Development of a Synthetic Database for Compact Neural Network Classification of Acoustic Scenes in Dementia Care Environments

Abigail Copiaco[*^], Christian Ritz[*], Stefano Fasciani[†] and Nidhal Abdulaziz[^]
[*]University of Wollongong, Australia
E-mail: abigailc@uow.edu.au, critz@uow.edu.au Tel: +61-2-4221-3555
[†]University of Oslo, Norway
E-mail: stefano.fasciani@uio.imv.no Tel: +47-22-85-50-50
[^]University of Wollongong in Dubai, United Arab Emirates
Email: abigailcopiaco@uowdubai.ac.ae, nidhalabdulaziz@uowdubai.ac.ae Tel: +9714-278-1800

***Abstract*—This paper focuses on automatic detection and classification of sounds occurring in dementia care facilities for monitoring a resident's safety and wellbeing. While there has been significant advances the field of domestic audio classification within the recent years and several audio databases exist, these have not been designed for dementia care environments and can be limited in terms of the amount of information they provide, such as the exact location of the sound sources, and the associated noise levels. This work details our approach to generating a synthetic database of sound scenes and events that is carefully curated to reflect a typical real-world dementia care environment. This includes background noise and room impulse responses based on a typical one-bedroom apartment (Hebrew SeniorLife Facility). The database contains clean and noisy excerpts from 11 classes with duration of 5-seconds and sampling rate of 16 kHz. Using this database, we also explore further development of a series compact neural network architecture through our baseline model which utilizes Continuous Wavelet Transform scalograms as features to the AlexNet. Our compact, MAlexNet-40 approach has achieved a 15x reduction in network size, and an improvement of about 3% on the weighted F1-score when compared to the traditional AlexNet model.**

I. INTRODUCTION

Dementia, a neurodegenerative ailment experienced by the elderly, is commonly associated with cognitive decline [1]. Due to its progressive nature, it affects how the resident perceives external stimuli, especially noise and light [2]. Hence, residents may experience distress, provided that they perceive external stimuli differently compared to those unaffected by dementia. Such distress may result in wandering and changes in behaviour [3,4]. For these reasons, consistent monitoring is crucial for maintaining a safe environment for the dementia resident. Monitoring systems are commonly used as a form of assistive technology to help inform caretakers of the residents' assistance requirement. However, visual monitoring systems are often subject to privacy concerns [5,6]. Thus, audio-based systems, which detect sounds that may indicate a resident's need for assistance, are generally less intimidating compared to visual monitoring. This paper focuses on audio-based systems that can automatically classify recorded sounds using neural network approaches.

Neural network-based sound classification systems require an appropriate database for training the network. The Sound Interfacing through the Swarm (SINS) database is a domestic acoustic scene database, which is composed of 9 different recording categories, sampled at 16 kHz [7]. Although this is sufficient to conduct an initial experiment, and to test the classification effectiveness of the proposed system, there are several limitations, especially when conducting an in-depth analysis of the system performance. Firstly, while the recording consisted of 13 nodes across a number of rooms, where each consist of a linear array of four microphones [7], the data provided publicly is extracted solely from the first four receiver nodes of only the living room and kitchen area. Further, the database was created for domestic audio environments rather than dementia care environments that might include sounds deemed dangerous to residents. Similarly, the exact locations of the sound sources are not provided in the SINS database.

Hence, this paper describes a new synthetic database that can address these limitations and includes additional disruptive sounds commonly faced in a dementia resident's environment. Further, it also allows the recreation of scenarios that could occur in real-world settings, including noisy environments, and various source-to-receiver distances. Furthermore, this will also provide the exact locations of the sound sources, which will be useful for sound location estimation purposes. Creating a synthetic database for dementia care environments also avoids ethical considerations that would arise if creating and sharing recordings made in a real environment. This allows for new systems to be developed and refined that might later be tested in real environments following appropriate ethical procedures.

Neural networks used for healthcare monitoring purposes are required to be compact, in order to fit mobile or embedded devices with limited resources. However, the majority of the compact networks developed in the recent years possess complex, Directed Acyclic Graph (DAG) architectures. Hence, customizations pose risks in affecting the overall network predictive power and potential overfitting [8].

Thus, in order to be able to develop an effective compact neural network while keeping overfitting possibilities at a minimum, using the dataset described in this paper, we look at

the series architecture, compact neural network model presented in our recent work [9]. This was developed through reducing the overall network complexity by modifications of layer architectures and parameters. The series architecture constitutes of layers arranged one after another, which allows for better customizability and lower complexity, which lessens the risks of overfitting[10]. In this work, the effects of normalization, learning and regularization parameter adjustment, and optimization algorithms to compact neural networks will be explored towards network optimization.

Section II of this paper provides describes the details for generating the synthetic database. Section III then discusses the process of synthesizing and refining the database, eliminating the potential of biasing and overfitting, as well as the integration of noise into the clean signals in order to create a more realistic database. Section IV then describes the theory behind the approaches examined for compact neural network development, followed by Section V, which details the relevant results yielded. Finally, the concluding section gives suggestions to improve and extend the scope for future work.

II. Generating the Synthetic Database

The following sections provide detailed information about the simulated sound scenes likely to happen in a realistic dementia resident care facility, the simulated recording setup and the synthesized Room Impulse Responses (RIRs).

