Ambient Sound Provides Supervision for Visual Learning

Andrew Owens[1], Jiajun Wu[1], Josh H. McDermott[1], William T. Freeman[1,2], and Antonio Torralba[1]

[1]Massachusetts Institute of Technology
[2]Google Research

Abstract. The sound of crashing waves, the roar of fast-moving cars – sound conveys important information about the objects in our surroundings. In this work, we show that ambient sounds can be used as a supervisory signal for learning visual models. To demonstrate this, we train a convolutional neural network to predict a statistical summary of the sound associated with a video frame. We show that, through this process, the network learns a representation that conveys information about objects and scenes. We evaluate this representation on several recognition tasks, finding that its performance is comparable to that of other state-of-the-art unsupervised learning methods. Finally, we show through visualizations that the network learns units that are selective to objects that are often associated with characteristic sounds.

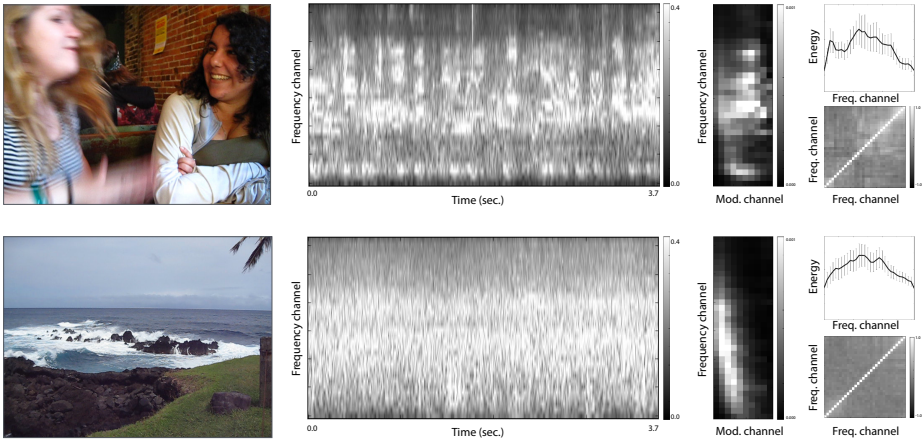
Keywords: Sound, convolutional networks, unsupervised learning.

1 Introduction

Sound conveys important information about the world around us – the bustle of a café tells us that there are many people nearby, while the low-pitched roar of engine noise tells us to watch for fast-moving cars [10]. Although sound is in some cases complementary to visual information, such as when we listen to something out of view, vision and hearing are often informative about the same structures in the world. Here we propose that as a consequence of these correlations, concurrent visual and sound information provide a rich training signal that we can use to learn useful representations of the visual world.

In particular, an algorithm trained to predict the sounds that occur within a visual scene might be expected to learn objects and scene elements that are associated with salient and distinctive noises, such as people, cars, and flowing water. Such an algorithm might also learn to associate visual scenes with the ambient sound textures [25] that occur within them. It might, for example, associate the sound of wind with outdoor scenes, and the buzz of refrigerators with indoor scenes.

Although human annotations are indisputably useful for learning, they are expensive to collect. The correspondence between ambient sounds and video is, by contrast, ubiquitous and free. While there has been much work on learning



(a) Video frame (b) Cochleagram (c) Summary statistics

Fig. 1: Visual scenes are associated with characteristic sounds. Our goal is to take an image (a) and predict time-averaged summary statistics (c) of a cochleagram (b). The statistics we use are (clockwise): the response to a bank of band-pass modulation filters; the mean and standard deviation of each frequency band; and the correlation between bands. We show two frames from the Flickr video dataset [34]. The first contains the sound of human speech; the second contains the sound of wind and crashing waves. The differences between these sounds are reflected in their summary statistics: e.g., the water/wind sound, which is similar to white noise, contains fewer correlations between cochlear channels.

from unlabeled image data [4,35,22], an audio signal may provide information that that is largely orthogonal to that available in images alone – information about semantics, events, and mechanics are all readily available from sound [10].

One challenge in utilizing audio-visual input is that the sounds that we hear are only loosely associated with what we see. Sound-producing objects often lie outside of our visual field, and objects that are capable of producing characteristic sounds – barking dogs, ringing phones – do not always do so. A priori it is thus not obvious what might be achieved by predicting sound from images.

In this work, we show that a model trained to predict held-out sound from video frames learns a visual representation that conveys semantically meaningful information. We formulate our sound-prediction task as a classification problem, in which we train a convolutional neural network (CNN) to predict a statistical summary of the sound that occurred at the time a video frame was recorded. We then validate that the learned representation contains significant information about objects and scenes.

We do this in two ways: first, we show that the image features that we learn through our sound-prediction task can be used for object and scene recognition. On these tasks, our features obtain similar performance to state-of-the-art unsupervised and self-supervised learning methods. Second, we show that the

intermediate layers of our CNN are highly selective for objects. This augments recent work [38] showing that object detectors "emerge" in a CNN's internal representation when it is trained to recognize scenes. As in the scene recognition task, object detectors emerge inside of our sound-prediction network. However, our model learns these detectors from an unlabeled audio-visual signal, without any explicit human annotation.

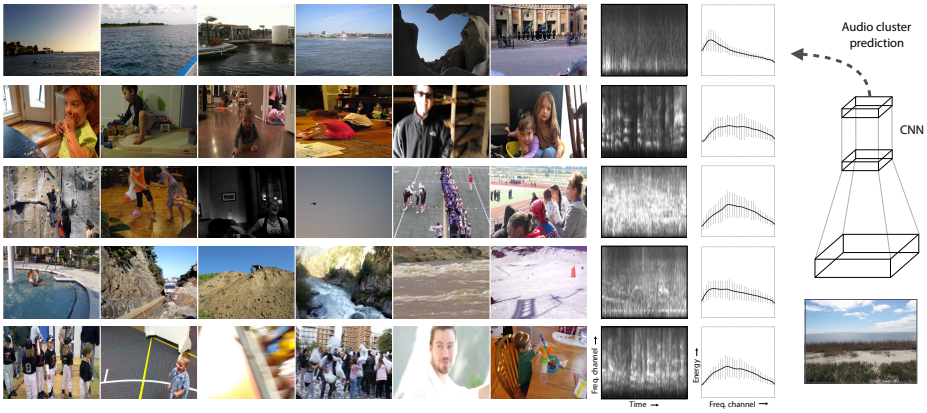
In this paper, we: (1) present a model based on visual CNNs and sound textures [25] that predicts a video frame's held-out sound; (2) demonstrate that the CNN learns units in its convolutional layers that are selective for objects, extending the methodology of Zhou et al. [38]; (3) validate the effectiveness of sound-based supervision by using the learned representation for object- and scene-recognition tasks. These results suggest that sound data, which is available in abundance from consumer videos, provides a useful training signal for visual learning.

