IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received October 2, 2020, accepted October 13, 2020, date of publication October 16, 2020, date of current version October 28, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3031685

A Comparative Analysis of Residual Block Alternatives for End-to-End Audio Classification

JAVIER NARANJO-ALCAZAR[1,2], (Graduate Student Member, IEEE), SERGI PEREZ-CASTANOS[1], IRENE MARTÍN-MORATÓ[3], (Graduate Student Member, IEEE), PEDRO ZUCCARELLO[1], FRANCESC J. FERRI[2], (Senior Member, IEEE), AND MAXIMO COBOS[2], (Senior Member, IEEE)

[1]Visualfy, 46181 Benisano, Spain
[2]Computer Science Department, Universitat de Valencia, 46100 Burjassot, Spain
[3]Computing Sciences, Tampere University, 33720 Tampere, Finland

Corresponding author: Javier Naranjo-Alcazar (janal2@alumni.uv.es)

This work was supported by the European Union's Horizon 2020 Research and Innovation Program under Grant 779158. The work of Javier Naranjo-Alcazar was supported in part by the Spanish Ministry of Science, Innovation, and Universities through the Torres Quevedo Fellowships under Grant DIN2018-009982. The work of Irene Martín-Morató was supported by the Spanish Ministry of Science, Innovation, and Universities under Grant FPU14/06329. The work of Pedro Zuccarello was supported in part by the Spanish Ministry of Science, Innovation, and Universities through the Torres Quevedo Fellowships under Grant PTQ-17-09106. The work of Francesc J. Ferri and Maximo Cobos was supported in part by the ERDF, in part by the Spanish Ministry of Science, Innovation, and Universities under Grant RTI2018-097045-B-C21, and in part by the Generalitat Valenciana under Grant AICO/2020/154 and Grant AEST/2020/012.

ABSTRACT Residual learning is known for being a learning framework that facilitates the training of very deep neural networks. Residual blocks or units are made up of a set of stacked layers, where the inputs are added back to their outputs with the aim of creating identity mappings. In practice, such identity mappings are accomplished by means of the so-called skip or shortcut connections. However, multiple implementation alternatives arise with respect to where such skip connections are applied within the set of stacked layers making up a residual block. While residual networks for image classification using convolutional neural networks (CNNs) have been widely discussed in the literature, their adoption for 1D end-to-end architectures is still scarce in the audio domain. Thus, the suitability of different residual block designs for raw audio classification is partly unknown. The purpose of this article is to compare, analyze and discuss the performance of several residual block implementations, the most commonly used in image classification problems, within a state-of-the-art CNN-based architecture for end-to-end audio classification using raw audio waveforms. Deep and careful statistical analyses over six different residual block alternatives are conducted, considering two well-known datasets and common input normalization choices. The results show that, while some significant differences in performance are observed among architectures using different residual block designs, the selection of the most suitable residual block can be highly dependent on the input data.

INDEX TERMS Audio classification, convolutional neural networks, residual learning, urbansound8k, ESC.

I. INTRODUCTION

Audio event classification (AEC) is the problem of categorizing an audio sequence into exclusive classes [1]–[3]. Basically, AEC is aimed at recognizing and understanding the acoustic environment based on sound information. This is usually treated as a supervised learning problem where a set of labels (such as siren, dog barking, etc.) describe the content of the different sound clips. In contrast to classical schemes based on feature extraction followed by classification, Deep Neural Networks (DNNs) [4] reduce these steps by working both as feature extractors and classifiers. Among the many different deep learning techniques, the ones based on Convolutional Neural Networks (CNNs) have shown very successful results in areas such as image classification and object detection [5]–[8]. CNNs are able to learn spatial or time invariant features from pixels (i.e. images) or from time-domain waveforms (i.e. audio signals). Several convolutional layers can be stacked to get different levels of representation of the input signal. As a result, CNNs have

The associate editor coordinating the review of this manuscript and approving it for publication was Tallha Akram.

been proposed to tackle audio related problems such as sound event detection or audio tagging [9]–[11].

Although audio signals are natively one-dimensional sequences, most state-of-the-art approaches to audio classification based on CNNs use a two-dimensional (2D) input [12], [13]. Usually, these 2D inputs computed from the audio signal are well-known time-frequency representations such as Mel-spectrograms [14]–[17] or the output of constant-Q transform [18] (CQT) filterbanks, among others. Time-frequency 2D audio representations are able to accurately extract acoustically meaningful patterns but require a set of parameters to be specified, such as the window type and length, hop size or the number of frequency bins. The choice of these hyperparameters can lead to different optimal settings depending on the particular problem being treated or the particular type of input signals [19]. In order to overcome these problems and providing an end-to-end solution, other approaches have proposed the use of 1D convolutions using the raw audio signals as input. Recent works have shown satisfactory results in this direction [20]–[28].

This article is focused on the analysis of the performance of a particular CNN architecture, called Residual Network (ResNet), fed with 1D audio data. The ResNet architecture was first introduced in [29] with the purpose of dealing with the vanishing gradient issue. The core idea of ResNet is to introduce the so-called *identity weight shortcut connection* that skips one or more layers and adds the input of such layers to their stacked output. After the first residual unit was presented in [29], an exhaustive analysis of different variations of such a configuration was done for CNNs with 2D input signals to tackle the image classification problem [30]. Nevertheless, although other works have studied the contribution of residual blocks in the context of 1D raw audio input waveforms [28], [31], a comprehensive analysis of how different residual block designs may affect the overall performance of audio recognition systems has not been provided so far.

