A MULTIMODAL WAVETRANSFORMER ARCHITECTURE CONDITIONED ON OPENL3 EMBEDDINGS FOR AUDIO-VISUAL SCENE CLASSIFICATION

Technical Report

Andreas Triantafyllopoulos[1,2], *Konstantinos Drossos*[3], *Alexander Gebhard*[2], *Alice Baird*[2], *Björn Schuller*[1,2,4]

[1]audEERING GmbH, Gilching, Germany
[2]EIHW – Chair of Embedded Intelligence for Healthcare and Wellbeing, University of Augsburg, Augsburg, Germany
[3]Audio Research Group, Tampere University, Tampere, Finland
[4]GLAM – Group on Language, Audio, and Music, Imperial College, London, UK
atriant@audeering.com

ABSTRACT

In this report, we present our submission systems to TASK1B of the DCASE2021 Challenge. We submit a total of four systems: one purely audio-based and three multimodal variants of the same architecture. The main module consists of the WaveTransformer architecture, which was recently introduced for automatic audio captioning (AAC). We first adapt the architecture to the task of acoustic scene classification (ASC), and then extend it to handle multimodal signals by globally conditioning all layers on multimodal OpenL3 embeddings. As data augmentation, we apply time- and frequency-bin masking, as well as random cropping. Our best-effort system achieves a log-loss of 0.568 and an accuracy of $79.5\,\%$.

***Index Terms*—** audio-visual scene classification, multimodal fusion, DCASE Challenge

1. INTRODUCTION

Obtaining an understanding of the acoustic scene has many real-world benefits, ranging from security [1] to workplace wellbeing [2]. However, accurate recognition of a given acoustic scene still presents a difficult challenge, in particular for closely-related classes. As changes to the underlying environment naturally affect both the auditory and the visual information streams, utilising multimodal information is an obvious research direction to solve this problem, and such approaches are recently becoming more popular [3, 4]. This general trend towards multimodal approaches has proven effective in related areas which were traditionally handled as unimodal as well. Speech enhancement, for example, can benefit from multimodal information [5], e. g., recognition of lip movement [6]. Such previous examples demonstrate that multimodal methods have great potential in improving scene understanding when information from all modalities is simultaneously present.

The recently released TAU Urban Audio-Visual Scenes 2021 dataset [7] constitutes an intriguing new benchmark allowing for the development of novel approaches to scene understanding. In this report, we outline our method, which builds on the WaveTransformer (WT) architecture [8] and adds multimodal information using OpenL3 embeddings [9, 10], which were found to be very effective for this task in the challenge baseline [7].

2. ARCHITECTURE

Our architecture is an extension of WT [8], an encoder-decoder based architecture recently introduced for the task of AAC [11]. As the AAC task requires the generation of a sequence of words for each audio segment, we had to adapt the decoder to work for the audio-visual scene classification (AVSC) case, which only requires the prediction of a single label. Additionally, as the input to the WT encoder was originally developed for audio, we extend it to handle multimodal information. In the following subsections, we present the unimodal architecture and its multimodal extension. An overview of WT is presented in Figure 1.

2.1. Unimodal architecture

In the unimodal case, the input to WT is a sequence of T feature vectors with F_{a} features, $\boldsymbol{X}_{\text{a}} \in \mathbb{R}^{T \times F_{\text{a}}}$. WT can be conceptually broken down into the following jointly-learnt processes:

- a dual-branch encoder, consisting of:
  - E_{tf}, a time-frequency (TF) based encoder which jointly processes both the feature and the time dimension
  - E_{temp}, a WaveNet-like encoder which operates only on the time dimension
  - E_{merge}, a 2D-CNN which processes the concatenated output of the two branches to produce one final output
- a transformer-based decoder, T_{d}
- temporal average pooling
- C_{cl}, a final output classification layer.

