Audio R&D. The authors express their sincere gratitude.

REFERENCES

1. Lahti S., Rintala M., Hamalainen T. D. Leveraging Modern C++ in High-level Synthesis, *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems,* 2023, Vol. 42, № 4, pp. 1123–1132. DOI: 10.1109/TCAD.2022.3193646.
2. Monson J., Wirthlin M., Hutchings B. L. Optimization techniques for a high level synthesis implementation of the Sobel filter, *International Conference on reconfigurable computing and FPGAs (ReConFig'13).* Cancun, Mexico, 9–11 December 2013, pp. 1–6. DOI: 10.1109/ReConFig.2013.6732315.
3. Plugariu O., Petrica L., Pirea R., Hobincu R. Hadoop ZedBoard cluster with GZIP compression FPGA acceleration, *11th International Conference on Electronics, Computers and Artificial Intelligence (ECAI'19).* Pitesti, Romania, 27–29 June 2019, pp. 1–5. DOI: 10.1109/ ecai46879.2019.9042006.
4. Rong X., Sun T., Zhang X., Hu Y., Zhu C. , Lu J. GTCRN: A speech enhancement model requiring ultralow computational resources, *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'2024)*. Seoul, Korea, 14–19 April 2024, pp. 971–975. DOI: 10.1109/icassp48485.2024.10448310.
5. Leiva L., Torrents-Barrena J., Vazquez M. FPGA-based accelerator for AI-toolbox reinforcement learning library, *IEEE Embedded Systems Letters*, 2023, Vol. 15, № 2, pp. 113–116. DOI: 10.1109/les.2022.3218168.
6. Fan H., Wang H., Che K., Wu Z. Design of FPGA deep neural network accelerator based on high-level synthesis, *5th International Academic Exchange Conference on Science and Technology Innovation (IAECST'23)*. Guangzhou, China, 8–10 December 2023, pp. 163–166. DOI: 10.1109/iaecst60924.2023.10502749.
7. Chang Y.-W., Huang C.-C., Hwang Y.-T. A face mask detection system based on high level synthesis and hardware software codesign, *IET International Conference on Engineering Technologies and Applications (IET-ICETA'22).* Changhua, Taiwan, 14–16 October 2022, pp. 1–2. DOI: 10.1109/ieticeta56553.2022. 9971488.
8. Riazati M., Daneshtalab M., Sjödin M. , Lisper B. DeepFlexiHLS: Deep neural network flexible high-level synthesis directive generator, *IEEE Nordic Circuits and Systems Conference (NorCAS'22)*. Oslo, Norway, 25–26 October 2022, pp. 1–6. DOI: 10.1109/norcas 57515.2022.9934617.

9. Doumanidis C. C., Anagnostou C., Arvaniti E.-S., Papadopoulou A. RNNoise-Ex: hybrid speech enhancement system based on rnn and spectral features [Electronic resource], 2021, pp. 1–5. Access mode: https://arxiv.org/pdf/2105.11813. DOI: 10.48550/arXiv. 2105.11813
10. Cheng B., Zhang G., Tao X. , Wang S., Wu N., Chen M. An improved real-time noise suppression method based on RNN and long-term speech information, *3rd international symposium on automation, information and computing (ICSPCC'22)*. Beijing, China, China, 9–11 December 2022, pp. 476–481. DOI: 10.1109/ICASSP40776.2020.9054597.
11. Reddy C.K.A., Beyrami E., Pool J., Cutler R., Srinivasan S., Gehrke J. A scalable noisy speech dataset and online subjective test framework, *International Conference "Interspeech 2019"*. Graz, Austria, 15–19 September 2019. pp. 1816–1820. DOI: 10.21437/interspeech. 2019-3087.
12. Chen B. , Zhou Y., Ma Y., Liu H. A new real-time noise suppression algorithm for far-field speech communication based on recurrent neural network, *IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC'22)*. Xi'an, China, 17–19 August 2021, pp. 1–5. DOI: 10.1109/icspcc52875. 2021.9564530.
13. Zenodo: Python library for audio and music analysis Librosa [Electronic resource], 2024. Access mode: https://zenodo.org/records/11192913.
14. Scheibler R., Bezzam E., Dokmanić I. Pyroomacoustics: A Python package for audio room simulation and array processing algorithms, *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'18*), Calgary. Canada, 15–20 April 2018, pp. 351–355. DOI: doi.org/10.1109/icassp.2018.8461310.
15. Upadhyay N., Jaiswal R. K. Single channel speech enhancement: using wiener filtering with recursive noise estimation, *Proceeding of the Seventh International Conference on Intelligent Human Computer Interaction,* 2016, Vol. 84, pp. 22–30. DOI: 10.1016/j.procs.2016.04.061.
16. Shkil A., Rahlis D., Filippenko I. , Kornijenko V., Rozhnova T. Automated design of embedded digital signal processing systems on SOC platform, *Innovative technologies and scientific solutions for industries,* 2024, No. 1 (27), pp. 192–203. DOI: https://doi.org/10.30837/ITSSI.2024.27.192
17. Naderi B., Cutler R. Subjective evaluation of noise suppression algorithms in crowdsourcing, *International Conference "Interspeech 2021"*. Brno, Czechia, 30 August –3 September 2021, pp. 2132–2136. DOI: 10.21437/interspeech.2021-343.
18. Catalin R. N., Saabas A., Cutler R., Naderi B. , Braun S., Branets S. Speech signal improvement challenge [Electronic resource], *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'24),* 2024. Access mode: https://arxiv.org/pdf/2401.14444. DOI: 10.48550/arXiv.2401.14444.

