

---

# Soundify: Matching Sound Effects to Video

---

David Chuan-En Lin<sup>1</sup>, Anastasis Germanidis<sup>2</sup>,  
Cristóbal Valenzuela<sup>2</sup>, Yining Shi<sup>2</sup>, Nikolas Martelaro<sup>1</sup>

<sup>1</sup>Carnegie Mellon University, <sup>2</sup>Runway

<sup>1</sup>chuanenl@cs.cmu.edu, nikmart@cmu.edu,

<sup>2</sup>{anastasis, cris, yining}@runwayml.com

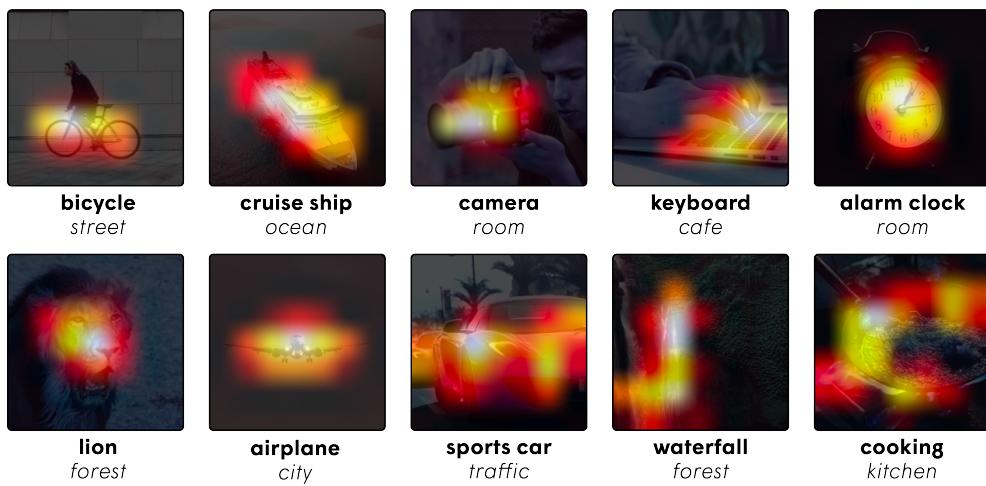


Figure 1: Soundify matches sound effects (**bold**) and ambients (*italics*) by detecting "sound emitters".

## 1 Introduction

In the art of video editing, sound is really half the story. A skilled video editor overlays sounds, such as effects and ambients, over footage to add character to an object or immerse the viewer within a space [1]. However, through formative interviews with 10 professional video editors, we found that this process can be extremely tedious and time-consuming. More specifically, video editors identified three key bottlenecks: (1) finding suitable sounds, (2) precisely aligning sounds to video, and (3) tuning parameters such as pan and gain frame-by-frame. To address these challenges, we introduce Soundify, a system that matches sound effects to video. Prior works have largely explored either learning audio-visual correspondence from large-scale data [3, 9, 8] or performing audio synthesis from scratch [5, 10, 4]. In this work, we take a different approach. By leveraging labeled, studio-quality sound effects libraries [2] and extending CLIP [6], a neural network with impressive zero-shot image classification capabilities, into a "zero-shot detector", we are able to produce high-quality results without resource-intensive correspondence learning or audio generation. We encourage you to have a look at, or better yet, have a *listen* to the results at <https://chuanenlin.com/soundify>.

## 2 Method

The following outlines our method (Figure 3). We implemented Soundify in PyTorch and used Decord, OpenCV, NumPy, and SciPy for image processing and Pydub for audio processing.

**Classify.** We match sound effects to a video by classifying "sound emitters" within (Figure 4). A sound emitter is simply an object or environment that produces sound and is defined based on

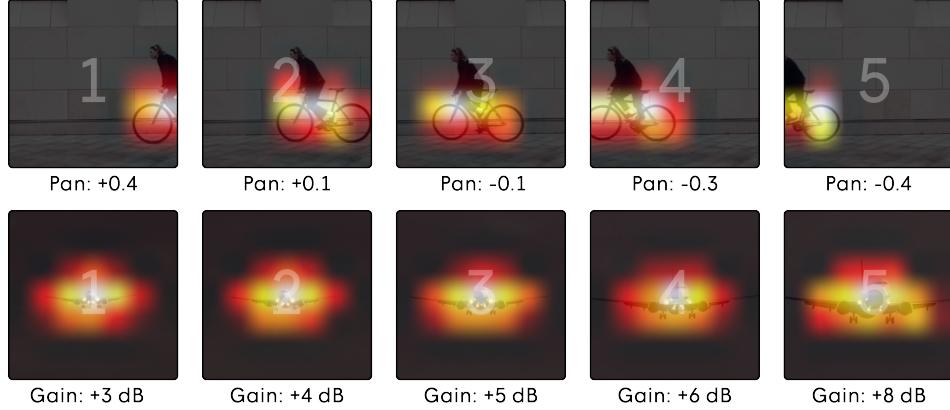


Figure 2: Soundify adapts pan (top row) and gain (bottom row) parameters over time.