E_{temp} and E_{tf} both accept $\boldsymbol{X}_{\text{a}}$ as input. E_{temp}, following the WaveNet architecture [12], operates over the time dimension and processes the input sequentially. In the present study, we make use of non-causal convolutions, as the task of AVSC does not require us to preserve temporal causality. Our non-causal convolutions consist of N_{temp} wave-blocks. Each wave-block consists of seven 1D convolutional layers denoted by 1D-CNN$^{t}_{n_{\text{temp}}}$, with $t = 1, \ldots, 7$ and $n_{\text{temp}} = 1, \ldots, N_{\text{temp}}$. Each 1D-CNN$^{t}_{n_{\text{temp}}}$ consists of two convolutional residual blocks using the gated activation unit non-linearity of the original WaveNet [12]. The output of the entire 1D-CNN$^{t}_{n_{\text{temp}}}$ is

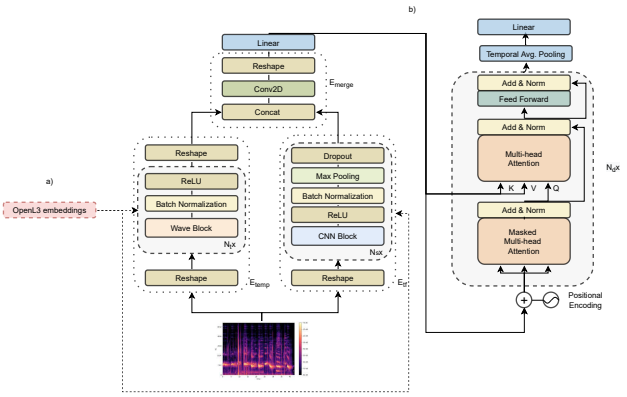


Figure 1: The proposed multimodal approach. OpenL3 embeddings (a) globally condition the encoder of WT (b). The output of the encoder is passed to the WT decoder, then a temporal average pooling layer, and finally to a classification layer. The unimodal architecture is identical without the multimodal conditioning part (a).

passed through batch normalization (BN) and a rectified linear unit (ReLU) activation. The output $\boldsymbol{H}_{n_{\text{temp}}}$ of the n_{temp}-th convolution block is computed according to

$$\boldsymbol{A}_{n_{\text{temp}}} = \text{1D-CNN}^1_{n_{\text{temp}}}(\boldsymbol{H}_{n_{\text{temp}}-1}), \tag{1}$$

$$\boldsymbol{B}_{n_{\text{temp}}} = \tanh(\text{1D-CNN}^2_{n_{\text{temp}}}(\boldsymbol{A}_{n_{\text{temp}}})) \odot \sigma(\text{1D-CNN}^3_{n_{\text{temp}}}(\boldsymbol{A}_{n_{\text{temp}}})), \tag{2}$$

$$\boldsymbol{C}_{n_{\text{temp}}} = \text{1D-CNN}^4_{n_{\text{temp}}}(\boldsymbol{B}_{n_{\text{temp}}}) + \boldsymbol{A}_{n_{\text{temp}}}, \tag{3}$$

$$\boldsymbol{D}_{n_{\text{temp}}} = \tanh(\text{1D-CNN}^5_{n_{\text{temp}}}(\boldsymbol{C}_{n_{\text{temp}}})) \odot \sigma(\text{1D-CNN}^6_{n_{\text{temp}}}(\boldsymbol{C}_{n_{\text{temp}}})), \tag{4}$$

$$\boldsymbol{E}_{n_{\text{temp}}} = \text{1D-CNN}^7_{n_{\text{temp}}}(\boldsymbol{D}_{n_{\text{temp}}}) + \boldsymbol{C}_{n_{\text{temp}}}, \text{ and} \tag{5}$$

$$\boldsymbol{H}_{n_{\text{temp}}} = \text{ReLU}(\text{BN}_{n_{\text{temp}}}(\boldsymbol{E}_{n_{\text{temp}}})) + \boldsymbol{H}_{n_{\text{temp}}-1}, \tag{6}$$

where σ is the sigmoid non-linearity, $\odot$ indicates the Hadamard product, $\boldsymbol{H}_{n_{\text{temp}}} \in \mathbb{R}^{T'_{n_{\text{temp}}} \times M_{n_{\text{temp}}}}$, and $M_{n_{\text{temp}}}$ are the amount of output channels of the $\text{1D-CNN}^7_{n_{\text{temp}}}$.

