Realtime Generation of Audible Textures Inspired by a Video Stream

Simone Mellace, Jérôme Guzzi, Alessandro Giusti, Luca M. Gambardella
Dalle Molle Institute for Artificial Intelligence, USI-SUPSI, Lugano (Switzerland)

Abstract

We showcase a model to generate a soundscape from a camera stream in real time. The approach relies on a training video with an associated meaningful audio track; a granular synthesizer generates a novel sound by randomly sampling and mixing audio data from such video, favoring timestamps whose frame is similar to the current camera frame; the semantic similarity between frames is computed by a pretrained neural network. The demo is interactive: a user points a mobile phone to different objects and hears how the generated sound changes.

Introduction

Our goal is to generate an audible texture (i.e., an immersive sound with a temporally uniform character, sometimes defined *soundscape*) using a video-only stream as input (e.g., from a webcam). The resulting audio should not necessarily be realistic or plausible for the given input, but instead it should evoke, in the listener, a similar scenario as the one visible in the input frames. Potential applications include: artistic performances; sonification of museum exhibitions; generation of relaxing soundscapes during travelling; assistance to visually impaired people.

Several deep learning approaches generate sounds from images. They differ for the audio representation used: from cochleagrams to generate sounds of materials hit or scratched with a stick (Owens et al. 2016), to raw waveforms to generate very realistic sounds associated with more general actions (Zhou et al. 2017).

We present a simpler and faster approach that combines deep learning for image analysis with more traditional audio synthesizers; in particular, we integrate two components: 1) a pre-trained deep convolutional neural networks for extracting high-level semantic features from video frames; 2) a granular audio synthesizer to procedurally generate coherent, continuous, non-looping soundscapes.

The approach relies on a dataset of aligned video and audio tracks, which are used as source data; in particular, for each input frame, the granular synthesizer is reprogrammed in order to generate a signal reminiscent of the audio corresponding to the dataset frames that are semantically most similar to the input frame.

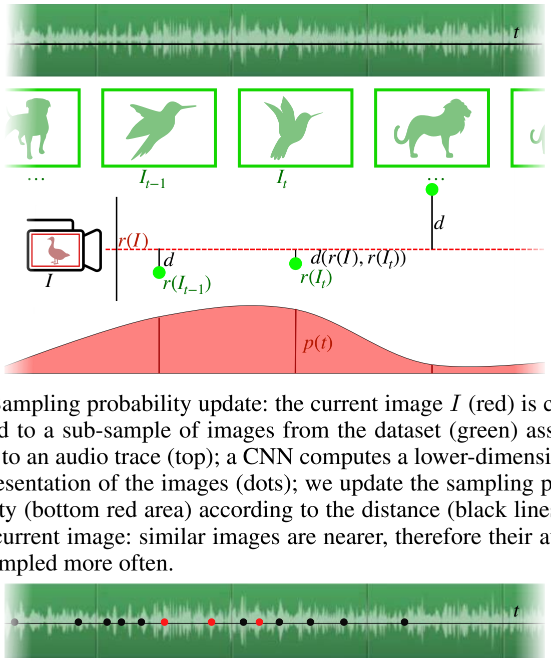
We propose an interactive demo in which the user points a camera to various objects in the environment (people, computers, cellphones, fans) or pictures (from printed postcards or screens) and hears how the generated sound changes in real time. The demo also visualizes the internal state of the system and the audio generation process, and serves as a captivating illustration of the inner workings of the neural network and the granular synthesis techniques.

Granular Synthesis

A granular synthesizer (Roads 1988) produces a continuous soundscape by randomly overlaying many short sound samples (*granules*) with a length between 1 and 100 ms; each granule is too short to be perceived as an individual entity, but long enough to meaningfully contribute to the resulting sound (*microsounds* (Roads 2004)). Granules are sampled at random times from a source audio and played with soft attack and release transients, in order to better blend them in the overall texture. In our demo, we use a granule length in the range between 50 and 100 ms, and we start the playback of new granules at approximately 100 Hz (so that about 5 to 10 granules are overlayed at any given time): at these settings, the character of the resulting sound is pleasant and reminds the character of the audio from which the granules are sampled from, without the source being recognizable. To affect the sound character, we manipulate in real time the probability distribution from which the synthesizer samples the starting points of the granules to be played next.