The main objective of this work is to analyze the influence on the performance of different residual block alternatives, the ones more commonly used in the image domain, within the context of 1D raw audio classification. To this end, a baseline architecture is slightly modified considering six different residual block implementations that have been shown to lead to satisfactory results in image classification problems. These blocks provide a varying scheme with regard to where identity mappings are created within the set of stacked layers that conform the block. The common baseline architecture is the one presented in [20], which proposed a 1D CNN for raw audio waveform classification using the public dataset UrbanSound8k.[1] For the sake of consistency, the same dataset will be considered in this work. Additionally, the public dataset ESC-50[2] (concretely, the ESC-10 subset) is also used in the experimentation to evaluate the potential differences arising over different datasets. The results suggest that the best performing blocks in the image domain are not the ones showing significant advantages in performance for raw audio classification [30], nor the one originally suggested in [20] for audio data using the baseline architecture.

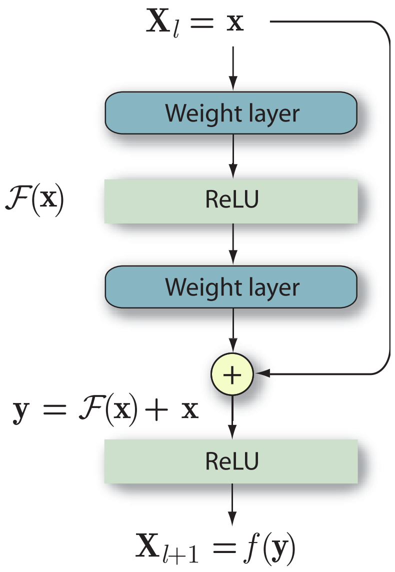

$\mathbf{X}_l = \mathbf{x}$
Weight layer
$\mathcal{F}(\mathbf{x})$
ReLU
Weight layer
+
$\mathbf{y} = \mathcal{F}(\mathbf{x}) + \mathbf{x}$
ReLU
$\mathbf{X}_{l+1} = f(\mathbf{y})$

FIGURE 1. Originally proposed residual block or unit [29].

II. BACKGROUND

Residual neural networks -or ResNets- can be understood as modular networks whose building blocks are the so-called residual units or blocks. These residual blocks (RB) are usually characterized by two or three convolutional layers and a shortcut connection that guarantees residual learning during the network training process. The original residual block proposed in [29] is shown in Fig. 1. Consider $\mathcal{H}(\mathbf{x})$ an underlying mapping to be fit by a set of stacked layers in a particular network module, where $\mathbf{x}$ is the input to the first of such layers. Residual blocks are designed to let such layers approximate a residual function, $\mathcal{F}(\mathbf{x}) := \mathcal{H}(\mathbf{x}) - \mathbf{x}$, which means that the original function can be expressed as $\mathcal{H}(\mathbf{x}) = \mathcal{F}(\mathbf{x})+\mathbf{x}$. Similar to predictive coding, the motivation of using residual blocks comes from the intuition that it may be easier to optimize the above residual mapping than to optimize the original, unreferenced mapping. A straightforward way of implementing residual learning is by adding shortcut connections performing identity mappings. In such connections, the input to the set of layers $\mathbf{x}$ is added back to their output, so that $\mathbf{y} = \mathbf{x} + \mathcal{F}(\mathbf{x})$. The function $\mathcal{F}(\mathbf{x})$ represents the residual to be learned by a set of stacked layers of the CNN, where the weight layers are convolutional. In the original residual block, Rectified Linear Unit (ReLU) activation is applied to the result after each identity mapping, resulting in a final output $f(\mathbf{y})$ that acts as input to the next residual block, where $f(\cdot)$ denotes the ReLU function. Thus, in general, the input to the l-th block, $\mathbf{X}_l$, is the output from the previous block and its output becomes the input to the next one, $\mathbf{X}_{l+1}$. Note that shortcut connections do not add extra

[1] https://urbansounddataset.weebly.com/urbansound8k.html
[2] https://github.com/karoldvl/ESC-50

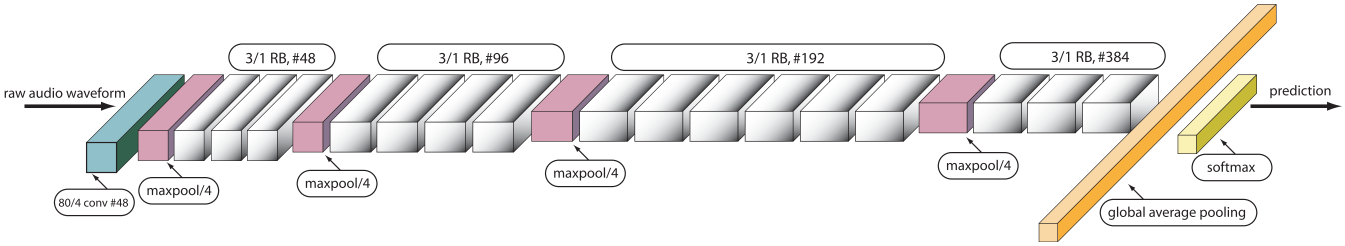


FIGURE 2. Network analyzed [20]. The architecture is explained as follows: [80/4, #48] denotes a layer with 48 filters, 80 of kernel size and stride equal to 4. RB blocks are indicated with kernel size, stride and number of filters.

parameters nor additional computational cost. Thus, deeper networks can be trained with little additional effort, substantially reducing vanishing-gradient problems. However, CNNs often include Batch Normalization (BN) layers and vary slightly with regard to where the activation function is applied. Therefore, the performance of residual learning may also depend both on the order followed by these layers and on the selected point at which shortcut connections are established. In [30], a careful discussion on identity mappings is provided, proposing the use of pre-activated residual units where f is an identity mapping, i.e. $\mathbf{X}_{l+1} = \mathbf{y}_l$. Such slight modification is shown to benefit the training process and to achieve better results in image recognition tasks. However, such analysis has only been performed for 2D architectures and, to the best of the authors' knowledge, a similar study analyzing residual blocks in 1D CNNs has not been addressed so far.

