DCASE 2019: CNN DEPTH ANALYSIS WITH DIFFERENT CHANNEL INPUTS FOR ACOUSTIC SCENE CLASSIFICATION

Technical Report

Javier Naranjo-Alcazar, Sergi Perez-Castanos, Pedro Zuccarello　　　　*Maximo Cobos*

Visualfy
Benisanó, València, Spain
{javier.naranjo, sergi.perez, pedro.zuccarello}@visualfy.com

Universitat de València
València, Spain
maximo.cobos@uv.es

ABSTRACT

The objective of this technical report is to describe the framework used in Task 1, Acoustic scene classification (ASC), of the DCASE 2019 challenge. The presented approach is based on Log-Mel spectrogram representations and VGG-based Convolutional Neural Networks (CNNs). Three different CNNs, with very similar architectures, have been implemented. The main difference is the number of filters in their convolutional blocks. Experiments show that the depth of the network is not the most relevant factor for improving the accuracy of the results. The performance seems to be more sensitive to the input audio representation. This conclusion is important for the implementation of real-time audio recognition and classification system on edge devices. In the presented experiments the best audio representation is the Log-Mel spectrogram of the harmonic and percussive sources plus the Log-Mel spectrogram of the difference between left and right stereo-channels ($L - R$). Also, in order to improve accuracy, ensemble methods combining different model predictions with different inputs are explored. Besides geometric and arithmetic means, ensembles aggregated with the Orness Weighted Averaged (OWA) operator have shown interesting and novel results. The proposed framework outperforms the baseline system by 14.34 percentage points. For Task 1a, the obtained development accuracy is 76.84%, being 62.5% the baseline, whereas the accuracy obtained in public leaderboard is 77.33%, being 64.33% the baseline.

Index Terms— Deep Learning, Convolutional Neural Networks, Acoustic Scene Classification, Mel-Spectrogram, HPSS, DCASE2019, Ensemble Methods, OWA

1. INTRODUCTION

Sounds carry a large amount of information about the everyday environment. Therefore, developing methods to automatically extract this information has a huge potential in relevant applications such as autonomous cars or home assistants. In [1], an audio scene is described as a collection of sound events on top of some ambient noise. Given a predefined set of tags where each tag describes a different audio scene (e.g. airport, public park, metro, etc.) and given an audio clip coming from a particular audio scene, Audio Scene Classification (ASC) consists in the automatic assignment of one single tag to describe the content of the audio clip.

The objective of the Task 1 of DCASE 2019 Challenge [2] is to encourage the participants to propose different solutions to tackle the ASC problem in a public tagged audio dataset. The first edition
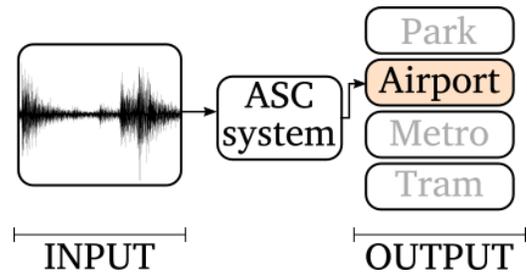

Figure 1: Acoustic Scene Classification framework. Given an audio input, the system must classify it into a given predefined class.

of the DCASE challenge took place in 2013 [3], showing the increasing interest on ASC among the scientific community. The following editions took place in 2016, 2017 and 2018. The challenge has been a backbone element for researches in the audio signal processing and machine learning area. While submissions in the first edition were mostly based on Gaussian mixture models (GMMs) [4] or Support Vector Machines (SVMs) [5], deep learning methods such as those based on Convolutional Neural Networks (CNNs) took the lead from 2016. The use of these techniques is directly correlated with the amount of data available. Task 1 of DCASE 2019 provides around 15000 audio clips per subtask, each one of 10 seconds duration, made available to the participants for algorithm development. This task contains 3 subtasks approaching different issues: (1) ASC, (2) ASC with mismatched recording devices and (3) Open-set ASC; this last one aimed at solving the problem of identifying audio clips that do not belong to any of the predefined classes used for training. Each subtask has its own training and test datasets: TAU Urban Acoustic Scenes 2019, TAU Urban Acoustic Scenes 2019 Mobile and TUT Urban Acoustic Scenes 2019 Openset respectively.

