[1]University of North Carolina at Chapel Hill, [2]Adobe Research

Abstract

As two of the five traditional human senses (sight, hearing, taste, smell, and touch), vision and sound are basic sources through which humans understand the world. Often correlated during natural events, these two modalities combine to jointly affect human perception. In this paper, we pose the task of generating sound given visual input. Specifically, we apply learning-based methods to generate raw waveform samples given input video frames. We evaluate our models on a dataset of videos containing a variety of sounds (such as ambient sounds and sounds from people/animals). Our experiments show that the generated sounds are fairly realistic and have good temporal synchronization with the visual inputs.

1. Introduction

The visual and auditory senses are arguably the most important channels through which humans perceive their surrounding environments, and they are often entertwined. From life-long observations of the natural world, people are able to learn the association between vision and sound. For instance, when seeing a flash of lightning in the sky, one might cover their ears subconsciously, knowing that the crack of thunder is coming next. Alternatively, hearing leaves rustling in the wind might conjure up a picture of a peaceful forest scene.

In this paper, we explore whether computational models can learn the relationship between visuals and sound. Models of this relationship could be fundamental for many applications such as combining videos with automatically generated ambient sound to enhance the experience of immersion in virtual reality; adding sound effects to videos automatically to reduce tedious manual sound editing work; Or enabling equal accessibility by associating sound with visual information for people with visual impairments (allowing them to "see" the world through sound). While all of these tasks require powerful high-level inference and reasoning ability, in this work we take a first step toward this goal, narrowing down the task to generating audio for video based on the viewable content. Specifically, we train models to directly predict raw audio signals (waveform samples) from input videos. The models are expected to learn associations between generated sound and visual inputs for various scenes and object interactions. Existing works [9, 2] handle sound generation given input of videos/images under experimental settings (e.g., to generate a hitting sound or where the input videos are recorded indoor with fixed background). In our work, we deal with generating natural sound from videos collected in the wild.

To enable learning, we introduce a dataset that is derived from AudioSet [5]. The dataset includes sounds of humans/animals and other natural sounds spanning 10 categories (Baby crying, Human snoring, Dog, Water flowing, Fireworks, Rail transport, Printers, Drum, Helicopter and Chainsaw).

	Number	Length (sec)
Dog	2785	6.3±3.2
Chainsaw	1824	7.3±3.2
Water flowing	2924	7.0±3.2
Rail transport	3259	7.8±3.0
Fireworks	3115	7.0±3.1
Printer	3673	7.2±3.2
Helicopter	3718	7.8±3.0
Snoring	2144	5.9±3.3
Drum	2606	7.4±3.2
Baby crying	2061	6.0±3.1

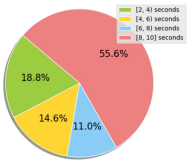

Figure 1. Dataset statistics: the table shows the number of videos with averaged length for each category, while the pie chart presents the distribution of video lengths.

Our model learns a mapping from video frames to audio using a video encoder plus sound generator structure. For sound generation, we use a hierarchical recurrent neural network proposed by [7]. We present 3 variants to encode the visual information, which can be combined with the sound generation network to form a complete framework (Sec. 3). To evaluate the proposed models and the generated results, we conduct both numerical evaluations and human experiments (Sec. 4). Please see our supplementary video to see and hear sound generation results. In addition, recent works [10, 8, 1, 16] learn sound source localization in visual scenes based on the concurrent property of visual and sound. We also apply an attention mechanism on our proposed model to see whether it can learn the localization as a by-product through generation task. However, we observe that the attention maps are not as reasonable as the carefully designed tasks mentioned above.

The innovations introduced by our paper are: 1) We propose a new problem of generating sounds from videos in the wild; 2) We release a dataset containing 28109 cleaned videos (55 hours in total) spanning 10 object categories; 3) We explore model variants for the generation architectures; 4) Numerical and human evaluations are provided as well as an analysis of generated results.

2. Visually Engaged and Grounded AudioSet

As mentioned by Sec. 1, we collect a video (with sound) dataset derived from AudioSet. Audioset is a dataset collected for audio event recognition but not ideal for our task because many of videos and audios are loosely related; the target sound might be covered by other sounds (like music); and the dataset contains some mis-classified videos. All of these sources of noise tend to deter the models from learning the correct mapping from video to audio. To alleviate these issues, we clean a subset of the data, including 28,109 videos in total with an average length of 7 seconds, by ver-

ifying the presence of the target objects for both videos and audios respectively (at 2 second intervals) to make them suitable for the generation task. Fig. 1 shows the number of videos and the average length with the standard deviation for each category.