A. RELATED WORK

The use of residual networks for audio-related tasks has already been explored in the literature, usually taking as input frame-level features such as the outputs from mel-scale or logarithmic filterbanks [32]–[34]. As in the present work, several variants of a CNN-based audio classification system accepting raw audio waveforms as input was proposed in [20], including a particular residual architecture. Similarly, end-to-end audio classification systems using residual networks were covered by Kim *et al.* in [28], [31], proposing as well the use of squeeze-and-excitation strategies [35] for increased accuracy. Such strategies are aimed at rescaling the convolutional feature maps by learning proper weightings using temporal aggregation (squeeze) and channel-wise recalibration (excitation). The residual blocks presented in these works, named Res$-n$ (purely residual) and ReSE$-n$ (combinining squeeze-and-excitation), considered the original residual design of [29] depicted in Fig. 1. Both works showed that CNN architectures making use of such blocks provided promising results for learning from raw data and analyzed in detail the effect of including squeeze-and-excitation recalibration. However, the influence of the specific residual block design, as considered in [30] for the image domain, has not been covered so far and its effect in 1D raw audio learning is still unclear.

III. NETWORK ARCHITECTURE

The experimentation conducted in this work considers as a baseline the architecture originally proposed in [20] for raw audio waveforms, consisting in a fully-convolutional network intercalating convolutional and pooling layers. Fully-convolutional networks can usually obtain better generalization properties, whereas, fully-connected layers at the end of the network are more prone to suffer from overfitting. In [20], the convolutional layers are configured with small receptive fields, with the exception of the first layer, whose receptive field is bigger in order to emulate a band-pass filter. Therefore, temporal resolution is reduced in the first two layers with large convolution and max pooling strides. After these layers, resolution reduction is complemented by doubling the number of filters in specific layers. Finally, after the last residual unit, global average pooling is applied to reduce each feature into a single value by averaging the activation across the input. To study the behavior of a given residual block (RB), this article focuses on the residual variant proposed in [20] (originally labeled as M34-res), which follows the general architecture shown in Fig. 2.

Six different RB implementation alternatives are analyzed: the original block proposed by He *et al.* [29] plus the other four blocks proposed by the same authors in [30] and the one introduced by Dai *et al.* in [20] (see Fig. 3). In ResNets, convolutional layers are replaced by different RBs. To isolate the effect of these blocks from the rest of parameters of the network, the number of filters, the receptive field size and the number of convolutional layers remain the same as in [20]. The analyzed residual blocks are the following:

- **RB1 [29]**: the input is first convolved and the output of the second convolution is the input of a batch normalization layer. After the addition, ReLU activation is applied.
- **RB2 [30]**: the input is first convolved and no post-processing is done after the second convolution. The only difference with respect to RB1 is that normalization is applied after adding the input and consequently f corresponds to the composition of BN and ReLU.
- **RB3 [30]**: the input is first convolved as in [20] and the activation is performed before the addition.
- **RB4 [30]**: the input is first passed through a ReLU activation layer and then normalized after the second convolution.
- **RB5 [30]**: the input is first normalized and there are no layers after the second convolution as well as after the addition. RB3-5 constitute a family in which there are no layers after the addition and consequently f is exactly the identity. The differences are in the order in the layers