First approaches to ASC relied exclusively on feature-engineering [6, 7]. Most research efforts tried to develop meaningful features to feed classical classifiers, such as GMMs or SVMs [8]. Over the last years, Deep Neural Networks (DNNs) and, particularly CNNs, have shown remarkable results in many different areas [9, 10, 11], thus being the most popular choice among researchers and application engineers. CNNs allow to solve both problems, feature extraction and classification, into one single structure. In fact, deep features extracted from the internal layers of a pre-trained CNN can be successfully used for transfer learning in audio classi-

fication [12].

Although several works for automatic audio classification have successfully proposed to feed the CNN with an 1D audio signal [13, 14, 15], most of the research on this field has been focused on using a 2D time-frequency representation as a starting point [16, 17, 18]. With this last approach, some parameters, such as window type, size and/or overlap, need to be chosen. The advantage of 2D time-frequency representations is that they can be treated and processed with CNNs that have shown successful results with images. In the present work, six different time-frequency representations based on the well-known Log Mel spectrogram are selected a inputs to CNNs. With the objective of aggregating classification probabilities [19] and analyze the impact over the accuracy of the network size, three different VGG-based CNNs have been implemented. A performance study has been carried out to match each input type with its most suitable CNN.

2. METHOD

This section describes the method proposed for this challenge. First, a brief background on CNNs is provided, explaining the most common layers used in their design. Then, the pre-processing applied over the audio clips before being fed to the network is explained. Finally, the submitted CNN is presented.

2.1. Convolutional Neural Networks

CNNs were first presented by LeCun *et al.* in 1998 [20] to classify digits. Since then, a large amount of research has been carried out to improve this technique, resulting in multiple improvements such as new layers, generalization techniques or more sophisticated networks like ResNet [21], Xception [22] or VGG [10]. The main feature of a CNN is the presence of convolutional layers. These layers perform filtering operations by shifting a small window (receptive field) across the input signal, either 1D or 2D. These windows contain kernels (weights) that change their values during training according to a cost function. Common choices for the non-linear activation function in convolutional layers are the Rectified Linear Unit (ReLU) or the Leaky Rectified Linear Unit (Leaky ReLU) [23]. The activations computed by each kernel are known as feature maps, which represent the output of the convolutional layer. Other layers commonly employed in CNNs are Batch normalization and Dropout. These layers are interspersed between the convolutions to achieve a greater regularization and generalization of the network.

Traditional CNNs include one or two fully-connected layers before the final output prediction layer [24, 25]. Nevertheless, recent approaches [26] do not include fully-connected layers before the output layer. Under this approach, known as fully-convolutional CNN, the feature map is reshaped before the output layer by using global max or average pooling techniques. This procedure maps each filter with just one feature, either the maximum value or the average [26]. Most common approaches reshape all feature maps to a 1D vector (Flatten) before the final fully-connected layer used for prediction [27].

2.2. Audio preprocessing

The way audio examples are presented to a neural network can be important in terms of system performance. In the system used in this challenge, a combination of the time-frequency representations detailed in Table 1 has been used as input to the CNN (see Table 2).

All of them are based on the Log Mel spectrogram [26, 24, 28] with 40 ms of analysis window, 50% of overlap between consecutive windows, Hamming asymmetric windowing, FFT of 2048 points and a 64-band normalized Mel-scaled filter bank. Each row, corresponding to a particular frequency band, of this Mel-spectrogram matrix is then normalized according to its mean and standard deviation.