2 Related Work

We take inspiration from work in psychology, such as Gaver's Everyday Listening [10], that studies the ways that humans learn about objects and events using sound. In this spirit, we would like to study the situations where sound tells us about visual objects and scenes. Work in auditory scene analysis [6,7,23] meanwhile has provided computational methods for recognizing structures in audio streams. Following this work, we use a sound representation [25] that has been applied to sound recognition [6] and synthesis tasks [25].

Recently, researchers have proposed many unsupervised learning methods that learn visual representations by solving prediction tasks (sometimes known as *pretext* tasks) for which the held-out prediction target is derived from a natural signal in the world, rather than from human annotations. This style of learning has been called "self supervision" [4] or "natural supervision" [30]. With these methods, the supervisory signal may come from video, for example by having the algorithm estimate camera motion [1,17] or track content across frames [35,27,12]. There are also methods that learn from static images, for example by predicting the relative location of image patches [4,16], or by learning invariance to simple geometric and photometric transformations [5]. The assumption behind these methods is that, in order to solve the pretext task, the model has to implicitly learn about semantics and, through this process, develop image features that are broadly useful.

While we share with this work the high-level goal of learning image representations, and we use a similar technical approach, our work differs in significant ways. In contrast to methods whose supervisory signal comes entirely from the imagery itself, ours comes from a modality (sound) that is complementary to vision. This is advantageous because sound is known to be a rich source of information about objects and scenes [10,6], and it is largely invariant to visual transformations, such as lighting, scene composition, and viewing angle. Predicting sound from images thus requires some degree of generalization to visual



(a) Images grouped by audio cluster (b) Clustered audio stats. (c) CNN model

Fig. 2: Visualization of some of the audio clusters used in one of our models (5 of 30 clusters). For each cluster, we show (a) the images in the test set whose sound textures were closest to the centroid (no more than one frame per video), and (b) we visualize aspects of the sound texture used to define the cluster centroid – specifically, the mean and standard deviation of the frequency channels. We also include a representative cochleagram (that of the leftmost image). Although the clusters were defined using audio, there are common objects and scene attributes in many of the images. We train a CNN to predict a video frame's auditory cluster assignment (c).

transformations. Moreover, our supervision task is based on solving a straightforward classification problem, which allows us to use a network design that closely resembles those used in object and scene recognition (rather than, for example, the siamese-style networks used in video methods).

Our approach is closely related to recent audio-visual work [30] that predicts soundtracks for videos that show a person striking objects with a drumstick. A key feature of this work is that the sounds are "visually indicated" by actions in video – a situation that has also been considered in other contexts, such as in the task of visually localizing a sound source [13,19,9] or in evaluating the synchronization between the two modalities [32]. In the natural videos that we use, however, the sound sources are frequently out of frame. Also, in contrast to other recent work in multi-modal representation learning [28,33,2], our technical approach is based on solving a self-supervised classification problem (rather than a generative model or autoencoder), and our goal is to learn visual representations that are generally useful for object recognition tasks.

3 Learning to predict ambient audio

We would like to train a model that, when given a frame of video, can predict its corresponding sound – a task that implicitly requires knowledge of objects and scenes.

3.1 Statistical sound summaries

A natural question, then, is how our model should represent sound. Perhaps the first approach that comes to mind would be to estimate a frequency spectrum at the moment in which the picture was taken, similar to [30]. However, this is potentially suboptimal because in natural scenes it is difficult to predict the precise timing of a sound from visual information. Upon seeing a crowd of people, for instance, we might expect to hear the sound of speech, but the precise timing and content of that speech might not be directly indicated by the video frames.

To be closer to the time scale of visual objects, we estimate a statistical summary of the sound, averaged over a few seconds. We do this using the sound texture model of McDermott and Simoncelli [25], which assumes that sound is stationary within a temporal window (we use 3.75 seconds). More specifically, we closely follow [25] and filter the audio waveform with a bank of 32 band-pass filters intended to mimic human cochlear frequency selectivity. We then take the Hilbert envelope of each channel, raise each sample of the envelope to the 0.3 power (to mimic cochlear amplitude compression), and resample the compressed envelope to 400 Hz. Finally, we compute time-averaged statistics of these subband envelopes: we compute the mean and standard deviation of each frequency channel, the mean squared response of each of a bank of modulation filters applied to each channel, and the Pearson correlation between pairs of channels. For the modulation filters, we use a bank of 10 band-pass filters with center frequencies ranging from 0.5 to 200 Hz, equally spaced on a logarithmic scale.

To make the sound features more invariant to gain (e.g., from the microphone), we divide the envelopes by the median energy (median vector norm) over all timesteps, and include this energy as a feature. As in [25], we normalize the standard deviation of each cochlear channel by its mean, and each modulation power by its standard deviation. We then rescale each kind of texture feature (i.e. marginal moments, correlations, modulation power, energy) inversely with the number of dimensions. The sound texture for each image is a 502-dimensional vector. In Figure 1, we give examples of these summary statistics for two audio clips. We provide more details about our audio representation in the supplementary material.

3.2 Predicting sound from images

We would like to predict sound textures from images – a task that we hypothesize leads to learning useful visual representations. Although multiple frames are available, we predict sound from a single frame, so that the learned image features

will be more likely to transfer to single-image recognition tasks. Furthermore, since the the actions that produce the sounds may not appear on-screen, motion information may not always be applicable.

While one option would be to regress the sound texture v_j directly from the corresponding image I_j, we choose instead to define explicit sound categories and formulate this visual recognition problem as a classification task. This also makes it easier to analyze the network, because it allows us to compare the internal representation of our model to object- and scene-classification models with similar network architecture (Section 4.1). We consider two labeling models: one based on a vector quantization, the other based on a binary coding scheme.

Clustering audio features In the *Clustering* model, the sound textures $\{v_j\}$ in the training set are clustered using k-means. These clusters define image categories: we label each sound texture with the index of the closest centroid, and train our CNN to label images with their corresponding labels.

We found that audio clips that belong to a cluster often contain common objects. In Figure 2, we show examples of such clusters, and in the supplementary material we provide their corresponding audio. We can see that there is a cluster that contains indoor scenes with children in them – these are relatively quiet scenes punctuated with speech sounds. Another cluster contains the sounds of many people speaking at once (often large crowds); another contains many water scenes (usually containing loud wind sounds). Several clusters capture general scene attributes, such as outdoor scenes with light wind sounds. During training, we remove examples that are far from the centroid of their cluster (more than the median distance to the vector, amongst all examples in the dataset).

Binary coding model For the other variation of our model (which we call the *Binary* model), we use a binary coding scheme [14,31,36] equivalent to a multi-label classification problem. We project each sound texture v_j onto the top principal components (we use 30 projections), and convert these projections into a binary code by thresholding them. We predict this binary code using a sigmoid layer, and during training we measure error using cross-entropy loss.

For comparison, we trained a model (which we call the *Spectrum* model) to approximately predict the frequency spectrum at the time that the photo was taken, in lieu of a full sound texture. Specifically, for our sound vectors v_j in this model, we used the mean value of each cochlear channel within a 33.3-millisecond interval centered on the input frame (approximately one frame of a 30 Hz video). For training, we used the projection scheme from the Binary model.