Epidemic Sound [2], a database of over 90,000 high-quality sound effects. To reduce the number of distinct sound emitters to classify simultaneously, we first split the video into scenes using a boundary detection algorithm based on absolute color histogram distances between neighboring frames. To construct a realistic soundscape, we then classify each scene for two types of sounds: *effects* (e.g. bicycle, camera, keyboard) and *ambients* (e.g. street, room, cafe). For a given scene, we run each frame through the CLIP image encoder and concatenate them into a representation for the entire scene. For each effects type label in the sound database, we run it through the CLIP text encoder. We then perform pairwise comparisons between the encoded scene and each encoded effects label with cosine similarities and obtain the top-5 matching effects labels for the scene. The user may select one or more recommended effects or, by default, the top-matching effect is assigned. For ambients type labels, we perform the same encoding and pairwise comparison steps. However, ambients classification can be more error-prone due to the background often being visually out of focus or occluded. Thus, we additionally run both the predicted ambients and the previously user-selected effect(s) through CLIP text encoders, and rerank the predicted ambients based on their cosine similarities (Figure 5). For example, `forest` may be ranked higher than `cafe` if the user had previously selected `waterfall`. The user may select one recommended ambient or, by default, the top-matching ambient is assigned.

**Sync.** A sound emitter may appear on screen for only a subset of the scene. Therefore, we want to synchronize effects to when their sound emitter appears (Figure 6). We pinpoint such intervals by comparing the effects label with each frame of the scene and identifying consecutive matches above a threshold. There may be multiple intervals, such as when a sound emitter disappears then reappears.

**Mix.** Video editors adjust sound according to the state of the scene. For instance, as a `bicycle` paddles from one side to another, we hear a shift in stereo panning. As an `airplane` glides up close, we experience a gain in sound intensity. Similarly, we mix an effect’s pan and gain parameters over time (Figure 2). To achieve this, we split an effects interval into around one-second chunks (Figure 6), mix “spatially-aware sound bits” for each chunk (Figure 7), and stitch the chunks smoothly with crossfades. A spatially-aware sound bit uses the first image frame of the chunk as the reference image. We run the reference image through Grad-CAM [7] on the ReLU activation of the last visual layer (ResNet-50) to generate an activation map. This *localizes* the sound emitter, functioning close to a coarse object detector. We then compute the pan parameter by the x-axis of its center of mass and the gain parameter by its normalized area. Next, we retrieve the effect’s corresponding .wav audio file and remix its pan and gain. For ambients, we assume a constant environment for each scene. Thus, we retrieve the corresponding .wav audio file and simply use it across the entire scene. Finally, we merge all audio tracks of effects and ambients for all scenes into one final audio track for the video.

### 3 Conclusion and Future Work

In this paper, we introduce Soundify, a system that automatically matches sound effects to video. Our next step is to evaluate our system through a within-subjects user study with professionals and novices to measure output quality, usability, workload, and satisfaction (Figures 8 and 9 show our user interface). For future work, it may be interesting to also explore ultra-fine synchronizations for certain sounds, such as individual footsteps, to make the matches even more seamless.

## Ethical Implications

The introduction of Soundify into the video editing process may also come with potential ethical implications. One example is bias. Several years ago, Google Photos went under criticism for mislabeling black people as gorillas. Similarly, there may be biases for Soundify in the sound domain that need to be carefully monitored and addressed over time.