For the case of harmonic and percussive features, a Short-Time Fourier Transform (STFT) of the time waveform with 40 ms of analysis window, 50% of overlap between consecutive windows, Hamming asymmetric windowing and FFT of 2048 points has been taken as starting point. This spectrogram was used to compute the harmonic and percussive sources using the median-filtering HPSS algorithm [29]. The resulting spectrogram was treated in 64 Log Mel frequency bands. Considering that the audio clips are 10 s long, the size of the final Log Mel feature matrix is 64×500 for all the cases. The audio preprocessing has been developed using the LibROSA library for Python [30].

2.3. Proposed Network

The network proposed in this work is inspired in the architecture of the VGG [10], since this has shown successful results in ASC [17, 31, 26, 32]. The convolutional layers were configured with small 3×3 receptive fields. After each convolutional layer, batch normalization and exponential linear unit (ELU) activation layers [33] were stacked. Thus, two consecutive convolutional layers, including their respective batch normalization and activation layers, plus a max pooling and dropout layer correspond to a *convolutional block*. The final network (see Table 3) is composed of three convolutional blocks plus two fully-connected layers acting as classifiers.

Three different values for the number of filters for the first *convolutional block* have been implemented and tested: 16, 32, and 64 (*Vfy-3L16*, *Vfy-3L16* and *Vfy-3L64* in Table 2). The detailed architecture is given in Table 3. The developed network is intended to be used in a real-time embedded system, therefore a compromise has been achieved between the number of parameters and the final classification accuracy.

2.4. Model ensemble

Combining predictions from different classifiers has become a popular technique to increase the accuracy [34]. This is known as *ensemble models*. For this work, different model probabilities have been aggregated using the arithmetic and geometric means as well as the *Orness Weighted Average* (OWA) operator [35]. The following two weight vectors have been used for OWA ensembles: $\mathbf{w}_1 = [0.05, 0.15, 0.8]^T$ and $\mathbf{w}_2 = [0.1, 0.15, 0.75]^T$.

3. RESULTS

3.1. Experimental details

The optimizer used was Adam [36] configured with $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$, $decay = 0.0$ and $amsgrad = True$. The models were trained with a maximum of 2000 epochs. Batch size was set to 32. The learning rate started with a value of 0.001 decreasing with a factor of 0.5 in case of no improvement in the validation accuracy after 50 epochs. If validation accuracy does not improve after 100 epochs, training is early stopped. Keras with Tensorflow backend was used to implement the models.

		Description
M	Mono	Log Mel spectrogram (computed as detailed in Sect. 2.2) of the arithmetic mean of the left and right audio channels.
L	Left	Log Mel spectrogram (computed as detailed in Sect. 2.2) of the left audio channel.
R	Right	Log Mel spectrogram (computed as detailed in Sect. 2.2) of the right audio channel.
D	Difference	Log Mel spectrogram (computed as detailed in Sect. 2.2) of the difference of the left and right audio channels $(L - R)$.
H	Harmonic	Harmonic Log Mel matrix (computed as detailed in Sect. 2.2) using the Mono signal as input.
P	Percussive	Percussive Log Mel matrix (computed as detailed in Sect. 2.2) using the Mono signal as input.

Table 1: Basic audio representations used in this work. Several combinations of the above alternatives have been used as input to the CNN (see Table 2 and Sect. 3).

Audio preprocessing		Networks					
		Vfy-3L16		*Vfy-3L32*		*Vfy-3L64*	
Audio representation	Channels	Dev set	Public LB	Dev set	Public LB	Dev set	Public LB
	Mono (M)	*70.47	70.00	70.07	-	70.49	-
	Left + Right + Difference (LRD)	72.69	-	*73.76	71.16	73.41	-
	Harmonic + Percussive (HP)	71.23	-	*71.85	-	72.04	-
Log Mel spectrogram	Harmonic + Percussive + Mono (HPM)	69.37	-	70.99	-	71.59	-
	Harmonic + Percussive + Difference (HPD)	72.64	-	***75.75**	-	75.44	-
	Harmonic + Percussive + Left + Right (HPLR)	71.57	-	*72.76	-	73.19	-
		PN (3D): 176,926		**PN (3D): 495,150**		**PN (3D): 1,560,142**	

Table 2: Network accuracy (%) in development stage. The accuracy for the development set (Dev set) was calculated using the first evaluation setup of 4185 samples. Public leaderboard accuracy (Public LB) was extracted from Kaggle's public leaderboard composed of 1200 samples. The models labeled with an (*) were used for ensembles (see Table 4).