Training We trained our models to predict audio on a 360,000-video subset of the Flickr video dataset [34]. Most of the videos in the dataset are personal video recordings containing natural audio, though many were post-processed, e.g. with added subtitles, title screens, and music. We divided our videos into training and test sets, and we randomly sampled 10 frames per video (1.8 million training images total). For our network architecture, we used the CaffeNet architecture [18] (a variation of Krizhevsky et al. [21]) with batch normalization [15]. We

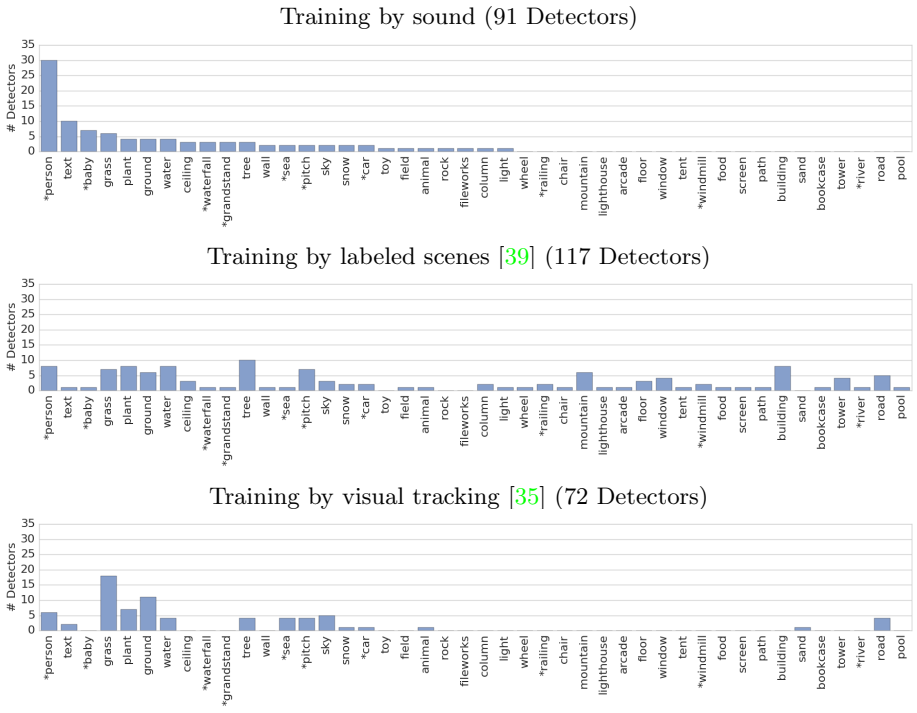


Fig. 3: Histogram of object-selective units in networks trained with different styles of supervision. From top to bottom: training to predict ambient sound (our Clustering model); training to predict scene category using the Places dataset [39]; and training to do visual tracking [35]. Compared to the tracking model, which was also trained without semantic labels, our network learns more high-level object detectors. It also has more detectors for objects that make characteristic sounds, such as *person*, *baby*, and *waterfall*, in comparison to the one trained on Places [39]. Categories marked with $*$ are those that we consider to make characteristic sounds.

trained our model with Caffe [18], using a batch size of 256, for 320,000 iterations of stochastic gradient descent.

4 Results

We evaluate the image representation that our model learned in multiple ways. First, we demonstrate that the internal representation of our model contains convolutional units (neurons) that are selective to particular objects, and we analyze those objects' distribution. We then empirically evaluate the quality of the learned representation for several image recognition tasks, finding that it

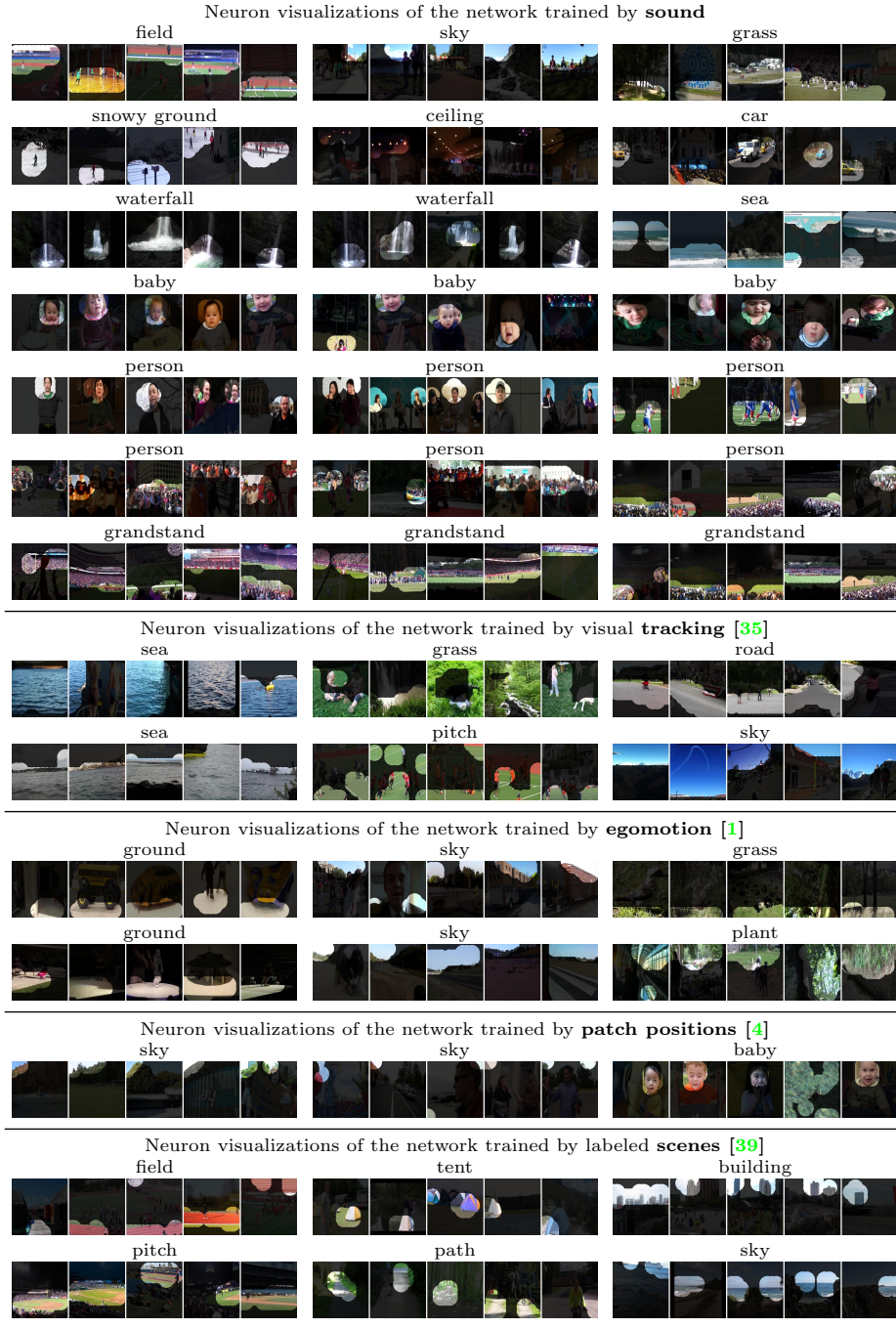


Fig. 4: Top 5 responses for neurons of various networks, tested on the Flickr dataset. Please see Section A2 for more visualizations.

achieves performance comparable to other feature-learning methods that were trained without human annotations.

