Deep Neural Networks for the Recognition and Classification of Heart Murmurs Using Neuromorphic Auditory Sensors

Juan P. Dominguez-Morales, *Member, IEEE*, Angel F. Jimenez-Fernandez, *Member, IEEE*, Manuel J. Dominguez-Morales, and Gabriel Jimenez-Moreno, *Member, IEEE*

***Abstract*—Auscultation is one of the most used techniques for detecting cardiovascular diseases, which is one of the main causes of death in the world. Heart murmurs are the most common abnormal finding when a patient visits the physician for auscultation. These heart sounds can either be innocent, which are harmless, or abnormal, which may be a sign of a more serious heart condition. However, the accuracy rate of primary care physicians and expert cardiologists when auscultating is not good enough to avoid most of both type-I (healthy patients are sent for echocardiogram) and type-II (pathological patients are sent home without medication or treatment) errors made. In this paper, the authors present a novel convolutional neural network based tool for classifying between healthy people and pathological patients using a neuromorphic auditory sensor for FPGA that is able to decompose the audio into frequency bands in real time. For this purpose, different networks have been trained with the heart murmur information contained in heart sound recordings obtained from nine different heart sound databases sourced from multiple research groups. These samples are segmented and preprocessed using the neuromorphic auditory sensor to decompose their audio information into frequency bands and, after that, sonogram images with the same size are generated. These images have been used to train and test different convolutional neural network architectures. The best results have been obtained with a modified version of the AlexNet model, achieving 97% accuracy (specificity: 95.12%, sensitivity: 93.20%, PhysioNet/CinC Challenge 2016 score: 0.9416). This tool could aid cardiologists and primary care physicians in the auscultation process, improving the decision making task and reducing type-I and type-II errors.**

***Index Terms*—Audio processing, Caffe, convolutional neural networks, deep learning, heart murmur, neuromorphic sensor, pattern recognition.**

This work was supported by the Spanish government under Grant (with support from the European Regional Development Fund) COFNET (TEC2016-77785-P). The work of J. P. Dominguez-Morales was supported by a Formación de Personal Universitario Scholarship from the Spanish Ministry of Education, Culture and Sport. This paper was recommended by Associate Editor S. Renaud. *(Corresponding author: Juan P. Dominguez-Morales.)*

The authors are with the Robotic and Technology of Computers Laboratory, Department of Architecture and Technology of Computers, University of Seville, Seville 41012, Spain (e-mail: jpdominguez@atc.us.es; ajimenez@atc.us.es; mdominguez@atc.us.es; gaji@atc.us.es).

I. Introduction

HEART disease is a major health problem and is one of the main causes of death in the world. Cardiovascular disease (CVD) causes nearly half of the deaths in Europe (48%) [1] and 34.3% in America (1 in 2.9 deaths in the United States) [2]. Detecting CVDs at an early stage is crucial for applying the corresponding treatment and reduce the potential risk factors. Auscultation is one of the most used techniques for this purpose, and can provide clues to the diagnosis of many cardiac abnormalities by listening and analyzing the heart sound components using a stethoscope. It is very cheap and requires minimal equipment. However, physicians need extensive training and experience for auscultating [3]. Moreover, the accuracy rate of primary care physicians and medical students on the auscultation process is between 20–40%, as reported in [4]–[7], and only roughly 80% is achieved by expert cardiologists [4], [7], [8].

Heart murmurs are sounds produced when blood flows across one of the heart valves that are loud enough to produce audible noise. Murmurs may be harmless (innocent), which are primarily due to physiologic conditions outside the heart, or abnormal, which may be a sign of a more serious heart condition or a structural defect in the heart itself. The most common problems that cause abnormal heart murmurs are mitral or aortic stenosis and mitral or aortic regurgitation. The sounds can also be categorized by timing, into systolic and diastolic, differing in the part of the heartbeat on which they can be heard (between the S1 and S2 heart sounds, or starting at or after S2 and ending before or at S1, respectively).

Heart murmurs are the most common abnormal finding when a patient visits the physician for auscultation. A heart murmur does not necessarily lead to having a CVD; it could be an innocent murmur instead of a pathological one, which does not represent current or future illness. The physician must decide if the patient is healthy or not, but, due to the fact that the accuracy is not great, the expert could be wrong, making type-I or type-II errors. A type-I error (alpha error) is the detection of an effect that is not present (i.e., healthy patients are sent for echocardiogram), while a type-II error (beta error) is failing on the detection of an effect that is present (i.e., pathological patients are sent home without medication or treatment). It is clear that, in this case, type-II errors are more important to avoid.

TABLE I
COMPARATIVE STUDY BETWEEN STATE-OF-THE-ART STUDIES ABOUT HEART SOUND DIAGNOSIS SYSTEMS

Ref.	Preprocessing	Classification method	Classes	Results	No. of samples
[9]	-Segmentation -Alignment -Spectrogram	ANN[1]	3: normal, AS[a] or AR[b]	85% (using simulated heart sounds) 48.7% (using real ones)	Train: 24 per class Test: 7 normal, 4 AS[a], 2 AR[b]
[10]	-Lowpass filtering -Segmentation (manually) -S-Transform	MLP[2] ANN[1]	5: normal, AS[a], AR[b], MS[c] or MR[d]	98% (using simulated heart sounds)	Train: 30 per class Test: 20 per class
[12]	-Wavelet transform -Normalization -Segmentation -Normalized Average Shannon Energy -Envelope extraction algorithm	FNNSL[3]	2: normal or abnormal	100% (using real heart sounds)	Train: 1 normal and 10 abnormal Test: 2 normal and 2 abnormal
[14]	-Segmentation -Trimmed Mean Spectrogram	PNN[4]	2: normal or abnormal	96.3% (using real heart sounds)	Train: 25 non-pathological, 36 pathological Test: 18 non-pathological, 37 pathological
[15]	-Band-pass filtering -Short-time Fourier transform -Features manually extracted from spectrogram -Interesting areas were manually selected	Statistical analysis	2: innocent or pathological	90.9% (using real heart sounds)	447 innocent, 272 pathological
[16]	-Short-time Fourier transform -Mean value of the signal segments -Band-pass filtering -Segmentation	SVM[5]	2: innocent or pathological	96.07% (using real heart sounds)	Train: 20 innocent, 20 pathological Test: 5 innocent, 5 pathological
[17]	-Wavelet decomposition -Feature detection using MIR toolbox from Matlab	Statistical analysis	7: normal, PDA[e], PS[f], MR[d], ASD[g], MS[c] or TOF[h]	85.08% (using real heart sounds)	20 normal, 9 PDA[e], 6 PS[f], 12 MR[d], 13 ASD[g], 17 MS[c], 13 TOF[h]
[18]	-Resampling -Band-pass filtering -Segmentation	AdaBoost + CNN[6]	2: normal or abnormal	94.24% sensitivity and 77.81% specificity. PhysioNet/CinC Challenge 2016 score: 86.02% (using real heart sounds)	Train: 2575 normal, 665 abnormal Test: 984 normal, 153 abnormal (PhysioNet/CinC Challenge 2016 dataset)
[19]	-18 features extracted from time, frequency and time-frequency domains based on a wrapper feature selection scheme.	Ensemble of SVMs[5]	2: normal or abnormal	86.91% sensitivity and 84.90% specificity. PhysioNet/CinC Challenge 2016 score: 85.90% (using real heart sounds)	Train: 2301 normal, 570 abnormal Test: 984 normal, 153 abnormal (PhysioNet/CinC Challenge 2016 dataset)

[1]: Artificial Neural Network. [2]: Multilayer Perceptron. [3]: Fuzzy Neural Network with Structure Learning. [4]: Probabilistic Neural Network. [5]: Support Vector Machine. [6]: Convolutional Neural Network.
[a]: Aortic Stenosis. [b]: Aortic Regurgitation. [c]: Mitral Stenosis. [d]: Mitral Regurgitation. [e]: Patent Ductus Arteriosus. [f]: Pulmonary Stenosis.
[g]: Atrial Septal Defect. [h]: Tetralogy of Fallot.