A. Sound Scenes in Dementia Care Environments

Monitoring disruptive noises for dementia residents can be challenging, even for healthcare professionals, as sound levels acceptable to staff may be distressing for dementia residents. This is due to the fact that dementia may worsen the effects of sensory changes, as the progressive nature of this ailment may alter how the resident perceives external stimuli, such as acoustic noise pollution [11]. A summary of possible negative impacts caused by disruptive noise levels to dementia residents is provided below:

- As hearing is linked to balance, aural disruption could lead to greater risks of falls, either through loss of balance [12], or through an increase in disorientation as a result of people trying to orientate themselves in an overstimulating environment [11].
- It has been proven that dementia residents respond on a more sensory level, rather than intellectually. For example, they note the body language or tone of voice, rather than what people actually say [13]. Since people with dementia have a reduced ability to understand their sensory environment, when combined with age-related deterioration in hearing, it can be overwhelming.
- Other research suggests that wandering behaviour in dementia residents may be their way to try to remove themselves from an overstimulating situation [14].

Table I. Dry Sample Sources and Licensing Summary.

Database	Categories Used	License
DESED Synthetic Soundscapes [15]	Alarm, Blender, Frying, Shaver, Water	MIT Open Source Initiative
Kaggle: Audio Cats & Dogs [16]	Cat, Dog	CC BY-SA 3.0
Open SLR: 64 and 70 [17]	Speech (Marathi Multi-speaker, English)	CC BY-SA 4.0
FSDKaggle2019 [18]	Scream, Slam, Shatter	CC BY-4.0
FSD50K [19]	Dishes	CC BY-4.0
SINS Database [7]	Absence / Silence	CC BY-NC 4.0
UrbanSound8k [20]	Background Noises	CC BY-NC 3.0

The following are examples of disruptive noises that residents normally experience [11]. Considering these, Table 1 summarizes the sources of the dry sample excerpts from which this database was generated along with the types of sounds extracted from them:

- **Sudden noises**: such as toilet flushes, alarms, glass shattering.
- **Unnecessary noises**: such as television that is not being watched, people talking, and loud music. Eliminating unnecessary noise can reduce the risks of aggression in noisy environments.
- **Sounds in open spaces**: some sounds appear louder in open spaces, for example, noises from a kitchen and dining area, the wheels of a tea trolley or the sound of conversations or laughing.
- **Inappropriate noise timing**: acoustic noise pollution at night can result in disturbed sleep which in turn can lead to problems during the day, such as lack of concentration, and difficulty communicating and performing during the day.

B. Simulated Recording Setup

The generation of the synthetic database is based on a 999 square-foot one-bedroom apartment in Hebrew Senior Life Facility [21], illustrated in Figure 1. We assume a 3-m height for the ceiling of the apartment. Multi-channel recordings were created using node receivers placed on every four corners of each of the six rooms concerned, at 0.2 m below the ceiling. Each node receiver is a microphone array composed of four linearly arranged omnidirectional microphones with 5 cm inter-microphone spacing, as shown in Figure 2. In turn, this creates 4-channel array recordings for every node. In this work, we consider each recording separately when inputting into the neural network.

The room dimensions, source and receiver locations, wall reflectance, and other relevant information were used in order to compute the impulse responses for each room. These are then convolved with the dry sounds, specifying their location, in order to create the synthetic data. Details regarding the process of sound synthesis are provided in the succeeding sub-section of this paper.

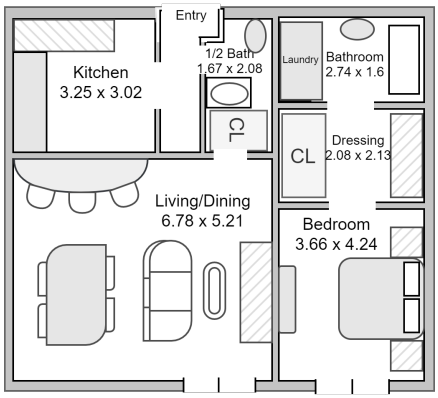

Fig 1 One-bedroom apartment in Hebrew SeniorLife Facility [21].

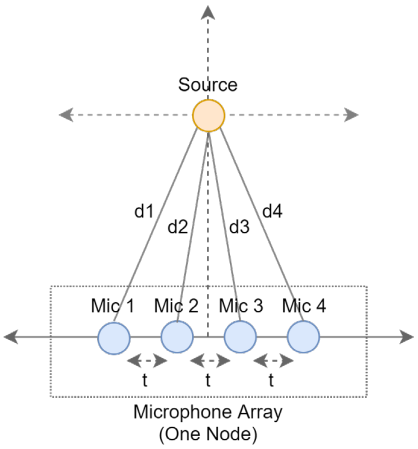

Fig 2 Microphone Array Geometry for a single node: Four linearly spaced microphones.

C. *Synthesized Room Impulse Responses (RIRs)*

The impulse response were synthesized at a sampling rate of 16 kHz, using the image method and using the implementation for directional sound sources [22]. As well as sounds with fixed source positions, such as flowing water through a sink and an alarm clock, moving sounds, such as speech and animal sounds, were created using multiple impulse responses corresponding to different source locations. All relevant information regarding the room dimensions, as well as source and receiver locations, are provided in the technical documentation of the DASEE dataset [23].

