Learning Features by Watching Objects Move

Deepak Pathak[1,2,*], Ross Girshick[1], Piotr Dollár[1], Trevor Darrell[2], and Bharath Hariharan[1]

[1]Facebook AI Research (FAIR)
[2]University of California, Berkeley

Abstract

This paper presents a novel yet intuitive approach to unsupervised feature learning. Inspired by the human visual system, we explore whether low-level motion-based grouping cues can be used to learn an effective visual representation. Specifically, we use unsupervised motion-based segmentation on videos to obtain segments, which we use as 'pseudo ground truth' to train a convolutional network to segment objects from a single frame. Given the extensive evidence that motion plays a key role in the development of the human visual system, we hope that this straightforward approach to unsupervised learning will be more effective than cleverly designed 'pretext' tasks studied in the literature. Indeed, our extensive experiments show that this is the case. When used for transfer learning on object detection, our representation significantly outperforms previous unsupervised approaches across multiple settings, especially when training data for the target task is scarce.

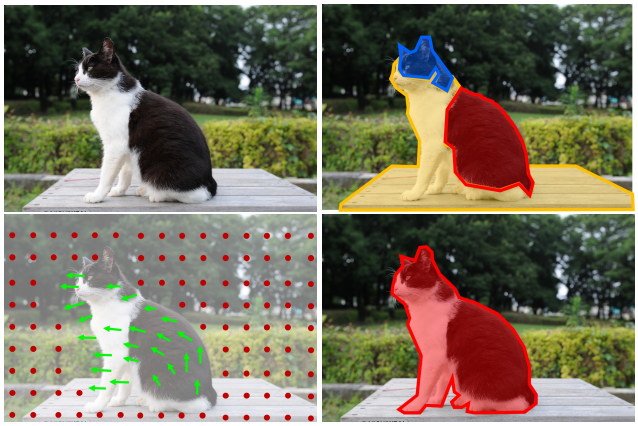


Figure 1. Low-level appearance cues lead to incorrect grouping (top right). Motion helps us to correctly group pixels that move together (bottom left) and identify this group as a single object (bottom right). We use unsupervised motion-based grouping to train a ConvNet to segment objects in *static images* and show that the network learns strong features that transfer well to other tasks.

1. Introduction

ConvNet-based image representations are extremely versatile, showing good performance in a variety of recognition tasks [9, 15, 19, 50]. Typically these representations are trained using *supervised learning* on large-scale image classification datasets, such as ImageNet [41]. In contrast, animal visual systems do not require careful manual annotation to learn, and instead take advantage of the nearly infinite amount of unlabeled data in their surrounding environments. Developing models that can learn under these challenging conditions is a fundamental scientific problem, which has led to a flurry of recent work proposing methods that learn visual representations without manual annotation.

A recurring theme in these works is the idea of a 'pretext task': a task that is not of direct interest, but can be used to obtain a good visual representation as a byproduct of training. Example pretext tasks include reconstructing the input [4, 20, 44], predicting the pixels of the next frame in a video stream [17], metric learning on object track endpoints [46], temporally ordering shuffled frames from a video [29], and spatially ordering patches from a static image [8, 30]. The challenge in this line of research lies in cleverly designing a pretext task that causes the ConvNet (or other representation learner) to learn high-level features.

In this paper, we take a different approach that is motivated by human vision studies. Both infants [42] and newly sighted congenitally blind people [32] tend to *oversegment* static objects, but can group things properly when they *move* (Figure 1). To do so, they may rely on the Gestalt principle of common fate [34, 47]: pixels that move together tend to belong together. The ability to parse static scenes improves [32] over time, suggesting that while motion-based grouping appears early, static grouping is acquired later, possibly bootstrapped by motion cues. Moreover, experiments in [32] show that shortly after gaining sight, human subjects are better able to name objects that tend to be seen

*Work done during an internship at FAIR.

in motion compared to objects that tend to be seen at rest.

Inspired by these human vision studies, we propose to train ConvNets for the well-established task of object foreground *vs*. background segmentation, using unsupervised motion segmentation to provide 'pseudo ground truth'. Concretely, to prepare training data we use optical flow to group foreground pixels that move together into a single object. We then use the resulting segmentation masks as automatically generated targets, and task a ConvNet with predicting these masks from *single, static frames without any motion information* (Figure 2). Because pixels with different colors or low-level image statistics can still move together and form a single object, the ConvNet cannot solve this task using a low-level representation. Instead, it may have to *recognize* objects that tend to move and identify their shape and pose. Thus, we conjecture that this task forces the ConvNet to learn a high-level representation.

We evaluate our proposal in two settings. First, we test if a ConvNet can learn a good feature representation when learning to segment from the high-quality, manually labeled segmentations in COCO [27], without using the class labels. Indeed, we show that the resulting feature representation is effective when transferred to PASCAL VOC object detection. It achieves state-of-the-art performance for representations trained without any semantic category labels, performing within 5 points AP of an ImageNet pretrained model and 10 points higher than the best unsupervised methods. This justifies our proposed task by showing that *given good ground truth segmentations, a ConvNet trained to segment objects will learn an effective feature representation*.

Our goal, however, is to learn features *without manual supervision*. Thus in our second setting we train with *automatically generated* 'pseudo ground truth' obtained through unsupervised motion segmentation on uncurated videos from the Yahoo Flickr Creative Commons 100 million (YFCC100m) [43] dataset. When transferred to object detection, our representation retains good performance even when most of the ConvNet parameters are frozen, significantly outperforming previous unsupervised learning approaches. It also allows much better transfer learning when training data for the target task is scarce. Our representation quality tends to increase logarithmically with the amount of data, suggesting the possibility of outperforming ImageNet pretraining given the countless videos on the web.

2. Related Work

Unsupervised learning is a broad area with a large volume of work; Bengio *et al*. [5] provide an excellent survey. Here, we briefly revisit some of the recent work in this area.