4.1 What does the network learn to detect?

Previous work [38] has shown that a CNN trained to predict scene categories will learn convolutional units that are selective for objects – a result that follows naturally from the fact that scenes are often defined by the objects that compose them. We ask whether a model trained to predict ambient sound, rather than explicit human labels, would learn object-selective units as well. For these experiments, we used our Clustering model, because its network structure is similar to that of the scene-recognition model used in [38].

Quantifying object-selective units Similar to the method in [38], we visualized the images that each neuron in the top convolutional layer (conv5) responded most strongly to. To do this, we sampled a pool of 200,000 images from our Flickr video test set. We then collected, for each convolutional unit, the 60 images in this set that gave the unit the largest activation. Next, we applied the so-called synthetic visualization technique of [38] to approximately superimpose the unit's receptive field onto the image. Specifically, we found all of the spatial locations in the layer for which the unit's activation strength was at least half that of its maximum response. We then masked out the parts of the image that were not covered by the receptive field of one of these high-responding spatial units. We assumed a circle-shaped receptive field, obtaining the radius from [38]. To examine the effect of the data used in the evaluation, we also applied this visualization technique to other datasets (please see the supplementary material).

Next, for each neuron we showed its masked images to three human annotators on Amazon Mechanical Turk, and asked them: (1) whether an object is present in many of these regions, and if so, what it is; (2) to mark the images whose activations contain these objects. Unlike [38], we only considered units that were selective to objects, ignoring units that were selective to textures. For each unit, if at least 60% of its top 60 activations contained the object, we considered it to be selective for the object (or following [38], we say that it is a *detector* for that object). We then manually labeled the unit with an object category, using the category names provided by the SUN database [37]. We found that 91 of the 256 units in our model were object-selective in this way, and we show a selection of them in Figure 4.

We compared the number of these units to those of a CNN trained to recognize human-labeled scene categories on Places [38]. As expected, this model – having been trained with explicit human annotations – contained more object-selective units (117 units). We also asked whether object-selective neurons appear in the convolutional layers when a CNN is trained on other tasks that do not use human labels. As a simple comparison, we applied the same methodology to the egomotion-based model of Agrawal et al. [1] and to the tracking-based method of Wang and Gupta [35]. We applied these networks to whole images

Method	Sound	Places
# Detectors	91	117
# Detectors for objects with characteristic sounds	49	26
Videos with object sound	43.7%	16.9%
Characteristic sound rate	81.2%	75.9%

Table 1: Row 1: the number of detectors (i.e. units that are selective to a particular object); row 2: the number of detectors for objects with characteristic sounds; row 3: fraction of videos in which an object's sound is audible (computed only for object classes with characteristic sounds); row 4: given that an activation corresponds to an object with a characteristic sound, the probability that its sound is audible. There are 256 units in total for each method.

(in all cases resizing the input image to 256×256 pixels and taking the center 227×227 crop), though we note that they were originally trained on cropped image regions.

We found that the tracking-based method also learned object-selective units, but that the objects that it detected were often textural "stuff," such as grass, ground, and water, and that there were fewer of these detection units in total (72 of 256). The results were similar for the egomotion-based model, which had 27 such units. In Figure 3 and in the supplementary material, we provide the distribution of the objects that the units were selective to. We also visualized neurons from the method of Doersch et al. [4] (as before, applying the network to whole images, rather than to patches). We found a significant number of the units were selective for position, rather than to objects. For example, one convolutional unit responded most highly to the upper-left corner of an image – a unit that may be useful for the training task, which involves predicting the relative position of image patches. In Figure 4, we show visualizations of a selection of object-detecting neurons for all of these methods.

The differences between the objects detected by these methods and our own may have to do with the requirements of the tasks being solved. The other unsupervised methods, for example, all involve comparing multiple input images or sub-images in a relatively fine-grained way. This may correspondingly change the representation that the network learns in its last convolutional layer – requiring its the units to encode, say, color and geometric transformations rather than object identities. Moreover, these networks may represent semantic information in other (more distributed) ways that would not necessarily be revealed through this visualization method.

Analyzing the types of objects that were detected Next, we asked what kinds of objects our network learned to detect. We hypothesized that the object-selective neurons were more likely to respond to objects that produce (or are closely associated with) characteristic sounds. To evaluate this, we (an author) labeled the SUN object categories according to whether they were closely

associated with a characteristic sound. We denote these categories with a $*$ in Figure 3. Next, we counted the number of units that were selective to these objects, finding that our model contained significantly more such units than a scene-recognition network trained on the Places dataset, both in total number and as a proportion (Table 1). A significant fraction of these units were selective to people (adults, babies, and crowds).

Finally, we asked whether the sounds that these objects make were actually present in the videos that these video frames were sampled from. To do this, we listened to the sound of the top 30 video clips for each unit, and recorded whether the sound was made by the object that the neuron was selective to (e.g., human speech for the *person* category). We found that 43.7% of these videos contained the objects' sounds (Table 1).

4.2 Evaluating the image representation

We have seen through visualizations that a CNN trained to predict sound from an image learns units that are highly selective for objects. Now we evaluate, experimentally, how well the CNN's internal representation conveys information that is useful for recognizing objects and scenes.

Since our goal is to measure the amount of semantic information provided by the learned representation, rather than to seek absolute performance, we used a simple evaluation scheme. In most experiments, we computed image features using our CNN and trained a linear SVM to predict object or scene category using the activations in the top layers.

Object recognition First, we used our CNN features for object recognition on the PASCAL VOC 2007 dataset [8]. We trained a one-vs.-rest linear SVM to detect the presence of each of the 20 object categories in the dataset, using the activations of the upper layers of the network as the feature set (pool5, fc6, and fc7). To help understand whether the convolutional units considered in Section 4.1 directly convey semantics, we also created a global max-pooling feature (similar to [29]), where we applied max pooling over the entire convolutional layer. This produces a 256-dimensional vector that contains the maximum response of each convolutional unit (we call it *max5*). Following common practice, we evaluated the network on a center 227×227 crop of each image (after resizing the image to 256×256), and we evaluated the results using mean average precision (mAP). We chose the SVM regularization parameter for each method by maximizing mAP on the validation set using grid search (we used $\{0.5^k \mid 4 \leq k < 20\}$).

