Classificação de Cenas Acústicas Utilizando Técnicas de Aprendizagem Profunda

Daniel Gomes de Pinho Zanco, Walter Antônio Gontijo e Eduardo Luiz Ortiz Batista

***Resumo*— Neste artigo são apresentadas contribuições que visam aumentar o desempenho do classificador de cenas acústicas *baseline* proposto no contexto do *Detection and Classification of Acoustic Scenes and Events 2018* (DCASE2018) *Challenge*. Tais contribuições consistem em alterações na estrutura da rede neural convolucional como também na utilização de estratégias de data augmentation e ensemble de modelos. Os resultados obtidos mostram uma acurácia de 72,04% para o conjunto de desenvolvimento e 68,5% para o conjunto de avaliação, significativamente superiores às do *baseline*, que é de 59,7% e 61,0%, respectivamente.**

***Palavras-Chave*— Classificação de Cenas Acústicas, DCASE, Aprendizagem Profunda, Redes Neurais Convolucionais.**

***Abstract*— This paper presents new contributions aiming at enhancing the performance of the baseline acoustic scene classifier proposed in the context of the Detection and Classification of Acoustic Scenes and Events 2018 (DCASE2018) Challenge. These contributions consist on modifications of the structure of the baseline convolutional neural network as well as on the use of data augmentation and model ensemble strategies. As a result, an accuracy of 72.04% is obtained for the development dataset, whereas 68.5% accuracy is obtained for the evaluation dataset. This corresponds to a significantly better performance in comparison with the baseline system, which attained 59.7% and 61.0%, accuracies for the development and evaluation datasets, respectively.**

***Keywords*— Acoustic Scene Classification, DCASE, Deep Learning, Convolutional Neural Networks.**

I. Introdução

Cenas acústicas são formadas da combinação, ou mistura, de diferentes fontes sonoras, tipicamente de um cenário real, como um parque ou um escritório. A classificação de cenas acústicas, ou *Acoustic Scene Classification* (ASC), por sua vez, é a tarefa de reconhecer ambientes a partir dos sons adquiridos nos mesmos [1], [2]. Os sistemas de ASC possuem diferentes campos de aplicação, tais como monitoramento [3], navegação robótica [4] e dispositivos portáteis inteligentes [5].

Atualmente, o *Detection and Classification of Acoustic Scenes and Events* (DCASE) *Workshop/Challenge* [6], [7], [8] tem sido um evento de destaque em ASC, no qual pesquisadores competem no desenvolvimento de sistemas de ASC a partir de conjuntos de dados padronizados. Nas diferentes edições desse evento, vários modelos baseados em técnicas de aprendizagem de máquina têm sido propostos, tais como modelos baseados em máquinas de vetores de suporte [9], modelos ocultos de Markov [10] e, mais recentemente, redes neurais convolucionais (CNNs) [11]. Quanto aos atributos de sinais de áudio, a representação em tempo-frequência foi a escolha de diversos dos sistemas envolvidos no DCASE *Challenge* de 2017, sendo que os sistemas de melhor desempenho utilizaram espectrogramas mel-logarítmicos (*log mel spectrograms*) [11], [12]. Técnicas de *data augmentation*, tais como *pitch shifting* [13] e redes adversárias generativas (GANs) [12], também vêm sendo bastante utilizadas visando aumentar a eficiência a partir de novos dados gerados artificialmente.

No contexto do DCASE *Challenge* (*Task 1A*) de 2018 [14], o conjunto de dados (*dataset*) disponibilizado foi o *TUT Urban Acoustic Scenes 2018* (TUT2018). Esse *dataset* é constituído de gravações realizadas em diferentes locais de seis cidades europeias e 10 tipos de cenas acústicas distintas. O áudio está dividido em trechos de 10 segundos, com indicações de cidade, cena acústica e local de gravação. Os trechos são estéreo com taxa de amostragem de 48 kHz e 24 bits de resolução. No total, há 24 horas de gravações com 864 segmentos de cada cena acústica (144 minutos de áudio). O TUT2018 [14] é ainda organizado em conjuntos de desenvolvimento (*development set*) e avaliação (*evaluation set*). O primeiro conjunto é sub-dividido em treinamento e teste. Já o segundo é utilizado pelos organizadores do desafio para a avaliação dos algoritmos propostos. Além disso, é disponibilizado um sistema de classificação *baseline* como referência, para fins de comparação com os sistemas dos participantes.

