

# Visually Indicated Sounds

Andrew Owens<sup>1</sup>  
Antonio Torralba<sup>1</sup>

<sup>1</sup>MIT

Phillip Isola<sup>2,1</sup>  
Edward H. Adelson<sup>1</sup>

<sup>2</sup>U.C. Berkeley

Josh McDermott<sup>1</sup>  
William T. Freeman<sup>1,3</sup>

<sup>3</sup>Google Research

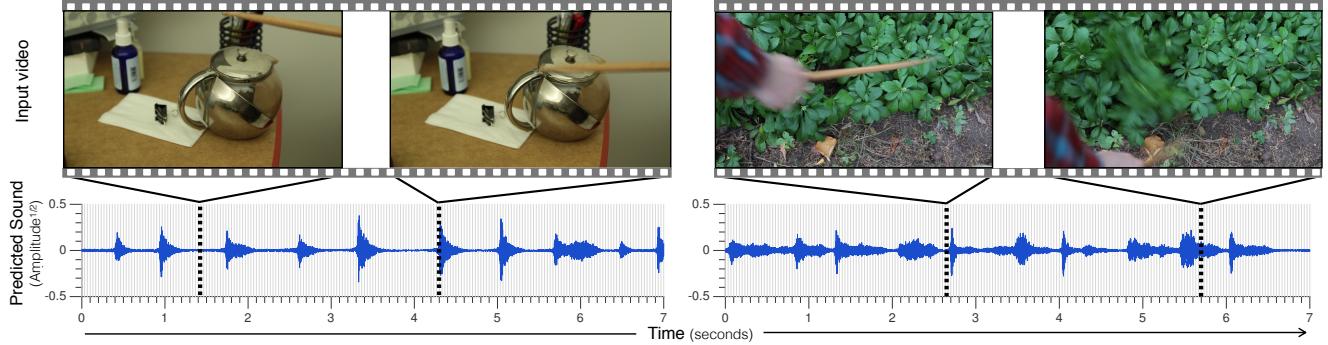


Figure 1: We train a model to synthesize plausible impact sounds from silent videos, a task that requires implicit knowledge of material properties and physical interactions. In each video, someone probes the scene with a drumstick, hitting and scratching different objects. We show frames from two videos and below them the predicted audio tracks. The locations of these sampled frames are indicated by the dotted lines on the audio track. The predicted audio tracks show seven seconds of sound, corresponding to multiple hits in the videos.

## Abstract

*Objects make distinctive sounds when they are hit or scratched. These sounds reveal aspects of an object’s material properties, as well as the actions that produced them. In this paper, we propose the task of predicting what sound an object makes when struck as a way of studying physical interactions within a visual scene. We present an algorithm that synthesizes sound from silent videos of people hitting and scratching objects with a drumstick. This algorithm uses a recurrent neural network to predict sound features from videos and then produces a waveform from these features with an example-based synthesis procedure. We show that the sounds predicted by our model are realistic enough to fool participants in a “real or fake” psychophysical experiment, and that they convey significant information about material properties and physical interactions.*

## 1. Introduction

From the clink of a ceramic mug placed onto a saucer, to the squelch of a shoe pressed into mud, our days are filled with visual experiences accompanied by predictable sounds. On many occasions, these sounds are not just statistically associated with the content of the images – the way, for example, that the sounds of unseen seagulls are associated with a view of a beach – but instead are directly caused

by the physical interaction being depicted: you *see* what is making the sound.

We call these events *visually indicated sounds*, and we propose the task of predicting sound from videos as a way to study physical interactions within a visual scene (Figure 1). To accurately predict a video’s held-out soundtrack, an algorithm has to know about the physical properties of what it is seeing and the actions that are being performed. This task implicitly requires material recognition, but unlike traditional work on this problem [4, 38], we never explicitly tell the algorithm about materials. Instead, it learns about them by identifying statistical regularities in the raw audio-visual signal.

We take inspiration from the way infants explore the physical properties of a scene by poking and prodding at the objects in front of them [36, 3], a process that may help them learn an intuitive theory of physics [3]. Recent work suggests that the sounds objects make in response to these interactions may play a role in this process [39, 43].

We introduce a dataset that mimics this exploration process, containing hundreds of videos of people hitting, scratching, and prodding objects with a drumstick. To synthesize sound from these videos, we present an algorithm that uses a recurrent neural network to map videos to audio features. It then converts these audio features to a waveform, either by matching them to exemplars in a database

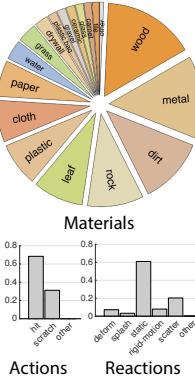


Figure 2: *Greatest Hits: Volume 1* dataset. What do these materials sound like when they are struck? We collected 977 videos in which people explore a scene by hitting and scratching materials with a drumstick, comprising 46,577 total actions. Human annotators labeled the actions with material category labels, the location of impact, an action type label (hit vs. scratch), and a reaction label (shown on right). These labels were used only for analyzing what our sound prediction model learned, not for training it. We show images from a selection of videos from our dataset for a subset of the material categories (here we show examples where it is easy to see the material in question).

and transferring their corresponding sounds, or by parametrically inverting the features. We evaluate the quality of our predicted sounds using a psychophysical study, and we also analyze what our method learned about actions and materials through the task of learning to predict sound.

## 2. Related work

Our work closely relates to research in sound and material perception, and to representation learning.

**Foley** The idea of adding sound effects to silent movies goes back at least to the 1920s, when Jack Foley and collaborators discovered that they could create convincing sound effects by crumpling paper, snapping lettuce, and shaking cellophane in their studio<sup>1</sup>, a method now known as Foley. Our algorithm performs a kind of automatic Foley, synthesizing plausible sound effects without a human in the loop.

**Sound and materials** In the classic mathematical work of [26], Kac showed that the shape of a drum could be partially recovered from the sound it makes. Material properties, such as stiffness and density [37, 31, 14], can likewise be determined from impact sounds. Recent work has used these principles to estimate material properties by measuring tiny vibrations in rods and cloth [8], and similar methods have been used to recover sound from high-speed video of a vibrating membrane [9]. Rather than using a camera as an instrument for measuring vibrations, we infer a plausible sound for an action by recognizing what kind of sound this action would normally make in the visually observed scene.

Impact sounds have been used in other work to recognize objects and materials. Arnab *et al.* [2] recently presented a semantic segmentation model that incorporates audio from impact sounds, and showed that audio information could

help recognize objects and materials that were ambiguous from visual cues alone. Other work recognizes objects using audio produced by robotic interaction [41, 29].

**Sound synthesis** Our technical approach resembles speech synthesis methods that use neural networks to predict sound features from pre-tokenized text features and then generate a waveform from those features [30]. There are also methods, such as the FoleyAutomatic system, for synthesizing impact sounds from physical simulations [45]. Work in psychology has studied low-dimensional representations for impact sounds [7], and recent work in neuroimaging has shown that silent videos of impact events activate the auditory cortex [19].