The other unsupervised (or self-supervised) models in our comparison [4,1,35] use different network designs. In particular, [4] was trained on image patches, so following their experiments we resized its convolutional layers for 227×227 images and removed the model's fully connected layers[1]. Also, since the model

[1] As a result, this model has a larger pool5 layer than the other methods: 7×7 vs. 6×6. Likewise, the fc6 layer of [35] is smaller (1,024 dims. vs. 4,096 dims.).

Method	VOC Cls. (%mAP)				SUN397 (%acc.)			
	max5	pool5	fc6	fc7	max5	pool5	fc6	fc7
Sound (cluster)	36.7	45.8	44.8	44.3	**17.3**	**22.9**	20.7	14.9
Sound (binary)	**39.4**	**46.7**	**47.1**	**47.4**	17.1	22.5	**21.3**	**21.4**
Sound (spect.)	35.8	44.0	44.4	44.4	14.6	19.5	18.6	17.7
Texton-CNN	28.9	37.5	35.3	32.5	10.7	15.2	11.4	7.6
K-means [20]	27.5	34.8	33.9	32.1	11.6	14.9	12.8	12.4
Tracking [35]	33.5	42.2	42.4	40.2	14.1	18.7	16.2	15.1
Patch pos. [4]	26.8	46.1	-	-	9.8	22.2	-	-
Egomotion [1]	22.7	31.1	-	-	9.1	11.3	-	-
ImageNet [21]	**63.6**	**65.6**	**69.6**	**73.6**	29.8	34.0	37.8	37.8
Places [39]	59.0	63.2	65.3	66.2	**39.4**	**42.1**	**46.1**	**48.8**

(a) Image classification with linear SVM

Method	(%mAP)
Random init. [20]	41.3
Sound (cluster)	44.1
Sound (binary)	43.3
Motion [35,20]	44.0
Egomotion [1,20]	41.8
Patch pos. [4,20]	46.6
Calib. + Patch [4,20]	**51.1**
ImageNet [21]	**57.1**
Places [39]	52.8

(b) Finetuning detection

Method	aer	bk	brd	bt	btl	bus	car	cat	chr	cow	din	dog	hrs	mbk	prs	pot	shp	sfa	trn	tv
Sound (cluster)	68	**47**	38	54	15	45	**66**	45	42	23	37	28	73	58	**85**	25	26	32	67	42
Sound (binary)	69	45	38	56	**16**	**47**	65	45	41	**25**	37	28	**74**	**61**	**85**	26	**39**	32	**69**	38
Sound (spect.)	65	40	35	54	14	42	63	41	39	24	32	25	72	56	81	**27**	33	28	65	40
Texton-CNN	65	35	28	46	11	31	63	30	41	17	28	23	64	51	74	9	19	33	54	30
K-means	61	31	27	49	9	27	58	34	36	12	25	21	64	38	70	18	14	25	51	25
Motion [35]	67	35	41	54	11	35	62	35	39	21	30	26	70	53	78	22	32	37	61	34
Patches [4]	**70**	44	**43**	**60**	12	44	**66**	**52**	**44**	24	**45**	**31**	73	48	78	14	28	**39**	62	**43**
Egomotion [1]	60	24	21	35	10	19	57	24	27	11	22	18	61	40	69	13	12	24	48	28
ImageNet [21]	79	**71**	**73**	75	**25**	60	80	**75**	51	**45**	60	**70**	**80**	**72**	**91**	42	**62**	56	82	62
Places [39]	**83**	60	56	**80**	23	**66**	**84**	54	**57**	40	**74**	41	**80**	68	90	**50**	45	**61**	**88**	**63**

(c) Per class mAP for image classification on PASCAL VOC 2007

Table 2: (a) Mean average precision for PASCAL VOC 2007 classification, and accuracy on SUN397. Here we trained a linear SVM using the top layers of different networks. We note in Section 4.2 that the shape of these layers varies between networks. (b) Mean average precision on PASCAL VOC 2007 using Fast-RCNN [11]. We initialized the CNN weights using those of our learned sound models. (c) Per-class AP scores for the VOC 2007 classification task with pool5 features (corresponds to mAP in (a)).

of Agrawal et al. [1] did not have a pool5 layer, we added one to it. We also considered CNNs that were trained with human annotations: object recognition on ImageNet [3] and scene categories on Places [39]. Finally, we considered using the k-means weight initialization method of [20] to set the weights of a CNN model (we call this the *K-means* model).

We found that our best-performing of our model (the binary-coding method) obtained comparable performance to other unsupervised learning methods, such as [4]. Both models based on sound textures (Clustering and Binary) outperformed the model that predicted only the frequency spectrum. This suggests that the additional time-averaged statistics from sound textures are helpful. For these models, we used 30 clusters (or PCA projections): in Figure A1a, we consider varying the number of clusters, finding that there is a small improvement from increasing it, and a substantial decrease in performance when using just

two clusters. The sound-based models significantly outperformed other methods when we globally pooled the conv5 features, suggesting that the convolutional units contain a significant amount of semantic information (and are well suited to being used at this spatial scale).

Scene recognition We also evaluated our model on a scene recognition task using the SUN dataset [37], a large classification benchmark that involves recognizing 397 scene categories with 7,940 training and test images provided in multiple splits. Following [1], we averaged our classification accuracy across 3 splits, with 20 examples per scene category. We chose the linear SVM's regularization parameter for each model using 3-fold cross-validation.

We again found that our features' performance was comparable to other models. In particular, we found that the difference between our models was smaller than in the object-recognition case, with both the Clustering and Binary models obtaining performance comparable to the patch-based method with pool5 features.

Pretraining for object detection Following recent work [35,4,20], we used our model to initialize the weights of a CNN-based object detection system (Fast R-CNN [11]), verifying that the results improved over random initialization. We followed the training procedure of Krähenbühl et al. [20], using 150,000 iterations of backpropagation with an initial learning rate of 0.002, and we compared our model with other published results (we report the numbers provided by [20]).

Our best-performing model (the Clustering model) obtains similar performance to that of Wang and Gupta's tracking-based model [35], while the overall best results were from variations of Doersch et al.'s patch-based model [4,20]. We note that the network changes substantially during fine-tuning, and thus the performance is fairly dependent on the parameters used in the training procedure. Moreover all models, when fine-tuned in this way, achieve results that are close to those of a well-chosen random initialization (within 6% mAP). Recent work [20,26] has addressed these optimization issues by rescaling the weights of a pretrained network using a data-driven procedure. The unsupervised method with the best performance combines the rescaling method of [20] with the patch-based pretraining of [4].

Sound prediction We also asked how well our model learned to solve its sound prediction task. We found that on our test set, the clustering-based model (with 30 clusters) chose the correct sound label 15.8% of the time. Pure chance in this case is 3.3%, while the baseline of choosing the most commonly occurring label is 6.6%.