No presente trabalho, são apresentadas contribuições ao sistema *baseline* do TUT2018 [14] com o objetivo de se obter um melhor desempenho de classificação. Nesse contexto, são propostas estratégias de *data augmentation* e de conjunto (*ensemble*) de modelos, além de modificações na estrutura da rede neural utilizada pelo sistema [15].

Este artigo é organizado como segue: Na Seção II, o sistema de classificação *baseline* é descrito. Na Seção III, são descritos os detalhes do sistema proposto. Na Seção IV, são apresentados os resultados experimentais. Finalmente, na Seção V, são apresentadas as conclusões do trabalho.

II. Sistema de Classificação Baseline

O sistema de classificação *baseline* é apresentado em [14] e se baseia na arquitetura de CNNs proposta por [16]. Tal sistema, cuja estrutura está descrita na Tabela I, obteve um desempenho de destaque no DCASE *Challenge* de 2016. No *baseline*, cada trecho de áudio de 10 segundos é convertido de estéreo para mono e então o respectivo *log mel spectrogram* é calculado. Os trechos de áudio são divididos em 500 *frames* de

Daniel Gomes de Pinho Zanco, Walter Antônio Gontijo e Eduardo Luiz Ortiz Batista, LINSE - Laboratório de Circuitos e Processamento de Sinais, Departamento de Engenharia Elétrica e Eletrônica, Universidade Federal de Santa Catarina (UFSC), Florianópolis, SC, Brasil, e-mails: zanco@linse.ufsc.br, walter@linse.ufsc.br e ebatista@linse.ufsc.br.

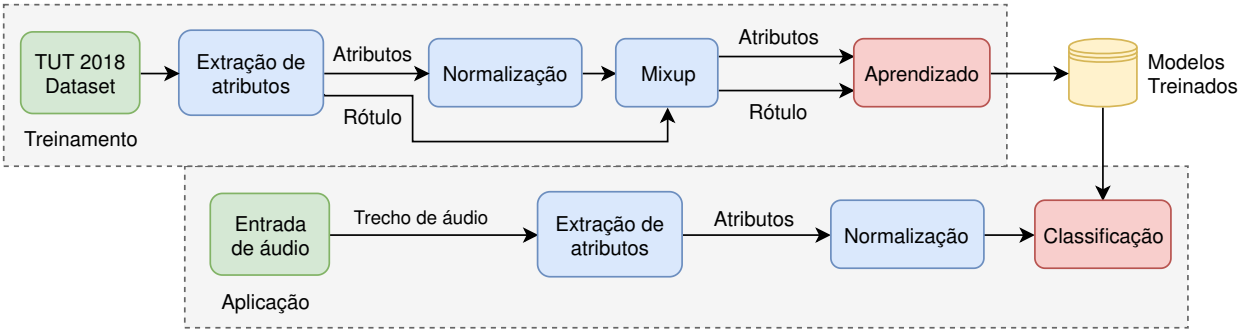

Fig. 1. Diagrama de blocos do sistema proposto de classificação de cenas acústicas.

40 ms, com *overlap* de 20 ms, e, para cada *frame*, são extraídas 40 bandas mel, resultando em representações de dimensão $(40, 500)$. Ao final desse processo, é feita a normalização dos atributos sobre todos os dados do conjunto de treinamento.

Camada	Detalhes		Saída
Entrada	Log mel spectrogram	shape: (40, 500)	(40, 500, 1)
CNN #1	Conv2D	32 filtros, (7,7)	(40, 500, 32)
	Batch Normalization	-	(40, 500, 32)
	Ativação ReLU	-	(40, 500, 32)
	Max Pooling 2D	pool size: (5,5)	(8, 100, 32)
	Dropout	taxa: 30%	(8, 100, 32)
CNN #2	Conv2D	64 filtros, (7,7)	(8, 100, 64)
	Batch Normalization	-	(8, 100, 64)
	ReLU	-	(8, 100, 64)
	Max Pooling 2D	pool size: (4,100)	(2, 1, 64)
	Dropout	taxa: 30%	(2, 1, 64)
Flatten	-	-	(128)
MLP #1	MLP	ReLU, 100 un.	(100)
	Dropout	taxa: 30%	(100)
Saída	MLP	Softmax, 10 un.	(10)

TABELA I
ESTRUTURA DA REDE NEURAL DO SISTEMA BASELINE.