**Learning visual representations from natural signals** Previous work has explored the idea of learning visual representations by predicting one aspect of a raw sensory signal from another. For example, [11, 22] learned image representations by predicting the spatial relationship between image patches, and [1, 23] by predicting the egocentric motion between video frames. Several methods have also used temporal proximity as a supervisory signal [33, 17, 47, 46]. Unlike in these approaches, we learn to predict one sensory modality (sound) from another (vision). There has also been work that trains neural networks from multiple modalities. For example, [34] learned a joint model of audio and video. However, while they study speech using an autoencoder, we focus on material interaction, and we use a recurrent neural network to predict sound features from video.

A central goal of other methods has been to use a proxy signal (*e.g.*, temporal proximity) to learn a generically useful representation of the world. In our case, we predict a signal – sound – known to be a useful representation for many tasks [14, 37], and we show that the output (*i.e.* the predicted sound itself, rather than some internal representation

<sup>1</sup>To our delight, Foley artists really do knock two coconuts together to fake the sound of horses galloping [6].

in the model) is predictive of material and action classes.

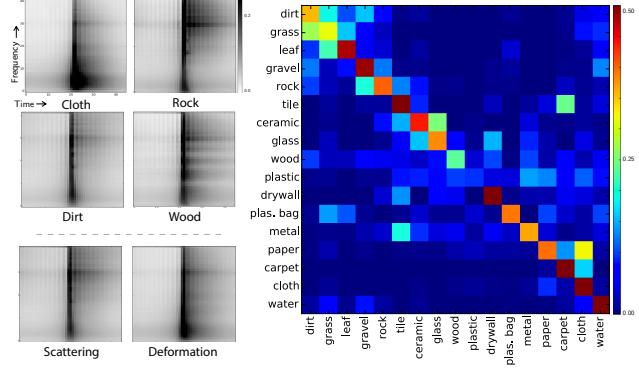
### 3. The *Greatest Hits* dataset

In order to study visually indicated sounds, we collected a dataset containing videos of humans (the authors) probing environments with a drumstick – hitting, scratching, and poking different objects in the scene (Figure 2). We chose to use a drumstick so that we would have a consistent way of generating the sounds. Moreover, since the drumstick does not occlude much of a scene, we can also observe what happens to the object after it is struck. This motion, which we call a *reaction*, can be important for inferring material properties – a soft cushion, for example, will deform more than a firm one, and the sound it produces will vary with it. Similarly, individual pieces of gravel will scatter when they are hit, and their sound varies with this motion (Figure 2, right).

Unlike traditional object- or scene-centric datasets, such as ImageNet [10] or Places [48], where the focus of the image is a full scene, our dataset contains close-up views of a small number of objects. These images reflect the viewpoint of an observer who is focused on the interaction taking place (similar to an egocentric viewpoint). They contain enough detail to see fine-grained texture and the reaction that occurs after the interaction. In some cases, only part of an object is visible, and neither its identity nor other high-level aspects of the scene are easily discernible. Our dataset is also related to robotic manipulation datasets [41, 35, 15]. While one advantage of using a robot is that its actions are highly consistent, having a human collect the data allows us to rapidly (and inexpensively) capture a large number of physical interactions in real-world scenes.

We captured 977 videos from indoor (64%) and outdoor scenes (36%). The outdoor scenes often contain materials that scatter and deform, such as grass and leaves, while the indoor scenes contain a variety of hard and soft materials, such as metal, plastic, cloth, and plastic bags. Each video, on average, contains 48 actions (approximately 69% hits and 31% scratches) and lasts 35 seconds. We recorded sound using a shotgun microphone attached to the top of the camera and used a wind cover for outdoor scenes. We used a separate audio recorder, without auto-gain, and we applied a denoising algorithm [20] to each recording.

**Detecting impact onsets** We detected amplitude peaks in the denoised audio, which largely correspond to the onset of impact sounds. We thresholded the amplitude gradient to find an initial set of peaks, then merged nearby peaks with the mean-shift algorithm [13], treating the amplitude as a density and finding the nearest mode for each peak. Finally, we used non-maximal suppression to ensure that onsets were at least 0.25 seconds apart. This is a simple onset-detection method that most often corresponds to drumstick impacts when the impacts are short and contain a single



(a) Mean cochleograms      (b) Sound confusion matrix

Figure 3: (a) Cochleograms for selected classes. We extracted audio centered on each impact sound in the dataset, computed our subband-envelope representation, and then estimated the mean for each class. (b) Confusion matrix derived by classifying sound features. Rows correspond to confusions made for a single category. The row ordering was determined automatically, by similarity in material confusions (see Section A1.2).

peak<sup>2</sup>. In many of our experiments, we use short video clips that are centered on these amplitude peaks.

**Semantic annotations** We also collected annotations for a sample of impacts (approximately 62%) using online workers from Amazon Mechanical Turk. These include material labels, action labels (hit vs. scratch), reaction labels, and the pixel location of each impact site. To reduce the number of erroneous labels, we manually removed annotations for material categories that we could not find in the scene. During material labeling, workers chose from finer-grained categories. We then merged similar, frequently confused categories (please see Section A2 for details). Note that these annotations are used only for analysis: we train our models on raw audio and video. Examples of several material and action classes are shown in Figure 2.

### 4. Sound representation

Following work in sound synthesis [42, 32], we compute our sound features by decomposing the waveform into subband envelopes – a simple representation obtained by filtering the waveform and applying a nonlinearity. We apply a bank of 40 band-pass filters spaced on an equivalent rectangular bandwidth (ERB) scale [16] (plus a low- and high-pass filter) and take the Hilbert envelope of the responses. We then downsample these envelopes to 90Hz (approximately 3 samples per frame) and compress them. More specifically, we compute envelope  $s_n(t)$  from a wave-

<sup>2</sup>Scratches and hits usually satisfy this assumption, but splash sounds often do not – a problem that could be addressed with more sophisticated onset-detection methods [5].

form  $w(t)$  and a filter  $f_n$  by taking:

$$s_n = D(|(w * f_n) + jH(w * f_n)|)^c, \quad (1)$$

where  $H$  is the Hilbert transform,  $D$  denotes downsampling, and the compression constant  $c = 0.3$ . In Section A1.2, we study how performance varies with the number of frequency channels.

The resulting representation is known as a *cochleagram*. In Figure 3(a), we visualize the mean cochleagram for a selection of material and reaction classes. This reveals, for example, that cloth sounds tend to have more low-frequency energy than those of rock.

How well do impact sounds capture material properties in general? To measure this empirically, we trained a linear SVM to predict material class for the sounds in our database, using the subband envelopes as our feature vectors. We resampled our training set so that each class contained an equal number of impacts (260 per class). The resulting material classifier has 45.8% (chance = 5.9%) class-averaged accuracy (*i.e.*, the mean of per-class accuracy values), and its confusion matrix is shown in Figure 3(b). These results suggest that impact sounds convey significant information about materials, and thus if an algorithm could learn to accurately predict these sounds from images, it would have implicit knowledge of material categories.