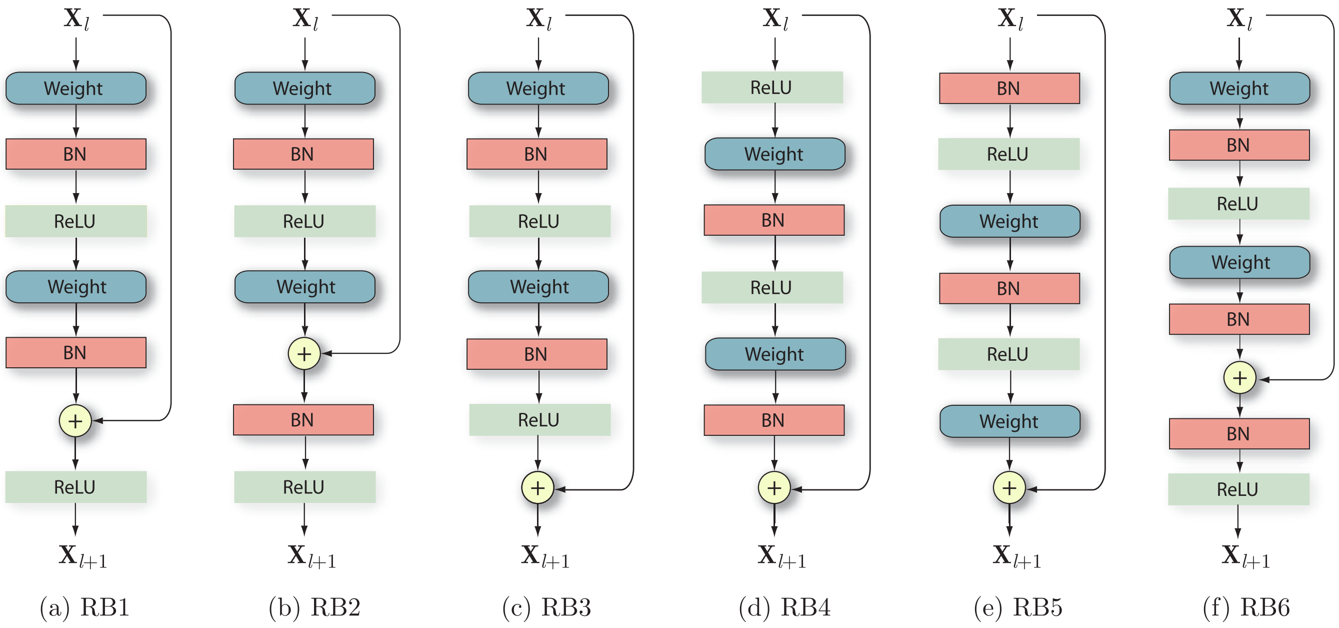


FIGURE 3. Residual units implemented in this work. RB1 to RB5 (a-e) were first introduced in [30], whereas RB6 (f) was presented in [20].

ranging from post-activation (RB3) to pre-activation (RB5).

- **RB6 [20]**: the input is first convolved and the output of the second convolution is the input of a batch normalization layer. After the addition, a new normalization is applied followed by ReLU activation which constitutes a very slight variation of RB2.

The M34-res presented in [20] has 4,001,242 parameters because it uses RB6. When using RB5 the network has 3,988,570 parameters while using RB1-4 the network is composed by 3,989,914 parameters. Dropout layers [36] have not been implemented neither after the pooling layers nor in the residual block, as set out in [20].

IV. EXPERIMENTAL DETAILS

A. DATASETS AND AUDIO PRE-PROCESSING

As in [20], the experimental setup of the present work is based on UrbanSound8k (UBS8k) [37], a public sound-database that contains 8732 sound clips of duration of up to 4 seconds with 10 different classes such as dog barking, car horn, drilling, etc. The dataset is partitioned into 10 different folds and the last one is commonly used as a test while the previous ones are left for training and validation. Additionally, the ESC-10 dataset [38], a public sound-database that contains 400 clips of 5 seconds of duration with 10 different categories (40 samples each category), is also considered. This dataset contains the same number of categories than UBS8k, making the comparison more precise. This dataset is also officially partitioned into different folds (5 in this case).

Clips from both datasets were resampled to 8 kHz and padded with zeros to reach 4 s or 5 s length if necessary after being pre-processed. Once an audio sequence has been read, two different pre-processings have been carried out to check how these can affect the behavior of the final system. The first processing is the scaling of the audio to the maximum absolute value (Scalemax). The second processing consists in normalizing to a signal with zero mean and unit standard deviation (Mean 0 Std 1) as in [20]. As mentioned earlier, padding is done once the signal has been accordingly pre-processed.

B. EXPERIMENTAL SETUP

Instead of using only the last fold of each dataset as a test, a full k-fold cross validation analysis will be carried out in order to obtain more accurate averaged measurements related to the generalization capabilities of the systems under study. The value of k is 10 and 5 for UBS8k and ESC-10, respectively.

Due to the stochastic nature of the experiments and to account for variability, the k-fold cross validation run is repeated a number of times for each dataset (5 and 10 for UBS8k and ESC-10, respectively) so that a total of 50 models are fully trained for each dataset. The final performance measures correspond to the classification accuracy over the whole dataset and averaged over all repetitions along with the corresponding standard deviation.

C. IMPLEMENTATION DETAILS

The optimizer used was Adam [39]. The models were trained with a maximum of 400 epochs. Batch size was set to 128. The learning rate started with a value of 0.001 decreasing with a factor of 0.2 in case of no improvement in the validation accuracy after 15 epochs. The training is early stopped if the validation accuracy does not improve during 50 epochs. The initialization method was glorot-uniform [40] and all weight parameters were subject to L2 regularization with a 0.0001 coefficient as in [20]. Keras with Tensorflow backend was used to implement the models in the experiments. The audio manipulation module used in this work was LibROSA [41].

TABLE 1. Averaged accuracies of the different blocks presented in this article depending on the pre-processing of the audio and dataset used for the experimentation.

Pre-processing	Dataset	RB1	RB2	RB3	RB4	RB5	RB6
Scalemax	UBS8k	**0.68**±0.01	**0.68**±0.00	0.64±0.02	**0.68**±0.00	**0.68**±0.01	**0.68**±0.01
	ESC-10	**0.77**±0.02	0.75±0.01	0.68±0.06	**0.79**±0.02	0.56±0.05	0.74±0.02
Mean 0 Std 1	UBS8k	0.68±0.01	0.68±0.01	0.59±0.03	**0.69**±0.01	0.68±0.01	0.68±0.01
	ESC-10	0.75±0.05	**0.79**±0.01	0.59±0.05	0.75±0.05	0.58±0.04	**0.79**±0.01

V. RESULTS

Given the number of folds and repetitions in the two datasets considered, a total number of 50 independent results are available in each case. With these we have carried out a careful analysis, first comparing averaged accuracies, and second performing a rank-based analysis. Note that the results can not be fairly compared to other previous published results (e.g. [20]) that are more challenge-oriented, but instead the followed procedure allows to compare the different alternatives more accurately.