O sistema *baseline* é treinado por 200 épocas com *minibatches* de 16 exemplos. Um otimizador Adam [17] com taxa de aprendizagem de 10^{-3} é usado em tal treinamento. Utiliza-se, ainda, um conjunto de validação que consiste de aproximadamente 30% do conjunto de treinamento, selecionado de forma que ambos conjuntos não tenham gravações de um mesmo local. O desempenho do modelo é avaliado a cada época sobre o conjunto de validação, e o modelo de melhor desempenho é escolhido, alcançando uma acurácia de 59,7% ($\pm$ 0,7) sobre o conjunto de teste do TUT2018 [14].

III. Sistema Proposto

Neste trabalho são propostas modificações ao *baseline* a fim de obter um sistema com melhor desempenho [15]. Na Figura 1, o diagrama de blocos do sistema proposto é ilustrado, o qual é constituído das etapas de treinamento e aplicação.

A. Extração de Atributos

Tal qual no sistema *baseline*, é realizada a conversão dos segmentos de áudio para mono e, em seguida, a extração de atributos *log mel spectrogram*, conforme apresentado em [14]. Similarmente, para o cálculo do espectrograma, os trechos de áudio são divididos em *frames* de 40 ms, com *overlap* de 20 ms. Na Figura 2, o diagrama de blocos do processo de extração de atributos é mostrado.

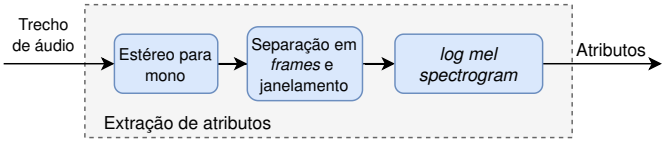

Fig. 2. Diagrama de blocos da extração de atributos.

Observa-se, no processo de extração de atributos, a existência de uma relação de compromisso na escolha da quantidade de bandas mel por *frame*. Um maior detalhamento da representação em frequência do sinal pode ser obtido com um número elevado de bandas mel, o que pode facilitar a diferenciação entre cenas acústicas. Entretanto, a complexidade do modelo de aprendizagem também cresce, o que pode deixá-lo mais suscetível a *overfitting*. Por esse motivo, optou-se por escolher experimentalmente o número de bandas mel extraídas, de forma que os atributos extraídos são um tensor de dimensões $(n_{\text{mel}}, n_{\text{frames}})$. Por diferenças de implementação entre o *baseline* e o sistema proposto, a separação do áudio em *frames* resulta em 1 *frame* extra, assim o número de *frames* (n_{frames}) é 501.

B. Estrutura da Rede Neural

A estrutura da rede neural proposta é baseada na arquitetura do *baseline*, com algumas modificações, e está descrita na Tabela II. Dentre tais modificações, destaca-se a adição de uma terceira camada convolucional (CNN #3). Com o aumento da profundidade da rede, espera-se que o modelo represente relações mais complexas entre sua entrada e saída [18], o que pode resultar em uma melhora no desempenho do classificador.

Além disso, o formato do *kernel* dos filtros e do *pool size* foram escolhidos experimentalmente de forma a operar somente sobre a dimensão temporal, que compreende os *frames* do tensor de entrada. Dessa forma, o número de bandas mel (n_{mel}) do tensor de entrada influencia diretamente no número de parâmetros da camada MLP #1 e pode ser visto como um hiperparâmetro de capacidade da rede proposta.

C. Data Augmentation

Para melhorar a generalização de um modelo de aprendizagem, o ideal é treiná-lo com a maior quantidade de dados possível [19]. Entretanto, a quantidade de dados disponível