Audio supervision It is natural to ask what role audio plays in the learning process. Perhaps, for example, our training procedure would produce equally good features if we replaced the hand-crafted sound features with hand-crafted *visual* features, computed from the images themselves. To study this, we replaced our sound texture features with (512-dimensional) visual texton histograms [24],

using the parameters from [37], and we used them to train a variation of our Clustering model.

As expected, the images that belong to each cluster are visually coherent, and share common objects. However, we found that the network performed significantly worse than the audio-based method on the object- and scene-recognition metrics (Table 2a). Moreover, we found that its convolutional units rarely were selective for objects (generally they responded responded to "stuff" such as grass and water). Likely this is because the network simply learned to approximate the texton features, obtaining low labeling error without high-level generalization. In contrast, the audio-based labels – despite also being based on another form of hand-crafted feature – are largely invariant to visual transformations, such as lighting and scale, and therefore predicting them requires some degree of generalization (one benefit of training with multiple, complementary modalities).

5 Discussion

Sound has many properties that make it useful as a supervisory training signal: it is abundantly available without human annotations, and it is known to convey information about objects and scenes. It is also complementary to visual information, and may therefore convey information not easily obtainable from unlabeled image analysis.

In this work, we proposed using ambient sound to learn visual representations. We introduced a model, based on convolutional neural networks, that predicts a statistical sound summary from a video frame. We then showed, with visualizations and experiments on recognition tasks, that the resulting image representation contains information about objects and scenes.

Here we considered one audio representation, based on sound textures, but it is natural to ask whether other audio representations would lead the model to learn about additional types of objects. To help answer this question, we would like to more systematically study the situations when sound does (and does not) tell us about objects in the visual world. Ultimately, we would like to know what object and scene structures are detectable through sound-based training, and we see our work as a step in this direction.

Acknowledgments. This work was supported by NSF grants #1524817 to A.T; NSF grants #1447476 and #1212849 to W.F.; a McDonnell Scholar Award to J.H.M.; and a Microsoft Ph.D. Fellowship to A.O. It was also supported by Shell Research, and by a donation of GPUs from NVIDIA. We thank Phillip Isola and Carl Vondrick for the helpful discussions, and the anonymous reviewers for their comments (in particular, for suggesting the comparison with texton features in Section 4.2).

References

1. Agrawal, P., Carreira, J., Malik, J.: Learning to see by moving. In: ICCV (2015) 3, 8, 9, 11, 12, 13, 16, 19

2. Andrew, G., Arora, R., Bilmes, J.A., Livescu, K.: Deep canonical correlation analysis. In: ICML (2013) 4
3. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: CVPR (2009) 12
4. Doersch, C., Gupta, A., Efros, A.A.: Unsupervised visual representation learning by context prediction. In: ICCV (2015) 2, 3, 8, 10, 11, 12, 13
5. Dosovitskiy, A., Springenberg, J.T., Riedmiller, M., Brox, T.: Discriminative unsupervised feature learning with convolutional neural networks. In: NIPS (2014) 3
6. Ellis, D.P., Zeng, X., McDermott, J.H.: Classifying soundtracks with audio texture features. In: ICASSP (2011) 3
7. Eronen, A.J., Peltonen, V.T., Tuomi, J.T., Klapuri, A.P., Fagerlund, S., Sorsa, T., Lorho, G., Huopaniemi, J.: Audio-based context recognition. IEEE TASLP 14(1), 321–329 (2006) 3
8. Everingham, M., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The pascal visual object classes (voc) challenge. IJCV 88(2), 303–338 (2010) 11
9. Fisher III, J.W., Darrell, T., Freeman, W.T., Viola, P.A.: Learning joint statistical models for audio-visual fusion and segregation. In: NIPS (2000) 4
10. Gaver, W.W.: What in the world do we hear?: An ecological approach to auditory event perception. Ecological psychology 5(1), 1–29 (1993) 1, 2, 3
11. Girshick, R.: Fast r-cnn. In: ICCV (2015) 12, 13
12. Goroshin, R., Bruna, J., Tompson, J., Eigen, D., LeCun, Y.: Unsupervised feature learning from temporal data. arXiv preprint arXiv:1504.02518 (2015) 3
13. Hershey, J.R., Movellan, J.R.: Audio vision: Using audio-visual synchrony to locate sounds. In: NIPS (1999) 4
14. Indyk, P., Motwani, R.: Approximate nearest neighbors: towards removing the curse of dimensionality. In: STOC (1998) 6
15. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: ICML (2015) 6
16. Isola, P., Zoran, D., Krishnan, D., Adelson, E.H.: Learning visual groups from co-occurrences in space and time. In: ICLR Workshop (2016) 3
17. Jayaraman, D., Grauman, K.: Learning image representations tied to ego-motion. In: ICCV (2015) 3
18. Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., Darrell, T.: Caffe: Convolutional architecture for fast feature embedding. In: MM (2014) 6, 7
19. Kidron, E., Schechner, Y.Y., Elad, M.: Pixels that sound. In: CVPR (2005) 4
20. Krähenbühl, P., Doersch, C., Donahue, J., Darrell, T.: Data-dependent initializations of convolutional neural networks. In: ICLR (2016) 12, 13
21. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: NIPS (2012) 6, 12
22. Le, Q.V., Ranzato, M.A., Monga, R., Devin, M., Chen, K., Corrado, G.S., Dean, J., Ng, A.Y.: Building high-level features using large scale unsupervised learning. In: ICML (2012) 2
23. Lee, K., Ellis, D.P., Loui, A.C.: Detecting local semantic concepts in environmental sounds using markov model based clustering. In: ICASSP (2010) 3
24. Leung, T., Malik, J.: Representing and recognizing the visual appearance of materials using three-dimensional textons. IJCV 43(1), 29–44 (2001) 13
25. McDermott, J.H., Simoncelli, E.P.: Sound texture perception via statistics of the auditory periphery: evidence from sound synthesis. Neuron 71(5), 926–940 (2011) 1, 3, 5, 17, 19