Unsupervised learning by generating images. Classical unsupervised representation learning approaches, such as autoencoders [4, 20] and denoising autoencoders [44], attempt to learn feature representations from which the original image can be decoded with a low error. An alternative to reconstruction-based objectives is to train generative models of images using generative adversarial networks [16]. These models can be extended to produce good feature representations by training jointly with image encoders [10, 11]. However, to generate realistic images, these models must pay significant attention to low-level details while potentially ignoring higher-level semantics.

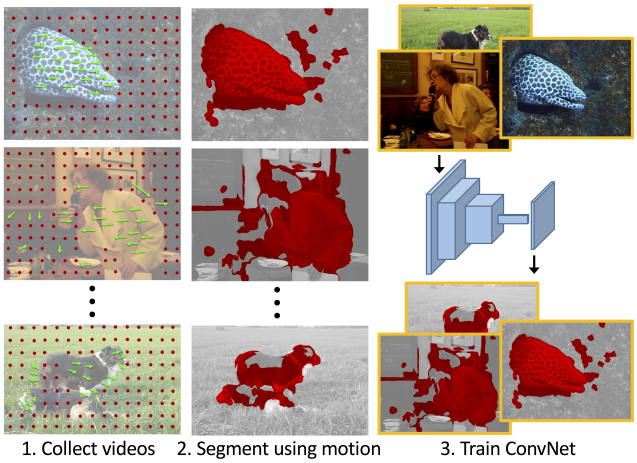


Figure 2. Overview of our approach. We use motion cues to segment objects in videos *without any supervision*. We then train a ConvNet to predict these segmentations from *static frames*, *i.e*. without any motion cues. We then transfer the learned representation to other recognition tasks.

Self-supervision via pretext tasks. Instead of producing images, several recent studies have focused on providing alternate forms of supervision (often called 'pretext tasks') that do not require manual labeling and can be algorithmically produced. For instance, Doersch *et al*. [8] task a ConvNet with predicting the relative location of two cropped image patches. Noroozi and Favaro [30] extend this by asking a network to arrange shuffled patches cropped from a 3×3 grid. Pathak *et al*. [35] train a network to perform an image inpainting task. Other pretext tasks include predicting color channels from luminance [25, 51] or vice versa [52], and predicting sounds from video frames [7, 33]. The assumption in these works is that to perform these tasks, the network will need to recognize high-level concepts, such as objects, in order to succeed. We compare our approach to all of these pretext tasks and show that the proposed natural task of object segmentation leads to a quantitatively better feature representation in many cases.

Learning from motion and action. The human visual system does not receive static images; it receives a continuous video stream. The same idea of defining auxiliary pretext tasks can be used in unsupervised learning from videos too. Wang and Gupta [46] train a ConvNet to distinguish be-

tween pairs of tracked patches in a single video, and pairs of patches from different videos. Misra *et al*. [29] ask a network to arrange shuffled frames of a video into a temporally correct order. Another such pretext task is to make predictions about the next few frames: Goroshin *et al*. [17] predict pixels of future frames and Walker *et al*. [45] predict dense future trajectories. However, since nearby frames in a video tend to be visually similar (in color or texture), these approaches might learn low-level image statistics instead of more semantic features. Alternatively, Li *et al*. [26] use motion boundary detection to bootstrap a ConvNet-based contour detector, but find that this does not lead to good feature representations. Our intuitions are similar, but our approach produces semantically strong representations.

Animals and robots can also sense their own motion (proprioception), and a possible task is to predict this signal from the visual input alone [2, 14, 21]. While such cues undoubtedly can be useful, we show that strong representations can be learned even when such cues are unavailable.

3. Evaluating Feature Representations

To measure the quality of a learned feature representation, we need an evaluation that reflects real-world constraints to yield useful conclusions. Prior work on unsupervised learning has evaluated representations by using them as initializations for fine-tuning a ConvNet for a particular isolated task, such as object detection [8]. The intuition is that a good representations should serve as a good starting point for task-specific fine-tuning. While fine-tuning for each task can be a good solution, it can also be impractical.

For example, a mobile app might want to handle multiple tasks on device, such as image classification, object detection, and segmentation. But both the app download size and execution time will grow linearly with the number of tasks unless computation is shared. In such cases it may be desirable to have a general representation that is shared between tasks and task-specific, lightweight classifier 'heads'.

Another practical concern arises when the amount of labeled training data is too limited for fine-tuning. Again, in this scenario it may be desirable to use a fixed general representation with a trained task-specific 'head' to avoid overfitting. Rather than emphasizing any one of these cases, in this paper we aim for a broader understanding by evaluating learned representations under a variety of conditions:

1. **On multiple tasks:** We consider object detection, image classification and semantic segmentation.
2. **With shared layers:** We fine-tune the pretrained ConvNet weights to different extents, ranging from only the fully connected layers to fine-tuning everything (see [30] for a similar evaluation on ImageNet).
3. **With limited target task training data:** We reduce the amount of training data available for the target task.

4. Learning Features by Learning to Group

The core intuition behind this paper is that training a ConvNet to *group pixels in static images into objects without any class labels* will cause it to learn a strong, high-level feature representation. This is because such grouping is difficult from low-level cues alone: objects are typically made of multiple colors and textures and, if occluded, might even consist of spatially disjoint regions. Therefore, to effectively do this grouping is to implicitly *recognize* the object and understand its location and shape, even if it cannot be *named*. Thus, if we train a ConvNet for this task, we expect it to learn a representation that aids recognition.

To test this hypothesis, we ran a series of experiments using high-quality manual annotations on static images from COCO [27]. Although *supervised*, these experiments help to evaluate a) how well our method might work under ideal conditions, b) how performance is impacted if the segments are of lower quality, and c) how much data is needed. We now describe these experiments in detail.

4.1. Training a ConvNet to Segment Objects

We frame the task as follows: given an image patch containing a single object, we want the ConvNet to segment the object, *i.e*., assign each pixel a label of 1 if it lies on the object and 0 otherwise. Since an image contains multiple objects, the task is ambiguous if we feed the ConvNet the entire image. Instead, we sample an object from an image and crop a box around the ground truth segment. However, given a precise bounding box, it is easy for the ConvNet to cheat: a blob in the center of the box would yield low loss. To prevent such degenerate solutions, we jitter the box in position and scale. Note that a similar training setup was used for recent segmentation proposal methods [37, 38].