E_{tf} is made of N_{tf} 2D convolution blocks based on the depth wise separable variant of convolutional neural networks (CNNs) [13] which was introduced for the task of sound event detection (SED) in [14]. Each convolution block in E_{tf} consists of a cross-channel CNN, $\text{S-CNN}_{n_{\text{tf}}}$, a leaky ReLU (LU) activation, and a pointwise convolution CNN, $\text{P-CNN}_{n_{\text{tf}}}$, followed by a ReLU, a BN, max-pooling process ($\text{MP}_{n_{\text{tf}}}$) operating over the feature dimension, and dropout ($\text{DR}_{n_{\text{tf}}}$) with a probability of 0.2. $\text{S-CNN}_{n_{\text{tf}}}$ has a kernel size of 5×5, with unit stride and a zero padding of 2 frames, and its goal is to learn TF patterns from each channel separately. $\text{P-CNN}_{n_{\text{tf}}}$ operates on all channels of its input and learns to combine their information, using a 3×3 kernel with unit stride and a zero padding of 2. The output of E_{tf} is computed as

$$\boldsymbol{A}_{n_{\text{tf}}} = \text{P-CNN}_{n_{\text{tf}}}(\text{BN}_{n_{\text{tf}}}(\text{LU}(\text{S-CNN}(\boldsymbol{H}_{n_{\text{tf}}-1})))) \text{ and} \tag{7}$$

$$\boldsymbol{H}_{n_{\text{tf}}} = \text{DR}_{n_{\text{tf}}}(\text{MP}_{n_{\text{tf}}}(\text{BN}_{n_{\text{tf}}}(\text{ReLU}(\boldsymbol{A}_{n_{\text{tf}}})))), \tag{8}$$

where $\boldsymbol{H}_{n_{\text{tf}}} \in \mathbb{R}^{M_{n_{\text{tf}}} \times T'_{n_{\text{tf}}} \times F'_{n_{\text{tf}}}}$, where $M_{n_{\text{tf}}}$ are the amount of output channels of the n_{tf}-th 2D CNN block. As E_{temp} and E_{tf} are designed to leave the sequence length of its output equal to its input, it follows that $T'_{N_{\text{temp}}} = T'_{N_{\text{tf}}} = T$.

Subsequently, $\boldsymbol{H}_{N_{\text{temp}}}$ is transformed to $1 \times T'_{N_{\text{temp}}} \times M_{N_{\text{temp}}}$ dimensions, and $\boldsymbol{H}_{N_{\text{tf}}}$ to $M_{N_{\text{tf}}} \times T'_{N_{\text{tf}}} \times F'_{N_{\text{tf}}}$, with $M_{N_{\text{temp}}} = F'_{N_{\text{tf}}}$. Then, $\boldsymbol{H}_{N_{\text{temp}}}$ and $\boldsymbol{H}_{N_{\text{tf}}}$ are concatenated in their first dimension, resulting to $\boldsymbol{H}_{\text{merge}} \in \mathbb{R}^{(M_{N_{\text{tf}}}+1) \times T \times M_{N_{\text{temp}}}}$. $\boldsymbol{H}_{\text{merge}}$ is given as an input to E_{merge}, which consists of a 2D convolution with a single output feature map, followed by a linear layer. The convolution layer has a kernel size of 5×5, unit stride and dilation, and a zero padding of 2, whereas the linear layer does not change the dimensionality of its input. This results in a final sequence of feature vector $\boldsymbol{Z}_{enc} \in \mathbb{R}^{T \times F'_{\text{merge}}}$.

The output of the encoder $\boldsymbol{Z}_{enc}$ is passed to the Transformer decoder [15] shown in Figure 1. The decoder architecture follows the implementation of the original Transformer [15], where a series N_{d} of blocks learns to attend to the encoder output $\boldsymbol{Z}_{enc}$. That is, each block accepts as input the output of the previous block and outputs a key value to attend to $\boldsymbol{Z}_{enc}$. The first block is initialised with $\boldsymbol{Z}_{enc}$ as input. Each block is made of two multi-head self-attention layers with N_{att} attention heads, and a feed-forward layer, all followed by layer normalisation and residual connections. $\boldsymbol{Z}_{enc}$ is thus passed through a series of blocks to produce $\boldsymbol{Z}_{dec} \in \mathbb{R}^{T \times F'}$. The sequence is then averaged over the time dimension, and passed to a final (linear) classification layer which computes the output class probabilities.

Similar to the original architecture [8], the hyperparameters of the unimodal approach are set to $N_{\text{temp}} = 4$, $N_{\text{tf}} = 3$, $N_{\text{d}} = 3$, $N_{\text{att}} = 4$.