However, echocardiograms cost between $750 and $1500 [4] per patient, making type-I errors also important to avoid. The probability of needing this costly procedure could be reduced for both healthy people and pathological patients if a reliable (with a high accuracy rate) diagnostic tool were available as an aide for physicians.

The classification of heart sounds is not a new topic. Many studies have worked toward designing practical murmur classifier systems to improve the diagnostic accuracy of physicians. Most of them use neural networks (NNs), support vector machines (SVMs) or some complex preprocessing algorithms to carry out this task [7]–[17], [18], [19]. Many studies like [10], [11], [15] have used a processing step where a person selects the best portion of the sound signal that should be used as input to the system, making this solution not ideal for a real scenario because of the need of human interaction. Some of them have used NNs to classify between different kinds of heart murmurs [7], [9]–[11], but have only trained the network with simulated heart sounds with no noise, obtaining very bad accuracy results when testing the classifier with real heart sounds (48.5%). Others have used only a small amount of real heart sounds [10], [12], [14]–[17], which is not representative when it comes to testing it in a real scenario. Table I summarizes the main information about the preprocessing and the classification steps that have been performed in some of the state-of-the-art studies that have been discussed in this section, along with the two leading approaches from the PhysioNet/CinC Challenge 2016. Works like [20] use similar preprocessing techniques and classification algorithms, but focusing on cough sounds identification instead of heart murmurs.

The main aim of this work is to develop a classifier system using a Convolutional Neural Network (CNN) that accepts heart sound recordings directly after preprocessing the information, and classifies the input to identify if the person whose heart sound is acquired, is either a healthy person or a pathological patient. The preprocessing step automatically divides the heart sound recordings into windows of a specific time length. Heart murmurs are located in the 195 Hz band [7], [9], but can reach up to 700 Hz [10], [21], which confirms that they can be identified and extracted from the heart sound signal in the frequency domain. For this purpose, these segments of the original sound are sent to a Neuromorphic Auditory Sensor (NAS) [22], which tries to mimic the way in which the inner ear works, decomposing the audio into frequency bands, and packetizes the information using the Address-Event Representation (AER) communication protocol [23]. Then, this information is converted to sonogram images, which are then used as input to the CNN for further classification using deep learning algorithms.

The rest of the paper is structured as follows: Section II presents an overview of the system architecture using a block diagram to explain each of the components in it. Then, Section III describes the Neuromorphic Auditory Sensor (NAS) [22] and how its output information is saved into AEDAT files [24] in the computer using a USBAERmini2 board [25]. After this, in Section IV, the dataset acquisition is explained, describing the heart sound database that has been used in this work. Section V presents the preprocessing algorithms executed using the data before applying them as input to the classifier system. **Caffe** [26], which is one of the most used deep learning frameworks, is described in Section VI along with the Convolutional Neural Networks (CNNs) that have been trained and tested in order to classify the heart sound dataset. Section VII presents the classification results and the comparison between the different experiments that have been carried out in this work. Finally, the conclusions of this work are presented in Section VIII.

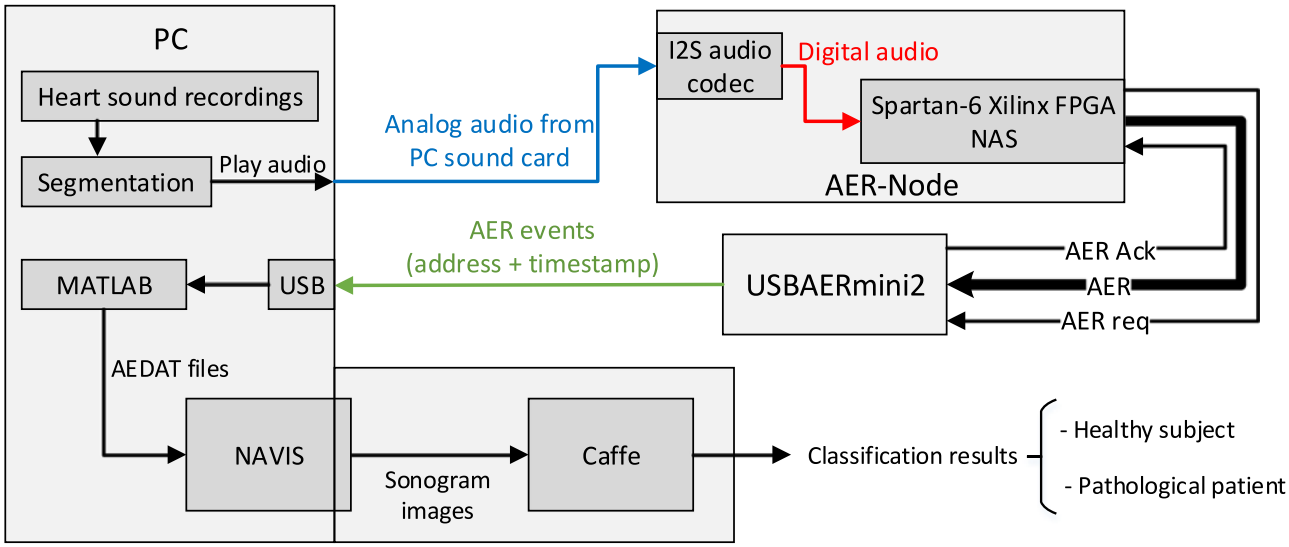


Fig. 1. Block diagram of the system architecture.

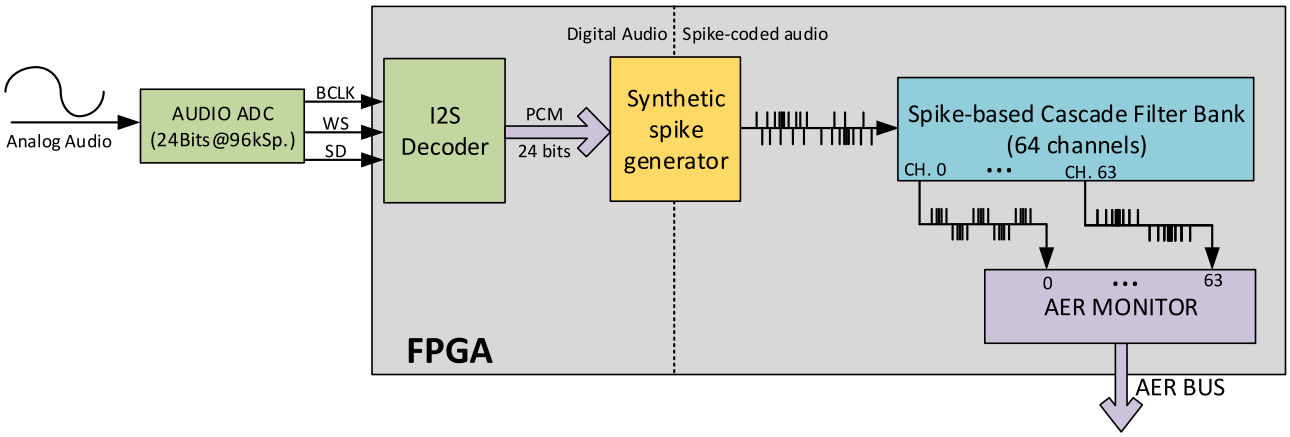


Fig. 2. Mono-aural Neuromorphic Auditory Sensor for FPGA with an I2S audio ADC and AER interface.

II. System Overview

The system consists of different modules and steps to achieve its purpose. Most of them are carried out in the computer; however, one of the most important parts of the preprocessing is done outside of it, on a Field Programmable Gate Array (FPGA). A block diagram of the whole system is presented in Fig. 1.