Camada	Detalhes		Saída
Entrada	Log mel spectrogram	shape: (n_{mel}, 501)	(n_{mel}, 501, 1)
CNN #1	Conv2D	32 filtros, (1,7)	(n_{mel}, 501, 32)
	Batch Normalization	-	(n_{mel}, 501, 32)
	Ativação ReLU	-	(n_{mel}, 501, 32)
	Max Pooling 2D	pool size: (1,5)	(n_{mel}, 100, 32)
	Dropout	taxa: 30%	(n_{mel}, 100, 32)
CNN #2	Conv2D	64 filtros, (1,7)	(n_{mel}, 100, 64)
	Batch Normalization	-	(n_{mel}, 100, 64)
	Ativação ReLU	-	(n_{mel}, 100, 64)
	Max Pooling 2D	pool size: (1,5)	(n_{mel}, 20, 64)
	Dropout	taxa: 30%	(n_{mel}, 20, 64)
CNN #3	Conv2D	32 filtros, (1,7)	(n_{mel}, 20, 32)
	Batch Normalization	-	(n_{mel}, 20, 32)
	Ativação ReLU	-	(n_{mel}, 20, 32)
	Max Pooling 2D	pool size: (1,16)	(n_{mel}, 1, 32)
	Dropout	taxa: 30%	(n_{mel}, 1, 32)
Flatten	-	-	(32 n_{mel})
MLP #1	MLP	ReLU, 100 un.	(100)
	Dropout	taxa: 30%	(100)
Saída	MLP	Softmax, 10 un.	(10)

TABELA II
ESTRUTURA DA REDE NEURAL DO SISTEMA PROPOSTO.

sempre é limitada. Logo, é útil criar dados artificialmente e adicioná-los ao conjunto de treinamento, uma técnica chamada de *data augmentation*. Essa técnica foi utilizada no contexto de classificação de cenas acústicas em [12] com GANs, alcançando o melhor desempenho do DCASE *Challenge* de 2017.

No presente trabalho, é utilizada a técnica *mixup data augmentation* [20], também usada em [21] e [22]. No *mixup*, exemplos virtuais são gerados a partir da interpolação linear entre dois pares exemplo/rótulo amostrados aleatoriamente do conjunto de treinamento. Assim, novos pares atributo/rótulo $(\widetilde{\mathbf{x}}, \widetilde{\mathbf{y}})$ são gerados de acordo com as seguintes equações:

$$\begin{aligned} \widetilde{\mathbf{x}} &= \lambda \mathbf{x}^{(i)} + (1-\lambda)\mathbf{x}^{(j)}, \\ \widetilde{\mathbf{y}} &= \lambda \mathbf{y}^{(i)} + (1-\lambda)\mathbf{y}^{(j)}, \end{aligned} \tag{1}$$

onde $(\mathbf{x}^{(i)}, \mathbf{y}^{(i)})$ e $(\mathbf{x}^{(j)}, \mathbf{y}^{(j)})$ são dois exemplos escolhidos aleatoriamente do conjunto de treinamento, e $\lambda \in [0, 1]$ é uma variável aleatória amostrada da distribuição Beta(α, α), para $\alpha \in (0, \infty)$.

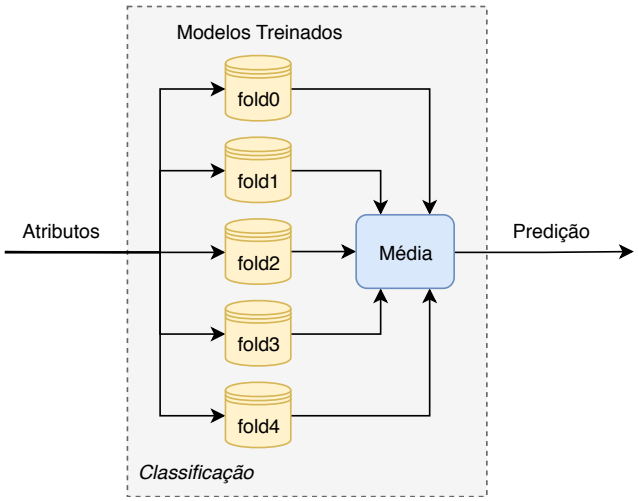

Fig. 3. Diagrama de blocos do processo de classificação utilizado na etapa de aplicação.

D. Treinamento e Aplicação

Na etapa de treinamento do sistema, é utilizado um esquema de validação cruzada *k-fold* [23], com $k = 5$. Em tal processo o conjunto de treinamento é amostrado aleatoriamente em k partições de mesmo tamanho. Assim como no caso do conjunto de validação do sistema *baseline*, as k partições são selecionadas de forma que diferentes partições não contenham gravações de um mesmo local. Cada uma das partições é utilizada como conjunto de validação e as $k - 1$ restantes como de treinamento. É realizado o treinamento de 5 modelos de aprendizagem, um para cada partição, sem repetir os conjuntos de validação. Além disso, para cada partição, é feita a normalização dos atributos a partir da média e desvio padrão sobre o respectivo conjunto de treinamento.