## References

- [1] Foley and Ambience Sounds in Film’s Sound Design, 2020. URL <https://iashik.com/foley-and-ambience-sounds-in-films-sound-design>.
- [2] Epidemic Sound: Royalty Free Music and Sound Effects, 2021. URL <https://www.epidemicsound.com/>.
- [3] Relja Arandjelovic and Andrew Zisserman. Look, Listen and Learn. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 609–617, 2017.
- [4] Sanchita Ghose and John J Prevost. AutoFoley: Artificial Synthesis of Synchronized Sound Tracks for Silent Videos with Deep Learning. *IEEE Transactions on Multimedia*, 2020.
- [5] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. WaveNet: A Generative Model for Raw Audio. *arXiv preprint arXiv:1609.03499*, 2016.
- [6] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning Transferable Visual Models from Natural Language Supervision. *arXiv preprint arXiv:2103.00020*, 2021.
- [7] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 618–626, 2017.
- [8] Yapeng Tian, Jing Shi, Bochen Li, Zhiyao Duan, and Chenliang Xu. Audio-Visual Event Localization in Unconstrained Videos. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 247–263, 2018.
- [9] Hang Zhao, Chuang Gan, Andrew Rouditchenko, Carl Vondrick, Josh McDermott, and Antonio Torralba. The Sound of Pixels. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 570–586, 2018.
- [10] Yipin Zhou, Zhaowen Wang, Chen Fang, Trung Bui, and Tamara L Berg. Visual to Sound: Generating Natural Sound for Videos in the Wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3550–3558, 2018.

## A System Diagrams

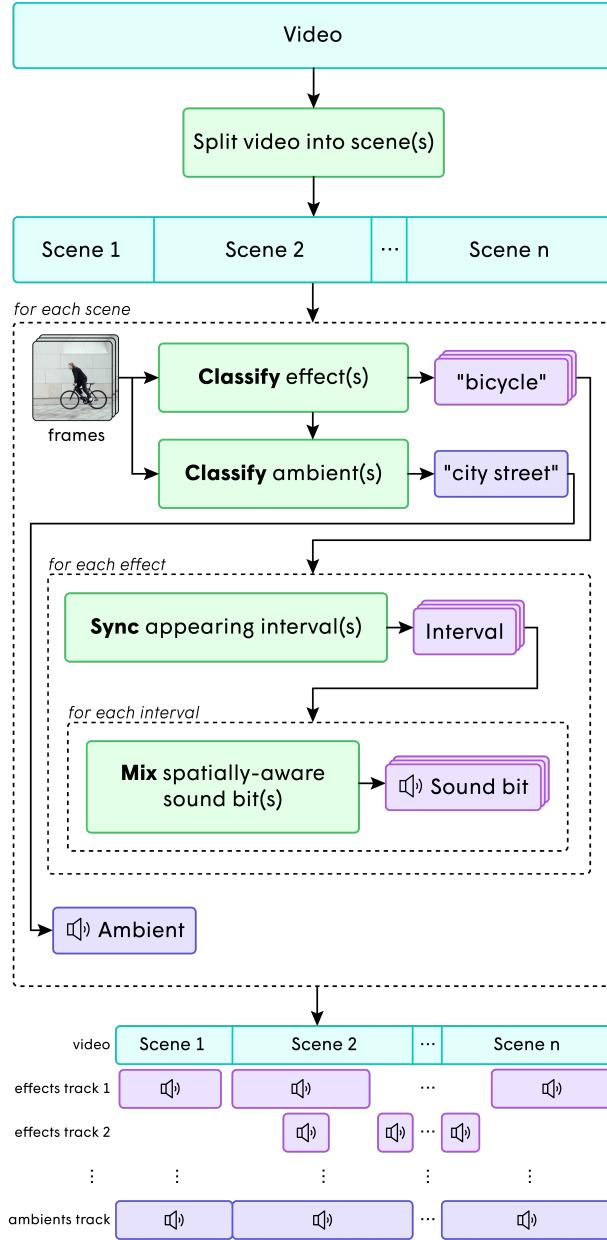


Figure 3: Overview of Soundify. Soundify first splits a video into scenes. For each scene, Soundify classifies for effects and ambients. The matched ambient is used for the entire scene. For each matched effect, Soundify performs more fine-grained synchronization by identifying their appearing intervals. For each interval, Soundify mixes spatially-aware sound bits with computed pan and gain parameters. The final result consists of one or more effects tracks and an ambient track.