We use a straightforward ConvNet architecture that takes as input a $w \times w$ image and outputs an $s \times s$ mask. Our network ends in a fully connected layer with s^2 outputs followed by an element-wise sigmoid. The resulting s^2 dimensional vector is reshaped into an $s \times s$ mask. We also downsample the ground truth mask to $s \times s$ and sum the cross entropy losses over the s^2 locations to train the network.

4.2. Experiments

To enable comparisons to prior work on unsupervised learning, we use AlexNet [24] as our ConvNet architecture. We use $s = 56$ and $w = 227$. We use images and annotations from the trainval set of the COCO dataset [27], *discarding the class labels* and only using the segmentations.

Does training for segmentation yield good features? Following recent work on unsupervised learning, we perform experiments on the task of object detection on PASCAL VOC 2007 using Fast R-CNN [15].[1] We use multi-

[1] https://github.com/rbgirshick/py-faster-rcnn

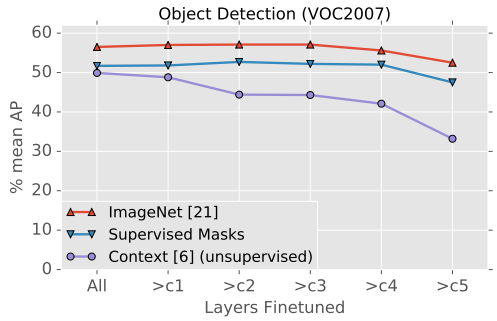


Figure 3. Our representation trained on manually-annotated segments from COCO (without class labels) compared to ImageNet pretraining and context prediction (unsupervised) [8], evaluated for object detection on PASCAL VOC 2007. '>cX': all layers above convX are fine-tuned; 'All': the entire net is fine-tuned.

scale training and testing [15]. In keeping with the motivation described in Section 3, we measure performance with ConvNet layers frozen to different extents. We compare our representation to a ConvNet trained on image classification on ImageNet, and the representation trained by Doersch *et al*. [8]. The latter is competitive with the state-of-the-art. (Comparisons to other recent work on unsupervised learning appear later.) The results are shown in Figure 3.

We find that our supervised representation outperforms the unsupervised context prediction model across all scenarios by a large margin, which is to be expected. Notably though, our model maintains a fairly small gap with ImageNet pretraining. This result is state-of-the-art for a model trained without semantic category labels. Thus, given high-quality segments, our proposed method can learn a strong representation, which validates our hypothesis.

Figure 3 also shows that the model trained on context prediction degrades rapidly as more layers are frozen. This drop indicates that the higher layers of the model have become overly specific to the *pretext* task [49], and may not capture the high-level concepts needed for object recognition. This is in contrast to the stable performance of the ImageNet trained model even when most of the network is frozen, suggesting the utility of its higher layers for recognition tasks. We find that this trend is also true for our representation: *it retains good performance even when most of the ConvNet is frozen*, indicating that it has indeed learned high-level semantics in the higher layers.

Can the ConvNet learn from noisy masks? We next ask if the quality of the learned representation is impacted by the quality of the ground truth, which is important since the segmentations obtained from unsupervised motion-based grouping will be imperfect. To simulate noisy segments, we train the representation with degraded masks from COCO. We consider two ways of creating noisy segments: introducing noise in the boundary and truncating the mask.

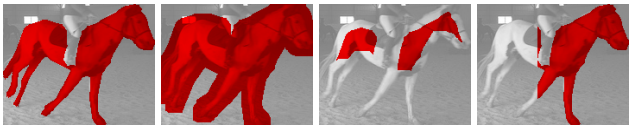


Figure 4. We degrade ground truth masks to measure the impact of segmentation quality on the learned representation. From left to right, the original mask, dilated and eroded masks (boundary errors), and a truncated mask (truncation can be on any side).

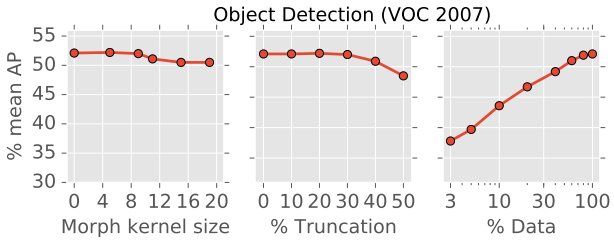


Figure 5. VOC object detection accuracy using our *supervised* ConvNet as noise is introduced in mask boundaries, the masks are truncated, or the amount of data is reduced. Surprisingly, the representation maintains quality even with large degradation.

Noise in the segment boundary simulates the foreground leaking into the background or vice-versa. To introduce such noise during training, for each cropped ground truth mask, we randomly either erode or dilate the mask using a kernel of fixed size (Figure 4, second and third images). The boundaries become noisier as the kernel size increases.

Truncation simulates the case when we miss a part of the object, such as when only part of the object moves. Specifically, for each ground truth mask, we zero out a strip of pixels corresponding to a fixed percentage of the bounding box area from one of the four sides (Figure 4, last image).

We evaluate the representation trained with these noisy ground truth segments on object detection using Fast R-CNN with all layers up to and including conv5 frozen (Figure 5). We find that the learned representation is surprisingly resilient to both kinds of degradation. Even with large, systematic truncation (up to 50%) or large errors in boundaries, the representation maintains its quality.

How much data do we need? We vary the amount of data available for training, and evaluate the resulting representation on object detection using Fast-RCNN with all conv layers frozen. The results are shown in the third plot in Figure 5. We find that performance drops significantly as the amount of training data is reduced, suggesting that good representations will need large amounts of data.

In summary, these results suggest that training for segmentation leads to strong features even with imprecise object masks. However, building a good representation requires significant amounts of training data. These observations strengthen our case for learning features in an unsupervised manner on large unlabeled datasets.