Existing works [8, 1, 4] utilize the subset of Audioset to learn multisensory representation, sound localization or audio source separation. While the goal of this work is to generate realistic sound based on video content and simple object activities. We expect visual and sound are directly related (predicting dog sound when seeing a dog) most of the time. Due to the verified properties of our dataset, we call it the Visually Engaged and Grounded AudioSet (VEGAS)

3. Approaches

In this work, we formulate the task as estimating conditional generation probability:

$$p(y_1, y_2, ..., y_n | x_1, x_2, ..., x_m) \quad (1)$$

where $x_1, ..., x_m$ represent input video frame representations and $y_1, ..., y_n$ are output waveform values. Note that typically $m << n$ because the sampling rate of audio is much higher than that of video.

We adopt an encoder-decoder architecture in model design and experiment with three variants of this type. In general, our models consist of two parts: video encoder and sound generator. In the following sections, we first discuss the sound generator in Sec. 3.1, then we talk about three different variations of encoding visual information and the concrete systems in Sec. 3.2, Sec. 3.3 and Sec. 3.4. For model/parameter details, please refer [17].

3.1. Sound generator

We apply the recently proposed SampleRNN [7] as our sound generator. Specifically, Fig. 2(a) (upper-left corner brown box) shows the simplified overview of the SampleRNN model. This model consists of multiple tiers, the fine tier (bottom layer) is a multilayer perceptron (MLP) which takes the output from the last recurrent tier (upper layer) and the previous k samples to generate a new sample (k = 4 in this work).

3.2. Frame-to-frame method

For the video encoder component, we first propose a straight-forward frame-to-frame encoding method. We represent the video frames as $x_i = V(f_i)$, where f_i is the i^{th} frame and x_i is the corresponding representation. Here, $V(.)$ is the operation to extract the $fc6$ feature of VGG19 network [12] which has been pre-trained on ImageNet [3] and x_i is a 4096-dimensional vector.

In this model, we encode the visual information by uniformly concatenating the frame representation with the nodes (samples) of the coarsest tier RNN of the sound generator as shown in Fig. 2(b) (content in dotted green box). Due to the difference of sampling rates between the two modalities, to maintain the alignment between them, for each x_i, we duplicate it s times. Here $s = ceiling[sr_{audio}/sr_{video}]$, where sr_{audio} is the sampling rate of audio, sr_{video} is that of video.

3.3. Sequence-to-sequence method

Our second model design has a sequence to sequence type of architecture [15]. In this sequence-to-sequence model, the video encoder and sound generator are clearly separated, and connected via a bottleneck representation, which feeds encoded visual information to the sound generator. As Fig. 2(c) (content in the middle red dotted box) shows, we build a recurrent neural network to encode video features. Here the same deep feature ($fc6$ layer of VGG19) is used to represent video frames as in Sec. 3.2. After visual encoding (i.e., deep feature extraction and recurrent processing), we use the last hidden state from the video encoder to initialize the hidden state of the coarsest tier RNN of the sound generator, then sound generation starts. Therefore the sound generation task becomes:

$$p(y_1, ..., y_n | x_1, ..., x_m) = \prod_{i=1}^{n} p(y_i | H, y_1, ..., y_{i-1}) \quad (2)$$

where H represents the last hidden state of the video encoding RNN or equivalently the initial hidden state of the coarsest tier RNN of the sound generator.

3.4. Flow-based method

Our third model further improves the visual representation to better capture the content and motion in input videos. To explicitly capture the motion signal, we add an optical flow-based deep feature to the visual encoder and call this method the flow-based method. The overall architecture of the current method is identical to the sequence-to-sequence model (as Fig. 2(c) shows), which encodes video features x_i recurrently through RNN and decodes with SampleRNN. The only difference is that here $x_i = cat[V(f_i), F(o_i)]$ ($cat[.]$ indicates concatenation operation); o_i is the optical flow of i^{th} frame; and F(.) is the function to extract the optical flow-based deep feature. We pre-compute optical flow between video frames using [14] and feed the flows to the temporal ConvNets from [11], which has been pre-trained on optical flows of UCF-101 video activity dataset [13], to get the deep feature. We extract the $fc6$ layer of temporal ConvNets, a 4096-dimensional vector.