26. Mishkin, D., Matas, J.: All you need is a good init. arXiv preprint arXiv:1511.06422 (2015) 13
27. Mobahi, H., Collobert, R., Weston, J.: Deep learning from temporal coherence in video. In: ICML (2009) 3
28. Ngiam, J., Khosla, A., Kim, M., Nam, J., Lee, H., Ng, A.Y.: Multimodal deep learning. In: ICML (2011) 4
29. Oquab, M., Bottou, L., Laptev, I., Sivic, J.: Is object localization for free?-weakly-supervised learning with convolutional neural networks. In: CVPR (2015) 11
30. Owens, A., Isola, P., McDermott, J., Torralba, A., Adelson, E.H., Freeman, W.T.: Visually indicated sounds. In: CVPR (2016) 3, 4, 5
31. Salakhutdinov, R., Hinton, G.: Semantic hashing. International Journal of Approximate Reasoning 50(7), 969–978 (2009) 6
32. Slaney, M., Covell, M.: Facesync: A linear operator for measuring synchronization of video facial images and audio tracks. In: NIPS (2000) 4
33. Srivastava, N., Salakhutdinov, R.R.: Multimodal learning with deep boltzmann machines. In: NIPS (2012) 4
34. Thomee, B., Shamma, D.A., Friedland, G., Elizalde, B., Ni, K., Poland, D., Borth, D., Li, L.J.: The new data and new challenges in multimedia research. arXiv preprint arXiv:1503.01817 (2015) 2, 6
35. Wang, X., Gupta, A.: Unsupervised learning of visual representations using videos. In: ICCV (2015) 2, 3, 7, 8, 9, 11, 12, 13, 16, 17, 18
36. Weiss, Y., Torralba, A., Fergus, R.: Spectral hashing. In: NIPS (2009) 6
37. Xiao, J., Hays, J., Ehinger, K.A., Oliva, A., Torralba, A.: Sun database: Large-scale scene recognition from abbey to zoo. In: CVPR (2010) 9, 13, 14
38. Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., Torralba, A.: Object detectors emerge in deep scene cnns. In: ICLR 2015 (2014) 3, 9, 16, 17
39. Zhou, B., Lapedriza, A., Xiao, J., Torralba, A., Oliva, A.: Learning deep features for scene recognition using places database. In: NIPS (2014) 7, 8, 12, 18

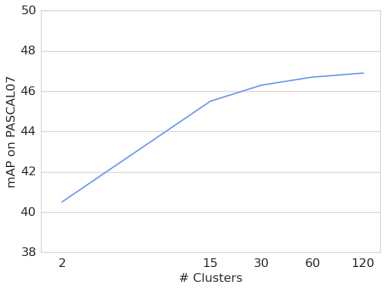
A1 Sound label space

Why do detectors for certain objects – e.g., people, water, and infants – emerge in our model? To help answer this question, we visualized the audio clusters that are used to define our model's label space (Section 3). The results, including sound clips, are provided on our webpage. In Figure A1a, we also examine how the quality of the learned image features varies as a function of the number of clusters, as measured by performance on the object recognition task.

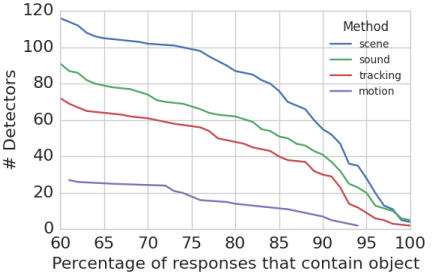
A2 Additional unit visualizations

In Figure A5, we provide visualizations of additional object-selective neurons in our model. In Figure A4 we provide object-detector histograms for additional unsupervised methods [1,35] (an extension of Figure 3). We also show how the number of object-selective units changes as a function of the threshold used to define whether a unit is selective (Figure A1b).

To examine the effect of the dataset used to create the neuron visualizations, we applied the same neuron visualization technique to 200,000 images sampled



(a) Varying the number of clusters



(b) Varying the threshold for selectivity

Fig. A1: (a) Object recognition performance (recognition performance on PASCAL VOC2007) increases with the number of clusters used to define the audio label space. For our experiments, we used 30 clusters. (b) The number of object-selective units for each method, as we increase the threshold used to determine whether a unit is object-selective. This threshold corresponds to the fraction of images that contain the object in question, amongst the images with the 60 largest activations. For our analysis in Section 4, we used a threshold of 60%.

equally from the SUN and ImageNet datasets (as in [38]). As expected, we found that the distribution of objects was similar to that of the Flickr dataset (Figure A2). Notably, there were fewer detectors in total (67 vs. 91), and there were some categories (e.g., *baby*) that appeared relatively less often. This may be due to the differences in the underlying distribution of objects in the datasets. For example, SUN focuses on scenes and contains more objects labeled *tree*, *lamp*, and *window* than objects labeled *person* [38]. We also computed a detector histogram for the model of [35], finding that the total number of detectors was similar to the sound-based model (61 detectors), but that, as before, the dominant categories were textural "stuff" (e.g., grass, plants).

A3 Sound textures

We now describe, in more detail, how we computed sound textures from audio clips. For this, we closely follow the work of McDermott and Simoncelli [25].

Subband envelopes To compute the cochleagram features $\{c_i\}$, we filter the input waveform s with a bank of bandpass filters $\{f_i\}$.

$$c_i(t) = |(s * f_i) + jH(s * f_i)|, \tag{1}$$

where H is the Hilbert transform and $*$ denotes cross-correlation. We then resample the signal to 400Hz and compress it by raising each sample to the 0.3 power (examples in Figure 1).

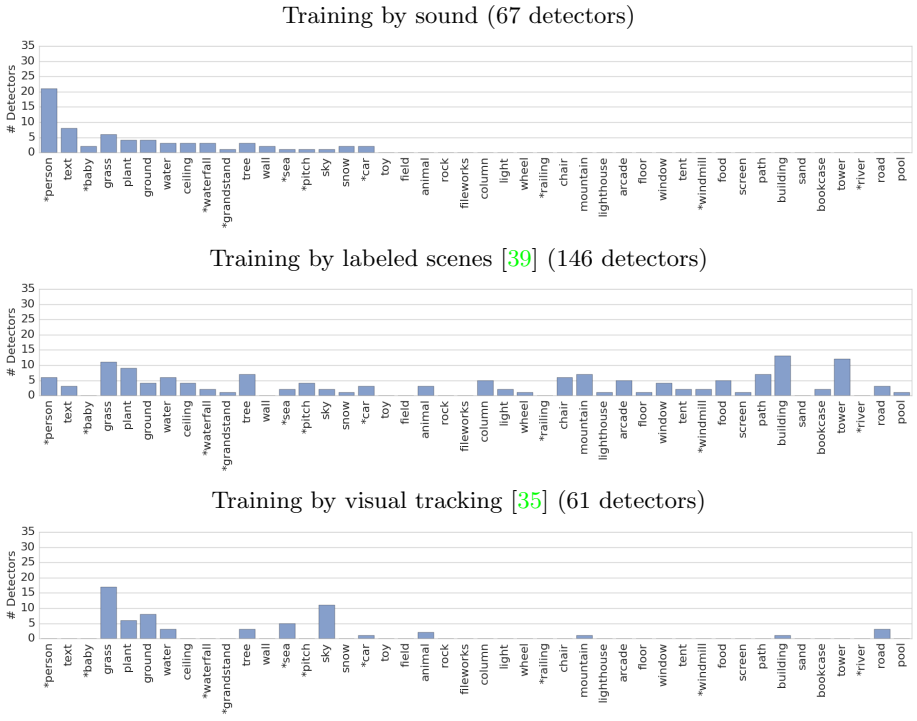


Fig. A2: The number of object-selective per category, when evaluating the model on the SUN and ImageNet datasets (cf. Figure 3, in which the models were evaluated on the Flickr video dataset).