The wall reflection coefficients utilized in the convolution process also vary for each room, depending on the percentage of obstruction by furniture, and whether it is a regular wall, floor, or ceiling. Table 2 provides the average room reflectance depending on the percentage of walls that are obstructed, using common wall reflectance coefficients [24]. Similarly, according to the European Standard EN 12464, ceilings have a typical wall reflectance coefficient of 0.7-0.9, walls have 0.5-0.8, and floors have 0.2-0.4 [25].

Table 2. Average room reflectance for varying wall reflectance and obstruction percentages [24].

Walls	**Wall Reflectance**						
Obstruct	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**
20%	0.475	0.505	0.535	0.565	0.596	0.626	0.656
30%	0.488	0.515	0.541	0.569	0.594	0.620	0.647
40%	0.502	0.524	0.547	0.570	0.592	0.615	0.638
50%	0.515	0.534	0.553	0.572	0.591	0.610	0.628
60%	0.529	0.544	0.559	0.574	0.589	0.604	0.619
70%	0.542	0.553	0.565	0.576	0.587	0.599	0.610
80%	0.555	0.563	0.571	0.578	0.586	0.593	0.601

Table 3. Wall reflectance coefficients used to synthesize the DASEE database

Room	**Reflectance**	**Obstruction (%)**	**Reflectance Used**
Bedroom	Walls – 0.5 Ceiling – 0.7 Floor – 0.2	Walls – 30, 50, 70, 30 Ceiling – 0 Floor – 30	Walls – 0.568, 0.572, 0.576, 0.568 Ceiling – 0.7 Floor - 0.488
Living or Dining	Walls – 0.5 Ceiling – 0.7 Floor – 0.2	Walls – 30, 50, 50, 20 Ceiling – 0 Floor – 30	Walls – 0.568, 0.572, 0.572, 0.568 Ceiling – 0.7 Floor – 0.488
Kitchen	Walls – 0.6 Ceiling – 0.8 Floor – 0.3	Walls – 30, 30, 30, 30 Ceiling – 0 Floor – 20	Walls – 0.594, 0.594, 0.594, 0.594 Ceiling – 0.8 Floor – 0.515
Bath	Walls – 0.7 Ceiling – 0.8 Floor – 0.4	Walls – 20, 30, 0, 0 Ceiling – 0 Floor – 30	Walls – 0.626, 0.62, 0.7, 0.7 Ceiling – 0.8 Floor – 0.541
Half-bath	Walls – 0.7 Ceiling – 0.8 Floor – 0.4	Walls – 20, 20, 0, 0 Ceiling – 0 Floor – 30	Walls – 0.626, 0.626, 0.7, 0.7 Ceiling – 0.8 Floor – 0.541
Dressing Room	Walls – 0.5 Ceiling – 0.7 Floor – 0.2	Walls – 80, 80, 20, 20 Ceiling – 0 Floor - 30	Walls – 0.578, 0.578, 0.565, 0.565 Ceiling – 0.7 Floor – 0.488

According to these guidelines, taking into consideration the wall type and obstruction percentages, the wall reflectance coefficients utilized in the setup for the generation of the RIR are seen in Table 3. The same wall reflectance coefficient was used for the four sides of the walls, and different coefficients were used for the ceiling, and the floor, as per the European Standard EN [24].

III. Data Synthesis and Refinement

For the DASEE database, only single source excerpts were taken from the three databases (DESED, Freesound and SINS) and, if required, converted through bandpass interpolation to a common 16 kHz sampling frequency. After this, they are subject to a six-step synthesis and refinement method, as summarized in Figure 3.

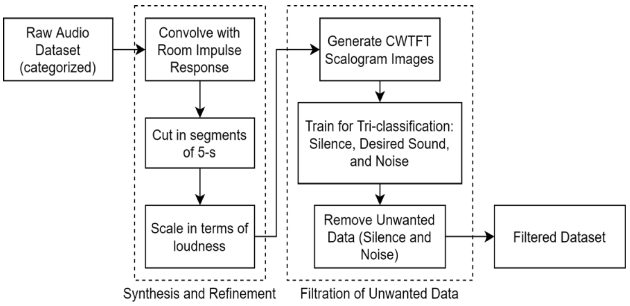

Fig 3 DASEE Database Synthesis and Refinement Process.

As illustrated in Figure 3, raw audio data is first convolved with the relevant RIRs generated per channel. All excerpts are then cut into segments with 5-s duration each. Finally, these are then scaled to have the same loudness.

However, since longer durations are divided into segments of 5-s, some of these segments are not guaranteed to contain the desired sound event. Therefore, a neural network-based filtration method is utilized in order to remove unwanted audio files. Excerpts of 1000 audio files that do not contain sound events and scenes are categorized as 'Silence', while 1000 audio files that contain desired sounds are labelled as 'Desired Sounds'. Lastly, another set containing 1000 files is categorized with the label 'Noise'. Through this, a three-level classifier was developed through the FFT-based Continuous Wavelet Transform (CWTFT) scalograms and CNN method via AlexNet pre-trained network, which was derived from our previous work [27], where this combination was found to provide accurate results for domestic acoustic classification [26, 27]. This network is then used to classify the entire synthesized database. Only those that fell under the 'Desired Sounds' category are kept, and those that fall under the two are filtered out as misclassified data.

A. Background Noise Integration

In order to reflect a realistic environment, recordings with background acoustic noise are also included in the database. For the background noise, we used excerpts from the Noise Urban Sound 8K database [20]. Data from noise sounds that are more relevant for a dementia resident's environment is selected from this database, including: air conditioner, children playing, and street music. Aside from this, white noise is also added as background noise for some files.