2.2. Multimodal encoders

In the multimodal setting, we add additional information to the encoder, as shown in Figure 1. The multimodal features, $\boldsymbol{X}_{\text{m}} \in \mathbb{R}^{1 \times F_{\text{m}}}$ are assumed to be of unit length and dimensionality F_{m}, thus encapsulating aggregated information over the entire sequence length and used to globally condition the main network. The intuition behind the global conditioning of a network operating on short-time features, e. g., spectrograms, is that a network trying to learn both long- and short-term information from a series of such features can benefit from information about the overall context. Different conditioning mechanisms have been explored in several fields in the past, e. g., speech synthesis [12], speech enhancement [16], and style conversion [17]. In a nutshell, all these approaches modulate the outputs of several layers in a deep neural network (DNN) by using a single fixed-length vector to modify all elements of a multidimensional tensor (the output of a given layer) across one dimension.

We use $\boldsymbol{X}_{\text{m}}$ to condition both E_{temp} and E_{tf}. For E_{tf}, we always use the same conditioning mechanism, which has been found to work well for convolutional networks operating on TF representations [16]. This mechanism works by adding an extra bias to each layer of every n_{tf} block:

$$\boldsymbol{V}_{n_{\text{tf}}} = \text{FFN}_{n_{\text{tf}}}(\boldsymbol{X}_{\text{m}}) \text{ and} \tag{9}$$

$$\boldsymbol{H}'_{n_{\text{tf}}} = \boldsymbol{H}_{n_{\text{tf}}} + \boldsymbol{V}_{n_{\text{tf}}}, \tag{10}$$

where $\boldsymbol{X}_{\text{m}}$ is first projected to the appropriate dimension for each block using a linear layer, and then added across all timesteps of $\boldsymbol{H}_{n_{\text{tf}}}$. For E_{temp}, we evaluate three different conditioning mechanisms, thus presenting three variants of the multimodal WT, all differing only in how they condition the wave-blocks of E_{temp}:

Table 1: Log loss and accuracy [%] results on the DCASE2021 Task1b evaluation set.

Scene class	Baseline		WT *System 1*		MWT-FiLM *System 2*		MWT-Bias *System 3*		MWT-Wave *System 4*	
Airport	0.963	66.8%	1.307	35.4%	0.883	71.8%	1.532	57.6%	1.452	55.6%
Bus	0.396	85.9%	0.540	82.3%	0.414	83.7%	0.275	91.3%	0.540	76.9%
Metro	0.541	80.4%	1.125	59.6%	0.670	77.1%	0.379	90.0%	0.677	75.8%
Metro station	0.565	80.8%	2.013	33.1%	0.365	88.6%	0.485	86.3%	1.219	61.6%
Park	0.710	77.2%	0.473	89.0%	0.344	87.9%	0.244	91.2%	0.748	71.6%
Public square	0.732	71.1%	1.934	34.0%	0.629	73.9%	0.593	76.4%	1.093	64.9%
Shopping mall	0.839	72.6%	0.808	74.0%	0.488	83.3%	0.469	81.6%	0.296	89.4%
Street pedestrian	0.877	72.7%	1.219	56.3%	0.786	71.3%	1.045	63.1%	1.113	61.5%
Street traffic	0.296	89.6%	0.644	82.1%	0.450	85.5%	0.775	78.1%	0.359	88.8%
Tram	0.659	73.1%	1.498	40.1%	0.740	68.3%	1.512	38.3%	0.524	77.4%
Average	0.658	77.0%	1.153	59.4%	**0.568**	79.5%	0.704	76.2%	0.796	72.6%

- **MWT-FiLM**, using feature-wise linear modulation (FiLM) conditioning [17]
- **MWT-Bias**, operating similar to FiLM, but only modulating the output bias [16]
- **MWT-Wave**, applying the global conditioning approach described in [12].

For all approaches, $\boldsymbol{X}_{\mathrm{m}}$ is first projected to the appropriate dimension using a linear layer, as

$$\boldsymbol{V}_{n_{\text{temp}}} = \mathrm{FFN}_{n_{\text{temp}}}(\boldsymbol{X}_{\mathrm{m}}), \tag{11}$$

where $\mathrm{FFN}_{n_{\text{temp}}}$ is the above-mentioned linear layer.