The heart sound recordings used in this work are obtained from the PhysioNet/CinC Challenge database [27]. It consists of 3,126 heart sound recordings, lasting from 5 to over 120 seconds. The main idea is that, after some preprocessing functions are applied to the information, one image is obtained for each of the audio samples contained in the dataset, so that it could be used as input to feed the CNN. In order to generate images with the same width and height, the first preprocessing step is to divide the heart sound recordings into segments of the same length. In this work, the accuracy of the system has been tested using segmentation windows of 1, 1.25 and 1.5 seconds (without overlapping), which were chosen because they are large enough to contain the information from, at least, one full cardiac cycle, but at the same time small enough to generate as many samples as possible.

After this process is completed (generating 77573, 61518 and 51009 samples when using a segmentation window of 1, 1.25 and 1.5 seconds length, respectively), audio samples are sent to the audio input of an AER-Node platform [28]. A 64-channel mono NAS (Neuromorphic Auditory Sensor) [22] is programmed on the Spartan-6 FPGA that the AER-Node board has, which decomposes the audio signal into frequency bands and packetizes the information using the AER (Address-Event Representation) protocol [23]. An USBAERmini2 board [25] receives this information and sends it to the computer through a USB port. Then, a script running on MATLAB collects the AER packets received and stores them into AEDAT files [24] (one file per audio sample), which is the standard format used for storing this kind of information.

A grayscale sonogram image is generated for each AEDAT file using Neuromorphic Auditory VISualizer Tool (NAVIS) [29], which is a desktop software application that is able to load AEDAT files and postprocess the information obtained from the NAS, generating useful charts like the cochleogram, sonogram, histogram, etc. The whole set of images obtained are then divided into three different datasets: one for training the

CNN (75% of the total amount of images), a second one for validation (15%) and the last one to test the CNN and obtain the accuracy ratio of the system (10%). Different CNN models have been trained and tested using **Caffe** and their accuracy results have been compared. Each of these elements and steps will be described in detail in the next sections.

III. Neuromorphic Auditory Sensor

Neuromorphic Auditory Sensor (NAS) is an audio sensor for FPGAs inspired by Lyon's model of the biological cochlea [30]. This sensor is able to process an excitatory audio signal using Spike Signal Proccesing (SSP) techniques [31], decomposing incoming audio in its frequency components, and providing this information as a stream of events using the Address-Event Representation (AER) [23]. Current state-of-the-art of silicon cochleae process audio in an analog way [32], using a bank of low-pass filters (modeling the basilar membrane), and convert the filters' output to spikes (modeling the inner hair cells). However, NAS works in the opposite way: first, it converts the incoming audio to spikes, and directly processes these spikes using a Spike Low-pass Filter (SLPF) bank with a cascade topology. Due to the use of SSP filters, circuits are very simple and do not need complex operating units or dedicated resources (e.g. floating point ALUs, hardware multipliers, RAM memory, etc...). As a consequence, NAS designers are able to replicate SLPFs in low-cost FPGAs, building large scale NAS with a low-clock frequency working fully in parallel.

To digitalize audio signals we use a commercial analog-to-digital audio converter (CS5344, with a resolution of 24 bits and a sample rate of 96 kSamples/sec.), that provides the audio samples using an I2S bus. Inside the FPGA, audio samples from the I2S bus are decoded to 24 bits digital words with two's complement. Digital audio samples are written in a synthetic spike generator (SSG), which provides a spike stream with a frequency that is proportional to the digital amplitude. These spikes are used as input to a bank of 64 SSP filters with a cascade topology, known as Cascade Filter Bank (CFB), which processes audio spikes decomposing them in frequency. Finally, output spikes from CFB are connected to an AER-Monitor [33]. This gives a unique address to the fired spikes following the Address-Event Representation, and propagates them using an asynchronous AER bus. Fig. 2 shows the block diagram of the architecture of a mono-aural NAS.

A 64-channel mono-aural NAS for FPGA with a cascade topology has been used together with a USB-AERmini2 interface [25], as can be seen in Fig. 3. NAS response is stored as AEDAT files and the output information can be seen in the second image (b) of Fig. 4, where each dot corresponds to an event that has been fired in a particular AER address at a specific time.

IV. Dataset Acquisition

The heart sound dataset used in this work contains the recordings used in the PhysioNet/CinC Challenge 2016 [34], [27], which comprises nine heart sound databases from different research groups. Heart sound recordings were sourced from several contributors around the world from both healthy subjects

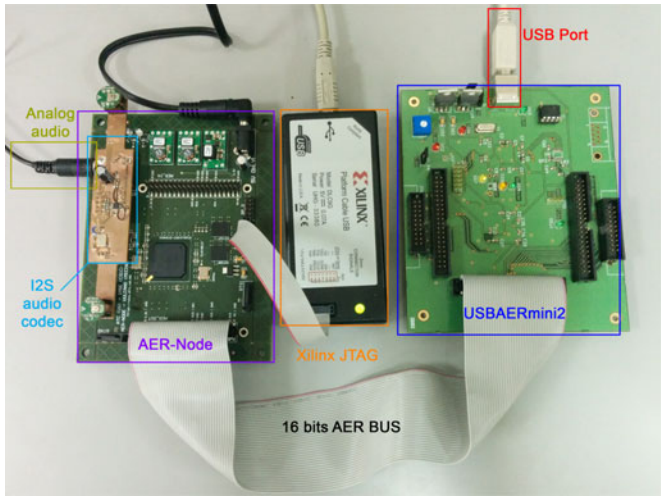


Fig. 3. NAS connected to an USBAERmini2.

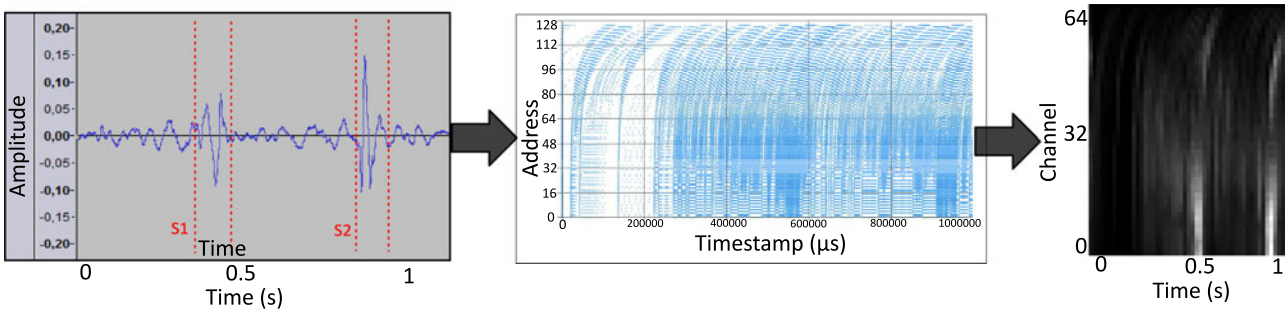
and pathological patients including children and adults, and contains a total of 3,126 heart sound recordings, lasting from 5 seconds to over 120 seconds. The heart sound recordings were collected from different locations on the body: aortic area, pulmonic area, tricuspid area and mitral area. These recordings are divided into two types: normal and abnormal heart sound recordings. The normal recordings were from healthy subjects and the abnormal ones were from patients with a confirmed cardiac diagnosis, which is not specified, but typically they are coronary artery diseases and heart valve defects like mitral valve prolapse, mitral regurgitation, aortic stenosis and valvular surgery.

Audio recordings were resampled to 2000 Hz and have been divided into three different sets of mutually exclusive populations, using 75% of them to train the network, 15% for validation and 10% to test the network. These recordings are not clean and contain noise from various sources due to the uncontrolled environment, such as talking, breathing, stethoscope motion and intestinal sounds, which is important to note because training the system with these real sounds will make it more robust and noise tolerant.