Figure 6. From left to right: a video frame, the output of uNLC that we use to train our ConvNet, and the output of our ConvNet. uNLC is able to highlight the moving object even in potentially cluttered scenes, but is often noisy, and sometimes fails (last two rows). Nevertheless, our ConvNet can still learn from this noisy data and produce significantly better and smoother segmentations.

5. Learning by Watching Objects Move

We first describe the motion segmentation algorithm we use to segment videos, and then discuss how we use the segmented frames to train a ConvNet.

5.1. Unsupervised Motion Segmentation

The key idea behind motion segmentation is that if there is a single object moving with respect to the background through the entire video, then pixels on the object will move differently from pixels on the background. Analyzing the optical flow should therefore provide hints about which pixels belong to the foreground. However, since only a part of the object might move in each frame, this information needs to be aggregated across multiple frames.

We adopt the NLC approach from Faktor and Irani [12]. While NLC is unsupervised with respect to video segmentation, it utilizes an edge detector that was trained on labeled edge images [39]. In order to have a purely unsupervised method, we replace the trained edge detector in NLC with unsupervised superpixels. To avoid confusion, we call our implementation of NLC as uNLC. First uNLC computes a per-frame saliency map based on motion by looking for either pixels that move in a mostly static frame or, if the frame contains significant motion, pixels that move in a direction different from the dominant one. Per-pixel saliency is then averaged over superpixels [1]. Next, a nearest neighbor graph is computed over the superpixels in the video using location and appearance (color histograms and HOG [6]) as features. Finally, it uses a nearest neighbor voting scheme to propagate the saliency across frames.

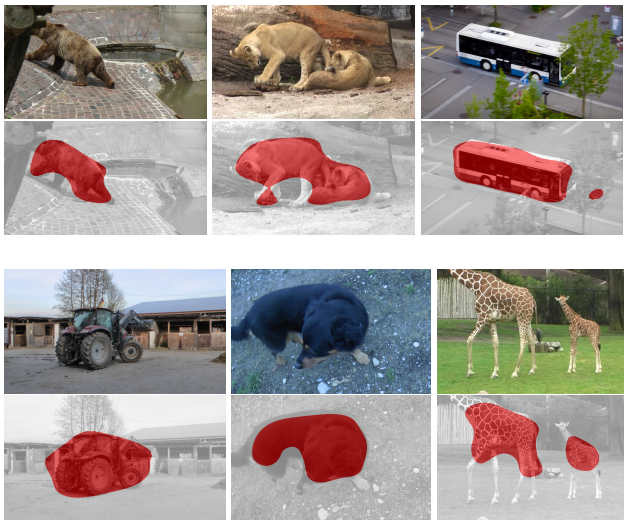


Figure 7. Examples of segmentations produced by our ConvNet on held out images. The ConvNet is able to identify the motile object (or objects) and segment it out from a single frame. Masks are not perfect but they do capture the general object shape.

We find that uNLC often fails on videos in the wild. Sometimes this is because the assumption of there being a single moving object in the video is not satisfied, especially in long videos made up of multiple shots showing different objects. We use a publicly available appearance-based shot detection method [40] (also unsupervised) to divide the video into shots and run uNLC separately on each shot.

Videos in the wild are also often low resolution and have compression artifacts, which can degrade the resulting segmentations. From our experiments using strong supervision, we know our approach can be robust to such noise. Nevertheless, since a large video dataset comprises a massive collection of frames, we simply discard badly segmented frames based on two heuristics. Specifically, we discard: (1) frames with too many (>80%) or too few (<10%) pixels marked as foreground; (2) frames with too many pixels (>10%) within 5% of the frame border that are marked as foreground. In preliminary tests, we found that results were not sensitive to the precise thresholds used.

We ran uNLC on videos from YFCC100m [43], which contains about 700,000 videos. After pruning, we ended up with 205,000 videos. We sampled 5-10 frames per shot from each video to create our dataset of 1.6M images, so we have slightly more frames than images in ImageNet. However, note that our frames come from fewer videos and are therefore more correlated than images from ImageNet.

We stress that our approach in generating this dataset is completely unsupervised, and does not use any form of supervised learning in any part of the pipeline. The code for the segmentation and pruning, together with our automatically generated dataset of frames and segments, will be made publicly available soon.

Our motion segmentation approach is far from state-of-the-art, as can be seen by the noisy segments shown in Figure 6. Nevertheless, we find that our representation is quite resilient to this noise (as shown below). As such, we did not aim to improve the particulars of our motion segmentation.

5.2. Learning to Segment from Noisy Labels

As before, we feed the ConvNet cropped images, jittered in scale and translation, and ask it to predict the motile foreground object. Since the motion segmentation output is noisy, we do not trust the absolute foreground probabilities it provides. Instead, we convert it into a *trimap* representation in which pixels with a probability <0.4 are marked as negative samples, those with a probability >0.7 are marked as positives, and the remaining pixels are marked as "don't cares" (in preliminary experiments, our results were found to be robust to these thresholds). The ConvNet is trained with a logistic loss only on the positive and negative pixels; don't care pixels are ignored. Similar techniques have been successfully explored earlier in segmentation [3, 22].

Despite the steps we take to get good segments, the uNLC output is still noisy and often grossly incorrect, as can be seen from the second column of Figure 6. However, if there are no *systematic* errors, then these motion-based segments can be seen as perturbations about a true latent segmentation. Because a ConvNet has finite capacity, it will not be able to fit the noise perfectly and might instead learn something closer to the underlying correct segmentation.

Some positive evidence for this can be seen in the output of the trained ConvNet on its training images (Fig. 6, third column). The ConvNet correctly identifies the motile object and its rough shape, leading to a smoother, more correct segmentation than the original motion segmentation.