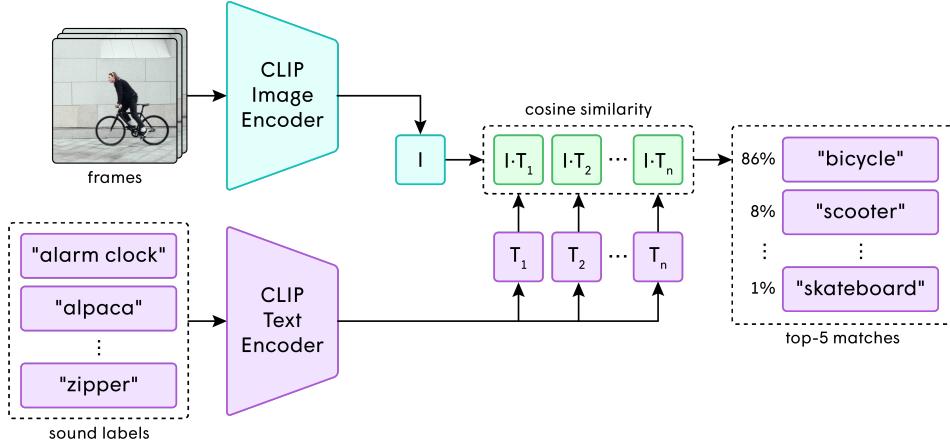


Figure 4: The Classify step of Soundify. Given the frames of a scene and a database of sound labels, Soundify performs pairwise comparisons to predict the top-5 matching sounds.

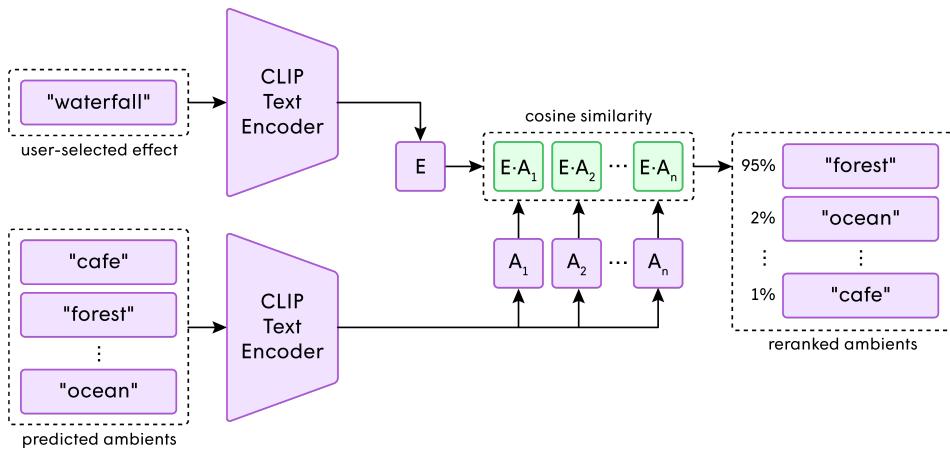


Figure 5: Since ambients classification can be more error-prone, given the user-select effects label and predicted ambients labels, Soundify performs pairwise comparisons to rerank the ambients.

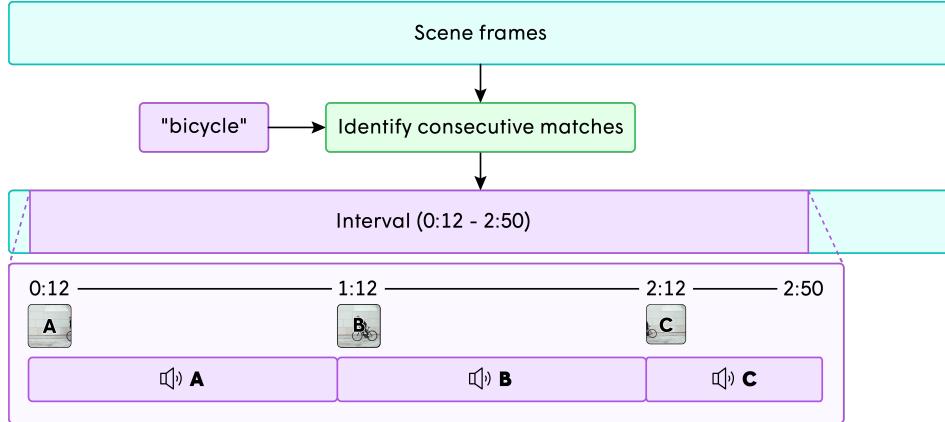


Figure 6: The Sync step of Soundify. Given the frames of a scene and a sound label, Soundify identifies appearing intervals. An interval is split into chunks. Each chunk takes the first frame as its reference frame.

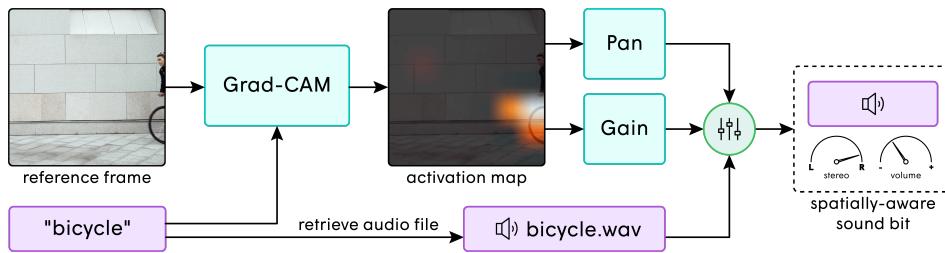


Figure 7: The Mix step of Soundify. Given a reference frame and a sound label, Soundify retrieves the relevant audio file and mixes its pan and gain parameters, by referencing the activation map, to generate a spatially-aware sound bit.