Using only 75% of the samples that this dataset has (which is only a total of 2345 heart recordings) for training the CNN is not sufficient if we want our system to be robust enough for a test with different recordings that are not included in that collection. Moreover, working with audio files with variable lengths is neither appropriate nor optimal for training a CNN: dividing these files into shorter ones (in terms of duration) would generate more samples that could be used to both train and test the network, making the system more reliable. For this purpose, the heart recordings obtained from the PhysioNet dataset were segmented using a fixed window length. The segmentation is one of the steps that have been carried out in the preprocessing phase, which is described in the next section.

V. Preprocessing of the Information

Sound recordings from the PhysioNet database do not have the same length (each file lasts from 5 to 120 seconds) and CNNs need the input images to have the same width and height for



(a) (b) (c)

Fig. 4. Outputs of the different preprocessing steps: the first image (a) is the original audio signal after the segmentation process; the second one (b) is the AER information obtained from the NAS' output; and the last one (c) is the grayscale sonogram image obtained with NAVIS, where a whiter tone in a specific section means that that section has more activity. (a) Audio signal. (b)Cochleogram. (c) Sonogram.

Algorithm 1: Sonogram calculation.

```
integPeriod = 20 ms
sonogram = zeros(max(in_addr), max(in_tStamp)/
  integPeriod)
for i = 1: max(in_addr) do
    sonogram(in_addr(i), in_tStamp(i)/integPeriod)++
end for
```

training and testing the network. For this purpose, a segmentation algorithm is applied to each of the samples before sending the audio signal to the NAS' audio input connector. In this work, different experiments have been carried out, using 1, 1.25 and 1.5 second-long windows in the segmentation process, obtaining 77573, 61518 and 51009 samples, respectively. This way, the number of samples available is also increased (more than 16 times the amount of samples in the default heart recordings database), which will provide more information in the training process of the CNN (these algorithms need a huge amount of images to train the system more robustly). Each of these three datasets has been used to feed different CNN models and the classification results are presented in Section VII.

These length values were selected due to the fact that they can contain the information from a full cardiac cycle at least (from the phase of relaxation diastole to the phase of contraction systole; or, in terms of sound, the whole "lub-dub" sequence including S1 and S2).

As was presented in the introduction (Section I), heart murmurs are located in the 195 Hz band [9], but can reach up to 700 Hz [21], which confirms that they can be identified and extracted from the heart sound signal in the frequency domain. For this purpose, each of the audio segments obtained from the original sound in the previous step are sent to a NAS, which mimics the way in which the inner ear works, decomposing the audio into frequency bands, and packetizes the information using the AER communication protocol. These packets are sent to the computer through a USB port using the USBAERmini2 board. A script in MATLAB is then used to generate an AEDAT file, which is the standard format used for storing this kind of information, for each of the audio samples. These files contain information about the address and timestamp of every event that has been fired in the NAS when feeding its input with an analog audio signal.

NAVIS is a GPL-licensed desktop software application that allows to post-process the information obtained from a NAS. This tool implements a set of charts that allows to represent the auditory information as cochleograms, histograms and sonograms, among others. It can also split the auditory information into different sets depending on the activity level of the spike streams. Due to the open-source nature of the project [35], it has been modified to automatically take the AER information contained in the AEDAT files that were obtained after sending each of the segmented samples to the NAS, and generate grayscale sonogram images based on the activity levels of the sound recordings in the frequency domain across the NAS' channels.

The pseudocode shown in Algorithm 1 presents the algorithm that has been used to calculate the sonogram's matrix of values (pixels of the image). These values are then normalized between 0 and 255, and a grayscale tone is set based on each value (0 being black, and 255 being white). Image (c) in Fig. 4 shows the output sonogram from one of the 1 second-long heart sound recordings.

The whole preprocessing step can be seen in Fig. 4. The first image (a) shows the audio signal that corresponds to one of the 1-second samples after being segmented from the original heart recording. Then, the second one (b) is the cochleogram of the information contained in the AEDAT file that was obtained after sending the audio signal to the NAS and capturing the output information using MATLAB and the USBAERmini2 board. Each dot of the cochleogram is an event that has been fired for a particular AER address (there are 128 addresses in a 64-channel mono NAS: each channel has two addresses, for positive and negative spikes) at a specific time (timestamp). The sonogram of the AEDAT file (c) was calculated using the equation that was previously described, resulting in a grayscale image with a width of 50 pixels (using time windows of 20000 μs in length for integrating the information) and a height of 64 pixels (the number of both negative and positive spikes from the same channel add up).

VI. Caffe

Caffe (Convolutional Architecture for Fast Feature Embedding) is a customizable framework for state-of-the-art deep learning algorithms. It allows to train and deploy general pur-

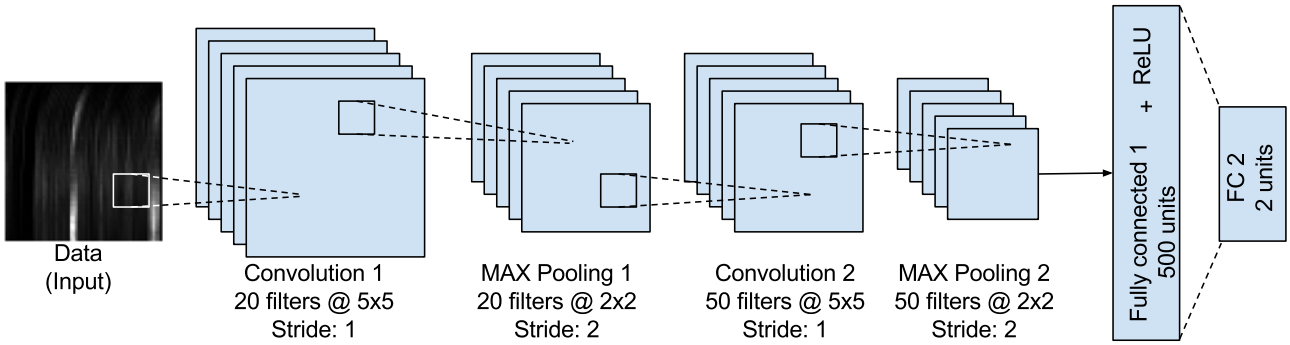


Fig. 5. Block diagram of the LeNet-5 model architecture.

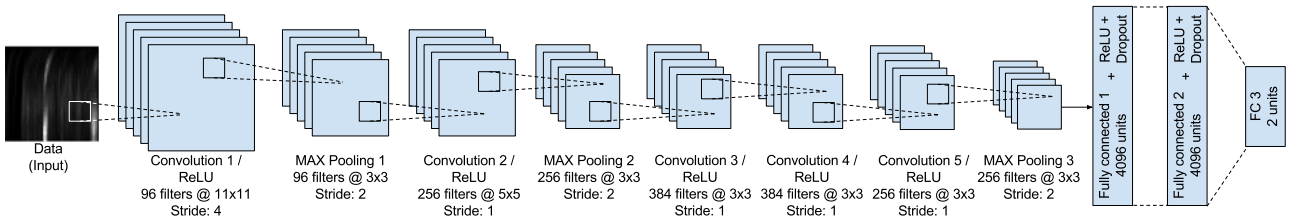


Fig. 6. Block diagram of the AlexNet model architecture.