The ConvNet is also able to *generalize* to unseen images. Figure 7 shows the output of the ConvNet on frames from the DAVIS [36], FBMS [31] and VSB [13] datasets, which were not used in training. Again, it is able to identify the moving object and its rough shape from just a single frame. When evaluated against human annotated segments in these datasets, we find that the ConvNet's output is significantly *better* than the uNLC segmentation output as shown below:

Metric	uNLC	ConvNet (unsupervised)
Mean IoU (%)	13.1	**24.8**
Precision (%)	15.4	**29.9**
Recall (%)	45.8	**59.3**

These results confirm our earlier finding that the ConvNet is able to learn well even from noisy and often incorrect ground truth. However, the goal of this paper is not segmentation, but representation learning. We evaluate the learned representation in the next section.

6. Evaluating the Learned Representation

6.1. Transfer to Object Detection

We first evaluate our representation on the task of object detection using Fast R-CNN. We use VOC 2007 for cross-validation: we pick an appropriate learning rate for each method out of a set of 3 values $\{0.001, 0.002 \text{ and } 0.003\}$. Finally, we train on VOC 2012 train and test on VOC 2012 val exactly once. We use multi-scale training and testing and discard difficult objects during training.

We present results with the ConvNet parameters frozen to different extents. As discussed in Section 3, a good representation should work well both as an initialization to fine-tuning and also when most of the ConvNet is frozen.

We compare our approach to ConvNet representations produced by recent prior work on unsupervised learning [2, 8, 10, 30, 33, 35, 46, 51]. We use publicly available models for all methods shown. Like our ConvNet representation, all models have the AlexNet architecture, but differ in minor details such as the presence of batch normalization layers [8] or the presence of grouped convolutions [51].

We also compare to two models trained with strong supervision. The first is trained on ImageNet classification. The second is trained on manually-annotated segments (without class labels) from COCO (see Section 4).

Results are shown in Figure 8(a) (left) and Table 1 (left). We find that our representation learned from unsupervised motion segmentation performs on par or better than prior work on unsupervised learning across all scenarios.

As we saw in Section 4.2, in contrast to ImageNet supervised representations, the representations learned by previous unsupervised approaches show a large decay in performance as more layers are frozen, owing to the representation becoming highly specific to the pretext task. Similar to our *supervised* approach trained on segmentations from COCO, we find that our *unsupervised* approach trained on motion segmentation also shows *stable* performance as the layers are frozen. Thus, unlike prior work on unsupervised learning, the upper layers in our representation learn high-level abstract concepts that are useful for recognition.

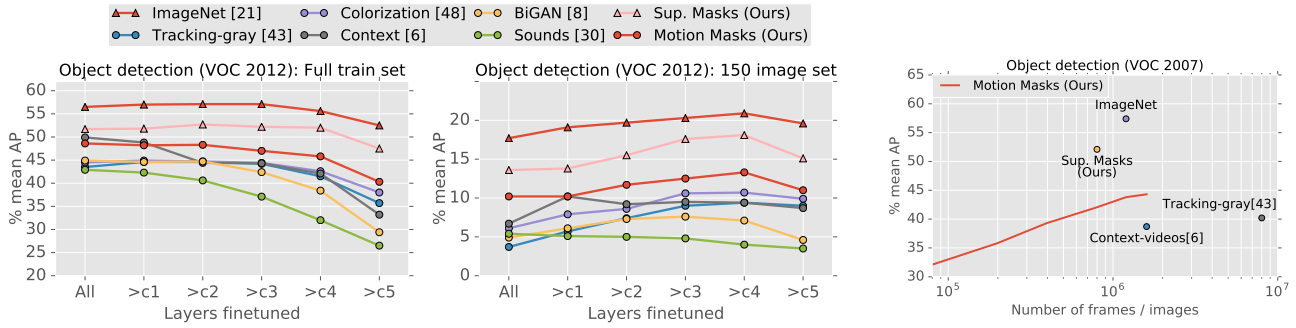
It is possible that some of the differences between our method and prior work are because the training data is from different domains (YFCC100m videos *vs*. ImageNet images). To control for this, we retrained the model from [8] on frames from our video dataset (see Context-videos in Table 1). The two variants perform similarly: 33.4% mean AP when trained on YFCC with conv5 and below frozen compared to 33.2% for the ImageNet version. This confirms that the different image sources do not explain our gains.

6.2. Low-shot Transfer

A good representation should also aid learning when training data is scarce, as we motivated in Section 3. Fig-

Method	Full train set All	>c1	>c2	>c3	>c4	>c5	150 image set All	>c1	>c2	>c3	>c4	>c5	#wins
Supervised													
Imagenet	56.5	57.0	57.1	57.1	55.6	52.5	17.7	19.1	19.7	20.3	20.9	19.6	NA
Sup. Masks (Ours)	51.7	51.8	52.7	52.2	52.0	47.5	13.6	13.8	15.5	17.6	18.1	15.1	NA
Unsupervised													
Jigsaw‡ [30]	49.0	50.0	48.9	47.7	45.8	37.1	5.9	8.7	8.8	10.1	9.9	7.9	NA
Kmeans [23]	42.8	42.2	40.3	37.1	32.4	26.0	4.1	4.9	5.0	4.5	4.2	4.0	0
Egomotion [2]	37.4	36.9	34.4	28.9	24.1	17.1	–	–	–	–	–	–	0
Inpainting [35]	39.1	36.4	34.1	29.4	24.8	13.4	–	–	–	–	–	–	0
Tracking-gray [46]	43.5	44.6	44.6	44.2	41.5	35.7	3.7	5.7	7.4	9.0	9.4	9.0	0
Sounds [33]	42.9	42.3	40.6	37.1	32.0	26.5	5.4	5.1	5.0	4.8	4.0	3.5	0
BiGAN [10]	44.9	44.6	44.7	42.4	38.4	29.4	4.9	6.1	7.3	7.6	7.1	4.6	0
Colorization [51]	44.5	44.9	44.7	44.4	42.6	38.0	6.1	7.9	8.6	10.6	10.7	9.9	0
Split-Brain Auto [52]	43.8	45.6	45.6	46.1	44.1	37.6	3.5	7.9	9.6	10.2	11.0	10.0	0
Context [8]	**49.9**	**48.8**	44.4	44.3	42.1	33.2	6.7	**10.2**	9.2	9.5	9.4	8.7	3
Context-videos† [8]	47.8	47.9	46.6	**47.2**	44.3	33.4	6.6	9.2	10.7	12.2	11.2	9.0	1
Motion Masks (Ours)	48.6	48.2	**48.3**	47.0	**45.8**	**40.3**	**10.2**	**10.2**	**11.7**	**12.5**	**13.3**	**11.0**	**9**