## 5. Predicting visually indicated sounds

We formulate our task as a regression problem – one where the goal is to map a sequence of video frames to a sequence of audio features. We solve this problem using a recurrent neural network that takes color and motion information as input and predicts the subband envelopes of an audio waveform. Finally, we generate a waveform from these sound features. Our neural network and synthesis procedure are shown in Figure 4.

### 5.1. Regressing sound features

Given a sequence of input images  $I_1, I_2, \dots, I_N$ , we would like to estimate a corresponding sequence of sound features  $\vec{s}_1, \vec{s}_2, \dots, \vec{s}_T$ , where  $\vec{s}_t \in \mathbb{R}^{42}$ . These sound features correspond to blocks of the cochleagram shown in Figure 4. We solve this regression problem using a recurrent neural network (RNN) that takes image features computed with a convolutional neural network (CNN) as input.

**Image representation** We found it helpful to represent motion information explicitly in our model using a two-stream approach [12, 40]. While two-stream models often use optical flow, it is challenging to obtain accurate flow estimates due to the presence of fast, non-rigid motion. Instead, we compute *spacetime* images for each frame – images whose three channels are grayscale versions of the previous, current, and next frames. This image representation

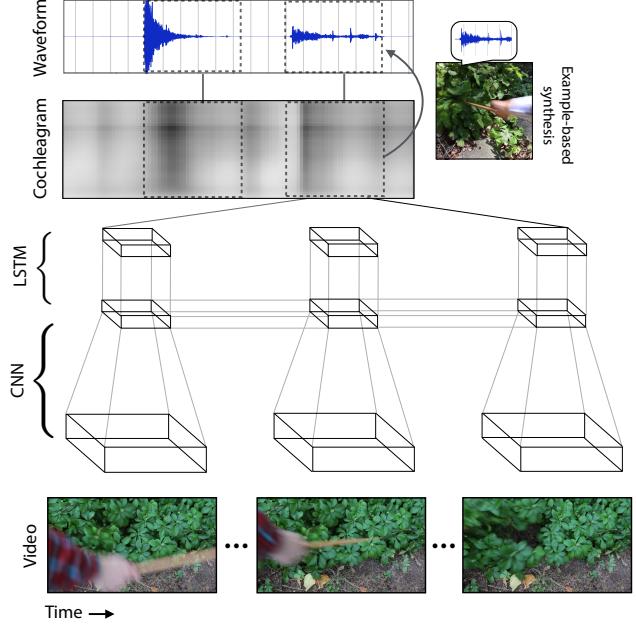


Figure 4: We train a neural network to map video sequences to sound features. These sound features are subsequently converted into a waveform using either parametric or example-based synthesis. We represent the images using a convolutional network, and the time series using a recurrent neural network. We show a subsequence of images corresponding to one impact.

is closely related to 3D video CNNs [24, 27], as derivatives across channels correspond to temporal derivatives.

For each frame  $t$ , we construct an input feature vector  $x_t$  by concatenating CNN features for the spacetime image at frame  $t$  and the color image from the first frame<sup>3</sup>:

$$x_t = [\phi(F_t), \phi(I_1)], \quad (2)$$

where  $\phi$  are CNN features obtained from layer  $fc_7$  of the AlexNet architecture [28] (its penultimate layer), and  $F_t$  is the spacetime image at time  $t$ . In our experiments (Section 6), we either initialized the CNN from scratch and trained it jointly with the RNN, or we initialized the CNN with weights from a network trained for ImageNet classification. When we used pretraining, we precomputed the features from the convolutional layers and fine-tuned only the fully connected layers.

**Sound prediction model** We use a recurrent neural network (RNN) with long short-term memory units (LSTM) [18] that takes CNN features as input. To compensate for the difference between the video and audio sampling rates, we replicate each CNN feature vector  $k$  times, where  $k = \lfloor T/N \rfloor$  (we use  $k = 3$ ). This results in a sequence of CNN features  $x_1, x_2, \dots, x_T$  that is the same length as the sequence of audio features. At each timestep of the RNN, we use the current image feature vector  $x_t$  to update the

<sup>3</sup>We use only the first color image to reduce the computational cost.

vector of hidden variables  $h_t$ <sup>4</sup>. We then compute sound features by an affine transformation of the hidden variables:

$$\begin{aligned}\vec{s}_t &= Wh_t + b \\ h_t &= \mathcal{L}(x_t, h_{t-1}),\end{aligned}\quad (3)$$

where  $\mathcal{L}$  is a function that updates the hidden state [18]. During training, we minimize the difference between the predicted and ground-truth predictions at each timestep:

$$E(\{\vec{s}_t\}) = \sum_{t=1}^T \rho(\|\vec{s}_t - \tilde{\vec{s}}_t\|_2), \quad (4)$$

where  $\tilde{\vec{s}}_t$  and  $\vec{s}_t$  are the true and predicted sound features at time  $t$ , and  $\rho(r) = \log(\epsilon + r^2)$  is a robust loss that bounds the error at each timestep (we use  $\epsilon = 1/25^2$ ). We also increase robustness of the loss by predicting the square root of the subband envelopes, rather than the envelope values themselves. To make the learning problem easier, we use PCA to project the 42-dimensional feature vector at each timestep down to a 10-dimensional space, and we predict this lower-dimensional vector. When we evaluate the network, we invert the PCA transformation to obtain sound features. We train the RNN and CNN jointly using stochastic gradient descent with Caffe [25, 12]. We found it helpful for convergence to remove dropout [44] and to clip large gradients. When training from scratch, we augmented the data by applying cropping and mirroring transformations to the videos. We also use multiple LSTM layers (the number depends on the task; please see Section A1.1).

## 5.2. Generating a waveform

We consider two methods for generating a waveform from the predicted sound features. The first is the simple parametric synthesis approach of [42, 32], which iteratively imposes the subband envelopes on a sample of white noise (we used just one iteration). This method is useful for examining what information is captured by the audio features, since it represents a fairly direct conversion from features to sound. However, for the task of generating plausible sounds to a human ear, we find it more effective to impose a strong natural sound prior during conversion from features to waveform. Therefore, we also consider an example-based synthesis method that snaps a window of sound features to the closest exemplar in the training set. We form a query vector by concatenating the predicted sound features  $\vec{s}_1, \dots, \vec{s}_T$  (or a subsequence of them), searching for its nearest neighbor in the training set as measured by  $L_1$  distance, and transferring the corresponding waveform.

## 6. Experiments

We applied our sound-prediction model to several tasks, and evaluated it with a combination of human studies and automated metrics.

<sup>4</sup>To simplify the presentation, we have omitted the LSTM’s hidden cell state, which is also updated at each timestep.

### 6.1. Sound prediction tasks

In order to study the problem of detection – that is, the task of determining when and whether an action that produces a sound has occurred – separately from the task of sound prediction, we consider two kinds of videos. First, we focus on the *prediction* problem and consider only videos centered on audio amplitude peaks, which often correspond to impact onsets (Section 3). We train our model to predict sound for 15-frame sequences (0.5 sec.) around each peak.