Visualfy Network Architecture - Vfy-3LX
[conv (3x3, #X), batch normalization, ELU(1.0)] x2
MaxPooling(2,10)
Dropout(0.3)
[conv (3x3, #$2X$), batch normalization, ELU(1.0)] x2
MaxPooling(2,5)
Dropout(0.3)
[conv (3x3, #$3X$), batch normalization, ELU(1.0)] x2
MaxPooling(2,5)
Dropout(0.3)
Flatten
[Dense(100), batch normalization, ELU(1.0)]
Dropout(0.4)
[Dense(10), batch normalization, softmax]

Table 3: Network architecture proposed for this challenge. The name indicates the number of convolutional blocks in the network and the number of filters of the first convolutional block. For example, *Vfy-3L16* means 3 convolutional blocks and the first one starts with 16 filters, the second with 32 filters and the last one with 64 filters.

3.2. Results on the development dataset

Table 2 shows the results obtained for the development dataset with the three networks detailed in Table 3, combined with the different inputs specified in Table 1.

Table 2 shows that when the network is fed with one channel input (M), the shallower network shows the same results as the deepest. On the other hand, when input is fed with more than one channel, a deeper network improves the accuracy with different improvements depending on the selected audio representation. The most suitable representation was HPD. As far as this group is aware, this combination has not been proposed before [34, 37].

On the other hand, when ensembles are used (see subsection 2.4) the accuracy is improved. As shown in Table 4, the combination showing the highest accuracy is LRD + HPD + HPLR. Although *Vfy-3L64* shows, in some cases, better accuracy in Dev set, this improvement is not correlated in Public Leaderborad. An interpretation for this could be that the network is more prone to overfitting due to the high number of parameters.

3.3. Subtask 1B

Although all the development has been focused on Task 1a, we have analyzed our networks using Task 1b data. The key of this task is that there is a mismatch among recording devices. These new devices are commonly customer devices such as smartphones or sport cameras. The main differences are the sample rate and the number of channels. Therefore, we decided to work with mono signals in this subtask, resampling them to 32 kHz before any pre-processing. The networks are fed with mono, harmonic and percussive spec-

Ensemble Network	Method	Dev set	Public LB
E_M_LRD_HP	arith. mean	74.74	-
E_M_LRD	arith. mean	75.02	-
E_LRD_HP	arith. mean	75.11	75.83
E_M_HP	arith. mean	72.56	-
E_LRD_HPD	arith. mean	76.48	77.00
E_LRD_HPLR	arith. mean	75.51	-
E_HPD_HPLR	arith. mean	76.81	-
*E_LRD_HPD_HPLR	arith. mean	**76.84**	**77.33**
*E_LRD_HPD_HPLR	geom. mean	77.06	77.00
*E_LRD_HPD_HPLR	OWA1	76.84	76.33
*E_LRD_HPD_HPLR	OWA2	76.94	76.50

Table 4: Ensemble accuracy (%). The name is composed using the abbreviation of the models ensemble shown in Table 2. Initial letter E stands for *ensemble*. Models labeled with an (*) have been submitted for challenge rank.

Network	Method	Public LB
Baseline		43.83
M		53.16
E_M_HP	arith. mean	58.33
E_HP_HPM	arith. mean	60.33
E_M_HP_HPM	arith. mean	**60.66**
E_M_HP_HPM	geom. mean	59.50
E_M_HP_HPM	OWA1	60.33
E_M_HP_HPM	OWA2	60.50

Table 5: Network accuracies (%). The name is composed using the abbreviation of the models ensemble shown in Table 2. Ensemble has been carried out using sum technique. The first "E" is used as initial for *ensemble*.

trograms. As explained in Subsection 3.2, the network used for ensemble with HP or HPM input is *Vfy-3L32* and *Vfy-3L16* in case of mono. This analysis has only been run in Public Leaderboard.