Table 1. Object detection AP (%) on PASCAL VOC 2012 using Fast R-CNN with various pretrained ConvNets. All models are trained on `train` and tested on `val` using consistent Fast R-CNN settings. '–' means training didn't converge due to insufficient data. Our approach achieves the best performance in the majority of settings. †Doersch *et al*. [8] trained their original context model using ImageNet images. The Context-videos model is obtained by retraining their approach on our video frames from YFCC. This experiment controls for the effect of the distribution of training images and shows that the image domain used for training does not significantly impact performance. ‡Noroozi *et al*. [30] use a more computationally intensive ConvNet architecture (>2× longer to finetune) with a finer stride at conv1, preventing apples-to-apples comparisons. Nevertheless, their model works significantly worse than our representation when either layers are frozen or in case of limited data and is comparable to ours when network is finetuned with full training data.



(a) Performance vs. Finetuning

(b) Performance vs. Data

Figure 8. Results on object detection using Fast R-CNN. (a) VOC 2012 object detection results when the ConvNet representation is frozen to different extents. We compare to other unsupervised and supervised approaches. **Left**: using the full training set. **Right**: using only 150 training images (note the different y-axis scales). (b) Variation of representation quality (mean AP on VOC 2007 object detection with conv5 and below frozen) with number of training frames. A few other methods are also shown. Context-videos [8] is the representation of Doersch *et al*. [8] retrained on our video frames. Note that most other methods in Table 1 use ImageNet as their train set.

ure 8(a) (right) and Table 1 (right) show how we compare to other unsupervised and supervised approaches on the task of object detection when we have few (150) training images. We observe that in this scenario it actually hurts to fine-tune the entire network, and the best setup is to leave some layers frozen. Our approach provides the best AP overall (achieved by freezing all layers up to and including conv4) among all other representations from recent unsupervised learning methods by a large margin. The performance in other low-shot settings is presented in Figure 10.

Note that in spite of its strong performance relative to prior unsupervised approaches, our representation learned without supervision on video trails both the strongly supervised mask and ImageNet versions by a significant margin. We discuss this in the following subsection.

6.3. Impact of Amount of Training Data

The quality of our representation (measured by Fast R-CNN performance on VOC 2007 with all conv layers frozen) grows roughly logarithmically with the number of

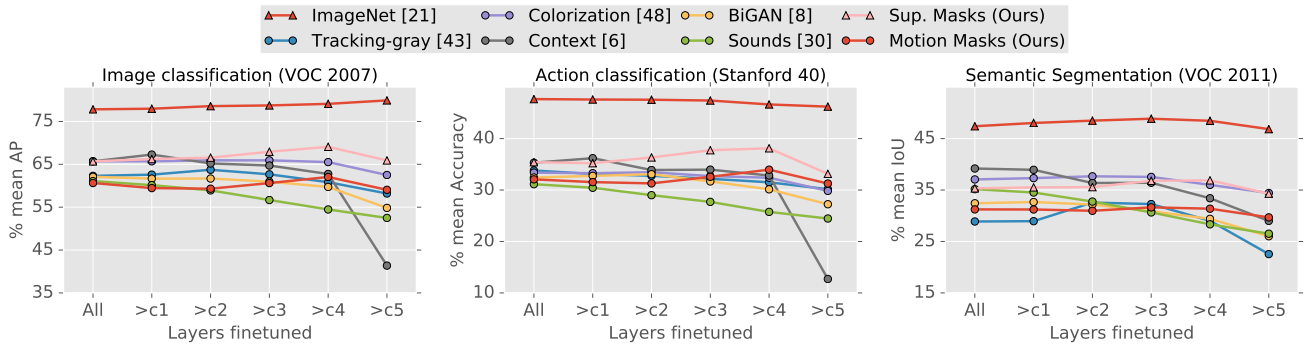


Figure 9. Results on image (object) classification on VOC 2007, single-image action classification on Stanford 40 Actions, and semantic segmentation on VOC 2011. Results shown with ConvNet layers frozen to different extents (note that the metrics vary for each task).

frames used. With 396K frames (50K videos), it is already better than prior state-of-the-art [8] trained on a million ImageNet images, see Figure 8(b). With our full dataset (1.6M frames) accuracy increases substantially. If this logarithmic growth continues, our representation will be on par with one trained on ImageNet if we use about 27M frames (or 3 to 5 million videos, the same order of magnitude as the number of images in ImageNet). Note that frames from the same video are very correlated. We expect this number could be reduced with more algorithmic improvements.

6.4. Transfer to Other Tasks

As discussed in Section 3, a good representation should generalize across tasks. We now show experiments for two other tasks: image classification and semantic image segmentation. For image classification, we test on both object and action classification.

Image Classification. We experimented with image classification on PASCAL VOC 2007 (object categories) and Stanford 40 Actions [48] (action labels). To allow comparisons to prior work [10, 51], we used random crops during training and averaged scores from 10 crops during testing (see [10] for details). We minimally tuned some hyperparameters (we increased the step size to allow longer training) on VOC 2007 validation, and used the same settings for both VOC 2007 and Stanford 40 Actions. On both datasets, we trained with different amounts of fine-tuning as before. Results are in the first two plots in Figure 9.

Semantic Segmentation. We use fully convolutional networks for semantic segmentation with the default hyperparameters [28]. All the pretrained ConvNet models are finetuned on union of images from VOC 2011 train set and additional SBD train set released by Hariharan *et al*. [18], and we test on the VOC 2011 val set after removing overlapping images from SBD train. The last plot in Figure 9 shows the performance of different methods when the number of layers being finetuned is varied.