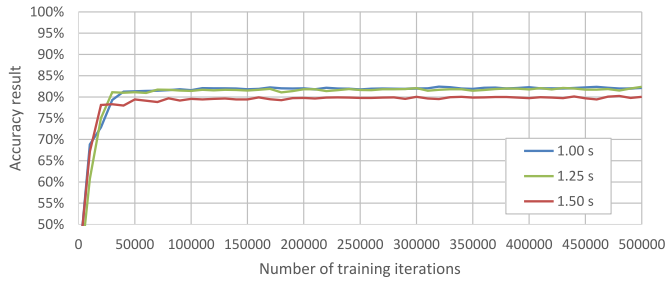


Fig. 7. Accuracy results achieved for each dataset (1s in blue, 1.25 s in green and 1.5 s in red) per 10000 training iterations using the default LeNet-5 model. Accuracy ratios obtained after 500000 training iterations: 82.11%, 82.39% and 80.00%, respectively.

pose CNNs and other deep models efficiently and in an easy way. **Caffe** is capable of processing over 40 million images a day on a single K40 or Titan GPU (∼2.5 ms per image) thanks to CUDA GPU computation. It has been used in many research fields like vision, speech recognition, robotics, neuroscience and astronomy.

Caffe provides a complete toolkit for training, testing and deploying models, which can be described using the BSD-licensed C++ library with Python and Matlab bindings. The framework also provides a collection of reference models and well-documented examples for all of these tasks, including the "AlexNet" ImageNet model [36] and the "LeNet" MNIST model [37]. These models can be modified, allowing to add/remove layers to/from the network, change the input dataset format and train it with different activation functions and parameters, which are already implemented. **Caffe** model definitions are written using the Protocol Buffer language [38], which is a language-neutral platform-neutral and easy to use mechanism for serializing structured data.

In this work, a modified version of the LeNet-5 CNN [37] has been used, where the number of outputs has been changed to two, as the goal is to distinguish between two classes: healthy subject and pathological patient. This model was designed for handwritten and machine-printed character recognition, but it is also well known for its high accuracy results for image recognition and feature extraction. Many studies have used this model for a wide variety of purposes, like freehand sketch recognition [39], Alzheimer's disease recognition [40] or even horse gait classification [41], obtaining very good results.

Fig. 5 shows the block diagram representation of the LeNet-5 model. The input dataset and the input image size have been set to match our requirements.

Several tests have been performed, using different values on some of the parameters of the Solver Prototxt file (which is the file that contains the network's training configuration) for each of the three datasets that were obtained after the preprocessing step (using 1 second, 1.25 seconds and 1.5 seconds audio length windows on the segmentation phase). The parameters that have been changed from the Solver Prototxt file are: (1) the base learning rate of the network (base_lr); (2) the momentum, which indicates how much of the previous weight will be retained in the new calculation (momentum); (3) the weight decay, which is the factor of penalization of large weights (weight_decay); (4) the test interval, which has been set to 10000 training iterations (test_interval); (5) the number of test iterations that should occur per test_interval (test_iter), to match the number of samples that the dataset has; and (6) the maximum training iterations, indicating when the network should stop training, which has been set to 500000 (max_iter). These parameters were optimized by repetition and comparison. The solver mode has been changed from CPU to GPU, due to the fact that the training process has been carried out using a NVIDIA GeForce GTX 1060 with 6GB of GDDR5 memory, and CUDA Toolkit 8.

TABLE II
TRAINING PARAMETERS AND LAYER CONFIGURATIONS FOR EACH OF THE CNNS USED

	base learning rate	learning policy	momentum	weight decay	Conv. layers	Pool. layers
Default LeNet-5	0.01	inv	0.9	0.0005	-Kernel sizes: 5 and 5 -Strides: 1 and 1	-Kernel sizes: 2 and 2 -Strides: 2 and 2
Modified LeNet-5	0.013	inv	0.6	0.000875	-Kernel sizes: 3 and 3 -Strides: 1 and 1	-Kernel sizes: 2 and 2 -Strides: 1 and 1
Default AlexNet	0.01	step (10000 iter)	0.9	0.0005	-Kernel sizes: 11, 5, 3, 3 and 3 -Strides: 4, 1, 1, 1 and 1	-Kernel sizes: 3, 3 and 3 -Strides: 2, 2, and 2
Modified AlexNet	0.013	step (10000 iter)	0.6	0.000875	-Kernel sizes: 3, 3, 3, 3 and 3 -Strides: 2, 1, 1, 1 and 1	-Kernel sizes: 3, 3 and 3 -Strides: 1, 1, and 1

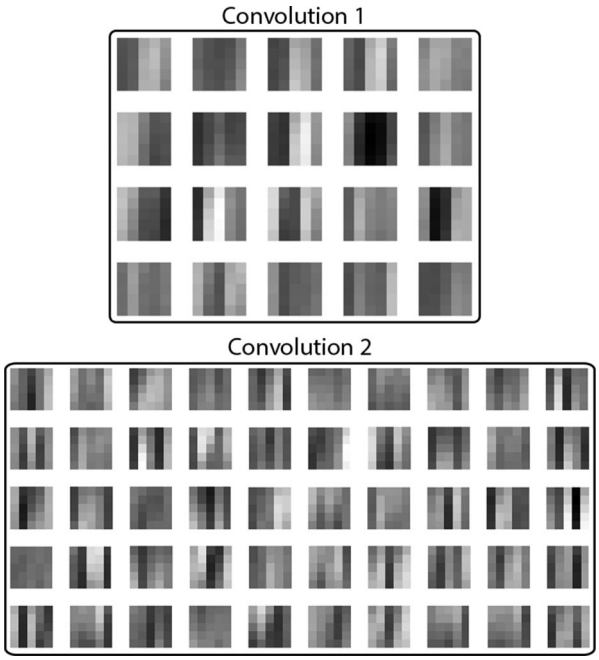


Fig. 8. Features learned for the two convolution layers (20 and 50 filters, respectively) with the default version of the LeNet-5 model.

Other CNNs like the AlexNet (Fig. 6), which is a much more complex network, has also been tested for this purpose and the accuracy results and comparison between this and the LeNet-5 models are presented in the next sections.

VII. RESULTS AND DISCUSSION

Three different window lengths have been used in the segmentation process in this work: 1, 1.25 and 1.5 seconds. As presented in the section where the preprocessing of the information is described, using these three sample lengths leads to obtaining up to 77573, 61518 and 51009 samples, respectively, which is enough for training and testing a CNN. In this work, modified versions of two widely-known CNN models have been used. The accuracy of the network has been obtained for each of the experiments. The sensitivity (Se), specificity (Sp) and the PhysioNet/Computing in Cardiology Challenge 2016 score (MAcc) have been calculated for the approaches that achieved the best accuracy results using the equations that are defined in [42].

A. Using the LeNet-5 Model

First, the accuracy of the system was tested using the LeNet-5 model [37]. The architecture of the model is presented in Fig. 5: it consists of a convolutional layer followed by a pooling layer, another convolutional layer followed by a pooling layer, and then two fully connected layers similar to the conventional multilayer perceptrons. The classifier was trained and tested using each of the three datasets described before without applying any modification to the training parameters or to the configuration of the CNN's layers. The accuracy results can be seen in Fig. 7 for every 10000 training iterations up to a total of 500000 using a base learning rate of 0.01, the *inv* learning policy, 0.9 as momentum and 0.0005 as weight decay. The *inv* learning policy updates the learning rate based on the equation shown in (1), where *gamma* is set to 0.0001 and *power* to 0.75. Table II summarizes the training parameters and layer configurations (kernel sizes and strides for each convolution and pooling layer) for each of the CNN models used in this work.

$$l\_rate = l\_rate * (1 + gamma * iter)^{(-power)} \quad (1)$$

After the default LeNet-5 CNN was trained and tested, 82.11% was achieved for the 1-second dataset, 82.39% for the 1.25-seconds dataset, and 80.00% for the 1.5-seconds dataset. Se, Sp and MAcc were calculated for the dataset that achieved the best accuracy, obtaining 83.26%, 78.58% and 0.8092, respectively. Even though the model was not modified from its default state to improve the classification, the obtained results were very similar to the accuracy that expert cardiologists are able to achieve when auscultating. Fig. 8 shows the features that the default LeNet-5 model is learning on each of its convolution layers. It can be seen that the first layer extract vertical information from the images and the second one is able to detect more complex patterns. However, the results obtained could be improved by changing the network configuration.