MWT-FiLM applies FiLM conditioning to the output of each block, thus linearly modulating the feature maps with a constant scale and bias factor. This is implemented by adding an extra equation to Equations (1) to (5) that modifies their output as follows:

$$\boldsymbol{H}'_{n_{\text{temp}}} = \boldsymbol{W}^{s}_{n_{\text{temp}}} \boldsymbol{H}_{n_{\text{temp}}} + \boldsymbol{W}^{b}_{n_{\text{temp}}}, \tag{12}$$

where $\boldsymbol{W}^{s}_{n_{\text{temp}}}$ and $\boldsymbol{W}^{b}_{n_{\text{temp}}}$ are the scale and bias terms of the FiLM layer, respectively, and are both fixed-sized vectors of dimensionality equal to the number of channels in each wave-block. In practice, they are both created as a single vector using the linear layer described by Equation (11), and then split to two equally-sized ones. The same bias and scaling factors are applied to all time-steps.

MWT-Bias instead modifies Equation (5) by adding a simple bias term projected over all time-steps, before the application of ReLU and BN as

$$\boldsymbol{Q}_{n_{\text{temp}}} = \text{1D-CNN}^{7}_{n_{\text{temp}}}(\boldsymbol{D}_{n_{\text{temp}}}) + \boldsymbol{C}_{n_{\text{temp}}} + \boldsymbol{V}_{n_{\text{temp}}} \text{ and} \tag{13}$$

$$\boldsymbol{H}_{n_{\text{temp}}} = \mathrm{ReLU}(\mathrm{BN}_{n_{\text{temp}}}(\boldsymbol{Q}_{n_{\text{temp}}}). \tag{14}$$

Finally, MWT-Wave follows the approach outlined by Oord *et al.* [12] for conditioning a WaveNet architecture, thus substituting Equation (2) and Equation (4) with their multimodal equivalents as:

$$\begin{aligned}\boldsymbol{B}_{n_{\text{temp}}} = &\tanh(\text{1D-CNN}^{2}_{n_{\text{temp}}}(\boldsymbol{A}_{n_{\text{temp}}}) + \boldsymbol{V}_{n_{\text{temp}}}) \odot \\ &\sigma(\text{1D-CNN}^{3}_{n_{\text{temp}}}(\boldsymbol{A}_{n_{\text{temp}}}) + \boldsymbol{V}_{n_{\text{temp}}}) \text{ and}\end{aligned} \tag{15}$$

$$\begin{aligned}\boldsymbol{D}_{n_{\text{temp}}} = &\tanh(\text{1D-CNN}^{5}_{n_{\text{temp}}}(\boldsymbol{C}_{n_{\text{temp}}}) + \boldsymbol{V}_{n_{\text{temp}}}) \odot \\ &\sigma(\text{1D-CNN}^{6}_{n_{\text{temp}}}(\boldsymbol{C}_{n_{\text{temp}}}) + \boldsymbol{V}_{n_{\text{temp}}}).\end{aligned} \tag{16}$$

3. DATASET

The dataset used in this work is the official DCASE2021 audio-visual scene recognition dataset [7], which contains data for 10 scenes spread across different locations in 10 cities. As validation split, we use the one suggested in the challenge baseline [18]. Results are reported on the official evaluation split of the development dataset.

3.1. Features

All models are trained on 1 s of audio data. During training, the audio files, each of a 10 s duration, are randomly cropped to a fixed duration of 1 s. During evaluation on the development set, the validation files also have a duration of 10 s. As specified by the challenge protocol [18], we evaluate and report results for each 1 s of audio.