Analysis. Like object detection, all these tasks require semantic knowledge. However, while in object detection the ConvNet is given a tight crop around the target object, the input in these image classification tasks is the entire image, and semantic segmentation involves running the ConvNet in a sliding window over all locations. This difference appears to play a major role. Our representation was trained on object crops, which is similar to the setup for object detection, but quite different from the setups in Figure 9. This mismatch may negatively impact the performance of our representation, both for the version trained on motion segmentation and the strongly supervised version. Such a mismatch may also explain the low performance of the representation trained by Wang *et al*. [46] on semantic segmentation.

Nevertheless, when the ConvNet is progressively frozen, our approach is a strong performer. When all layers until conv5 are frozen, our representation is better than other approaches on action classification and second only to colorization [51] on image classification on VOC 2007 and semantic segmentation on VOC 2011. Our higher performance on action classification might be due to the fact that our video dataset has many people doing various actions.

7. Discussion

We have presented a simple and intuitive approach to unsupervised learning by using segments from low-level motion-based grouping to train ConvNets. Our experiments show that our approach enables effective transfer especially when computational or data constraints limit the amount of task-specific tuning we can do. Scaling to larger video datasets should allow for further improvements.

We noted in Figure 6 that our network learns to refine the noisy input segments. This is a good example of a scenario where ConvNets can learn to extract signal from large amounts of noisy data. Combining the refined, single-frame output from the ConvNet with noisy motion cues extracted from the video should lead to better pseudo ground truth, and can be used by the ConvNet to bootstrap itself. We leave this direction for future work.

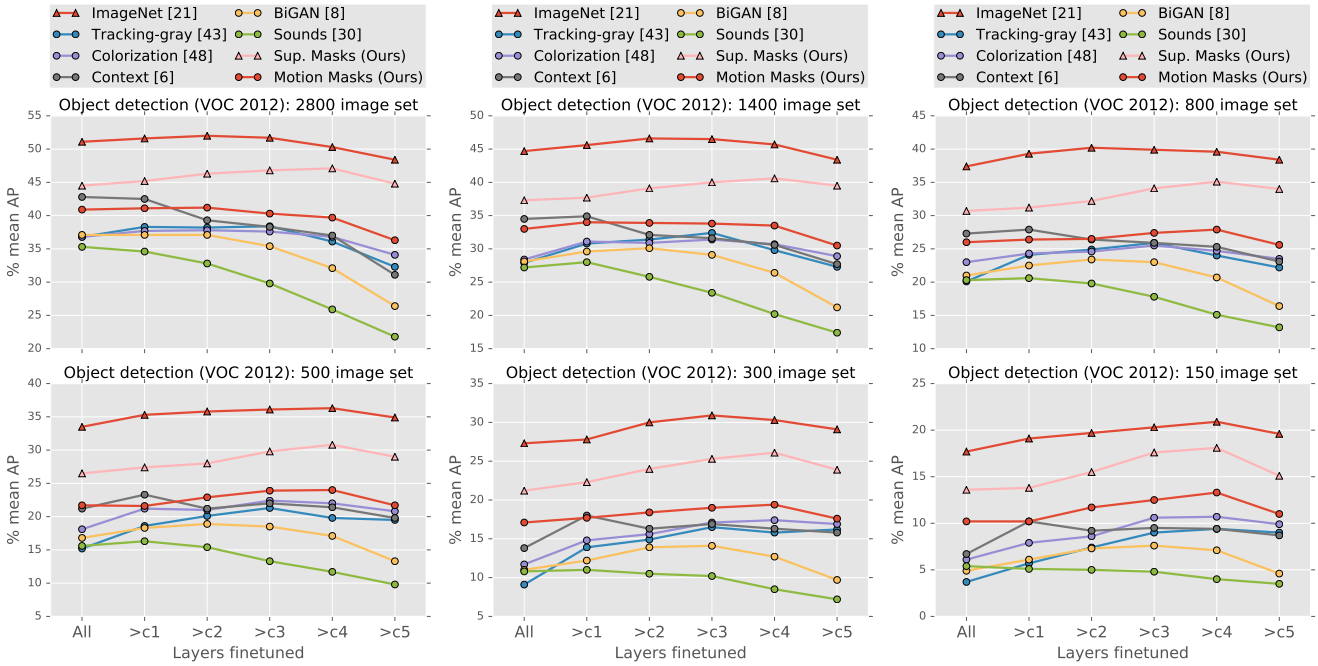


Figure 10. Results for object detection on Pascal VOC 2012 using Fast R-CNN and varying number of images available for finetuning. Each plot shows the comparison of different unsupervised learning methods as the number of layers being finetuned is varied. Different plots depict this variation for different amounts of data available for finetuning Fast R-CNN (please note the different y-axis scales for each plot). As the data for finetuning decreases, it is actually better to freeze more layers. Our method works well across all the settings and scales and as the amount of data decreases. When layers are frozen or data is limited, our method significantly outperforms other methods. This suggests that features learned in the higher layers of our model are good for recognition.

References

[1] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk. SLIC superpixels compared to state-of-the-art superpixel methods. *TPAMI*, 2012. 5

[2] P. Agrawal, J. Carreira, and J. Malik. Learning to see by moving. *ICCV*, 2015. 3, 6, 7

[3] C. Arteta, V. Lempitsky, and A. Zisserman. Counting in the wild. In *ECCV*, 2016. 6

[4] Y. Bengio. Learning deep architectures for AI. *Foundations and trends in Machine Learning*, 2009. 1, 2

[5] Y. Bengio, A. Courville, and P. Vincent. Representation learning: A review and new perspectives. *TPAMI*, 35(8), 2013. 2

[6] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. *CVPR*, 2005. 5

[7] V. R. de Sa. Learning classification with unlabeled data. *NIPS*, 1994. 2

[8] C. Doersch, A. Gupta, and A. A. Efros. Unsupervised visual representation learning by context prediction. *ICCV*, 2015. 1, 2, 3, 4, 6, 7, 8