O treinamento de cada modelo é realizado com *minibatches* de 100 exemplos, otimizador Adam [17] com taxa de aprendizagem de 10^{-3} e *mixup* [20] com $\alpha = 0{,}2$, conforme utilizado em [22]. O modelo é treinado até não apresentar melhora na acurácia sobre o conjunto de validação no decorrer das últimas 100 épocas, com um máximo de 600 épocas, em um processo de parada antecipada [24]. O modelo com o melhor desempenho sobre o conjunto de validação é selecionado.

Com os 5 modelos treinados, na etapa de aplicação, ilustrada na Figura 1, é realizado o processo de classificação, descrito na Figura 3. Tal processo consiste no *ensemble* [25] dos modelos obtidos na validação cruzada, onde a saída é a média aritmética da predição dos 5 modelos. A integração dos métodos de *ensemble* com a validação cruzada *k-fold* é sugerida por [21], que observou melhorias de desempenho significativas ao utilizar tal técnica.

E. Implementação

A implementação do sistema proposto foi realizada na linguagem de programação PYTHON e está disponível em repositório do GITHUB do autor[1]. Mais especificamente, o processo de extração de atributos das amostras de áudio foi realizado com o auxílio da biblioteca LIBROSA [26], a qual fornece implementações de funções comumente usadas em análise de música e áudio, incluindo o cálculo de *log mel spectrogram*.

Além disso, para a definição da arquitetura e treinamento de redes neurais, utilizou-se as bibliotecas TENSORFLOW [27] e KERAS [28]. A primeira realiza a definição e execução de expressões matemáticas dos algoritmos de aprendizagem de máquina, além de oferecer um custo computacional reduzido ao permitir o uso de GPUs para os cálculos. A KERAS fornece um nível mais alto de abstração sobre o TENSORFLOW, permitindo que modelos profundos sejam facilmente definidos e treinados.

IV. RESULTADOS EXPERIMENTAIS

O desempenho do sistema proposto foi avaliado sob variações do número de bandas mel extraídas por *frame* e do uso de *mixup data augmentation*. Foram treinadas arquiteturas do sistema com 20, 40, 100 e 200 bandas mel com *mixup*, além de uma versão de 100 bandas mel treinada sem o uso de *mixup*.