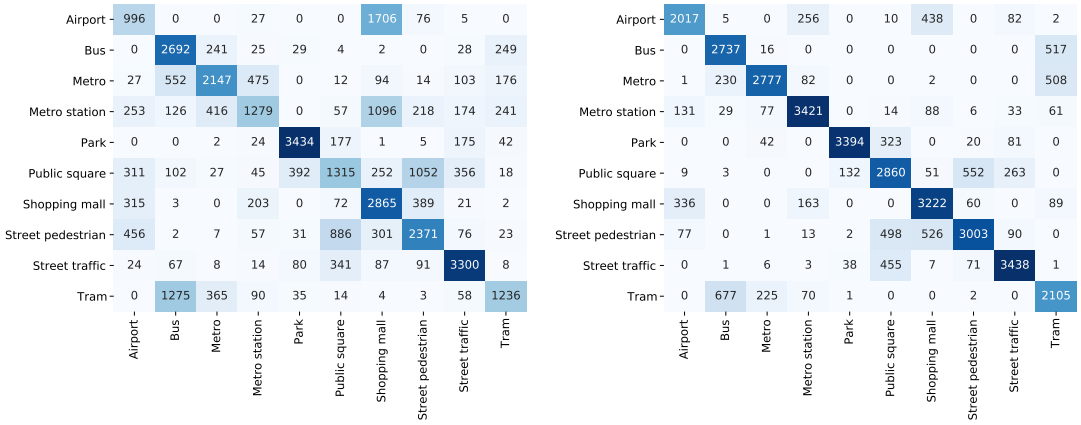
As audio features, we use log-Mel spectrograms extracted with 128 bins, a window size of 32 ms, and a hop size of 10 ms. As the dataset contains stereo recordings, we average the spectral magnitudes of both channels to get the final input to the network. During the training phase, we apply random time- and frequency-bin masking [19]. For each batch, we randomly pick a time mask of size 20 and a frequency mask of size 16 to be applied uniformly over all segments in it.

For multimodal conditioning, we use both audio and video embeddings extracted with OpenL3 [9, 10][1]. Same as the baseline [18], we use the model trained for environmental recognition with 256 mels for audio extraction, and extract embeddings with a hop size of 10 ms. For each 1 s we average the 10 corresponding embeddings to provide one global context vector.

4. EXPERIMENTS

All models are trained for 60 epochs with a categorical cross-entropy loss, and the best model is selected based on the validation set log-loss. We use a batch size of 16 and a learning rate of 0.00005 and weight decay of 0.0001 with the Adam optimiser [20]. Similar to Tran *et al.* [8], we use gradient clipping such that the 2-norm of the gradients does not exceed the value of 1.

[1] https://github.com/marl/openl3



(a) Confusion matrix on the evaluation set of the DCASE2021 challenge dataset for the unimodal WaveTransformer

(b) Confusion matrix on the evaluation set of the DCASE2021 challenge dataset for the multimodal WaveTransformer

Figure 2: Confusion matrices on the evaluation set of the DCASE2021 challenge dataset for the unimodal and the multimodal WaveTransformer with FiLM conditioning.

Table 2: Log loss results on the DCASE2021 Task1b evaluation set when conditioning only on audio (A) or video (V) OpenL3 embeddings compared to using both audio and video (A+V).

Embeddings	MWT-FiLM	MWT-Bias	MWT-Wave
A	0.887	0.961	0.927
V	1.028	0.887	0.829
A+V	0.568	0.704	0.796

Class-wise and average results for our four proposed systems, as well as the challenge baseline [18], are presented in Table 1. We can see that the best performing architecture is MWT-FiLM, which outperforms the baseline with a log-loss of 0.568 and an accuracy of $79.5\,\%$. The other two approaches, MWT-Bias and MWT-Wave, although substantially improving upon the unimodal WT, are performing worse than the baseline. Interestingly, the conditioning mechanism proposed by the original WaveNet authors [12] is showing the worst performance in this setting.

Confusion matrices for the baseline WT and MWT-FiLM are shown in Figure 2. These show that multimodal information has not only improved overall performance, but also helped with misclassifications across similar classes. For example, the baseline architecture often confuses 'airport' with 'shopping mall'; this is corrected through the use of multimodal information, though to the detriment of 'airport' vs 'metro station'. Similarly, the misclassifications between 'bus', 'metro', and 'tram', all belonging to the 'public transport' category as taxonomised in TASK1 of the DCASE2020 challenge [21], have also improved. This demonstrates that multimodal information can help disambiguate classes that are similar in one modality.

Finally, Table 2 shows model performance when using OpenL3 embeddings from only one of the two modalities to condition WaveTransformer. Results are worse for all conditioning mechanisms, indicating that multimodal OpenL3 embeddings contain complimentary information that is necessary to get good performance. This indicates that multimodal pre-training, where a network is trained to jointly model two modalities [9], helps learn generalizable representations for multimodal downstream tasks.