Received 23.08.2024.
Accepted 22.10.2024.

УДК 681.326

ВИКОРИСТАННЯ АДАПТИВНОЇ ФІЛЬТРАЦІЇ ТА МЕТОДІВ МАШИННОГО НАВЧАННЯ У СИСТЕМАХ ПРИДУШЕННЯ ШУМУ, РЕАЛІЗОВАНИХ НА ПЛАТФОРМІ SOC

Шкіль О. С. – канд. техн. наук, доцент кафедри автоматизації проєктування обчислювальної техніки, Харківський національний університет радіоелектроніки, Харків, Україна.

Філіпенко О. І. – канд. техн. наук, доцент кафедри інфокомунікаційної інженерії ім. В. В. Поповського, Харківський національний університет радіоелектроніки, Харків, Україна.

Рахліс Д. Ю. – канд. техн. наук, доцент кафедри автоматизації проєктування обчислювальної техніки, Харківський національний університет радіоелектроніки, Харків, Україна.

Філіпенко І. В. – канд. техн. наук, доцент кафедри автоматизації проєктування обчислювальної техніки, Харківський національний університет радіоелектроніки, Харків, Україна.

Пархоменко А. В. – канд. техн. наук, доцент кафедри програмних засобів, Національний університет «Запорізька політехніка», Запоріжжя, Україна.

Корнієнко В. Р. – аспірант кафедри автоматизації проектування обчислювальної техніки, Харківський національний університет радіоелектроніки, Харків, Україна.

АНОТАЦІЯ

Актуальність. Сучасні системи відео конференційного зв'язку працюють у різноманітному шумовому оточенні, тому актуальними завданнями є збереження чіткості мовлення та забезпечення швидкої адаптації до зміни цього оточення. При розробці вбудованих систем виникає необхідність знайти баланс між споживанням ресурсів, продуктивністю та якістю сигналу, отриманого після придушення шуму. Системи на кристалі дозволяють використовувати потужність як процесорних ядер, доступних на апаратній платформі, так і FPGA, для виконання складних обчислень, що сприяє підвищенню швидкодії або зменшенню навантаження на основні ядра SoC.

Мета. Проведення порівняльного аналізу якості придушення шуму у аудіо сигналах алгоритмом адаптивної фільтрації та алгоритмом фільтрації з використанням машинного навчання на основі нейронної мережі rnnoise в пристроях придушення шуму на технологічній платформі SoC.

Метод. Оцінка за допомогою об'єктивних метрик, аналіз спектрограм з використанням бібліотеки Librosa на Python. Навчання нейромережі та проєктування моделі виконується на основі інструментів Python та Torch. Для реалізації нейронної мережі на платформі SoC використовувався пакет Vitis IDE.

Результати. Виконано аналіз двох методів придушення шуму з використанням адаптивного фільтру Вейнера та нейронної мережі RNNoise. У розглянутих сценаріях було визначено що нейронна мережа показує кращі результати придушення шуму згідно до аналізу спектрограм та об'єктивних метрик.

Висновки. У роботі було виконано порівняльний аналіз ефективності алгоритмів придушення шуму на базі адаптивних фільтрів і нейронної мережі у сценаріях з різним шумовим оточенням. Були отримані результати об'єктивих метрик SIGMOS для оцінки якості отриманого аудіосигналу. Додатково була виконана перевірка можливості запуску нейронної мережі RNNoise на технологічній платформі SoC ZYNQ 7000.

КЛЮЧОВІ СЛОВА: вбудовані системи, системи на кристалі, FPGA, адаптивна фільтрація, алгоритми цифрової обробки сигналів, алгоритми придушення шуму, аудіо сигнали, машинне навчання, нейронні мережі.