Model

We consider a pre-built *reference dataset* consisting of a long video track with a corresponding audio track (green track in Figure 1a). The basic assumption is that the audio playing at a given time is somewhat associated to the corresponding frames, which is the case for many YouTube videos, but not for all, which often feature voice-overs or background music unrelated to the video track; for this reason, our reference dataset is composed by a concatenation of several hand-selected YouTube videos. Let I_t be the image frame at time t in the reference videos. We generate audio with a given character by sampling granules from the reference audio track according to a probability distribution $p(t)$ (Figure 1b) updated at runtime as illustrated in Figure 1a.



(a) Sampling probability update: the current image I (red) is compared to a sub-sample of images from the dataset (green) associated to an audio trace (top); a CNN computes a lower-dimensional representation of the images (dots); we update the sampling probability (bottom red area) according to the distance (black lines) to the current image: similar images are nearer, therefore their audio is sampled more often.

(b) Audio update: the granular synthesizer adds new granules (red) sampled randomly, from the dataset audio track, according to the current probability $p(t)$.

Figure 1: Two asynchronous loops update the sampling probability (a) and the ensemble of granules that compose the audio (b).

Sampling Probability Update

1. We acquire a new camera image I.
2. We input I in a CNN that has been pre-trained on ImageNet (Deng et al. 2009) to classify images in 1000 classes; we associate the activation of the second-last layer to a 2048-dimensional representation $r(I)$; such semantic representation captures high-level concepts in I but is not specific to the classes the network has been trained on.
3. We compute the euclidean distance d between $r(I)$ and $r(I_t)$ for a sub-samples of frames (e.g., one frame per second): semantically similar images will be at a lower distance, e.g., the image of a goose will be nearer to the image of another bird than to the image of a lion.
4. We monotonically map $d(r(I), r(I_t))$ to a probability $p(t)$, such that low distances map to higher probabilities.

Demo

Implementation The demo is implemented in Python using open-source libraries and models: a Resnet50 (He et al. 2016) model trained on ImageNet from Keras (Chollet and others 2015); and Pyo, a realtime audio synthesis library (Belanger 2016).

Setup The demo is interactive: the user is free to shoot a video in a scene containing several objects using the video camera of a provided mobile phone (Figure 2); we analyse the images as described above and produce an audio feedback to the user. We also provide a realtime GUI that gives an inside look at how the system works: which frames are nearest to the current images, how does the sampling probability looks like, ...

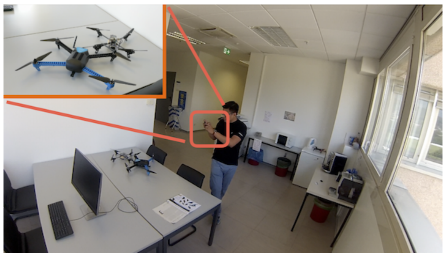


Figure 2: A screen-shot of the video demonstrating the system. The user captures a video with a mobile phone (orange) and listen to the generated soundscape.

Value The demo is captivating as users explore the behavior of a system that is only partially predictable: which objects produce the more realistic sound? what happens when different objects are in the image? how does my selfie sound?, ...

Moreover, the demo provides direct experience and insight into the inner workings of a CNN for image classification, and represents a novel approach to the sonification of the activations of its inner layers.

References

Belanger, O. 2016. Pyo, the python dsp toolbox. In *Proceedings of the ACM Multimedia Conference*, 1214–1217.

Chollet, F., et al. 2015. Keras. https://keras.io.

Deng, J.; Dong, W.; Socher, R.; Li, L.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 248–255.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 770–778.

Owens, A.; Isola, P.; McDermott, J.; Torralba, A.; Adelson, E. H.; and Freeman, W. T. 2016. Visually indicated sounds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2405–2413.

Roads, C. 1988. Introduction to granular synthesis. *Computer Music Journal* 12(2):11–13.

Roads, C. 2004. *Microsound*. The MIT Press.

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