5. CONCLUSION

We have adapted a state-of-the-art DNN architecture to the task of audio-visual scene recognition. Our main contribution lies in the use of multimodal embeddings to condition the two encoding branches, a WaveNet-like and a CNN based one, using three different conditioning mechanisms. Results show that the inclusion of multimodal information is necessary to improve performance and tell apart similar classes. In the future, we intend to explore more advanced ways to closely couple the two modalities.

6. ACKNOWLEDGMENT

Part of the work leading to this publication has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 957337, project MARVEL, as well as the DFG's Reinhart Koselleck project No. 442218748 (AUDI0NOMOUS).

7. References

[1] N. He and J. Zhu, "A weighted partial domain adaptation for acoustic scene classification and its application in fiber optic security system," *IEEE Access*, 2020.

[2] A. Jati, A. Nadarajan, R. Peri, K. Mundnich, T. Feng, B. Girault, and S. Narayanan, "Temporal dynamics of workplace acoustic scenes: Egocentric analysis and prediction," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 756–769, 2021.

[3] S. Parekh, S. Essid, A. Ozerov, N. Q. Duong, P. Pérez, and G. Richard, "Weakly supervised representation learning for audio-visual scene analysis," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 416–428, 2019.

[4] T. Rahman and L. Sigal, "Weakly-supervised audio-visual sound source detection and separation," *arXiv preprint arXiv:2104.02606*, 2021.

[5] D. Michelsanti, Z.-H. Tan, S.-X. Zhang, Y. Xu, M. Yu, D. Yu, and J. Jensen, "An overview of deep-learning-based audio-visual speech enhancement and separation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2021.

[6] I. Addarrazi, H. Satori, and K. Satori, "Lip movement modeling based on dct and hmm for visual speech recognition system," in *Embedded Systems and Artificial Intelligence*, Springer, 2020, pp. 399–407.

[7] S. Wang, A. Mesaros, T. Heittola, and T. Virtanen, "A curated dataset of urban scenes for audio-visual scene analysis," in *2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, accepted, IEEE, 2021. [Online]. Available: https://arxiv.org/abs/2011.00030.

[8] A. Tran, K. Drossos, and T. Virtanen, "Wavetransformer: An architecture for audio captioning based on learning temporal and time-frequency information," in *29th European Signal Processing Conference (EUSIPCO)*, 2021.

[9] R. Arandjelovic and A. Zisserman, "Look, listen and learn," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 609–617.

[10] J. Cramer, H.-H. Wu, J. Salamon, and J. P. Bello, "Look, listen, and learn more: Design choices for deep audio embeddings," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2019, pp. 3852–3856.

[11] K. Drossos, S. Adavanne, and T. Virtanen, "Automated audio captioning with recurrent neural networks," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, New Paltz, New York, U.S.A., Oct. 2017. [Online]. Available: https://arxiv.org/abs/1706.10006.

[12] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," in *9th ISCA Speech Synthesis Workshop*, pp. 125–125.

[13] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[14] K. Drossos, S. I. Mimilakis, S. Gharib, Y. Li, and T. Virtanen, "Sound event detection with depthwise separable and dilated convolutions," in *2020 International Joint Conference on Neural Networks (IJCNN)*, IEEE, 2020, pp. 1–7.

[15] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *arXiv preprint arXiv:1706.03762*, 2017.

[16] G. Keren, J. Han, and B. Schuller, "Scaling speech enhancement in unseen environments with noise embeddings," in *Proc. CHiME 2018 Workshop on Speech Processing in Everyday Environments*, 2018, pp. 25–29.

[17] E. Perez, F. Strub, H. De Vries, V. Dumoulin, and A. Courville, "Film: Visual reasoning with a general conditioning layer," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, 2018.

[18] S. Wang, T. Heittola, A. Mesaros, and T. Virtanen, *Audio-visual scene classification: Analysis of dcase 2021 challenge submissions*, 2021. arXiv: 2105.13675.

[19] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "Specaugment: A simple data augmentation method for automatic speech recognition," *Proc. Interspeech 2019*, pp. 2613–2617, 2019.

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

[21] T. Heittola, A. Mesaros, and T. Virtanen, "Acoustic scene classification in dcase 2020 challenge: Generalization across devices and low complexity solutions," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2020 Workshop (DCASE2020)*, Submitted, 2020. [Online]. Available: https://arxiv.org/abs/2005.14623.