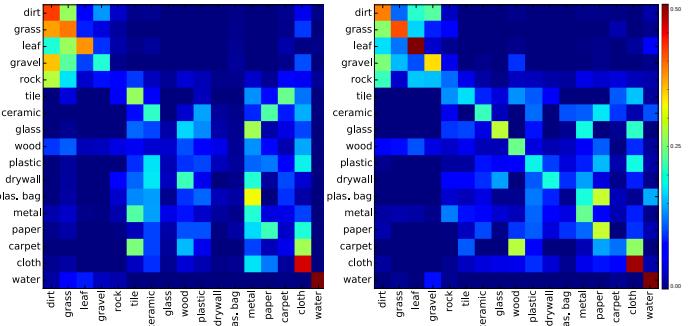
For the second task, which we call the *detection* problem, we train our model on longer sequences (approximately 2 sec. long) sampled from the training videos with a 0.5-second stride, and we subsequently evaluate this model on full-length videos. Since it can be difficult to discern the precise timing of an impact, we allow the predicted features to undergo small shifts before they are compared to the ground truth. We also introduce a two-frame lag in the RNN output, which allows the model to observe future frames before outputting sound features. Finally, before querying sound features, we apply a coloring procedure to account for statistical differences between the predicted and real sound features (*e.g.*, under-prediction of amplitude), using the silent videos in the test set to estimate the empirical mean and covariance of the network’s predictions. For these implementation details, please see Section A1.1. For both tasks, we split the full-length videos into training and test sets (75% training and 25% testing).

**Models** For the prediction task, we compared our model to image-based nearest neighbor search. We computed  $fc_7$  features from a CNN pretrained on ImageNet [28] for the center frame of each sequence, which by construction is the frame where the impact sound occurs. We then searched the training set for the best match and transferred its corresponding sound. We considered variations where the CNN features were computed on an RGB image, on (three-frame) spacetime images, and on the concatenation of both features. To understand the influence of different design decisions, we also considered several variations of our model. We included models with and without ImageNet pretraining; with and without spacetime images; and with example-based versus parametric waveform generation. Finally, we included a model where the RNN connections were broken (the hidden state was set to zero between timesteps).

For the RNN models that do example-based waveform generation (Section 5.2), we used the centered impacts in the training set as the exemplar database. For the prediction task, we performed the query using the sound features for the entire 15-frame sequence. For the detection task, this is not possible, since the videos may contain multiple, overlapping impacts. Instead, we detected amplitude peaks of the parametrically inverted waveform, and matched the sound features in small (8-frame) windows around each

Algorithm	Psychophysical study Labeled <i>real</i>	Loudness		Centroid	
		Err.	<i>r</i>	Err.	<i>r</i>
Full system	<b>40.01% ± 1.66</b>	<b>0.21</b>	<b>0.44</b>	<b>3.85</b>	<b>0.47</b>
- Trained from scratch	36.46% ± 1.68	0.24	0.36	4.73	0.33
- No spacetime	<b>37.88% ± 1.67</b>	0.22	0.37	4.30	0.37
- Parametric synthesis	34.66% ± 1.62	<b>0.21</b>	<b>0.44</b>	<b>3.85</b>	<b>0.47</b>
- No RNN	29.96% ± 1.55	1.24	0.04	7.92	0.28
Image match	32.98% ± 1.59	0.37	0.16	8.39	0.18
Spacetime match	31.92% ± 1.56	0.41	0.14	7.19	0.21
Image + spacetime	33.77% ± 1.58	0.37	0.18	7.74	0.20
Random impact sound	19.77% ± 1.34	0.44	0.00	9.32	0.02

(a) Model evaluation



(b) Predicted sound confusions

(c) CNN feature confusions

Figure 5: (a) We measured the rate at which subjects chose an algorithm’s synthesized sound over the actual sound. Our full system, which was pretrained from ImageNet and used example-based synthesis to generate a waveform, significantly outperformed models based on image matching. For the neural network models, we computed the auditory metrics for the sound features that were predicted by the network, rather than those of the inverted sounds or transferred exemplars. (b) What sounds like what, according to our algorithm? We applied a classifier trained on *real* sounds to the sounds produced by our algorithm, resulting in a confusion matrix (*c.f.* Fig. 3(b), which shows a confusion matrix for real sounds). It obtained 22.7% class-averaged accuracy. (c) Confusions made by a classifier trained on *fc*<sub>7</sub> features (30.2% class-averaged accuracy). For both confusion matrices, we used the variation of our model that was trained from scratch (see Fig. A1(b) for the sound confusions made with pretraining).

peak (starting the window one frame before the peak).

## 6.2. Evaluating the sound predictions

We assessed the quality of the sounds using psychophysical experiments and measurements of acoustic properties.

**Psychophysical study** To test whether the sounds produced by our model varied appropriately with different actions and materials, we conducted a psychophysical study on Amazon Mechanical Turk. We used a two-alternative forced choice test in which participants were asked to distinguish real and fake sounds. We showed them two videos of an impact event – one playing the recorded sound, the other playing a synthesized sound. We then asked them to choose the one that played the real sound. The sound-prediction algorithm was chosen randomly on a per-video basis. We randomly sampled 15 impact-centered sequences from each full-length video, showing each participant at most one impact from each one. At the start of the experiment, we revealed the correct answer to five practice videos.

We measured the rate at which participants mistook our model’s result for the ground-truth sound (Figure 5(a)), finding that our full system – with RGB and spacetime input, RNN connections, ImageNet pretraining, and example-based waveform generation – significantly outperformed the image-matching methods. It also outperformed a baseline that sampled a random (centered) sound from the training set ( $p < 0.001$  with a two-sided *t*-test). We found that the version of our model that was trained from scratch outperformed the best image-matching method ( $p = 0.02$ ). Finally, for this task, we did not find the difference between our full and RGB-only models to be significant ( $p = 0.08$ ).

We show results broken down by semantic category in

Algorithm	Labeled <i>real</i>	Features	Avg. Acc.
Full sys. + mat.	41.82% ± 1.46	Audio-supervised CNN	30.4%
Full sys.	39.64% ± 1.46	ImageNet CNN	42.0%
<i>fc</i> <sub>7</sub> NN + mat.	38.20% ± 1.47	Sound	45.8%
<i>fc</i> <sub>7</sub> NN	32.83% ± 1.41	ImageNet + sound	48.2%
Random + mat.	35.36% ± 1.42	ImageNet crop	52.9%
Random	20.64% ± 1.22	Crop + sound	59.4%
Real sound match	46.90% ± 1.49		

Figure 6: (a) We ran variations of the full system and an image-matching method (RGB + spacetime). For each model, we include an oracle model (labeled with “+ mat”) that draws its sound examples from videos with the same material label. (b) Class-averaged material recognition accuracy obtained by training an SVM with different image and sound features.

Figure 7. For some categories, such as grass and leaf, participants were frequently fooled by our results. Often when a participant was fooled, it was because the sound prediction was simple and prototypical (*e.g.*, a simple thud noise), while the actual sound was complex and atypical. True leaf sounds, for example, are highly varied and may not be fully predictable from a silent video. When they are struck, we hear a combination of the leaves themselves, along with rocks, dirt, and whatever else is underneath them. In contrast, the sounds predicted by our model tend to be closer to prototypical grass/dirt/leaf noises. Participants also sometimes made mistakes when the onset detection failed, or when multiple impacts overlapped, since this may have defied their expectation of hearing a single impact.