The air conditioning background noise was assumed to be placed near the walls, elevated slightly lower than the ceiling, while noises such as "children playing" and "street music" were placed near open windows and relevant RIRs were used to simulate this noise at the recording locations. Background noise was added at Signal-to-Noise (SNR) levels of 15 dB, 20 dB, and 25 dB and using Matlab functions from [28].

B. Curating an Unbiased Database

The database developed contains recordings from the four different nodes across each room. Further, there are 4 instances of these from the addition of the noise for the same sound at 3 different SNR levels. This is shown in Figure 4.

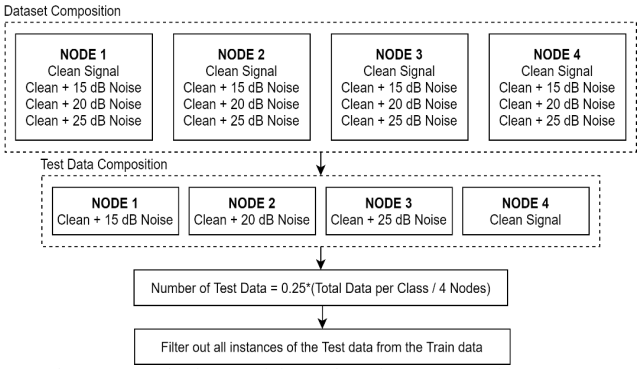

Fig 4 DASEE database Training and Testing Data Curation Process.

Table 4 Summary of the Training and Testing Data.

Category	Training Data	Testing Data
Absence / Silence	11286	876
Alarm	2765	260
Cat	11724	1080
Dog	6673	792
Kitchen_Activities	12291	1062
Scream	4308	376
Shatter	2877	370
Shaver_toothbrush	11231	1077
Slam	1565	268
Speech	30113	2374
Water	6796	829
TOTAL	**101629**	**9364**

As illustrated in Figure 4, the training and testing sets were constructed to avoid the chances of overfitting. In particular, each node is assigned with one of each specific noise levels, while the fourth nodes are assigned with the clean signal. This ensures that all four instances will have significant differences. Similarly, they will also have different noise levels, as it would in real life recordings, where a certain noise can be closer to a single node compared to the other nodes in the same room.

Table 4 summarises the database resulting from this curation process, where all instances of any recording that exists in the test set has been removed from the training set. It is important to note that smaller categories such as "Dishes" and "Frying", have been combined into one folder called "Kitchen Activities". This was found to help with biasing and overfitting.

IV. Exploring Compact Neural Network Factors

In our previous work [29], we developed the MAlexNet-33, a series architecture compact neural network model developed through constraining the model complexity via layer modification and hyperparameter adjustments of the original AlexNet model. In this section, we explore other factors that may influence, improve the performance, and lessen the possibilities of overfitting the pre-trained compact neural network model through regularization models. These include: normalization layers, convolutional layer learning and regularization parameter variations, and the effects of different optimization algorithms.

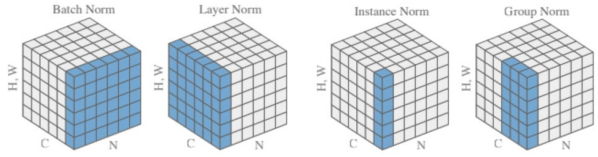

Fig 6 Comparison of the Different Normalization Layers in terms of the Activation [32]

A. Normalization Layers

Normalization is applied in neural network architectures in order to smoothen the gradients, achieve a faster training time and a better computational performance [30] and is typically applied in between convolutional layers and non-linearities, such as activation functions. Several normalization techniques currently exist [31] including batch normalization, layer normalization, group normalization, instance normalization, and cross-channel normalization, which is commonly applied post max-pooling layers.

Fig. 6 provides illustrates how the different normalization layers compare in terms of their activation [32]. Provided the activation of the shape, where N represents the observations and C represents the channels, the batch normalization layer normalizes within the N direction, whereas the layer and grouped normalization layers normalize within the C direction, provided that grouped normalization divides the channels into groups, prior to normalizing each of those groups individually. The instance normalization, however, normalizes an individual channel and at a particular observation one at a time.

Considering their functionalities, the batch normalization layer is advantageous for convolutional neural networks, however, this does not perform well in terms of recurrent neural networks due to the dependency on the previous mini-batches [31]. Nonetheless, the layer normalization layer removes such dependency by normalizing across the direction of the features as opposed to the mini batches [30], a technique which is also adopted by the group normalization layer.

B. Convolutional Layer Learning and Regularization Parameter Modification

Aside from the number of output parameters, convolutional layers also consist of other hyperparameters that can be customized in order to determine the best fit to the neural network. In this experiment, the response of the neural network architecture is examined in terms of modifying the learning and regularization hyperparameters of the 2D and grouped convolutional layers present within the network architecture, including the weight learn rate factor, weight L2 factor (weight decay), bias learn rate factor, and bias L2 factor [33].

By default, the traditional AlexNet network sets the learn rate and L2 factors of the weight to 1, while bias learn rate factor is set to 2. Finally, the bias L2 factor is set to 0. This is because regularization is used in order to avoid overfitting, and to smoothen the slopes of the weights. Since biases are considered to be “intercepts of the segregation” [34], they do not need smoothening or regularization.