A. AVERAGED PERFORMANCE ANALYSIS

Averaged rates of accuracy for all the experiments carried out are shown in Table 1 along with standard deviations across repetitions. Best results for each dataset and pre-processing method are marked in bold. We naively assume Gaussianity and perform a parametric multiple comparison test [42] that only discovers significant differences between RB3 and RB5 (shaded in the table) and the remaining options depending on dataset but regardless of pre-processing.

From this first analysis we can hardly observe differences among RBs but it is worth mentioning several surprising facts. First, the RB3 is significantly worse in all cases. Even though this was also the worst in the exact identity family (RB3-5) according to [30], its behavior in the image context was clearly better than that of RB2 which is now among the bests along with its slight variation RB6. Second, the full-preactivation option, RB5, which was the best in the image context is now significantly the worst for ESC-10.

It can be also observed that systems trained on the ESC-10 dataset seem to be more sensitive to the selected input pre-processing. Blocks RB1, RB3 and RB4 show better performance when the audios have been processed with Scalemax. On the other hand, blocks RB2, RB5 and RB6 show better performance when the audios have been normalized to zero mean and unit standard deviation.

Apart form putting forward normalization sensibility and the surprising dependence on data of RB5, the clearest conclusion that we can draw from comparing averaged rates is the very poor behavior of RB3. This could be somewhat expected as RB3 is the only block having a ReLU activation just before the addition leading to a non-negative output which is an unnatural option for a residual function. Note that having a non-negative residual function can have an undesirable impact on learned internal representations, which in turn may substantially affect the robustness and generalization capabilitites of the network.

B. NON-PARAMETRIC RANK-BASED COMPARISON

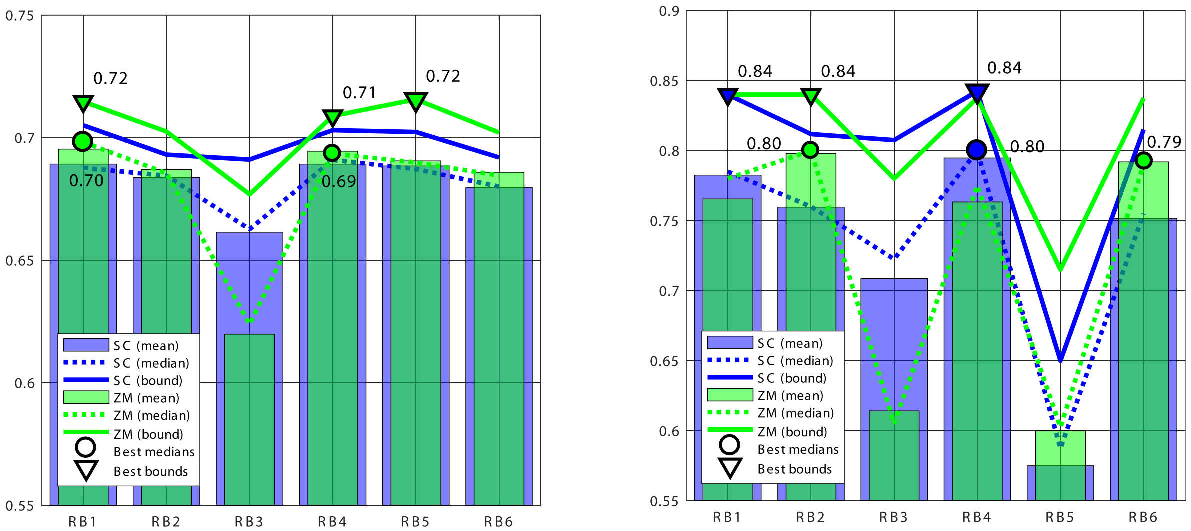
In order to provide more insight about the RB choice, a non-parametric Friedman test with Holm post-hoc has been carried out [43]. Moreover, medians of all repetitions and an optimistic bound obtained by selecting the best model for each fold have been computed and are shown in Fig. 4 along with averaged rates. Table 2 shows the test results including average ranks and corrected p-values. Significance level has been set to $\alpha = 0.05$. The value 0.00 means $p < 0.005$. Apart from the results for each dataset, we also show the ones corresponding to both datasets. Results that are significantly worse than the best according to the selected level appear as shaded in the table, with the corresponding p-values in bold.

The results of the non-parametric analysis confirm the findings from the parametric one and uncovers further differences among the best performing options. Unfortunately, and as previously observed, different datasets imply slightly different conclusions.

According to UBS8k results, the best performing blocks are RB1, RB4 and RB5, partially confirming the inappropriateness of RB2 as in [30]. Even though RB1 ranks the first and all means are indistinguishable we can still find some interesting differences. On the one hand, RB4 using both pre-processing options has almost the best median (0.69) which may suggest that the RB4 option is more robust. On the other hand, we obtain an optimistic bound of 0.72 both for RB5 and RB1 with Mean 0 Std 1 pre-processing. The value of this bound for the next best options is 0.71 for RB4 using the same preprocessing.