[9] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. *ICML*, 2014. 1

[10] J. Donahue, P. Krähenbühl, and T. Darrell. Adversarial Feature Learning. *ICLR*, 2017. 2, 6, 7, 8

[11] V. Dumoulin, I. Belghazi, B. Poole, A. Lamb, M. Arjovsky, O. Mastropietro, and A. Courville. Adversarially learned inference. *ICLR*, 2017. 2

[12] A. Faktor and M. Irani. Video Segmentation by Non-Local Consensus voting. *BMVC*, 2014. 5

[13] F. Galasso, N. Nagaraja, T. Cardenas, T. Brox, and B. Schiele. A unified video segmentation benchmark: Annotation, metrics and analysis. *ICCV*, 2013. 6

[14] R. Garg, V. K. B.G., G. Carneiro, and I. Reid. Unsupervised cnn for single view depth estimation: Geometry to the rescue. *ECCV*, 2016. 3

[15] R. Girshick. Fast R-CNN. *ICCV*, 2015. 1, 3, 4

[16] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. *NIPS*, 2014. 2

[17] R. Goroshin, M. Mathieu, and Y. LeCun. Learning to linearize under uncertainty. *NIPS*, 2015. 1, 3

[18] B. Hariharan, P. Arbeláez, L. Bourdev, S. Maji, and J. Malik. Semantic contours from inverse detectors. *ICCV*, 2011. 8

[19] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik. Hypercolumns for object segmentation and fine-grained localization. *CVPR*, 2015. 1

[20] G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 2006. 1, 2

[21] D. Jayaraman and K. Grauman. Learning image representations tied to ego-motion. *ICCV*, 2015. 3

[22] P. Kohli, P. H. Torr, et al. Robust higher order potentials for enforcing label consistency. *IJCV*, 2009. 6

[23] P. Krähenbühl, C. Doersch, J. Donahue, and T. Darrell. Data-dependent initializations of convolutional neural networks. *ICLR*, 2016. 7

[24] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet classification with deep convolutional neural networks. *NIPS*, 2012. 3

[25] G. Larsson, M. Maire, and G. Shakhnarovich. Learning representations for automatic colorization. *ECCV*, 2016. 2

[26] Y. Li, M. Paluri, J. M. Rehg, and P. Dollár. Unsupervised learning of edges. *CVPR*, 2016. 3

[27] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft COCO: Common objects in context. *ECCV*, 2014. 2, 3

[28] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. *CVPR*, 2015. 8

[29] I. Misra, C. L. Zitnick, and M. Hebert. Shuffle and Learn: Unsupervised Learning using Temporal Order Verification. *ECCV*, 2016. 1, 3

[30] M. Noroozi and P. Favaro. Unsupervised Learning of Visual Representations by Solving Jigsaw Puzzles. *ECCV*, 2016. 1, 2, 3, 6, 7

[31] P. Ochs, J. Malik, and T. Brox. Segmentation of moving objects by long term video analysis. *TPAMI*, 36(6), 2014. 6

[32] Y. Ostrovsky, E. Meyers, S. Ganesh, U. Mathur, and P. Sinha. Visual parsing after recovery from blindness. *Psychological Science*, 2009. 1

[33] A. Owens, J. Wu, J. H. McDermott, W. T. Freeman, and A. Torralba. Ambient sound provides supervision for visual learning. *ECCV*, 2016. 2, 6, 7

[34] S. E. Palmer. *Vision science: Photons to phenomenology*. MIT press, 1999. 1

[35] D. Pathak, P. Krähenbühl, J. Donahue, T. Darrell, and A. Efros. Context Encoders: Feature Learning by Inpainting. *CVPR*, 2016. 2, 6, 7

[36] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. V. Gool, M. Gross, and A. Sorkine-Hornung. A Benchmark Dataset and Evaluation Methodology for Video Object Segmentation. *CVPR*, 2016. 6

[37] P. O. Pinheiro, R. Collobert, and P. Dollr. Learning to Segment Object Candidates. *NIPS*, 2015. 3

[38] P. O. Pinheiro, T.-Y. Lin, R. Collobert, and P. Dollár. Learning to Refine Object Segments. *ECCV*, 2016. 3

[39] L. Z. Piotr Dollár. Structured forests for fast edge detection. *ICCV*, 2013. 5

[40] D. Potapov, M. Douze, Z. Harchaoui, and C. Schmid. Category-specific video summarization. *ECCV*, 2014. 5

[41] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *IJCV*, 2015. 1

[42] E. S. Spelke. Principles of object perception. *Cognitive science*, 14(1), 1990. 1

[43] B. Thomee, D. A. Shamma, G. Friedland, B. Elizalde, K. Ni, D. Poland, D. Borth, and L.-J. Li. YFCC100M: The new data in multimedia research. *Communications of the ACM*, 59(2), 2016. 2, 5

[44] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol. Extracting and composing robust features with denoising autoencoders. *ICML*, 2008. 1, 2

[45] J. Walker, C. Doersch, A. Gupta, and M. Hebert. An uncertain future: Forecasting from static images using variational autoencoders. *ECCV*, 2016. 3

[46] X. Wang and A. Gupta. Unsupervised learning of visual representations using videos. *ICCV*, 2015. 1, 2, 6, 7, 8

[47] M. Wertheimer. Laws of organization in perceptual forms. 1938. 1

[48] B. Yao, X. Jiang, A. Khosla, A. L. Lin, L. Guibas, and L. Fei-Fei. Human action recognition by learning bases of action attributes and parts. *ICCV*, 2011. 8

[49] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson. How transferable are features in deep neural networks? *NIPS*, 2014. 4

[50] N. Zhang, J. Donahue, R. Girshick, and T. Darrell. Part-based R-CNNs for fine-grained category detection. *ECCV*, 2014. 1

[51] R. Zhang, P. Isola, and A. A. Efros. Colorful Image Colorization. *ECCV*, 2016. 2, 6, 7, 8

[52] R. Zhang, P. Isola, and A. A. Efros. Split-brain autoencoders: Unsupervised learning by cross-channel prediction. *CVPR*, 2017. 2, 7