In this context, the next experiment consisted in modifying the same CNN model and its training parameters to improve the accuracy results of the system. As in the previous case, the input layer was adapted to be able to work with the proper image size that matches its corresponding dataset (50x64 for the 1 s sample length dataset, 63 $\times$ 64 for the 1.25 s dataset and 75 $\times$ 64 for the 1.5 s dataset). Moreover, kernel sizes were reduced from 5 to 3 and the stride from 2 to 1, for a more detailed analysis of the input images, which allows the extraction of more features

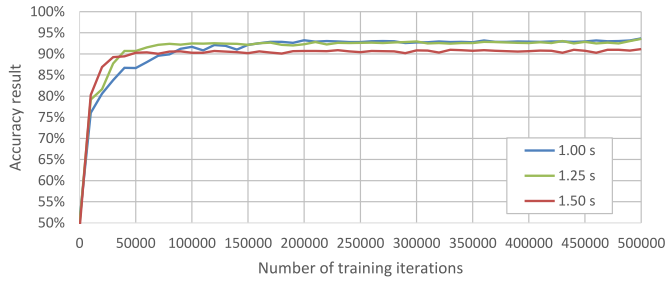


Fig. 9. Accuracy results achieved for each dataset (1s in blue, 1.25 s in green and 1.5 s in red) per 10000 training iterations using the modified version of the LeNet-5 model. Accuracy ratios obtained after 500000 training iterations: 93.68%, 93.57% and 91.14%, respectively, which are better that the ones obtained previously.

from them. Training parameters were optimized by repetition and comparison until the best results were obtained for each of the datasets.

Fig. 9 presents the accuracy results for every 10000 training iterations up to a total of 500000 using a base learning rate of 0.013, the *inv* learning policy, 0.6 as momentum and 0.000875 as weight decay. As can be seen, the 1 s dataset achieves the best result (93.68%), while the 1.25 s and the 1.5 s datasets achieve 93.57% and 91.14% accuracy ratios, respectively. The chart also shows that using smaller window length values in the segmentation step makes the network take a higher number of iterations to converge when training the CNN, due to the fact that more images are generated in the process. Se, Sp and MAcc were calculated for the 1.25 s dataset, obtaining 92.84%, 91.48% and 0.9216, respectively. Training the system took an average of four hours to complete when using the default model, and six hours (∼375 minutes) for the modified model, for each of the experiments and datasets with a NVIDIA GeForce GTX 1060 GPU. The first approaches were carried out using the CPU (3.2 GHz Intel i5-4460) instead of the GPU, which increased the training process execution time more than 24 hours. An average of 13.7% improvement over the default LeNet-5 model was achieved in this case.

B. Using the AlexNet Model

The same experiments that were performed using the LeNet-5 model were then tested with a more complex architecture: the AlexNet [36]. The network is made up of 5 convolutional layers, max-pooling layers, dropout layers and 3 fully connected layers. It was released in 2012 by Alex Krizhevsky and scaled the insights of the LeNet-5 model into a much deeper and wider neural network that could be used to learn much more complex objects. It was used to win by a large margin the 2012 ILSVRC (ImageNet Large-Scale Visual Recognition Challenge) [43].

First, the network was trained and tested without modifying the architecture or the training parameters (only the input and output layers were adapted to accept the image sizes that are being used in this work, and to classify between two different categories). The accuracy results can be seen in Fig. 10, where a base learning rate of 0.01 is used along with the *step* learning policy and 0.9 and 0.0005 as momentum and weight decay,

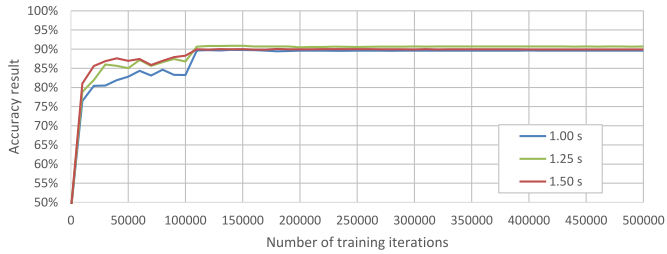


Fig. 10. Accuracy results achieved for each dataset (1s in blue, 1.25 s in green and 1.5 s in red) per 10000 training iterations using the default version of the AlexNet model. Accuracy ratios obtained after 500000 training iterations: 89.61%, 90.70% and 89.91%, respectively.

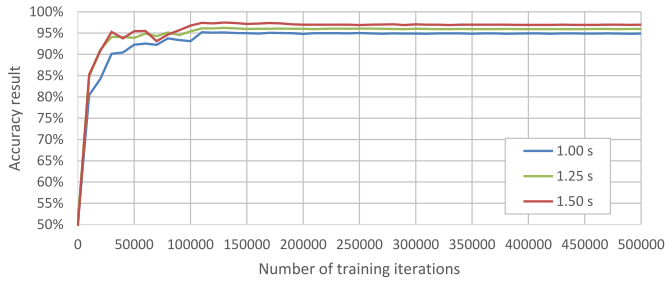


Fig. 11. Accuracy results achieved for each dataset (1s in blue, 1.25 s in green and 1.5 s in red) per 10000 training iterations using the modified version of the AlexNet model. Accuracy ratios obtained after 500000 training iterations: 94.88%, 95.95% and 97.05%, respectively.

respectively. Se, Sp and MAcc were calculated for the dataset that achieved the best accuracy, obtaining 94.52%, 90.48% and 0.9250, respectively. As can be seen, the results do not differ much from the ones obtained with the modified version of the LeNet-5 model while using the default training parameters. Other learning policies like *fixed* and *inv* (which is the one that the LeNet-5 model uses) were used without modifying the rest of the network, but the results did not improve significantly. The *step* learning policy updates the learning rate based on the equation shown in (2), where *gamma* is set to 0.1 and *step* to 100000.

$$l\_rate = l\_rate * gamma^{(floor(iter/step))} \quad (2)$$

In the next experiment, the AlexNet model was modified, reducing kernel sizes and the stride value for each convolutional layer. Training parameters were changed to the ones with whom the LeNet5 obtained the best results, and, after that, they were optimized by repetition and comparison. Fig. 11 presents the accuracy results for every 10000 training iterations up to a total of 500000 using a base learning rate of 0.013, the *step* learning policy, 0.6 as momentum and 0.000875 as weight decay. In this case, the 1.5 s dataset achieved the best result (97.05%), while the 1s and the 1.25 s datasets achieved 94.88% and 95.95% accuracy ratios, respectively. This could be due to the fact that training a more complex CNN like the AlexNet allows to extract more information from the 1.5 s images, which was not possible with the LeNet-5 model. Se, Sp and MAcc were calculated for the dataset that achieved the best accuracy, obtaining 95.12%, 93.20% and 0.9416, respectively.