When considering the ESC-10 results, the previous surprising behavior of RB5 is confirmed in all cases. Moreover, the more specific differences among methods also confirm that pre-processing affects the behavior of RB options for this dataset. In particular, RB1 and RB4 on one hand, and RB2 and RB6 on the other, are the best performing blocks depending on pre-processing, all with indistinguishable means. If we compute the medians as with the previous dataset we find slight differences between RB2 and RB4 (0.80) and RB1 and RB6 (0.79). Finally, the best options according to the optimistic bound when the best models are selected are RB4, RB2 and RB1 (0.84) for different pre-processing options. These bounds, together with the fact that RB1 exhibits significantly worse medians, suggest that both RB4 and RB2 constitute a more robust alternative.

Given these overall results, drawing a general conclusion looks difficult. The more remarkable fact is that the best block considered for the image domain RB5 is not, in general,



(a) UBS8k (b) ESC-10

FIGURE 4. **Means, medians and optimistic bounds on accuracies of the considered residual blocks on two datasets for different pre-processings: Scalemax (SC) in blue and Mean 0 Std 1 (ZM) in green. The best medians and optimistic bounds are marked as cercles and triangles, respectively.**

TABLE 2. **Ranking results of the different RB configurations.**

Pre-processing	Dataset								
	UBS8k			ESC-10			Both		
	R	AvRnk	pval	R	AvRnk	pval	R	AvRnk	pval
Scalemax	RB1	2.75	-	RB4	1.53	-	RB4	2.14	-
	RB4	3.00	0.98	RB1	2.02	0.09	RB1	2.37	0.46
	RB5	3.00	0.98	RB2	3.13	**0.00**	RB2	3.52	**0.00**
	RB2	3.80	**0.04**	RB6	3.71	**0.00**	RB6	3.80	**0.00**
	RB6	4.08	**0.01**	RB3	4.60	**0.00**	RB3	4.43	**0.00**
	RB3	4.38	**0.00**	RB5	6.00	**0.00**	RB5	4.74	**0.00**
Mean 0 Std 1	RB1	2.38	-	RB2	1.73	-	RB1	2.64	-
	RB4	2.77	0.42	RB6	2.24	0.61	RB2	2.72	1.00
	RB5	2.92	0.42	RB1	2.89	**0.01**	RB6	2.85	1.00
	RB2	3.67	**0.01**	RB4	3.16	**0.01**	RB4	2.96	0.93
	RB6	3.73	**0.01**	RB3	5.38	**0.00**	RB5	4.41	**0.00**
	RB3	5.60	**0.00**	RB5	5.60	**0.00**	RB3	5.43	**0.00**

among the bests. Also interesting is the fact that the block RB6 proposed in [20] and specially its close variant RB2 with normalized inputs are among the bests but only for one of the datasets. Finally, the original block RB1 is among the bests for all datasets even though it exhibits a dependence on pre-processing. Also among the bests for all datasets is the RB4 option that can be considered as a small variation of the RB5 recommended in [30]. If one had to put one of the options ahead of the others for 1D end-to-end audio classification from the experimentation carried out in the present work it would be the RB4 design. This ReLU-only pre-activation, as named in [30] had also a very good behavior in image classification. Moreover, it consistently produces median accuracies among the best in all datasets and pre-processing options (except Mean 0 Std 1 in the case of ESC-10).

VI. CONCLUSION

End-to-end 1D architectures are very convenient for addressing audio classification tasks, as they avoid making certain decisions related to the adoption of suitable input representations for the input audio data. As a result, raw audio waveforms can be fed directly into convolutional networks without the need for a prior feature extraction process. While residual learning has been widely demonstrated to be a successful approach for training deep neural networks, different residual block designs may affect the final performance of the classification system. In this context, while the study of the appropriateness of different residual block designs has been previously addressed in the image domain, similar analyses have not been previously reported when considering 1D audio data. In this work, it has been shown that previous results obtained for image classification can not be easily extrapolated to the audio domain. Moreover, significant differences in the performance provided by different residual blocks have been observed when considering different audio datasets and pre-processings. With the considered baseline architecture, some of the recommended residual blocks in the literature did not achieve the best performance, nor even the the most successful block recommended for image classification tasks.

REFERENCES

[1] S. Adavanne and T. Virtanen, "A report on sound event detection with different binaural features," 2017, *arXiv:1710.02997*. [Online]. Available: http://arxiv.org/abs/1710.02997

[2] H. Zhang, I. McLoughlin, and Y. Song, "Robust sound event recognition using convolutional neural networks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2015, pp. 559–563.

[3] P. Foggia, N. Petkov, A. Saggese, N. Strisciuglio, and M. Vento, "Reliable detection of audio events in highly noisy environments," *Pattern Recognit. Lett.*, vol. 65, pp. 22–28, Nov. 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167865515001981

[4] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, p. 436, 2015.

[5] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*. [Online]. Available: http://arxiv.org/abs/1409.1556

[6] 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.

[7] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2818–2826.

[8] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1251–1258.

[9] L. Zhang and J. Han, "Acoustic scene classification using multi-layer temporal pooling based on convolutional neural network," 2019, *arXiv:1902.10063*. [Online]. Available: http://arxiv.org/abs/1902.10063

[10] S. Hershey, S. Chaudhuri, D. P. W. Ellis, J. F. Gemmeke, A. Jansen, R. C. Moore, M. Plakal, D. Platt, R. A. Saurous, B. Seybold, M. Slaney, R. J. Weiss, and K. Wilson, "CNN architectures for large-scale audio classification," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Mar. 2017, pp. 131–135.

[11] Y. Xu, Q. Kong, W. Wang, and M. D. Plumbley, "Large-scale weakly supervised audio classification using gated convolutional neural network," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2018, pp. 121–125.