DOI 10.15588/1607-3274-2024-4-16

ЛІТЕРАТУРА

1. Lahti S. Leveraging Modern C++ in High-level Synthesis / S. Lahti, M. Rintala, T. D. Hamalainen // IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2023. – Vol. 42, № 4. – P. 1123–1132. DOI: 10.1109/TCAD.2022.3193646.
2. Monson J. Optimization techniques for a high level synthesis implementation of the Sobel filter / J. Monson, M. Wirthlin, B. L. Hutchings // International Conference on reconfigurable computing and FPGAs (ReConFig'13), Cancun, Mexico, 9–11 December 2013. – P. 1–6. DOI: 10.1109/ReConFig.2013.6732315.
3. Hadoop ZedBoard cluster with GZIP compression FPGA acceleration / [O. Plugariu, L. Petrica, R. Pirea, R. Hobincu] // 11th International Conference on Electronics, Computers and Artificial Intelligence (ECAI'19), Pitesti, Romania, 27–29 June 2019. – P. 1–5. DOI: 10.1109/ecai46879.2019.9042006.
4. GTCRN: A speech enhancement model requiring ultralow computational resources / [X. Rong, T. Sun, X. Zhang et al.] // IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'2024), Seoul, Korea, 14–19 April 2024. – P. 971–975. DOI: 10.1109/icassp48485.2024.10448310.
5. Leiva L. FPGA-based accelerator for AI-toolbox reinforcement learning library / L. Leiva, J. Torrents-Barrena, M. Vazquez // IEEE Embedded Systems Letters, 2023. – Vol. 15, № 2. – P. 113–116. DOI: 10.1109/les.2022.3218168.
6. Design of FPGA deep neural network accelerator based on high-level synthesis / [H. Fan, H. Wang, K. Che, Z. Wu] // 5th International Academic Exchange Conference on Science and Technology Innovation (IAECST'23), Guangzhou, China, 8–10 December 2023. – P. 163–166. DOI: 10.1109/iaecst60924.2023.10502749.
7. Chang Y.-W. A face mask detection system based on high level synthesis and hardware software codesign / Y.-W. Chang, C.-C. Huang, Y.-T. Hwang // IET International Conference on Engineering Technologies and Applications (IET-ICETA'22), Changhua, Taiwan, 14–16 October 2022. – P. 1–2. DOI: 10.1109/iet-iceta56553.2022.9971488.
8. DeepFlexiHLS: Deep neural network flexible high-level synthesis directive generator / [M. Riazati, M. Daneshtalab, M. Sjödin, B. Lisper] // IEEE Nordic Circuits and Systems Conference (NorCAS'22), Oslo, Norway, 25–26 October 2022. – P. 1–6. DOI: 10.1109/norcas57515.2022.9934617.
9. RNNoise-Ex: hybrid speech enhancement system based on rnn and spectral features [Electronic resource] / [C. C. Doumanidis, C. Anagnostou, E.-S. Arvaniti, A. Papadopoulou]. – 2021. – P. 1–5. – Access mode: https://arxiv.org/pdf/2105.11813. DOI: 10.48550/arXiv.2105.11813.
10. An improved real-time noise suppression method based on RNN and long-term speech information / [B. Cheng, G. Zhang, X. Tao et al.] // 3rd international symposium on automation, information and computing (ICSPCC'22), Beijing, China, China, 9–11 December 2022. – P. 476–481. DOI: 10.1109/ICASSP40776.2020.9054597.
11. A scalable noisy speech dataset and online subjective test framework / [C.K.A. Reddy, E. Beyrami, J. Pool et al.] // International Conference "Interspeech 2019", Graz, Austria, 15–19 September 2019. – P. 1816–1820. DOI: 10.21437/interspeech.2019-3087.
12. A new real-time noise suppression algorithm for far-field speech communication based on recurrent neural network / [B. Chen, Y. Zhou, Y. Ma, H. Liu] // IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC'22), Xi'an, China, 17–19 August 2021. – P. 1–5. DOI: 10.1109/icspcc52875.2021.9564530.
13. Zenodo: Python library for audio and music analysis Librosa [Electronic resource]. – 2024. – Access mode: https://zenodo.org/records/11192913.
14. Scheibler R. Pyroomacoustics: A Python package for audio room simulation and array processing algorithms / R. Scheibler, E. Bezzam, I. Dokmanić // IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'18), Calgary, Canada, 15–20 April 2018. – P. 351–355. DOI: 10.1109/icassp.2018.8461310.
15. Upadhyay N. Single channel speech enhancement: using wiener filtering with recursive noise estimation / N. Upadhyay, R. K. Jaiswal // Proceeding of the Seventh International Conference on Intelligent Human Computer Interaction. – 2016. – Vol. 84. – P. 22–30. DOI: 10.1016/j.procs.2016.04.061.
16. Автоматизоване проєктування вбудованих систем цифрового оброблення сигналів на платформі SoC / [О. С. Шкіль, Д. Ю. Рахліс, І. В. Філіпенко та ін.] // Сучасний стан наукових досліджень та технологій в промисловості. – 2024. – No. 1 (27). – С. 72–83. DOI: https://doi.org/10.30837/ITSSI.2024.27.192
17. Naderi B. Subjective evaluation of noise suppression algorithms in crowdsourcing / B. Naderi, R. Cutler // International Conference "Interspeech 2021", Brno, Czechia, 30 August – 3 September 2021. – P. 2132–2136. DOI: 10.21437/interspeech.2021-343.
18. Speech signal improvement challenge [Electronic resource] / [R. N. Catalin, A. Saabas, R. Cutler et al.] // IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'24). – 2024. – Access mode: https://arxiv.org/pdf/2401.14444. DOI: 10.48550/arXiv.2401.14444.

DOI 10.15588/1607-3274-2024-4-16