4. CONCLUSION

In order to embed an ASC classifier into an edge system, a study of the network depth becomes a crucial stage. Real-time devices usually work under sharp constraints concerning the classification time. In this technical report, a study of 3 different CNNs has been proposed. It turns out that deeper networks do not always present the best accuracy. In our study, we conclude that there is not a substantial difference in accuracy between a model of 0.5M parameters or 1.5M parameters. Therefore, taking into account that the majority of CNN architectures proposed in the literature follow a repeating structure that duplicates the number of filters in each repetition, it seems that an appropriate selection of the number of

filters in the first layer is essential for real-time applications. Although our network is not very deep for preventing overfitting, this aspect should be considered more carefully when designing deeper networks. In addition, a brief study on the use of different model ensembles has been presented, showing that even simple averaging already improves the final accuracy. Finally, we have discussed the importance of the right input (audio representation) when training a CNN. The novel HPD representation shows the best accuracy in all the three networks, which suggests that testing alternative input representations might be more worthy than tuning complex and consuming networks.

5. ACKNOWLEDGMENT

6. REFERENCES

[1] Y. Han and K. Lee, "Acoustic scene classification using convolutional neural network and multiple-width frequency-delta data augmentation," *arXiv preprint arXiv:1607.02383*, 2016.

[2] A. Mesaros, T. Heittola, and T. Virtanen, "A multi-device dataset for urban acoustic scene classification," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2018 Workshop (DCASE2018)*, November 2018, pp. 9–13. [Online]. Available: https://arxiv.org/abs/1807.09840

[3] D. Giannoulis, E. Benetos, D. Stowell, M. Rossignol, M. Lagrange, and M. D. Plumbley, "Detection and classification of acoustic scenes and events: An ieee aasp challenge," in *2013 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*. IEEE, 2013, pp. 1–4.

[4] M. Chum, A. Habshush, A. Rahman, and C. Sang, "Ieee aasp scene classification challenge using hidden markov models and frame based classification," *IEEE AASP Challenge on Detection and Classification of Acoustic Scenes and Events*, 2013.

[5] J. T. Geiger, B. Schuller, and G. Rigoll, "Large-scale audio feature extraction and svm for acoustic scene classification," in *2013 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*. IEEE, 2013, pp. 1–4.

[6] I. Martin-Morato, M. Cobos, and F. J. Ferri, "A case study on feature sensitivity for audio event classification using support vector machines," in *26th IEEE International Workshop on Machine Learning for Signal Processing, MLSP 2016, Vietri sul Mare, Salerno, Italy, September 13-16, 2016*, 2016, pp. 1–6. [Online]. Available: https://doi.org/10.1109/MLSP.2016.7738834

[7] D. Stowell, D. Giannoulis, E. Benetos, M. Lagrange, and M. D. Plumbley, "Detection and classification of acoustic scenes and events," *IEEE Transactions on Multimedia*, vol. 17, no. 10, pp. 1733–1746, Oct 2015.

[8] I. Martin-Morato, M. Cobos, and F. J. Ferri, "Adaptive mid-term representations for robust audio event classification," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 12, pp. 2381–2392, Dec 2018.

[9] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 1097–1105.

[10] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[11] A. Bhandare, M. Bhide, P. Gokhale, and R. Chandavarkar, "Applications of convolutional neural networks," *International Journal of Computer Science and Information Technologies*, vol. 7, no. 5, pp. 2206–2215, 2016.

[12] I. Martin-Morato, M. Cobos, and F. J. Ferri, "On the robustness of deep features for audio event classification in adverse environments," in *2018 14th IEEE International Conference on Signal Processing (ICSP)*, Aug 2018, pp. 562–566.

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

[14] S. Qu, J. Li, W. Dai, and S. Das, "Understanding audio pattern using convolutional neural network from raw waveforms," *arXiv preprint arXiv:1611.09524*, 2016.