We found that the model in which the RNN connections were broken was often unable to detect the timing of the hit, and that it under-predicted the amplitude of the sounds. As a

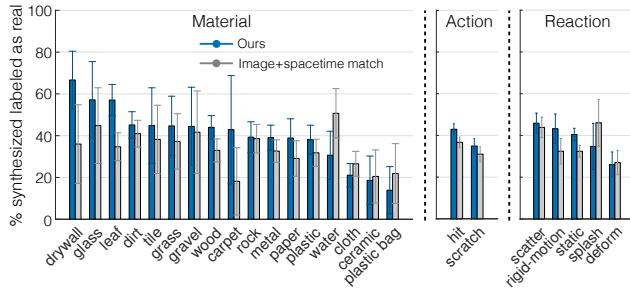


Figure 7: Semantic analysis of psychophysical study. We show the rate that our algorithm fooled human participants for each material, action, and reaction class. The error bars show 95% confidence intervals. Our approach significantly outperforms the highest-performing image-matching method (RGB + spacetime).

result, it performed poorly on automated metrics and failed to find good matches. The performance of our model with parametric waveform generation varied widely between categories. It did well on materials such as *leaf* and *dirt* that are suited to the relatively noisy sounds that the method produces but poorly on hard materials such as *wood* and *metal* (*e.g.*, a confusion rate of  $62\% \pm 6\%$  for *dirt* and  $18\% \pm 5\%$  for *metal*). On the other hand, the example-based approach was not effective at matching textural sounds, such as those produced by splashing water (Fig. 7).

**Auditory metrics** We measured quantitative properties of sounds for the prediction task. We chose metrics that were not sensitive to precise timing. First, we measured the loudness of the sound, which we took to be the maximum energy ( $L_2$  norm) of the compressed subband envelopes over all timesteps. Second, we compared the sounds’ spectral centroids, which we measured by taking the center of mass of the frequency channels for a one-frame (approx. 0.03 sec.) window around the center of the impact. We found that on both metrics, the network was more accurate than the image-matching methods, both in terms of mean squared error and correlation coefficients (Figure 5(a)).

**Oracle results** How helpful is material category information? We conducted a second study that controlled for material-recognition accuracy. Using the subset of the data with material annotations, we created a model that chose a random sound from the same class as the input video. We also created a number of oracle models that used these material labels (Table 6(a)). For the best-performing image-matching model (RGB + spacetime), we restricted the pool of matches to be those with the same label as the input (and similarly for the example-based synthesis method). We also considered a model that matched the ground-truth sound to the training set and transferred the best match. We found that, while knowing the material was helpful for each method, it was not sufficient, as the oracle models did not outperform our model. In particular, our model significantly outperformed the random-sampling oracle ( $p < 10^{-4}$ ).

**Impact detection** We also used our methods to produce sounds for long, uncentered videos, a problem setting that allows us to evaluate their ability to detect impact events. We provide qualitative examples in Figure 8 and on our webpage ([vis.csail.mit.edu](http://vis.csail.mit.edu)). To quantitatively evaluate its detection accuracy, we used the parametric synthesis method to produce a waveform, applied a large gain to that waveform, and then detected amplitude peaks (Section 3). We then compared the timing of these peaks to those of the ground truth, considering an impact to be detected if a predicted spike occurred within 0.1 seconds of it. Using the predicted amplitude as a measure of confidence, we computed average precision. We compared our model to an RGB-only model, finding that the spacetime images significantly improve the result, with APs of 43.6% and 21.6% respectively. Both models were pretrained with ImageNet.

### 6.3. Learning about material and action by predicting sounds

By learning to predict sounds, did the network also learn something about material and physical interactions? To assess this, we tested whether the network’s output sounds were informative about material and action class. We applied the same SVM that was trained to predict material/action class on *real* sound features (Sec. 4) to the sounds predicted by the model. Under this evaluation regime, it is not enough for the network’s sounds to merely be distinguishable by class: they must be close enough to real sounds so as to be classified correctly by an SVM that has never seen a predicted sound. To avoid the influence of pretraining, we used a network that was trained from scratch. We note that this evaluation method is different from that of recent unsupervised learning models [11, 1, 47] that train a classifier on the network’s feature activations, rather than on a ground-truth version of the output.

Using this idea, we classified the material category from predicted sound features. The classifier had class-averaged accuracy of 22.7%, and its confusion matrix is shown in Fig. 5(b). This accuracy indicates that our model learned an output representation that was informative about material, even though it was only trained to predict sound. We applied a similar methodology to classify action categories from predicted sounds, obtaining 68.6% class-averaged accuracy (chance = 50%), and 53.5% for classifying reaction categories (chance = 20%). We found that material and reaction recognition accuracy improved with ImageNet pretraining (to 28.8% and to 55.2%, respectively), but that there was a slight decrease for action classification (to 66.5%).

We also tested whether the predicted sound features convey information about the hardness of a surface. We grouped the material classes into superordinate *hard* and *soft* classes, and trained a classifier on real sound features (sampling 1300 examples per class), finding that it obtained

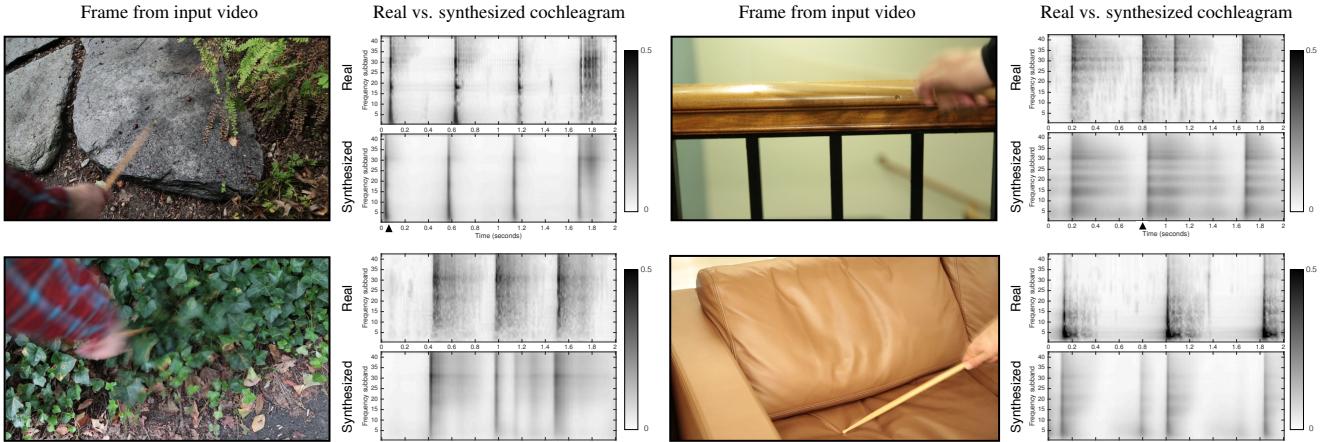


Figure 8: Automatic sound prediction results. We show cochleograms for a representative selection of video sequences, with a sample frame from each sequence on the left. The frame is sampled from the location indicated by the black triangle on the  $x$ -axis of each cochleogram. Notice that the algorithm’s synthesized cochleograms match the general structure of the ground truth cochleograms. Dark lines in the cochleograms indicate hits, which the algorithm often detects. The algorithm captures aspects of both the temporal and spectral structure of sounds. It correctly predicts staccato taps in rock example and longer waveforms for rustling ivy. Furthermore, it tends to predict lower pitched thuds for a soft couch and higher pitched clicks when the drumstick hits a hard wooden railing (although the spectral differences may appear small in these visualizations, we evaluate this with objective metrics in Section 6). A common failure mode is that the algorithm misses a hit (railing example) or hallucinates false hits (cushion example). This frequently happens when the drumstick moves erratically. *Please see our video for qualitative results.*