[12] E. Cakır, T. Heittola, and T. Virtanen, "Domestic audio tagging with convolutional neural networks," in *Proc. IEEE AASP Challenge Detection Classification Acoustic Scenes Events (DCASE)*, 2016. [Online]. Available: http://dcase.community/documents/challenge2016/technical_reports/DCASE2016_Cakir_4003.pdf

[13] E. Cakir, G. Parascandolo, T. Heittola, H. Huttunen, and T. Virtanen, "Convolutional recurrent neural networks for polyphonic sound event detection," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 25, no. 6, pp. 1291–1303, Jun. 2017.

[14] J. Naranjo-Alcazar, S. Perez-Castanos, P. Zuccarello, and M. Cobos, "Acoustic scene classification with squeeze-excitation residual networks," *IEEE Access*, vol. 8, pp. 112287–112296, 2020.

[15] J. Naranjo-Alcazar, S. Perez-Castanos, P. Zuccarello, F. Antonacci, and M. Cobos, "Open set audio classification using autoencoders trained on few data," *Sensors*, vol. 20, no. 13, p. 3741, Jul. 2020.

[16] M. Valenti, A. Diment, G. Parascandolo, S. Squartini, and T. Virtanen, "DCASE 2016 acoustic scene classification using convolutional neural networks," in *Proc. Workshop Detection Classification Acoust. Scenes Events*, 2016, pp. 95–99.

[17] A. Mesaros, T. Heittola, and T. Virtanen, "Acoustic scene classification in DCASE 2019 challenge: Closed and open set classification and data mismatch setups," in *Proc. Detection Classification Acoustic Scenes Events Workshop (DCASE)*, New York, NY, USA, Oct. 2019, pp. 164–168.

[18] T. Lidy and A. Schindler, "CQT-based convolutional neural networks for audio scene classification," in *Proc. Detection Classification Acoustic Scenes Events Workshop (DCASE)*, vol. 90, 2016, pp. 1032–1048.

[19] K. J. Piczak, "The details that matter: Frequency resolution of spectrograms in acoustic scene classification," in *Proc. Detection Classification Acoustic Scenes Events Workshop (DCASE)*, Nov. 2017, pp. 103–107.

[20] W. Dai, C. Dai, S. Qu, J. Li, and S. Das, "Very deep convolutional neural networks for raw waveforms," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Mar. 2017, pp. 421–425.

[21] S. Qu, J. Li, W. Dai, and S. Das, "Understanding audio pattern using convolutional neural network from raw waveforms," 2016, *arXiv:1611.09524*. [Online]. Available: http://arxiv.org/abs/1611.09524

[22] J. Lee, J. Park, K. Kim, and J. Nam, "SampleCNN: End-to-end deep convolutional neural networks using very small filters for music classification," *Appl. Sci.*, vol. 8, no. 1, p. 150, Jan. 2018.

[23] Y. Gong and C. Poellabauer, "How do deep convolutional neural networks learn from raw audio waveforms?" Tech. Rep., 2018. [Online]. Available: https://openreview.net/pdf?id=S1Ow_e-Rb

[24] J. Lee, J. Park, K. L. Kim, and J. Nam, "Sample-level deep convolutional neural networks for music auto-tagging using raw waveforms," 2017, *arXiv:1703.01789*. [Online]. Available: http://arxiv.org/abs/1703.01789

[25] J. J. Huang and J. J. A. Leanos, "AclNet: Efficient end-to-end audio classification CNN," 2018, *arXiv:1811.06669*. [Online]. Available: http://arxiv.org/abs/1811.06669

[26] J. Vera-Diaz, D. Pizarro, and J. Macias-Guarasa, "Towards end-to-end acoustic localization using deep learning: From audio signals to source position coordinates," *Sensors*, vol. 18, no. 10, p. 3418, Oct. 2018.

[27] S. Abdoli, P. Cardinal, and A. L. Koerich, "End-to-end environmental sound classification using a 1D convolutional neural network," *Expert Syst. Appl.*, vol. 136, pp. 252–263, Dec. 2019.

[28] T. Kim, J. Lee, and J. Nam, "Comparison and analysis of SampleCNN architectures for audio classification," *IEEE J. Sel. Topics Signal Process.*, vol. 13, no. 2, pp. 285–297, May 2019.

[29] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[30] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 630–645.

[31] T. Kim, J. Lee, and J. Nam, "Sample-level CNN architectures for music auto-tagging using raw waveforms," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2018, pp. 366–370.

[32] J.-W. Jung, H.-S. Heo, I. Yang, S.-H. Yoon, H.-J. Shim, and H.-J. Yu. (2017). *DNN-Based Audio Scene Classification for DCASE 2017: Dual Input Features, Balancing Cost, and Stochastic Data Duplication Detection and Classification of Acoustic Scenes and Events*. [Online]. Available: http://dcase.community/documents/workshop2017/proceedings/DCASE2017Work shop_Jung_187.pdf

[33] J. H. Yang, N. K. Kim, and H. K. Kim. (2018). *Se-Resnet With Gan-Based Data Augmentation Applied to Acoustic Scene Classification Detection and Classification of Acoustic Scenes and Events*. [Online]. Available: https://pdfs.semanticscholar.org/e95f/b1ac75c42943c4a74e5c082bfdcc07d90c1f.pdf

[34] M. Liu, W. Wang, and Y. Li. (2019). *The System for Acoustic Scene Classification Using Resnet Detection and Classification of Acoustic Scenes and Events*. [Online]. Available: http://dcase.community/documents/challenge2019/technical_reports/DCASE 2019_SCUT_19.pdf

[35] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7132–7141.