C. Optimization Algorithms

Training neural network models involve multiple iterations of minimizing losses and learning the parameters that converge to the desired function, which is achieved through an optimization algorithm [35]. During the training process, the model yields an output for every iteration, prior to calculating the difference between the yielded and desired output, aiming to minimize this variation as much as possible. There are various types of optimization algorithms available. However, for this experiment, the response to three specific optimization algorithms is examined, which include: [35]:

- *Stochastic Gradient Descent with Momentum (SGDM) Optimizer* [36]: calculates the gradient on one individual data element at a certain time. The incorporation of the momentum aids in accelerating the SGD into the correct direction, which dampens the oscillations accordingly and speeds up convergence. This also overcomes disadvantages concerning noise in weight updates, provided its denoising capabilities [37].
- *RMSProp Optimizer* [38]*:* that computes adaptive gradients, and accumulates these into an exponentially decaying weighted average.
- *Adaptive Moments Optimizer* [39]*:* is a fusion of the SGDM and the RMSProp optimizers, such that it accumulates an exponentially decaying weighted average, as per the RMSProp, in addition to retaining the exponentially decaying averages of the past gradients, as per SGDM [37]. A bias correction mechanism is also applied. Further, the update operation solely takes into consideration the smooth version of the gradient, and the decaying average is computed between the past gradient and the past squared gradient.

V. Results

This section details the yielded results and observations in relation to the developed series compact network. The following approaches use the FFT-based Continuous Wavelet Transform (CWT) scalograms as features to the model, which provided accurate results in previous work [27]. These scalogram features result from mapping the average of the four-channels of each recording. The performance was assessed through several evaluation metrics, inclusive of the Accuracy, Precision, Recall, and F1-scores. Further, the use of weighted, micro, and macro F1-score averaging was used to take into consideration the imbalance throughout the dataset developed.

A. Baseline Model

In this section, we provide a per-level and overall preliminary results report on the dataset presented in this paper using a baseline technique, which uses the AlexNet pre-trained model. The results for this is summarized in Table 5.

As observed, results attained using this dataset remains to be consistent with findings communicated in our previous work [27] using the SINS database, as per the DCASE 2018 Task 5 Challenge. The FFT-based CWT scalograms consistently

outperformed other feature sets. The slight inconsistency in the performance figures observed throughout the classes is due to the presence of both sound events and scenes in the dataset.

B. Exploring the Effects of Normalization Layers

Exploring the effects of the normalization layers within the current network architecture yielded results detailed in Table 6, where FC6 refers to the output of the first fully connected layer, and FC7 refers to the output of the second fully connected layer. The output of the last fully connected layer corresponds to the number of classes the system aims to identify.

For these experiments, an epsilon value of 10^{-5} is used for all normalization layers, while the global learning rate is specified as 10^{-5}. Similarly, learning and regularization hyperparameters for the normalization layers, including the offset learn rate factor, offset L2 factor, scale learn rate factor, and scale L2 factor are all set to 1 unless otherwise specified in the table. The offset initializer is set at 0, while the scale initializer is set at 1. In the case of the grouped normalization layer, the group division is kept at 2, provided that all grouped convolutional layers used in the network have two groups. For the case of the batch normalization layer, which requires two extra parameters in the form of mean and variance decay, the values for both of these are set at 0.1, respectively. Finally, the Leaky ReLU with a parameter value of 0.01 was consistently used as the activation function for the network, provided that this returned the best performance as per our recent work [29].

As observed, incorporating grouped normalization layers for each grouped convolutional layer yielded the best performance when compared to the other normalization layers. This can be justified by the suitability of this normalization layer to the grouped convolutional layers used in our model. True to the purpose of normalization layers, incorporating this to the model allowed for more consistent results in between cross-validations, and has also improved the performance of the previous version of our compact network from between 86-88% to between 87-89% weighted F1-score.

C. Convolutional Layer Learning and Regularization Parameter Modification

Subsequent to the investigation involving normalization layers, this section explores the effects of variations in convolutional layer learning and regularization parameter. This involves the weight and bias learning and L2 parameters between both convolutional and fully connected layers. The results for this are summarized in Table 7.

Table 5 Per-level and Overall Results Summary: FFT-based CWT scalograms

Category	Train	Test	TP	FP	FN	Accuracy	Precision	Recall	F1-score
Absence	11286	876	876	2	0	100.00%	99.77%	100.00%	99.89%
Alarm	2765	260	168	86	92	64.62%	66.14%	64.62%	65.37%
Cat	11724	1080	1054	193	26	97.59%	84.52%	97.59%	90.59%
Dog	6673	792	580	34	212	73.23%	94.46%	73.23%	82.50%
Kitchen	12291	1062	878	385	184	82.67%	69.52%	82.67%	75.53%
Scream	4308	376	317	88	59	84.31%	78.27%	84.31%	81.18%
Shatter	2877	370	289	58	81	78.11%	83.29%	78.11%	80.61%
Shaver	11231	1077	765	224	312	71.03%	77.35%	71.03%	74.06%
Slam	1565	268	178	54	90	66.42%	76.72%	66.42%	71.20%
Speech	30113	2374	2374	42	0	100.00%	98.26%	100.00%	99.12%
Water	6796	829	608	111	221	73.34%	84.56%	73.34%	78.55%
TOTAL	**101629**	**9364**	**8087**	**1277**	**1277**	**86.36%**	**86.36%**	**86.36%**	**86.36%**
					Weighted	**86.36%**	**86.72%**	**86.36%**	**86.24%**
					Macro	**81.03%**	**82.99%**	**81.03%**	**81.69%**