4. Experiments

In this section, we first introduce training details (Sec. 4.1). Then, we visualize the generated audio to qualitatively evaluate the results (Sec. 4.2). We report the loss

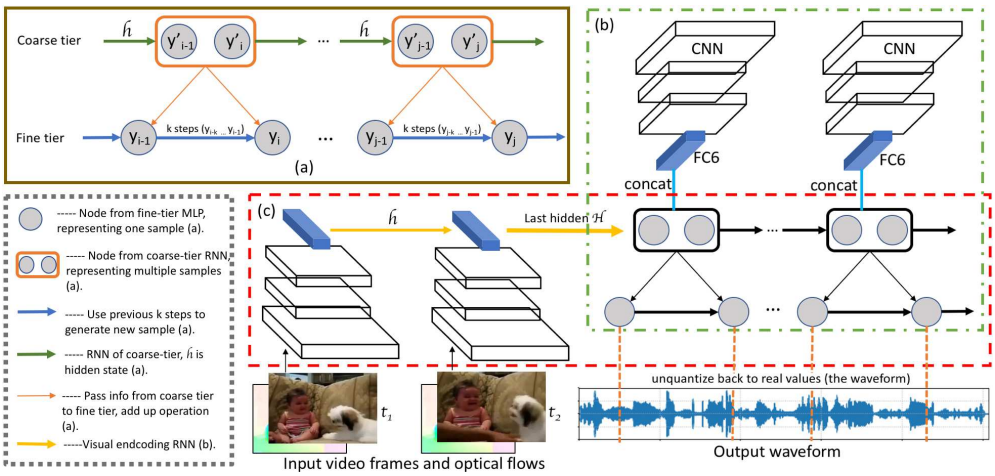

Figure 2. (a) (brown box) shows the simplified architecture of the sound generator, where the fine tier MLP takes as input k previously generated samples and output from the coarse tier to guide generation of new samples. (b) (green dotted box) presents the frame-to-frame structure, where we concatenate the visual representation (the blue FC6 cuboid) with the nodes from the coarsest tier. And (c) (red dotted box) shows the model architecture for sequence-to-sequence and flow-based methods, we recurrently embed visual representations and use the last encoding hidden state (the bold yellow arrow) to initialize the hidden state of the coarsest tier RNN of the sound generator. The MLP tier of the sound generator does 256-way classification to output integers within $[0, 255]$, which are linearly mapped to raw waveforms $[-1, 1]$. The legends in the bottom-left gray dotted box summarize the meaning of the visualization units and the letters in the end ((a)/(b)/(c)) point to the part where the unit can be found.

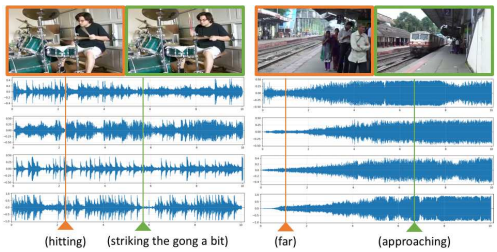

Figure 3. Waveforms of generated audio aligned with corresponding video key frames. For each case the 4 waveforms (from top to bottom) are from $Frame$, Seq, $Flow$ methods, and the original audio. The border color of the frames indicates which flagged position is shown and descriptions indicate what is happening in the video at that moment.

values and also run a human evaluation experiments to subjectively evaluate the results (Sec. 4.3). Finally, we discuss some additional experiments (Sec. 4.4).

4.1. Training details

We train the 3 proposed models on each of the 10 categories of our dataset independently. All training videos have been padded to the same length (10 secs) by duplicating and concatenating up to the target length. We sample the videos at 15.6 FPS (156 frames for 10 seconds) and sample the audios at approximately 16kHz, specifically 159744 times per 10 seconds. For the frame based method, step size s is set to 1024.

We randomly select 128 videos from each category for testing, leaving the remaining videos for training. No data augmentation has been applied. During training, we apply Adam Stochastic Optimization [6] with learning rate 0.001 and minibatch of size 128 for all models.

4.2. Qualitative visualization

We visualize the generated waveform results from the three proposed models as well as the original audio and corresponding video frames in Fig. 3. Results from two categories are shown from left to right: Drum, and Rail transport. We show more results in the supplementary video.

4.3. Numerical and human evaluation

In this section, we provide quantitative and human evaluations of the models.

Loss values: First we show the average cross-entropy loss for training and testing of $Frame$, Seq and $Flow$ models in Table. 1. We can see that $Flow$ and Seq methods achieve lower training and testing loss than $Frame$ method, and they are competitive. Specifically Seq method has the lowest training loss after converging, while $Flow$ works best on testing loss.

Real or fake determination: In this task, we would like to see whether the generated audios can fool people into thinking that they are real. We provide instructions to the Amazon mechanical turkers that the audio of the current video might be either real (originally belonging to this video) or fake (synthesis by computers). The criteria of being fake can be bad synchronization or poor quality such as contain-

	Frame	Seq	Flow
Training	2.6143	**2.5991**	2.6037
Testing	2.7061	2.6866	**2.6839**

Table 1. Average cross-entropy loss for training and testing of 3 methods. Frame represents frame-to-frame method; Seq means sequence-to-sequence method and Flow is flow based method.