[1] https://github.com/dangpzanco/dcase-task1

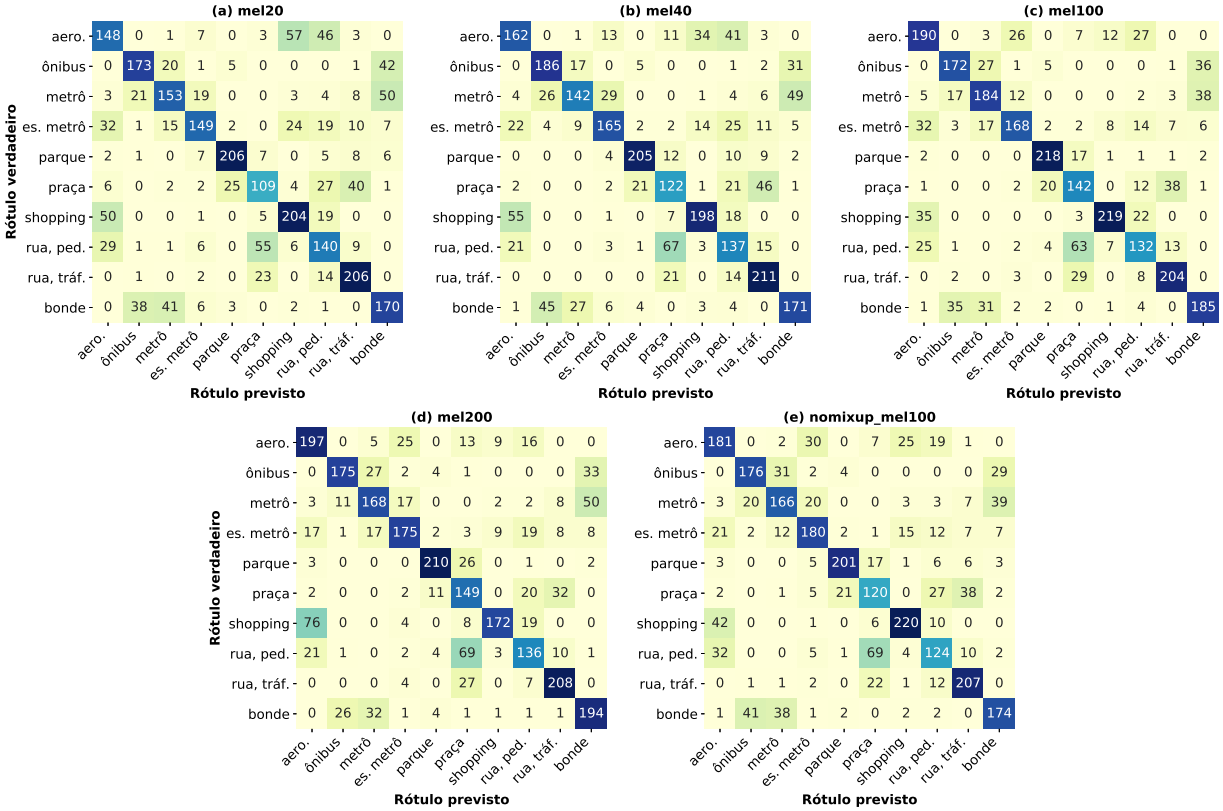

Fig. 4. Matrizes de confusão de diferentes variações do sistema proposto. Predições realizadas sobre o conjunto de teste do TUT2018, *development set*.

Por questão de limitação de memória da GPU, foram utilizados *minibatches* de 50 exemplos no treinamento da arquitetura com 200 bandas mel, mantendo os outros hiperparâmetros inalterados.

Os experimentos foram realizados na versão de avaliação do serviço de computação em nuvem *Google Cloud Platform* [29], que oferece uma máquina virtual *Debian Linux* com 2 CPUs virtuais, 13GB de RAM e uma GPU *Nvidia Tesla K80*.

Na Tabela III, são descritos o desempenho das diferentes variações do sistema proposto para cada *fold* da validação cruzada, assim como o respectivo número de épocas e tempo de treinamento. As arquiteturas com maior número de bandas mel tendem a apresentar um tempo de treinamento mais elevado, apesar de um número de épocas similar, em consequência do maior número de parâmetros.

Na Figura 4, são mostradas as matrizes de confusão das predições de cada variação do sistema proposto sobre o conjunto de teste do TUT2018 (desenvolvimento). É observado um bom desempenho para algumas cenas acústicas, tais como parque, shopping e rua com tráfego (rua, tráf.). No entanto, o sistema proposto não apresenta os mesmos resultados em outras cenas, como praça e rua com pedestres (rua, ped.).

Na Tabela IV, são descritos os desempenhos do *baseline* e das variações do sistema proposto. O sistema proposto obteve os melhores resultados em sua variação *mel100*, com ganhos significativos de acurácia em relação ao *baseline*, aumentando de 59,7% para 72,04% no conjunto de desenvolvimento e de 61,0% para 68,5% no conjunto de avaliação.

(a) 20 bandas mel por *frame*				
Fold	**Épocas**	**Tempo de treinamento**	**Acurácia (%)**	**Função custo**
0	160	14m17s	61,09	1,14
1	593	52m54s	65,44	0,96
2	333	29m57s	67,13	1,01
3	227	20m34s	57,90	1,16
4	307	27m48s	64,71	0,99
(b) 40 bandas mel por *frame*				
Fold	**Épocas**	**Tempo de treinamento**	**Acurácia (%)**	**Função custo**
0	504	1h17m4s	63,70	1,25
1	333	51m18s	67,32	0,94
2	253	38m59s	67,62	0,97
3	258	39m47s	57,74	1,22
4	202	31m23s	65,93	1,03
(c) 100 bandas mel por *frame*				
Fold	**Épocas**	**Tempo de treinamento**	**Acurácia (%)**	**Função custo**
0	401	2h21m19s	62,64	1,17
1	219	1h17m14s	60,44	1,18
2	181	1h4m12s	63,52	1,20
3	579	3h23m33s	57,17	1,21
4	392	2h19m4s	66,83	1,01
(d) 200 bandas mel por *frame*				
Fold	**Épocas**	**Tempo de treinamento**	**Acurácia (%)**	**Função custo**
0	477	5h42m16s	58,97	1,18
1	241	2h53m16s	57,82	1,29
2	318	3h49m24s	57,05	1,22
3	469	5h38m59s	55,70	1,25
4	443	5h20m3s	68,79	0,97
(e) Sem *mixup*, 100 bandas mel por *frame*				
Fold	**Épocas**	**Tempo de treinamento**	**Acurácia (%)**	**Função custo**
0	374	2h11m42s	64,36	1,52
1	258	1h31m2s	65,44	1,00
2	395	2h19m39s	68,20	1,14
3	297	1h44m57s	55,21	1,69
4	266	1h34m17s	65,77	1,17