## B User Interface

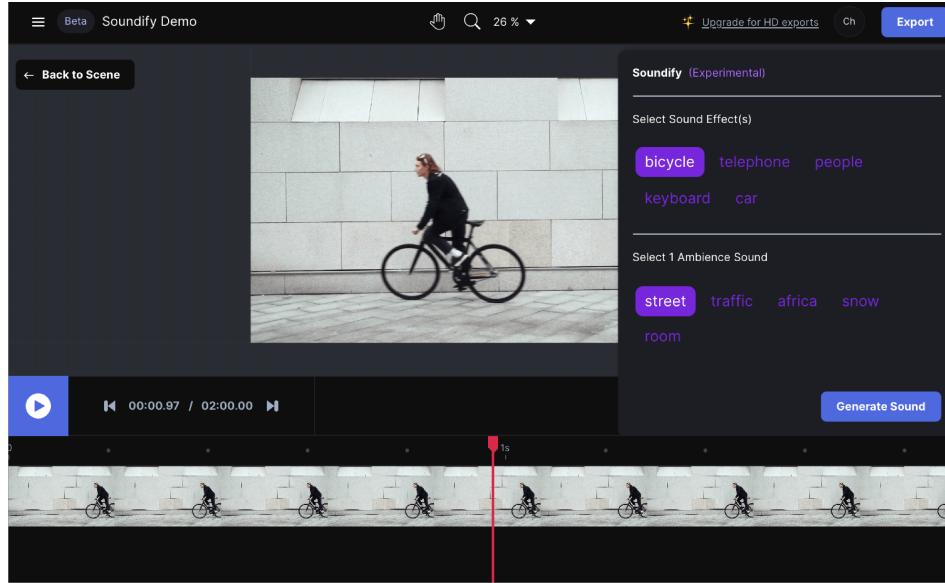


Figure 8: The Soundify interface in [Sequel](#), an ML-powered, web-based video editor developed by Runway. For each scene, Soundify recommends matching effects and ambients. The user may then select one or more effects and one ambient. By default, the top-matching effect and the top-matching ambient are selected.

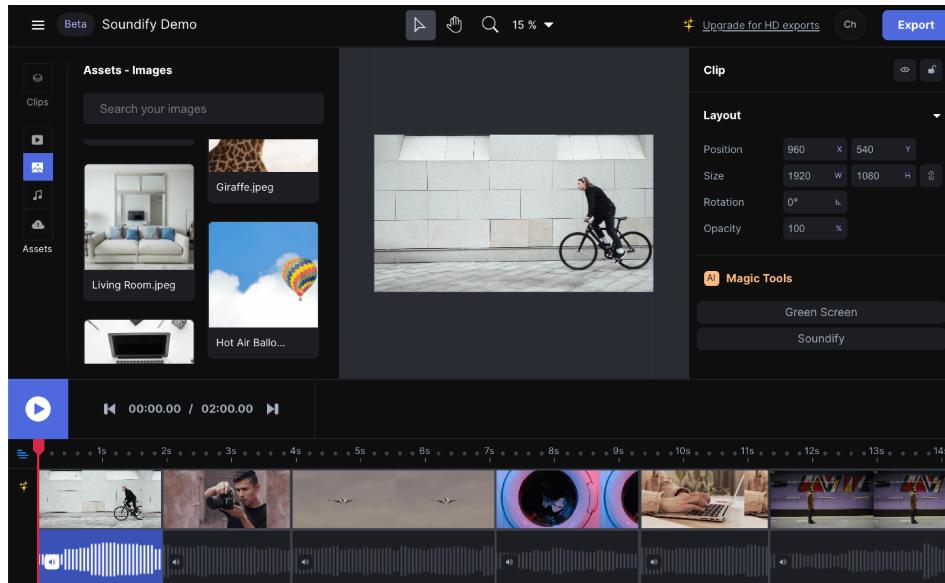


Figure 9: The main interface of Sequel, showing results generated with Soundify. From the bottom timeline, we see that the original video is split into scenes and populated with audio tracks matched with Soundify.