[15] Y. Aytar, C. Vondrick, and A. Torralba, "Soundnet: Learning sound representations from unlabeled video," in *Advances in neural information processing systems*, 2016, pp. 892–900.

[16] T. Lidy and A. Schindler, "Cqt-based convolutional neural networks for audio scene classification," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2016 Workshop (DCASE2016)*, vol. 90. DCASE2016 Challenge, 2016, pp. 1032–1048.

[17] Y. Su, K. Zhang, J. Wang, and K. Madani, "Environment sound classification using a two-stream cnn based on decision-level fusion," *Sensors*, vol. 19, no. 7, p. 1733, 2019.

[18] E. Cakır, T. Heittola, and T. Virtanen, "Domestic audio tagging with convolutional neural networks," *IEEE AASP Challenge on Detection and Classification of Acoustic Scenes and Events (DCASE 2016)*, 2016.

[19] C. N. Silla Jr, C. A. Kaestner, and A. L. Koerich, "Automatic music genre classification using ensemble of classifiers," in *2007 IEEE International Conference on Systems, Man and Cybernetics*. IEEE, 2007, pp. 1687–1692.

[20] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, *et al.*, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[21] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[22] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1251–1258.

[23] B. Xu, N. Wang, T. Chen, and M. Li, "Empirical evaluation of rectified activations in convolutional network," *arXiv preprint arXiv:1505.00853*, 2015.

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

[25] J. Salamon and J. P. Bello, "Deep convolutional neural networks and data augmentation for environmental sound classification," *IEEE Signal Processing Letters*, vol. 24, no. 3, pp. 279–283, 2017.

[26] Q. Kong, T. Iqbal, Y. Xu, W. Wang, and M. D. Plumbley, "Dcase 2018 challenge baseline with convolutional neural networks," *arXiv preprint arXiv:1808.00773*, 2018.

[27] M. Valenti, S. Squartini, A. Diment, G. Parascandolo, and T. Virtanen, "A convolutional neural network approach for acoustic scene classification," in *2017 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2017, pp. 1547–1554.

[28] Y. Xu, Q. Kong, W. Wang, and M. D. Plumbley, "Large-scale weakly supervised audio classification using gated convolutional neural network," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 121–125.

[29] D. FitzGerald, "Harmonic/percussive separation using median filtering," in *Proc. of the 13th Int. Conference on Digital Audio Effects (DAFx-10)*, Graz (Austria), September 2010.

[30] B. McFee, C. Raffel, D. Liang, D. P. Ellis, M. McVicar, E. Battenberg, and O. Nieto, "librosa: Audio and music signal analysis in python," in *Proceedings of the 14th python in science conference*, 2015, pp. 18–25.

[31] L. Zhang and J. Han, "Acoustic scene classification using multi-layer temporal pooling based on convolutional neural network," *arXiv preprint arXiv:1902.10063*, 2019.

[32] Y. Han and K. Lee, "Convolutional neural network with multiple-width frequency-delta data augmentation for acoustic scene classification," *IEEE AASP Challenge on Detection and Classification of Acoustic Scenes and Events*, 2016.

[33] D.-A. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and accurate deep network learning by exponential linear units (elus)," *arXiv preprint arXiv:1511.07289*, 2015.

[34] Y. Sakashita and M. Aono, "Acoustic scene classification by ensemble of spectrograms based on adaptive temporal divisions," *IEEE AASP Challenge on DCASE 2018 technical reports*, 2018.

[35] T. León, P. Zuccarello, G. Ayala, E. de Ves, and J. Domingo, "Applying logistic regression to relevance feedback in image retrieval systems," *Pattern Recognition*, vol. 40, no. 10, pp. 2621–2632, 2007.

[36] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[37] Y. Han, J. Park, and K. Lee, "Convolutional neural networks with binaural representations and background subtraction for acoustic scene classification," *the Detection and Classification of Acoustic Scenes and Events (DCASE)*, pp. 1–5, 2017.