Table 6 Effects of Various Normalization Layers

Normalization	Layers	FC6	FC7	Size	W-F1	Comments
Group	40	384	172	14.35 MB	88.50%	Applied a Grouped Normalization Layer for each grouped convolution, with a scale learning rate of 1.
Group	40	384	128	14.35 MB	88.65%	Same as the above experiment, but with FC7 output as 128.
Batch + Group	42	384	172	14.36 MB	85.97%	Added Batch Normalization Layers for every 2D convolution, and Group Normalization for every Group Convolution.
Instance	40	384	172	14.35 MB	83.26%	Applied Instance Normalization Layers for each grouped convolution.
Layer	40	384	172	14.35 MB	86.91%	Applied Layer Normalization Layers for each grouped convolution.
Instance + Group	42	384	172	14.36 MB	83.92%	Applied Instance Normalization Layers for every 2D convolution, and Group Normalization for every Group Convolution.
Layer + Group	42	384	172	14.36 MB	87.42%	Applied Layer Normalization Layers for every 2D convolution, and Group Normalization for every Group Convolution.
Group + Cross-channel	41	384	172	14.35 MB	88.11%	Applied Grouped Normalization Layers for each grouped convolution, and a Cross-channel Normalization before each max pooling layer.

Table 7 Learning and Regularization Hyperparameter Study

Properties	Experiment Number						
	1	**2**	**3**	**4**	**5**	**6**	**7**
FC6	384	384	384	384	**384**	384	192
FC7	128	128	96	96	**64**	64	48
Convolutional Learning and Regularization Parameters							
Weight Learn Rate	2	2	1	2	**1**	3	1
Weight L2 Factor	2	2	1	2	**1**	3	1
Bias Learn Rate	4	4	2	4	**2**	6	2
Bias L2 Factor	0	0	0	0	**0**	0	0
Fully-connected Learning and Regularization Parameters							
Weight Learn Rate	1	2	1	2	**1**	3	1
Weight L2 Factor	1	2	1	2	**1**	3	1
Bias Learn Rate	1	2	1	2	**1**	3	1
Bias L2 Factor	0	0	0	0	**0**	0	0
Weight F1 (%)	**87.48**	**88.72**	**89.03**	**88.58**	**89.46**	**89.08**	**88.39**
Net size (MB)	**14.28**	**14.28**	**14.23**	**14.23**	**14.18**	**14.18**	**14.10**

As observed, variations on the weight and bias learning and L2 factors do not make much significant impact on the system performance, both in terms of the network size and the accuracy. Nonetheless, the optimum combination observed uses a weight learn and L2 factor of 1 for both convolutional and fully-connected layers. Accordingly, a bias learn rate of 2 is used for convolutional layers, while a factor of 1 is utilized for fully-connected layers. For both layers, the bias L2 factor is kept at 0. As discussed earlier, biases do not require smoothening or regularization, provided that they are intercepts of segregation.

D. System Response to Various Optimization Algorithms

Finally, the system response to the three different optimization algorithms discussed in the previous section is considered. It is important to note that the previous experiments have been conducted using the SGDM optimizer. As per the results provided in Table 7, the optimum 40-layer compact network (MAlexNet-40) has yielded a weighted F1-score of 89.46%.

In this section, we consider training the same network through the other two optimization algorithms discussed (RMSProp and Adam optimizer). The results for these are provided in Figures 7 and 8 in the form of confusion matrices.

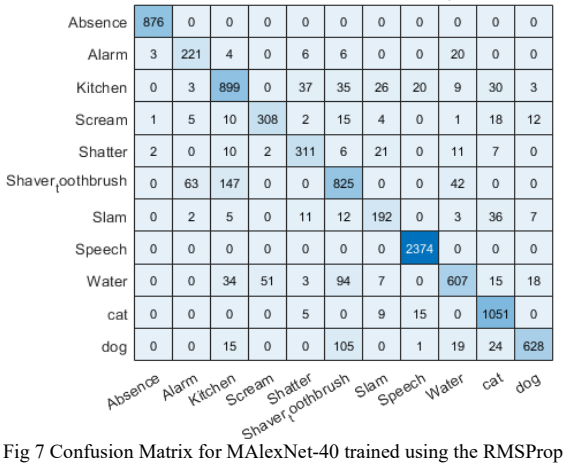

Fig 7 Confusion Matrix for MAlexNet-40 trained using the RMSProp Optimizer Algorithm

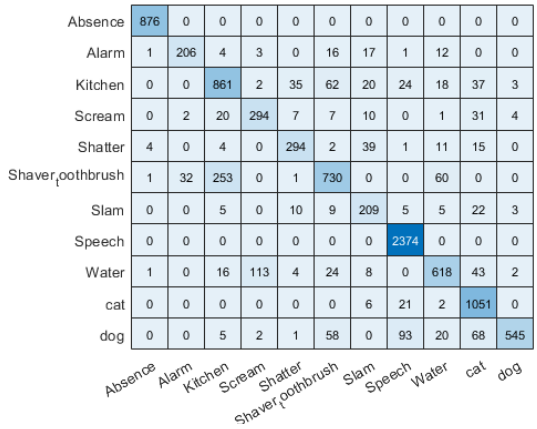

Fig 8 Confusion Matrix for MAlexNet-40 trained using the Adam Optimizer Algorithm

The above confusion matrices yielded a weighted F1-score of 88.55% for the RMSProp, and 86.05% for the Adam optimizer. As observed, the SGDM optimizer produced the best performance for the MAlexNet-40 network architecture at 89.46%, provided that it is a simpler, Series Network format. For more complex architectures, such as the DAG, the Adam optimizer is found to have a good performance [35]. As discussed previously, every optimization algorithm possesses relevant advantages and suitability that can be examined depending on the neural network architecture.