[36] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014. [Online]. Available: http://jmlr.org/papers/v15/srivastava14a.html

[37] J. Salamon, C. Jacoby, and J. P. Bello, "A dataset and taxonomy for urban sound research," in *Proc. 22nd ACM Int. Conf. Multimedia*, 2014, pp. 1041–1044.

[38] K. J. Piczak, "ESC: Dataset for environmental sound classification," in *Proc. 23rd ACM Int. Conf. Multimedia (MM)*, 2015, pp. 1015–1018.

[39] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: http://arxiv.org/abs/1412.6980

[40] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. Int. Conf. Artif. Intell. Statist. (AISTATS)*, 2010, pp. 249–256.

[41] B. McFee, C. Raffel, D. Liang, D. Ellis, M. McVicar, E. Battenberg, and O. Nieto, "Librosa: Audio and music signal analysis in Python," in *Proc. 14th Python Sci. Conf.*, 2015, pp. 18–25.

[42] J. W. Tukey, "Comparing individual means in the analysis of variance," *Biometrics*, vol. 5, no. 2, pp. 99–114, 1949. [Online]. Available: http://www.jstor.org/stable/3001913

[43] S. Garcia and F. Herrera, "An extension on 'statistical comparisons of classifiers over multiple data sets' for all pairwise comparisons," *J. Mach. Learn. Res.*, vol. 9, pp. 2677–2694, Dec. 2008.

JAVIER NARANJO-ALCAZAR (Graduate Student Member, IEEE) received the Telecommunications degree and the master's degree in telecommunications engineering from the Universitat Politècnica de València, Valencia, Spain, in 2016 and 2018, respectively. He is currently pursuing the Ph.D. degree with the Department of Computer Science, Universitat de València, funded by the Torres Quevedo Program and the valencian start-up Visualfy. His research interests include machine listening, few-shot learning, and open-set recognition. He was a recipient of the Best M.Sc. Thesis Award from the Regional Telecommunications Engineering Association in 2019.

SERGI PEREZ-CASTANOS received the Telecommunications degree and the master's degree in telecommunications engineering from the Universitat Politècnica de València, Valencia, Spain, in 2016 and 2018, respectively. He is currently working as a Machine Learning Engineer with Visualfy. His research interests include machine listening, anomaly detection, and audio captioning.

IRENE MARTÍN-MORATÓ (Graduate Student Member, IEEE) received the bachelor's (Hons.) and M.Sc. degrees in telecommunications and the Ph.D. degree in information technology, communications, and computing under the University Faculty Training Program (FPU) from the Universitat de València, in 2014, 2016, and 2019, respectively. She is currently a Postdoctoral Researcher with Tampere University, Finland. Her research interests include acoustic signal processing, machine learning, and audio event detection and classification.

PEDRO ZUCCARELLO received the Electronics Engineering degree from the University of Buenos Aires, Argentina, the M.Sc. degree in telecommunications from the Universitat Politècnica de València, Valencia, Spain, and the Ph.D. degree from the Universitat de València, Valencia. He developed most of his career as a Researcher in several public research and development institutions such as the Institute of Microelectronics of Barcelona, Barcelona, Spain, or the Institute of Corpuscular Physics, Valencia. From 2017 to June 2020, he has developed as the Head of the Artificial Intelligence Group, Visualfy, a private startup company. He currently works as a Senior Artificial Intelligence Researcher with Tyris IA private company. He has coauthored nearly 30 papers in international peer-review journals and conferences in topics that include artificial intelligence, machine learning, signal processing, electronics, and microelectronic design. He received several postdoctoral fellowships such as the Val-I+D, from the Valencian Government, or the Torres Quevedo, from the Spanish Ministry of Science and Education.

FRANCESC J. FERRI (Senior Member, IEEE) received the Licenciado degree in physics (electricity, electronics, and computer science) and the Ph.D. degree in pattern recognition from the Universitat de València, in 1987 and 1993, respectively. He has been with the Computer Science Department, Universitat de València, since 1986. His current research interests include feature selection, nonparametric classification methods, machine learning, computer vision, and image retrieval. He has authored or coauthored more than 100 technical papers in international conferences and well established journals in his fields of interest. He is a member of ACM and IAPR.

MAXIMO COBOS (Senior Member, IEEE) received the master's degree in telecommunications and the Ph.D. degree in telecommunications engineering from the Universitat Politècnica de València, Valencia, Spain, in 2007 and 2009, respectively. In 2011, he joined the Universitat de València, where he is currently an Associate Professor. His research interests include digital signal processing and machine learning for audio and multimedia applications. He has authored/coauthored more than 100 technical papers in international journals and conferences in his areas of interest. He is a member of the Audio Signal Processing Technical Committee of the European Acoustics Association. He completed with honors his studies under the University Faculty Training program (FPU) and was a recipient of the Ericsson Best Ph.D. Thesis Award from the Spanish National Telecommunications Engineering Association. In 2010, he received the Campus de Excelencia Postdoctoral Fellowship to work at the Institute of Telecommunications and Multimedia Applications. He serves as an Associate Editor for IEEE SIGNAL PROCESSING LETTERS and the *EURASIP Journal on Audio, Speech, and Music Processing*.