	Frame	Seq	Flow	Real
Dog	61.46%	**64.32%**	62.24%	89.06%
Chainsaw	71.09%	73.96%	**76.56%**	93.75%
Water flowing	70.83%	77.60%	**81.25%**	87.50%
Rail transport	79.69%	**83.33%**	80.47%	90.36%
Fireworks	76.04%	76.82%	**78.39%**	94.01%
Printer	**73.96%**	73.44%	71.35%	89.32%
Helicopter	71.61%	74.48%	**78.13%**	91.67%
Snoring	67.71%	**73.44%**	73.18%	90.63%
Drum	62.24%	64.58%	**70.83%**	93.23%
Baby crying	57.29%	**64.32%**	61.20%	94.79%
Average	68.69%	72.63%	**73.36%**	91.43%

Table 2. Human evaluation results: real or fake task where people judge whether a video-audio pair is real or generated. Percentages indicate the frequency of a pair being judged as real.

ing unpleasing noise. In addition to the generated results from our proposed methods, we also include videos with the original audio as a control. Each evaluation is performed by 3 turkers and we aggregate the votes.

The percentages for the audios being rated as real are shown in Table. 2 for all methods. Seq and $Flow$ methods outperform the $Frame$ method except for the printer category. One of the reasons that turkers consider some of the real cases as fake is that a few original audios might include light background music or other noise which appears not fitting with the visual content.

4.4. Additional experiments

Multi-category results: We train a multi-category model where we combine data from all categories and test the model on the VEGAS dataset by conducting the real/fake experiment in Sec 4.3 and find on average 46.29% of the generated sound can fool human (versus 73.63% of the best single-category model).

Comparison with [9]: [9] presents a CNN stacked with RNN structure to predict sound features (cochleagrams) at each time step, and audio samples are reconstructed by example-based retrieval. We implement an upper bound version by assuming the cochleagrams of ground truth sound are given for test videos. And we retrieve the sound from training data with the stride of 2s. This provides a baseline stronger than the method in [9]. We do not observe noticeable artifacts on the boundary of retrieved sound segments, but the synthesized audio does not synchronized very well with the visual content. We also conduct the same real/fake evaluation on the Dog and Drum categories, and the generated sound with this upper bound can fool 40.16% and 43.75% of human subjects respectively, which are largely outperformed by our results (64.32% and 70.83%).

On the other hand, we also test our model on the Greatest Hits dataset from [9]. Note that our model has been trained to generate much longer audios (10s) than those in [9] (0.5s). We evaluate the model via a similar psychology study as described in Sec 6.2 of [9]. 41.50% of our generated sounds are favored by humans over real sound, which is competitive with the method in [9] that achieves 40.01%.

References

[1] R. Arandjelovic and A. Zisserman. Objects that sound. *CoRR*, 2017. 1, 2

[2] L. Chen, S. Srivastava, Z. Duan, and C. Xu. Deep cross-modal audio-visual generation. *CoRR*, 2017. 1

[3] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 2

[4] R. Gao, R. Feris, and K. Grauman. Learning to separate object sounds by watching unlabeled video. *CoRR*, 2018. 2

[5] J. F. Gemmeke, D. P. W. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter. Audio set: An ontology and human-labeled dataset for audio events. In *ICASSP*, 2017. 1

[6] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *ICLR*, 2014. 3

[7] S. Mehri, K. Kumar, I. Gulrajani, R. Kumar, S. Jain, J. Sotelo, A. C. Courville, and Y. Bengio. Samplernn: An unconditional end-to-end neural audio generation model. *ICLR*, 2016. 1, 2

[8] A. Owens and A. A. Efros. Audio-visual scene analysis with self-supervised multisensory features. *CoRR*, 2018. 1, 2

[9] A. Owens, P. Isola, J. McDermott, A. Torralba, E. Adelson, and W. Freeman. Visually indicated sounds. In *CVPR*, 2016. 1, 4

[10] A. Senocak, T.-H. Oh, J. Kim, M.-H. Yang, and I. S. Kweon. Learning to localize sound source in visual scenes. In *CVPR*, 2018. 1

[11] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *NIPS*. 2014. 2

[12] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *ICLR*, 2015. 2

[13] K. Soomro, A. R. Zamir, M. Shah, K. Soomro, A. R. Zamir, and M. Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *CoRR*, 2012. 2

[14] D. Sun, S. Roth, and M. J. Black. Secrets of optical flow estimation and their principles. In *CVPR*, 2010. 2

[15] I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. NIPS, 2014. 2

[16] H. Zhao, C. Gan, A. Rouditchenko, C. Vondrick, J. McDermott, and A. Torralba. The sound of pixels. *CoRR*, 2018. 1

[17] Y. Zhou, Z. Wang, C. Fang, T. Bui, and T. L. Berg. Visual to sound: Generating natural sound for videos in the wild. *CoRR*, 2017. 2