TABELA III

DETALHES DO TREINAMENTO DAS REDES NEURAIS PARA VARIAÇÕES DO SISTEMA PROPOSTO. A ACURÁCIA E A FUNÇÃO CUSTO SÃO RELATIVAS AOS RESPECTIVOS CONJUNTOS DE VALIDAÇÃO.

	Acurácia (%)	
Sistema	**Desenvolvimento**	**Avaliação**
baseline	59,7	61,0
mel20	65,85	–
mel40	67,47	–
mel100	**72,04**	**68,50**
mel200	70,85	–
nomixup_mel100	69,46	–

TABELA IV

DESEMPENHO DO SISTEMA BASELINE E DAS VARIAÇÕES DO SISTEMA PROPOSTO. ACURÁCIA SOBRE O CONJUNTO DE TESTE DO TUT2018, CONJUNTOS DE DESENVOLVIMENTO E DE AVALIAÇÃO.

Pode-se observar ainda na Tabela IV, que os resultados obtidos nas variações do sistema proposto são superiores ao *baseline*, mostrando a eficácia das contribuições propostas.

Também é possível notar que o desempenho do sistema proposto melhora com o correspondente aumento no número de bandas mel (20-100) conforme apresentado na Tabela IV. É observada uma melhoria na acurácia de aproximadamente 2,5% entre as variações *nomixup_mel100* e *mel100*, mostrando a contribuição da técnica *mixup* de *data augmentation*.

V. CONCLUSÕES

Neste artigo são apresentadas contribuições a um sistema de classificação de cenas acústicas *baseline*, buscando melhorar o desempenho de classificação sobre o conjunto de dados *TUT Urban Acoustic Scenes 2018*. As contribuições propostas incluem estratégias de *data augmentation* e de *ensemble* de modelos, além de modificações do número de camadas convolucionais e de suas características na rede neural utilizada. Diferentes variações do sistema proposto foram avaliadas e, em sua melhor variação, foi obtida uma acurácia de 72,04% no *development set* e 68,5% no *evaluation set*, melhorando significativamente o desempenho em relação ao sistema *baseline*.

REFERÊNCIAS

[1] T. Virtanen, M. D. Plumbley, and D. Ellis, *Computational analysis of sound scenes and events*. Springer, 2018.

[2] D. Barchiesi, D. Giannoulis, D. Stowell, and M. D. Plumbley, "Acoustic scene classification: Classifying environments from the sounds they produce," *IEEE Signal Process. Mag.*, vol. 32, no. 3, pp. 16–34, May 2015.

[3] S. Goetze, J. Schroder, S. Gerlach, D. Hollosi, J.-E. Appell, and F. Wallhoff, "Acoustic monitoring and localization for social care," *Journal of Comput. Sci. and Eng. (JCSE)*, vol. 6, no. 1, pp. 40–50, March 2012.

[4] S. Chu, S. Narayanan, C. c. J. Kuo, and M. J. Mataric, "Where am i? scene recognition for mobile robots using audio features," in *2006 IEEE Int. Conf. on Multimedia and Expo (ICME)*, July 2006, pp. 885–888.

[5] Y. Xu, W. J. Li, and K. K. Lee, *Intelligent Wearable Interfaces*. New York, NY, USA: Wiley-Interscience, 2008.

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

[7] A. Mesaros, T. Heittola, E. Benetos, P. Foster, M. Lagrange, T. Virtanen, and M. D. Plumbley, "Detection and classification of acoustic scenes and events: Outcome of the dcase 2016 challenge," *IEEE/ACM Audio, Speech, Language Process.*, vol. 26, no. 2, pp. 379–393, Feb. 2018.

[8] A. Mesaros, T. Heittola, A. Diment, B. Elizalde, A. Shah, E. Vincent, B. Raj, and T. Virtanen, "DCASE 2017 challenge setup: Tasks, datasets and baseline system," in *Proc. of the Detection and Classification of Acoustic Scenes and Events 2017 Workshop (DCASE2017)*, Nov. 2017, pp. 85–92.