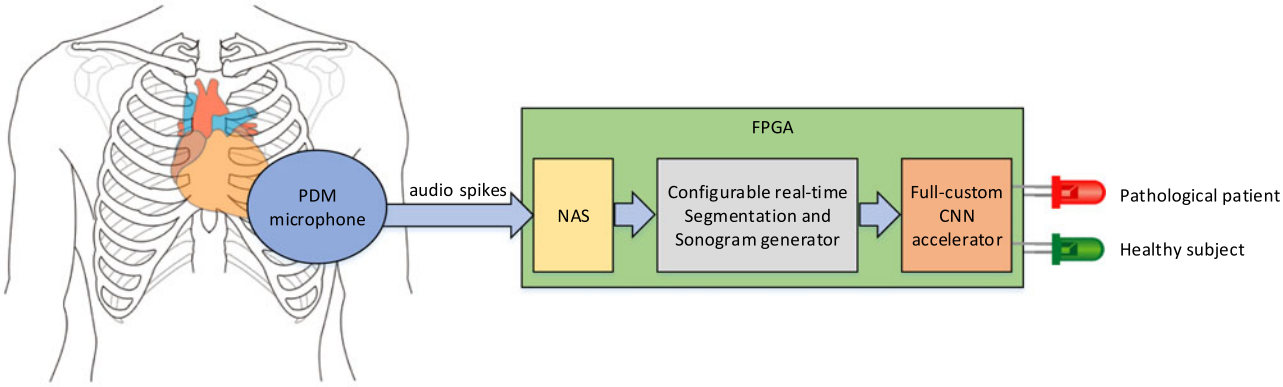


Fig. 12. Block diagram of the complete system implemented on an FPGA using a PDM microphone for real-time analysis of the heart sound directly from the patient.

TABLE III
ACCURACY, SENSITIVITY, SPECIFICITY AND PHYSIONET/CINC CHALLENGE 2016 SCORE OF THE DIFFERENT STUDIED APPROACHES

	Accuracy	Sensitivity*(Se)*	Specificity*(Sp)*	MAcc
Primary care physicians	40%	–	–	–
Expert cardiologists	80%	–	–	–
[18] Potes *et al.*	–	94.24%	77.81%	0.8602
[19] Zabihi *et al.*	–	86.91%	84.90%	0.8590
Default LeNet-5	82.39%	83.26%	78.58%	0.8092
Modified LeNet-5	93.68%	92.84%	91.48%	0.9216
Default AlexNet	90.70%	94.52%	90.48%	0.9250
Modified AlexNet	97.05%	95.12%	93.20%	0.9416

Best cases for the 1, 1.25 and 1.5 datasets are selected.

An average of 65 hours for the default model and 107 hours for the modified model were needed to train the AlexNet CNN using the GPU. The CPU was intended to be used instead of the GPU in the first place, but the training process was estimated around three months (for the default version) to complete per experiment, which is an unreasonable amount of time. However, as can be seen in the images, the system converges after the first 150000 training iterations, approximately, which corresponds to 20 and 32 hours, respectively. Hence, the whole system could be trained for less than half of the iterations and obtain a very similar accuracy while spending much less time in the training process.

The modified version of the AlexNet model achieved the best results. However, it is important to point out that this CNN only improves the accuracy of the modified version of the LeNet-5 (which is a much simpler CNN model) by around 3.5%, while taking almost eighteen times the time needed to train the second one.

VIII. CONCLUSION

In this work the authors have presented a useful tool to aid cardiologists and primary care physicians in the auscultation process. The system uses heart sound recordings from both healthy patients and pathological patients directly, which are first split using windows with a fixed length (1, 1.25 and 1.5 seconds) and then sent to a NAS where the frequency components of the audio are extracted. After this, sonogram images are generated for each of the samples using NAVIS. These images were used to feed different CNN models (LeNet-5 and AlexNet) capable of extracting interesting features from them, which have been trained and tested with different configurations in **Caffe** to classify between the two categories that were described.

The obtained results using different LeNet-5 and AlexNet configurations achieve up to 97.05% accuracy rate in the best case (with a modified version of the AlexNet model), and 80.00% in the worst case (with the default LeNet-5 configuration). These accuracy rates include the 80% accuracy level of an expert cardiologist (see Table III for a comparative study of the obtained results), proving that the system could be very useful as an aide for cardiologists and primary care physicians in the auscultation process, reducing the number of both type-I and type-II errors made. Thereby, the authors have presented a reliable diagnostic tool that could improve the detection of pathological heart murmurs when auscultating and, by aiding the physician, achieve almost 100% accuracy between both. Also, the results have been compared in terms of sensitivity, specificity and the PhysioNet/Computing in Cardiology Challenge 2016 score (obtaining 95.12%, 93.20% and 0.9416 for the best case, respectively) to the ones of leading approaches from the competition (Se: 94.24%, Sp: 77.81%, MAcc: 0.8602, in the best case), showing a clear improvement, especially in terms of specificity.

Using a NAS in this context instead of a traditional digital audio processing approach allows us not only to achieve a very good accuracy result, but also the possibility to develop a portable diagnosis device based on the system that has been described in this paper as the next step in this line of research. This device would be fully implemented in an FPGA (see Fig. 12) where a NAS, a configurable real-time segmentation and sonogram generator, and a full-custom CNN accelerator would be programmed. The input to this system would be generated by a PDM microphone that would be placed on each of the four main auscultatory areas: Aortic area, Pulmonic area, Tricuspid area, Mitral Area (Apex). The PDM microphone directly transmits the audio signal information in a spike-based codification, which would feed the NAS' input. The fact that this device uses a NAS to decompose the audio into frequency bands instead

of using a Fourier Transform leads to having a lower power consumption. As it is presented in [44], a low-power radix-2 FFT accelerator for FPGA achieves a power consumption of 125 mW; however, the NAS' is only 29.7 mW [22], which is less than 24% of the power consumption of the FFT. Additionally, the NAS could interface directly with Spiking Convolutional Neural Networks (SCNN) without the need of the segmentation of the information and the sonogram generation, processing the auditory information in a continuous way. When connected to an SCNN, the system would only need to compute and classify the input signal when spikes are being fired. This means that if there is no activity in the input, the power consumption of the device would be even less. This "neuromorphic stethoscope" would also consist of a button to start the analysis and two LEDs, which would indicate the result of the CNN's classification result in real time as either healthy subject or pathological patient.

REFERENCES

[1] M. Nichols, N. Townsend, P. Scarborough, R. Luengo-Fernandez, J. Leal, A. Gray, and M. Rayner, "European cardiovascular disease statistics 2012: European heart network. brussels," *Eur. Soci. Cardiology*, Sophia Antipolis, 2012.

[2] D. Lloyd-Jones *et al.*, "Heart disease and stroke statistics-2010 update a report from the American heart association," *Circulation*, vol. 121, no. 7, pp. e46–e215, 2010.

[3] D. Roy, J. Sargeant, J. Gray, B. Hoyt, M. Allen, and M. Fleming, "Helping family physicians improve their cardiac auscultation skills with an interactive CD-ROM," *J. Continuing Edu. Health Professions*, vol. 22, no. 3, pp. 152–159, 2002.

[4] E. Etchells, C. Bell, and K. Robb, "Does this patient have an abnormal systolic murmur?" *Jama*, vol. 277, no. 7, pp. 564–571, 1997.

[5] S. Mangione and L. Z. Nieman, "Cardiac auscultatory skills of internal medicine and family practice trainees: a comparison of diagnostic proficiency," *Jama*, vol. 278, no. 9, pp. 717–722, 1997.

[6] M. Lam *et al.*, "Factors influencing cardiac auscultation proficiency in physician trainees," *Singapore Med. J.*, vol. 46, no. 1, pp. 11–14, 2005.

[7] S. L. Strunic, F. Rios-Gutiérrez, R. Alba-Flores, G. Nordehn, and S. Bums, "Detection and classification of cardiac murmurs using segmentation techniques and artificial neural networks," in *Proc. IEEE Symp. Comput. Intell. Data Mining*, 2007, pp. 397–404.

[8] K. Ejaz, G. Nordehn, R. Alba-Flores, F. Rios-Gutierrez, S. Burns, and N. Andrisevic, "A heart murmur detection system using spectrograms and artificial neural networks." in *Proc. Int. Conf. Circuits, Signals, Syst.*, 2004, pp. 374–379.

[9] F. Rios-Gutierrez, R. Alba-Flores, and S. Strunic, "Recognition and classification of cardiac murmurs using ANN and segmentation," in *Proc. 22nd Int. Conf. Electr. Commun. Comput.*, 2012, pp. 219–223.