VI. Conclusion

This paper details the process we used to generate a synthetic domestic acoustic scene and event database for the design and evaluation of a neural network-based approach to classifying sounds commonly experienced in a dementia care environment. The database was simulated in an acoustic environment that was designed to closely match a real-world dementia care facility, with simulated recordings from microphone arrays and sound sources at various locations in different rooms. The database is released publicly in order to be utilized for future research. The paper also describes the design of a compact neural network, MAlexNet-40, for classifying sounds in this database and describes experiments designed to evaluate the impact on performance of different hyper parameter choices and other architectural modifications. MAlexNet-40 is a 40-layer series compact neural network model which produced an average weighted F1-score of 89.46% at a 14.35 MB network size. This is a considerable improvement from the AlexNet model, which returned an F1-score of 86.24% at 222.71 MB network size. The MAlexNet-40 is an improved version of our previously developed compact neural network model, the MAlexNet-33 [9]. Through the utilization of grouped normalization layers and refined hyperparameter factors with an SGDM optimization algorithm, this compact model maintains the customizable, series network format of the MAlexNet-33, while improving the consistency of the network performance upon cross-validations, and slightly increasing the average F1-score from 87.92% to 89.46%.

REFERENCES

[1] I. Korolev, "Alzheimer's Disease: A Clinical and Basic Science Review," *Medical Student Research Journal*, vol. 4, 2014, pp. 24-33.

[2] Social Care Institute for Excellence, *Dementia-friendly environments: Noise levels*, May 2015, Accessed on: Aug 20 2020, [online]. Available: https://www.scie.org.uk/dementia/supporting-people-with-dementia/dementia-friendly-environments/

[3] J. van Hoof, H.S.M. Kort, M.S.H. Duijnstee, P.G.S. Rutten, and J.L.M. Hensen, 'The indoor environment and the integrated design of homes for older people with dementia", *Building & Environment*, 45(5), 2010, pp. 1244–1261

[4] J.D. Price, D.G. Hermans, and J. Grimley Evans, "Subjective barriers to prevent the wandering of people cognitively impaired people", Cochrane Database of Systematic reviews, 3: CD001932, 2007..

[5] J. Cocco, "Smart home technology for the elderly and the need for regulation," *Journal of Environmental and Public Health Law*, 6(1), 2011, pp. 85-108

[6] B. Bennett, et al., "Assistive Technologies for People with Dementia: Ethical Considerations," *Bulletin of the World Health Organization*, 2017, pp. 1-12

[7] G. Dekkers et al., "The SINS database for detection of daily activities in a home environment using an acoustic sensor network," *DCASE*, 2017.

[8] Mathworks, "DAG Network, Matlab Documentation," 2017. [Online]. Available: https://www.mathworks.com/help/deeplearning/ref/dagnetwork.html. [Accessed March 2021].

[9] A. Copiaco; C. Ritz; N. Abdulaziz; and S. Fasciani, A Study of Features and Deep Neural Network Architectures and Hyper-Parameters for Domestic Audio Classification. *Appl. Sci.* **2021**, *11*, 4880. https://doi.org/10.3390/app11114880.

[10] Brownlee, J., "How to Avoid Overfitting in Deep Learning Neural Networks", 2019 [online], in Deep Learning Performance, Machine Learning Mastery, Available: https://machinelearningmastery.com/introduction-to-regularization-to-reduce-overfitting-and-improve-generalization-error/, [Accessed Jun 2021].

[11] Social Care Institute for Excellence , "Dementia-friendly environments: Noise levels", published May 2015, Accessible at: https://www.scie.org.uk/dementia/supporting-people-with-dementia/dementia-friendly-environments/noise.asp#:~:text=Of%20all%20the%20senses%2C%20hearing,such%20as%20noise%20and%20light, Accessed Aug 20, 2020.

[12] Hayne, M.J, and Fleming, R. "Acoustic design guidelines for dementia care facilities", Proceedings on 43rd International Congress on Noise Control Engineering: Internoise 2014, Melbourne, Australia, pp. 1-10.

[13] Van Hoof, J., Kort, H.S.M., Duijnstee, M.S.H., Rutten, P.G.S. and Hensen, J.L.M. 'The indoor environment and the integrated design of homes for older people with dementia', Building and Environment, 2010, vol 45, no 5, pp 1244–1261.

[14] Price, J.D., Hermans, D.G. and Grimley Evans, J. Subjective barriers to prevent the wandering of people cognitively impaired people, Cochrane Database of Systematic reviews, 3: CD001932, 2007.