66.8% class-averaged accuracy (chance = 50%). Here we have defined soft materials to be  $\{\text{leaf, grass, cloth, plastic bag, carpet}\}$  and hard materials to be  $\{\text{gravel, rock, tile, wood, ceramic, plastic, drywall, glass, metal}\}$ .

We also considered the problem of directly predicting material class from visual features. In Table 6(b), we trained a classifier using  $fc_7$  features – both those of the model trained from scratch, and of a model trained on ImageNet [28]. We concatenated color and spacetime image features, since we found that this improved performance. We also considered an oracle model that cropped a high-resolution ( $256 \times 256$ ) patch from the impact location using human annotations, and concatenated its features with those of the full image (we used color images). To avoid occlusions from the arm or drumstick, we cropped the patch from the final frame of the video. We found that performing these crops significantly increased the accuracy, suggesting that localizing the impact is important for classification. We also tried concatenating vision and sound features (similar to [2]), finding that this significantly improved the accuracy.

The kinds of mistakes that the visual classifier (video  $\rightarrow$  material) made were often different from those of the sound classifier (sound  $\rightarrow$  material). For instance, the visual classifier was able to distinguish classes that have a very different appearance, such as *paper* and *cloth*. These classes both make low-pitched sounds (e.g., cardboard and cushions), and were sometimes confused by the sound classifier. On the other hand, the visual classifier was more likely to confuse materials from outdoor scenes, such as rocks and leaves – materials that sound very different but which frequently co-occur in a scene. When we analyzed our model by classifying its sound predictions (video  $\rightarrow$  sound  $\rightarrow$  material), the resulting confusion matrix (Fig. 5(b)) contains both kinds of error: there are *visual analysis* errors when it misidentifies the material that was struck, and *sound synthesis* errors when it produces a sound that was not a convincing replica of the real sound.

7. Discussion

In this work, we proposed the problem of synthesizing visually indicated sounds – a problem that requires an algorithm to learn about material properties and physical interactions. We introduced a dataset for studying this task, which contains videos of a person probing materials in the world with a drumstick, and an algorithm based on recurrent neural networks. We evaluated the quality of our approach with psychophysical experiments and automated metrics, showing that the performance of our algorithm was significantly better than baselines.

We see our work as opening two possible directions for future research. The first is producing realistic sounds from videos, treating sound production as an end in itself. The second direction is to use sound and material interactions as steps toward physical scene understanding.

**Acknowledgments.** This work was supported by NSF grants 6924450 and 6926677, by Shell, and by a Microsoft Ph.D. Fellowship to A.O. We thank Carl Vondrick and Rui Li for the helpful discussions, and the workers at Middlesex Fells, Arnold Arboretum, and Mt. Auburn Cemetery for not asking too many questions while we were collecting the *Greatest Hits* dataset.