[10] H. M. Hadi, M. Y. Mashor, M. Z. Suboh, and M. S. Mohamed, "Classification of heart sound based on S-transform and neural network," in *Proc. 10th Int. Conf. Inf. Sci. Signal Process. Appl.*, 2010, pp. 189–192.

[11] H. Hadi, M. Mashor, M. Mohamed, and K. Tat, "Classification of heart sounds using wavelets and neural networks," in *Proc. 5th Int. Conf. Electr. Eng. Comput. Sci. Autom. Control*, 2008, pp. 177–180.

[12] L. Jia, D. Song, L. Tao, and Y. Lu, "Heart sounds classification with a fuzzy neural network method with structure learning," in *Proc. Int. Symp. Neural Netw.*, 2012, pp. 130–140.

[13] M. Singh and A. Cheema, "Heart sounds classification using feature extraction of phonocardiography signal," *Int. J. Comput. Appl.*, vol. 77, no. 4, pp. 13–17, 2013.

[14] T. Leung, P. White, W. Collis, E. Brown, and A. Salmon, "Classification of heart sounds using time-frequency method and artificial neural networks," in *Proc. 22nd Annul. Int. Conf. Eng. Med. Biol. Soc.*, 2000, vol. 2, pp. 988–991.

[15] A.-L. Noponen, S. Lukkarinen, A. Angerla, and R. Sepponen, "Phonospectrographic analysis of heart murmur in children," *BMC Pediatrics*, vol. 7, no. 1, pp. 23–33, 2007.

[16] M. Markaki, I. Germanakis, and Y. Stylianou, "Automatic classification of systolic heart murmurs," in *Proc. 2013 IEEE Int. Conf. Acoust., Speech Signal Process.*, 2013, pp. 1301–1305.

[17] I. S. Perera, F. A. Muthalif, M. Selvarathnam, M. R. Liyanaarachchi, and N. D. Nanayakkara, "Automated diagnosis of cardiac abnormalities using heart sounds," in *Proc. 2013 IEEE Point-of-Care Healthcare Technol.*, 2013, pp. 252–255.

[18] C. Potes, S. Parvaneh, A. Rahman, and B. Conroy, "Ensemble of feature-based and deep learning-based classifiers for detection of abnormal heart sounds," in *Proc. Comput. Cardiol. Conf.*, 2016, pp. 621–624.

[19] M. Zabihi, A. B. Rad, S. Kiranyaz, M. Gabbouj, and A. K. Katsaggelos, "Heart sound anomaly and quality detection using ensemble of neural networks without segmentation," in *Proc. Comput. Cardiol. Conf.*, 2016, pp. 613–616.

[20] J. Amoh and K. Odame, "Deep neural networks for identifying cough sounds," *IEEE Trans. Biomed. Circuits Syst.*, vol. 10, no. 5, pp. 1003–1011, Oct. 2016.

[21] C. N. Gupta, R. Palaniappan, S. Swaminathan, and S. M. Krishnan, "Neural network classification of homomorphic segmented heart sounds," *Appl. Soft Comput.*, vol. 7, no. 1, pp. 286–297, 2007.

[22] A. Jiménez-Fernández *et al.*, "A binaural neuromorphic auditory sensor for FPGA: A spike signal processing approach," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 4, pp. 804–818, Apr. 2016.

[23] The Adress-Event Representation communication protocol. [Online]. Available: https://www.ini.uzh.ch/ amw/scx/std002.pdf

[24] The Adress-Event Representation communication protocol, 1993. [Online]. Available: https://www.ini.uzh.ch/∼amw/scx/std002.pdf

[25] R. Berner, T. Delbruck, A. Civit-Balcells, and A. Linares-Barranco, "A 5 Meps $100 USB2.0 address-event monitor-sequencer interface," in *Proc. 2007 IEEE Int. Symp. Circuits Syst*, 2008, pp. 2451–2454.

[26] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *Proc. 22nd ACM Int. Conf. Multimedia*, ACM, Nov. 2014, pp. 675–678.

[27] A. L. Goldberger *et al.*, "Physiobank, physiotoolkit, and physionet components of a new research resource for complex physiologic signals," *Circulation*, vol. 101, no. 23, pp. e215–e220, 2000.

[28] T. Iakymchuk *et al.*, "An AER handshake-less modular infrastructure PCB with x8 2.5 Gbps LVDS serial links," in *Proc. IEEE Int. Symp. Circuits Syst.*, 2014, pp. 1556–1559.

[29] J. P. Dominguez-Morales, A. Jimenez-Fernandez, M. Dominguez-Morales, and G. Jimenez-Moreno, "NAVIS: Neuromorphic Auditory VI-Sualizer tool," *Neurocomputing*, vol. 237, pp. 418–422, 2017.

[30] R. F. Lyon and C. Mead, "An analog electronic cochlea," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 36, no. 7, pp. 1119–1134, Jul. 1988.

[31] A. Jimenez-Fernandez, A. Linares-Barranco, R. Paz-Vicente, G. Jiménez, and A. Civit, "Building blocks for spikes signals processing," in *Proc. Int. Joint Conf. Neural Netw.*, 2010, pp. 1–8.

[32] M. Yang, C. H. Chien, T. Delbruck, and S. C. Liu, "A 0.5 V 55 μW 64 $\times$ 2 channel binaural silicon cochlea for event-driven stereo-audio sensing," *IEEE J. Solid-State Circuits*, vol. 51, no. 11, pp. 2554–2569, Nov. 2016.

[33] E. Cerezuela-Escudero, M. J. Dominguez-Morales, A. Jiménez-Fernández, R. Paz-Vicente, A. Linares-Barranco, and G. Jiménez-Moreno, "Spikes monitors for FPGAs, an experimental comparative study," in *Proc. Int. Work-Conf. Artif. Neural Netw.*, 2013, pp. 179–188.

[34] C. Liu *et al.*, "An open access database for the evaluation of heart sound algorithms," *Physiological Meas.*, vol. 37, no. 12, pp. 2181–2213, 2016.

[35] NAVIS Tool GitHub, 2015. [Online]. Available: https://github.com/jpdominguez/NAVIS-Tool/

[36] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[37] Y. LeCun *et al.*, "LeNet-5, convolutional neural networks," 2015. [Online]. Available: http://yann. lecun. com/exdb/lenet.

[38] Protocol Buffers, 2008. [Online]. Available: https://developers.google.com/protocol-buffers/

[39] R. K. Sarvadevabhatla and R. V. Babu, "Freehand sketch recognition using deep features," *CoRR*, vol. abs/1502.00254, 2015. [Online]. Available: http://arxiv.org/abs/1502.00254

[40] S. Sarraf and G. Tofighi, "Deep learning-based pipeline to recognize alzheimer's disease using fMRI data," *bioRxiv*, 2016. [Online]. Available: http://www.biorxiv.org/content/early/2016/07/31/066910

[41] A. Rios-Navarro, J. P. Dominguez-Morales, R. Tapiador-Morales, M. Dominguez-Morales, A. Jimenez-Fernandez, and A. Linares-Barranco, "A sensor fusion horse gait classification by a spiking neural network on SpiNNaker," in *Proc. Int. Conf. Artif. Neural Netw*, 2016, pp. 36–44.

[42] G. D. Clifford *et al.*, "Classification of normal/abnormal heart sound recordings: The physionet/computing in cardiology challenge 2016," in *Proc. Comput. Cardiol. Conf.*, 2016, pp. 609–612.
[43] O. Russakovsky *et al.*, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vision*, vol. 115, no. 3, pp. 211–252, 2015.
[44] S. Mookherjee, L. DeBrunner, and V. DeBrunner, "A low power radix-2 FFT accelerator for FPGA," in *Proc. 49th Asilomar Conf. Signals, Syst. Comput.*, 2015, pp. 447–451.