[15] Turpault, N.; Serizel, R.; Shah, A.P., and Salamon, J. Sound event detection in domestic environments with weakly labeled data and soundscape synthesis. In Proc. DCASE Workshop, Oct 2019

[16] Takahashi, N.; Gygli, M.; Pfister, B.; and Van Gool, L. Deep Convolutional Neural Networks and Data Augmentation for Acoustic Event Recognition, Proc. Interspeech, San Fransisco, 2016.

[17] He, F. et al. Open-source Multi-speaker Speech Corpora for Building Gujarati, Kannada, Malayalam, Marathi, Tamil and Telugu Speech Synthesis Systems, Proceedings of The 12th Language Resources and Evaluation Conference (LREC), Marseille, France, 2020.

[18] Fonseca, E.; Plakal, M.; Font, F.; Ellis, D.P.W.; Serra, X. Audio tagging with noisy labels and minimal supervision. Proceedings of the DCASE 2019 Workshop, NYC, US, 2019.

[19] Fonseca, E.; Favory, X.; Pons, J.; Font, F.; and Serra, X. "FSD50K: an Open Dataset of Human-Labeled Sound Events", arXiv:2010.00475, 2020.

[20] Salamon, J.; Jacoby, C.; and Bello, J.P., A Dataset and Taxonomy for Urban Sound Research, 22nd ACM International Conference on Multimedia, Orlando USA, Nov. 2014.

[21] Hebrew SeniorLife. [Online]. Accessible at: https://www.hebrewseniorlife.org/newbridge/types-residences/independent-living/independent-living-apartments.

[22] Hafezi, S. Moore, A.H., and Naylor, P.A. Room Impulse Response for Directional source generator (RIRDgen), 2015. Accessible at: http://www.commsp.ee.ic.ac.uk/~ssh12/RIRD.htm

[23] Copiaco, A.; Ritz, C.; Fasciani, S.; and Abdulaziz, N., 2021, DASEE A Synthetic Database of Domestic Acoustic Scenes and Events in Dementia Patients Environment, arXiv preprint, arXiv: 2103.13

[24] Simm, S.; and Coley, D. The relationship between wall reflectance and daylight factor in real rooms, Architectural Science Review, vol. 54, no. 4, pp. 329-334, 2011.

[25] European Committee for Standardization, "EN 12464-1. Lighting of work places - Part 1: Indoor work places.," 2011.

[26] Copiaco, A.; Ritz, C.; Fasciani, S.; and Abdulaziz, N. Scalogram Neural Network Activations with Machine Learning for Domestic Multi-channel Audio Classification, Proceedings of the IEEE ISSPIT Conference, 2019, Ajman, United Arab Emirates, pp. 1-6.

[27] Copiaco, A.; Ritz, C.; Abdulaziz, N.; and Fasciani, S. Identifying Optimal Features for Multi-channel Acoustic Scene Classification, Proceedings of the ICSPIS Conference, 2019, Dubai, United Arab Emirates, pp. 1-4.

[28] Brookes, M., 'v_addnoise', Voicebox MATLAB Toolbox, Accessible at: http://www.ee.ic.ac.uk/hp/staff/dmb/voicebox/mdoc/v_mfiles/v_addnoise.html

[29] Copiaco, A.; Ritz, C.; Abdulaziz, N., and Fasciani, S.; A Study of Features and Deep Neural Network Architectures 2 and Hyper-parameters for Domestic Audio Classification, MDPI, Applied Sciences Journal, *under review,* 2021.

[30] J. Xu, X. Sun, Z. Zhang, G. Zhao and J. Lin, "Understanding and Improving Layer Normalization," in 33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada, 2019.

[31] N. Vijayrania, "Different Normalization Layers in Deep Learning," Towards Data Science, 10 December 2020. [Online]. Available: https://towardsdatascience.com/different-normalization-layers-in-deep-learning-1a7214ff71d6. [Accessed 2 April 2020].

[32] S. Qiao, H. Wang, C. Liu, W. Shen and A. Yuille, "Micro-Batch Training with Batch-Channel Normalization and Weight Standardization," Journal of Latex Class Files, vol. 14, no. 8, pp. 1-15, 2015.

[33] Mathworks, "MATLAB Documentation: convolution2DLayer," Mathworks, 2016. [Online]. Available: https://www.mathworks.com/help/deeplearning/ref/nnet.cnn.layer.convolution2dlayer.html;jsessionid=c7f49d801e8a242e369f28459e95. [Accessed 7 April 2021].

[34] S. Jadon, "Why we don't use Bias in Regularization?," Medium, 2 February 2018. [Online]. Available: https://medium.com/@shrutijadon10104776/why-we-dont-use-bias-in-regularization-5a86905dfcd6. [Accessed 7 April 2021].

[35] R. Zaheer and H. Shaziya, "A Study of the Optimization Algorithms in Deep Learning," in International Conference on Inventive Systems and Control (ICISC 2019), 2019.

[36] S. Ruder, "An overview of gradient descent optimization algorithms," Sebastian Ruder, 19 January 2016. [Online]. Available: https://ruder.io/optimizing-gradient-descent/index.html#momentum. [Accessed 11 April 2021].

[37] N. Chauhan, "Optimization Algorithms in Neural Networks," KDNuggets, 2020.

[38] M. Mukkamala and M. Hein, "Variants of RMSProp and Adagrad with Logarithmic Regret Bounds," in Proceedings of ICML, 2017.

[39] D. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," in 3rd International Conference for Learning Representations, San Diego, 2015.