[9] E. Marchi, D. Tonelli, X. Xu, F. Ringeval, J. Deng, S. Squartini, and B. Schuller, "Pairwise decomposition with deep neural networks and multiscale kernel subspace learning for acoustic scene classification," in *Proc. of the Detection and Classification of Acoustic Scenes and Events 2016 Workshop (DCASE2016)*, September 2016, pp. 65–69.

[10] 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.

[11] Y. Han and J. Park, "Convolutional neural networks with binaural representations and background subtraction for acoustic scene classification," DCASE2017 Challenge, Tech. Rep., September 2017.

[12] S. Mun, S. Park, D. Han, and H. Ko, "Generative adversarial network based acoustic scene training set augmentation and selection using SVM hyper-plane," DCASE2017 Challenge, Tech. Rep., September 2017.

[13] B. Lehner, H. Eghbal-Zadeh, M. Dorfer, F. Korzeniowski, K. Koutini, and G. Widmer, "Classifying short acoustic scenes with I-vectors and CNNs: Challenges and optimisations for the 2017 DCASE ASC task," DCASE2017 Challenge, Tech. Rep., September 2017.

[14] A. Mesaros, T. Heittola, and T. Virtanen, "A multi-device dataset for urban acoustic scene classification," in *Proc. 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

[15] D. G. de Pinho Zanco, "Classificação de cenas acústicas utilizando técnicas de aprendizagem profunda," TCC (graduação em Eng. Elétrica), Univ. Federal de Santa Catarina (UFSC), Florianópolis, SC, Brasil, 2018. [Online]. Available: https://repositorio.ufsc.br/handle/123456789/193287

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

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

[18] R. Eldan and O. Shamir, "The power of depth for feedforward neural networks," in *29th Annual Conf. on Learn. Theory (COLT)*, ser. Proc. of Machine Learn. Research, vol. 49. New York, NY, USA: PMLR, 23–26 Jun 2016, pp. 907–940. [Online]. Available: http://proceedings.mlr.press/v49/eldan16.html

[19] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, pp. 236–237, http://www.deeplearningbook.org.

[20] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," in *Int. Conf. on Learn. Representations (ICLR)*, 2018. [Online]. Available: https://openreview.net/forum?id=r1Ddp1-Rb

[21] S. Gharib, H. Derrar, D. Niizumi, T. Senttula, J. Tommola, T. Heittola, T. Virtanen, and H. Huttunen, "Acoustic scene classification: A competition review," in *2018 IEEE 28th Int. Workshop on Mach. Learn. for Signal Process. (MLSP)*. IEEE, 2018, pp. 1–6.

[22] M. Dorfer, B. Lehner, H. Eghbal-zadeh, H. Christop, P. Fabian, and W. Gerhard, "Acoustic scene classification with fully convolutional neural networks and I-vectors," DCASE2018 Challenge, Tech. Rep., September 2018.

[23] J. Friedman, T. Hastie, and R. Tibshirani, *The elements of statistical learning*, 2nd ed. Springer, 2009, vol. 1, no. 10, pp. 241–245.

[24] L. Prechelt, "Early stopping-but when?" in *Neural Networks: Tricks of the trade*. Springer, 1998, pp. 55–69.

[25] R. Caruana, A. Niculescu-Mizil, G. Crew, and A. Ksikes, "Ensemble selection from libraries of models," in *Proc. of the Twenty-first Int. Conf. on Machine Learning*, ser. ICML '04. New York, NY, USA: ACM, 2004, pp. 18–. [Online]. Available: http://doi.acm.org/10.1145/1015330.1015432

[26] Brian McFee, Colin Raffel, Dawen Liang, Daniel P.W. Ellis, Matt McVicar, Eric Battenberg, and Oriol Nieto, "librosa: Audio and Music Signal Analysis in Python," in *Proc. of the 14th Python in Sci. Conf. (SciPy 2015)*, Kathryn Huff and James Bergstra, Eds., 2015, pp. 18 – 25.

[27] M. Abadi *et al.* (2015) TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org. [Online]. Available: https://www.tensorflow.org/

[28] F. Chollet *et al.* (2015) Keras. [Online]. Available: https://keras.io

[29] Google Cloud Computing: AI Platform Deep Learning VM Image. [Online]. Available: https://cloud.google.com/deep-learning-vm/