Correlations As described in Section 3, we compute the correlation between bands using a subset of the entries in the cochlear-channel correlation matrix. Specifically, we include the correlation between channels c_j and c_k if $|j-k| \in \{1,2,3,5\}$. The result is a vector ρ of correlation values.

Modulation filters We also include modulation filter responses. To get these, we compute each band's response to a filter bank $\{m_i\}$ of 10 bandpass filters whose center frequencies are spaced logarithmically from 0.5 to 200Hz:

$$b_{ij} = \frac{1}{N}||c_i * m_j||^2, \tag{2}$$

where N is the length of the signal.

Marginal statistics We estimate marginal moments of the cochleagram features, computing the mean μ_i and standard deviation σ_i of each channel. We also estimate the loudness, l, of the sequence by taking the median of the energy at each timestep, i.e. $l = \text{median}(||c(t)||)$.

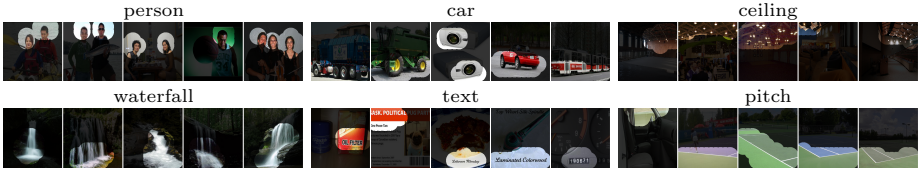


Fig. A3: A selection of object-selective neurons, obtained by testing our model on the SUN and ImageNet datasets. We show the top 5 activations for each unit.

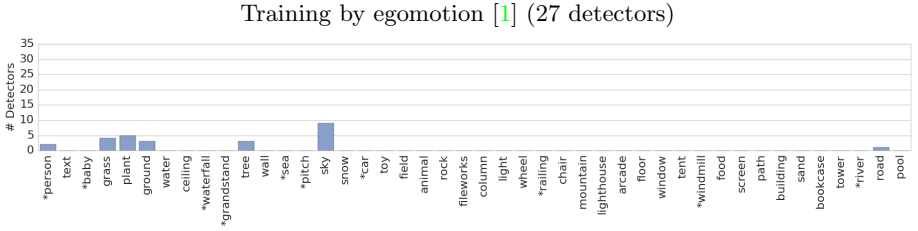


Fig. A4: Here we quantify the number of object-selective units for an additional method, using the Flickr video dataset (cf. Figure 3).

Normalization To account for global differences in gain, we normalize the cochleagram features by dividing by the loudness, l. Following [25], we normalize the modulation filter responses by the variance of the cochlear channel, computing $\tilde{b}_{ij} = \sqrt{\frac{b_{ij}}{\sigma_i^2}}$. Similarly, we normalize the standard deviation of each cochlear channel, computing $\tilde{\sigma}_i = \sqrt{\frac{\sigma_i^2}{\mu_i^2}}$. From these normalized features, we construct a sound texture vector: $[\mu, \tilde{\sigma}, \rho, \tilde{b}, l]$

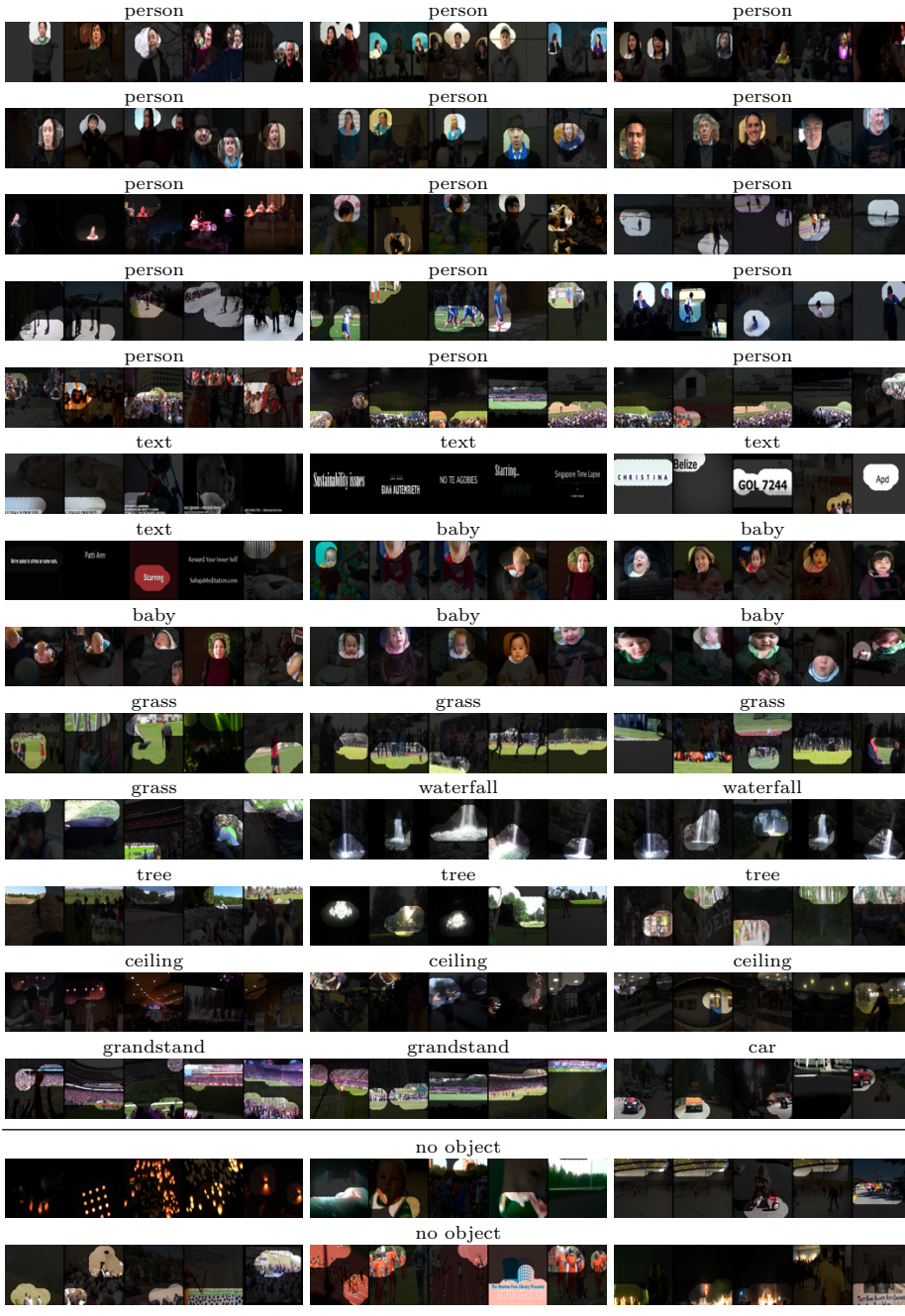


Fig. A5: Top 5 activations for units in our model (39 of 91 from common classes). The last two rows show neurons that were not selective to an object class.