## References

- [1] P. Agrawal, J. Carreira, and J. Malik. Learning to see by moving. In *ICCV*, 2015. [2](#), [7](#)
- [2] A. Arnab, M. Sapienza, S. Golodetz, J. Valentin, O. Miksik, S. Izadi, and P. H. S. Torr. Joint object-material category segmentation from audio-visual cues. In *BMVC*, 2015. [2](#), [8](#)
- [3] R. Baillargeon. The acquisition of physical knowledge in infancy: A summary in eight lessons. *Blackwell handbook of childhood cognitive development*, 1:46–83, 2002. [1](#)
- [4] S. Bell, P. Upchurch, N. Snavely, and K. Bala. Material recognition in the wild with the materials in context database. *CoRR*, abs/1412.0623, 2014. [1](#)
- [5] J. P. Bello, L. Daudet, S. Abdallah, C. Duxbury, M. Davies, and M. B. Sandler. A tutorial on onset detection in music signals. *Speech and Audio Processing, IEEE Transactions on*, 13(5):1035–1047, 2005. [3](#)
- [6] T. Bonebright. Were those coconuts or horse hoofs? visual context effects on identification and perceived veracity of everyday sounds. In *International Conference on Auditory Display*, 2012. [2](#)
- [7] S. Cavaco and M. S. Lewicki. Statistical modeling of intrinsic structures in impacts sounds. *The Journal of the Acoustical Society of America*, 121(6):3558–3568, 2007. [2](#)
- [8] A. Davis, K. L. Bouman, M. Rubinstein, F. Durand, and W. T. Freeman. Visual vibrometry: Estimating material properties from small motion in video. In *CVPR*, 2015. [2](#)
- [9] A. Davis, M. Rubinstein, N. Wadhwa, G. J. Mysore, F. Durand, and W. T. Freeman. The visual microphone: passive recovery of sound from video. *ACM Transactions on Graphics (TOG)*, 2014. [2](#)
- [10] 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. [3](#)
- [11] C. Doersch, A. Gupta, and A. A. Efros. Unsupervised visual representation learning by context prediction. *ICCV*, 2015. [2](#), [7](#)
- [12] J. Donahue, L. A. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell. Long-term recurrent convolutional networks for visual recognition and description. *CVPR*, 2015. [4](#), [5](#)
- [13] K. Fukunaga and L. D. Hostetler. The estimation of the gradient of a density function, with applications in pattern recognition. *Information Theory, IEEE Transactions on*, 21(1):32–40, 1975. [3](#)
- [14] W. W. Gaver. What in the world do we hear?: An ecological approach to auditory event perception. *Ecological psychology*, 1993. [2](#)
- [15] M. Gemici and A. Saxena. Learning haptic representation for manipulating deformable food objects. In *IROS*, 2014. [3](#)
- [16] B. R. Glasberg and B. C. Moore. Derivation of auditory filter shapes from notched-noise data. *Hearing research*, 47(1):103–138, 1990. [3](#)
- [17] R. Goroshin, J. Bruna, J. Tompson, D. Eigen, and Y. LeCun. Unsupervised feature learning from temporal data. *arXiv preprint arXiv:1504.02518*, 2015. [2](#)
- [18] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. [4](#), [5](#)
- [19] P.-J. Hsieh, J. T. Colas, and N. Kanwisher. Spatial pattern of bold fmri activation reveals cross-modal information in auditory cortex. *Journal of neurophysiology*, 2012. [2](#)
- [20] Y. Hu and P. C. Loizou. Speech enhancement based on wavelet thresholding the multitaper spectrum. *Speech and Audio Processing, IEEE Transactions on*, 12(1):59–67, 2004. [3](#)
- [21] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015. [10](#)
- [22] P. Isola, D. Zoran, D. Krishnan, and E. H. Adelson. Learning visual groups from co-occurrences in space and time. *arXiv preprint arXiv:1511.06811*, 2015. [2](#)
- [23] D. Jayaraman and K. Grauman. Learning image representations tied to ego-motion. In *ICCV*, December 2015. [2](#)
- [24] S. Ji, W. Xu, M. Yang, and K. Yu. 3d convolutional neural networks for human action recognition. *IEEE TPAMI*, 2013. [4](#)
- [25] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the ACM International Conference on Multimedia*, pages 675–678. ACM, 2014. [5](#)
- [26] M. Kac. Can one hear the shape of a drum? *The american mathematical monthly*, 1966. [2](#)
- [27] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In *CVPR*, 2014. [4](#)
- [28] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012. [4](#), [5](#), [8](#), [10](#)
- [29] E. Krotkov. Robotic perception of material. In *IJCAI*, 1995. [2](#)
- [30] Z.-H. Ling, S.-Y. Kang, H. Zen, A. Senior, M. Schuster, X.-J. Qian, H. M. Meng, and L. Deng. Deep learning for acoustic modeling in parametric speech generation: A systematic review of existing techniques and future trends. *IEEE Signal Processing Magazine*, 2015. [2](#)
- [31] R. A. Lutfi. Human sound source identification. In *Auditory perception of sound sources*, pages 13–42. Springer, 2008. [2](#)
- [32] J. H. McDermott and E. P. Simoncelli. Sound texture perception via statistics of the auditory periphery: evidence from sound synthesis. *Neuron*, 71(5):926–940, 2011. [3](#), [5](#)
- [33] H. Mobahi, R. Collobert, and J. Weston. Deep learning from temporal coherence in video. In *ICML*, 2009. [2](#)
- [34] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng. Multi-modal deep learning. In *ICML*, 2011. [2](#)
- [35] L. Pinto and A. Gupta. Supersizing self-supervision: Learning to grasp from 50k tries and 700 robot hours. *arXiv preprint arXiv:1509.06825*, 2015. [3](#)
- [36] L. Schulz. The origins of inquiry: Inductive inference and exploration in early childhood. *Trends in cognitive sciences*, 16(7):382–389, 2012. [1](#)
- [37] A. A. Shabana. *Theory of vibration: an introduction*. Springer Science & Business Media, 1995. [2](#)
- [38] L. Sharan, C. Liu, R. Rosenholtz, and E. H. Adelson. Recognizing materials using perceptually inspired features. *International journal of computer vision*, 103(3):348–371, 2013. [1](#)
- [39] M. H. Siegel, R. Magid, J. B. Tenenbaum, and L. E. Schulz. Black boxes: Hypothesis testing via indirect perceptual evidence. *Proceedings of the 36th Annual Conference of the Cognitive Science Society*, 2014. [1](#)
- [40] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in Neural Information Processing Systems*, 2014. [4](#)
- [41] J. Sinapov, M. Wiemer, and A. Stoytchev. Interactive learning of the acoustic properties of household objects. In *ICRA*, 2009. [2](#), [3](#)
- [42] M. Slaney. Pattern playback in the 90s. In *NIPS*, pages 827–834, 1994. [3](#), [5](#)
- [43] L. Smith and M. Gasser. The development of embodied cognition: Six lessons from babies. *Artificial life*, 11(1-2):13–29, 2005. [1](#)
- [44] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014. [5](#)
- [45] K. Van Den Doel, P. G. Kry, and D. K. Pai. Foleyautomatic: physically-based sound effects for interactive simulation and animation. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 537–544. ACM, 2001. [2](#)
- [46] C. Vondrick, H. Pirsiavash, and A. Torralba. Anticipating the future by watching unlabeled video. *arXiv preprint arXiv:1504.08023*, 2015. [2](#)
- [47] X. Wang and A. Gupta. Unsupervised learning of visual representations using videos. In *ICCV*, 2015. [2](#), [7](#)

- [48] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva. Learning deep features for scene recognition using places database. In *NIPS*, 2014. 3  
[49] L. Zitnick. 80,000 ms coco images in 5 minutes. In <https://www.youtube.com/watch?v=ZUIEOUoCLBo>. 11

## A1. Model implementation

We provide more details about our model and sound representation.

### A1.1. Detection model

We describe the variation of our model that performs the detection task (Section 6.1) in more detail.

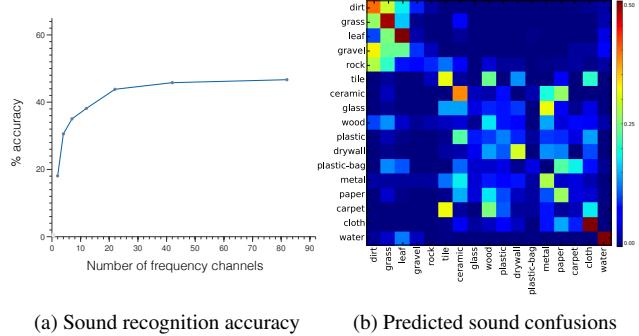
**Timing** We allow the sound features to undergo small time shifts in order to account for misalignments for the detection task. During each iteration of backpropagation, we shift the sequence so as to minimize the loss in Equation 4. We resample the feature predictions to create a new sequence  $\hat{s}_1, \hat{s}_2, \dots, \hat{s}_T$  such that  $\hat{s}_t = \tilde{s}_{t+L_t}$  for some small shift  $L_t$  (we use a maximum shift of 8 samples, approximately 0.09 seconds). During each iteration, we infer this shift by finding the optimal labeling of a Hidden Markov Model:

$$\sum_{t=1}^T w_t \rho(\|\hat{s}_t - \tilde{s}_t\|) + V(L_t, L_{t+1}), \quad (5)$$

where  $V$  is a smoothness term for neighboring shifts. For this, we use a Potts model weighted by  $\frac{1}{2}(\|\tilde{s}_t\| + \|\tilde{s}_{t+1}\|)$  to discourage the model from shifting the sound near high-amplitude regions. We also include a weight variable  $w_t = 1 + \alpha \delta(\tau \leq \|\tilde{s}_t\|)$  to decrease the importance of silent portions of the video (we use  $\alpha = 3$  and  $\tau = 2.2$ ). During each iteration of backpropagation, we align the two sequences, then propagate the gradients of the loss to the shifted sequence.

To give the RNN more temporal context for its predictions, we also delay its predictions, so that at frame  $f$ , it predicts the sound features for frame  $f - 2$ .

**Transforming features for neighbor search** For the detection task, the statistics of the synthesized sound features can differ significantly from those of the ground truth – for example, we found the amplitude of peaks in the predicted waveforms to be smaller than those of real sounds. We correct for these differences during example-based synthesis (Section 5.2) by applying a coloring transformation before the nearest-neighbor search. More specifically, we obtain a whitening transformation for the predicted sound features by running the neural network on the test videos and estimating the empirical mean and covariance at the detected amplitude peaks, discarding peaks whose amplitude is below a threshold. We then estimate a similar transformation for ground-truth amplitude peaks in the training set, and we use these transformations to color (*i.e.* transform the mean



(a) Sound recognition accuracy      (b) Predicted sound confusions

Figure A1: (a) Class-averaged accuracy for recognizing materials, with an SVM trained on real sounds. We varied the number of band-pass filters and adjusted their frequency spacing accordingly (we did not vary the temporal sampling rate). (b) Confusion matrix obtained by classifying the sounds predicted by our pretrained model, using a classifier trained on real sound features (*c.f.* the same model without pretraining in Figure 5(b).)

and covariance of) the predicted features into the space of real features before computing their  $L_1$  nearest neighbors. To avoid the influence of multiple, overlapping impacts on the nearest neighbor search, we use a search window that starts at the beginning of the amplitude spike.

**Evaluating the RNN for long videos** When evaluating our model on long videos, we run the RNN on 10-second subsequences that overlap by 30%, transitioning between consecutive predictions at the time that has the least sum-of-squares difference between the overlapping predictions.

### A1.2. Sound representation

We measured performance on the task of assigning material labels to ground-truth sounds after varying the number of frequency channels in the subband envelope representation. The result is shown in Figure A1. To obtain the ordering of material classes used in visualizations of the confusion matrices (Figure 3), we iteratively chose the material category that was most similar to the previously chosen class. When measuring the similarity between two classes, we computed Euclidean distance between rows of a (soft) confusion matrix – one whose rows correspond to the mean probability assigned by the classifier to each target class (averaged over all test examples).

### A1.3. Network structure

We used AlexNet [28] for our CNN architecture. For the pretrained models, we precomputed the *pool5* features and fine-tuned the model’s two fully-connected layers. For the model that was trained from scratch, we applied batch normalization [21] to each training mini-batch. For the centered videos, we used two LSTM layers with a 256-dimensional hidden state (and three for the detection model). When using multiple LSTM layers, we compen-

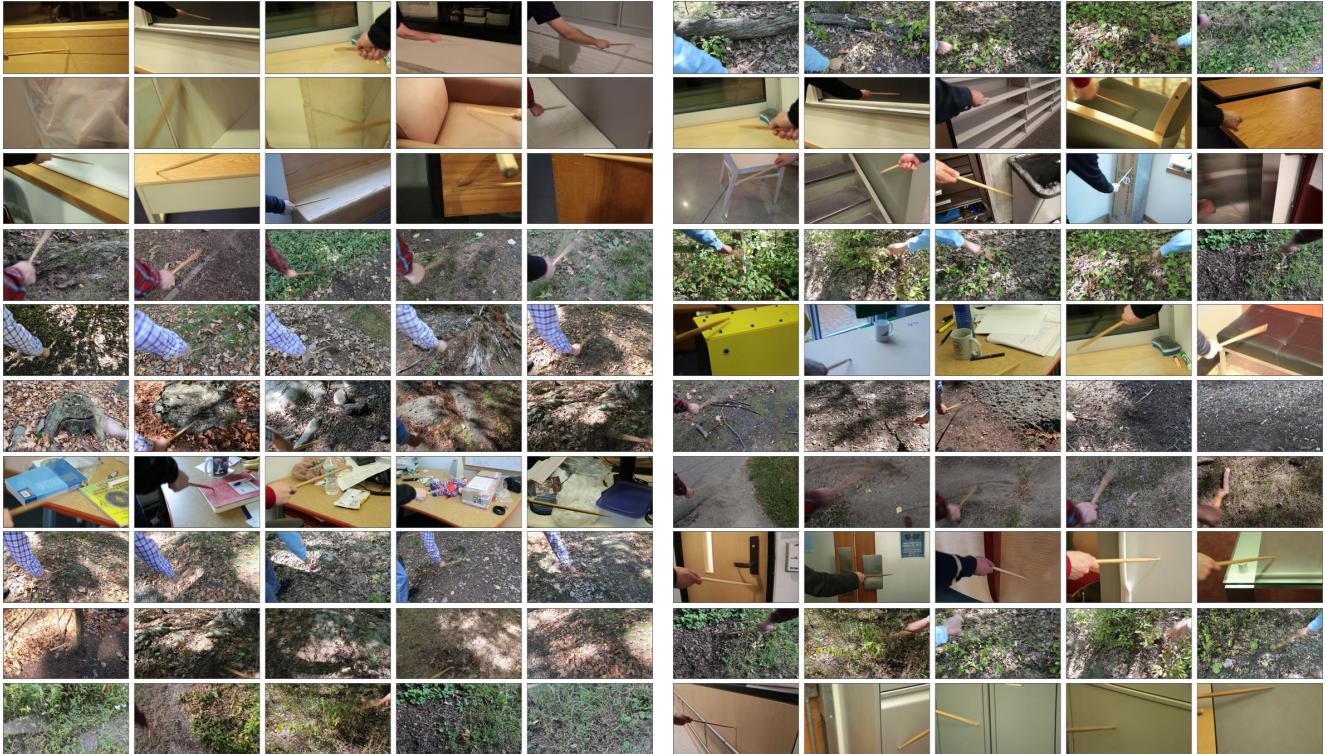


Figure A2: A “walk” through the dataset using AlexNet  $fc_7$  nearest-neighbor matches. Starting from the left, we matched an image with the database and placed its best match to its right. We repeat this 5 times, with 20 random initializations. We used only images taken at a contact point (the middle frames from the “centered” videos). To avoid loops, we removed videos when any of their images were matched. The location of the hit, material, and action often vary during the walk. In some sequences, the arm is the dominant feature that is matched between scenes.

sate for the difference in video and audio sampling rates by upsampling the input to the last LSTM layer (rather than up-sampling the CNN features), replicating each input  $k$  times (where again  $k = 3$ ).

## A2. Dataset details

In Figure A2, we show a “walk” through the dataset using  $fc_7$  features, similar to [49]. Our data was collected using two wooden (hickory) drumsticks, and an SLR camera with a 29.97 Hz framerate. We used a ZOOM H1 external audio recorder, and a Rode VideoMic Pro microphone. Online workers labeled the impacts by visually examining silent videos, without sound. We gave them finer-grained categories than, then merged similar categories that were frequently labeled inconsistently by workers. Specifically, we merged *cardboard* and *paper*; *concrete* and *rock*; *cloth* and *cushion* (often the former physically covers the latter); and *rubber* and *plastic*. To measure overall consistency between workers, we labeled a subset of the impacts with 3 or more workers, finding that their material labels agreed with the majority 87.6% of the time on the fine-grained categories. Common inconsistencies include confusing *dirt*

with *leaf* (confused 5% of the time); *grass* with *dirt* and *leaf* (8% each); and *cloth* with (the fine-grained category) *cushion* (9% of the time).