Self-Supervised Relative Depth Learning for Urban Scene Understanding

Huaizu Jiang[1], Gustav Larsson[2], Michael Maire[2,3]
Greg Shakhnarovich[3], and Erik Learned-Miller[1]

[1] UMass Amherst {hzjiang, elm}@cs.umass.edu
[2] University of Chicago {larsson, mmaire}@cs.uchicago.edu
[3] TTI-Chicago gregory@ttic.edu

Abstract. As an agent moves through the world, the apparent motion of scene elements is (usually) inversely proportional to their depth.[4] It is natural for a learning agent to associate image patterns with the magnitude of their displacement over time: as the agent moves, faraway mountains don't move much; nearby trees move a lot. This natural relationship between the appearance of objects and their motion is a rich source of information about the world. In this work, we start by training a deep network, using fully automatic supervision, to predict relative scene depth from *single images*. The relative depth training images are automatically derived from simple videos of cars moving through a scene, using recent motion segmentation techniques, and no human-provided labels. The *proxy* task of predicting relative depth from a single image induces features in the network that result in large improvements in a set of downstream tasks including semantic segmentation, joint road segmentation and car detection, and monocular (absolute) depth estimation, over a network trained from scratch. The improvement on the semantic segmentation task is greater than that produced by any other automatically supervised methods. Moreover, for monocular depth estimation, our unsupervised pre-training method even outperforms supervised pre-training with ImageNet. In addition, we demonstrate benefits from learning to predict (again, completely unsupervised) relative depth in the specific videos associated with various downstream tasks (*e.g.*, KITTI). We adapt to the specific scenes in those tasks in an unsupervised manner to improve performance. In summary, for semantic segmentation, we present state-of-the-art results among methods that do not use supervised pre-training, and we even exceed the performance of supervised ImageNet pre-trained models for monocular depth estimation, achieving results that are comparable with state-of-the-art methods.

Keywords: self-supervised learning, unsupervised domain adaptation, urban scene understanding, semantic segmentation, monocular depth estimation

1 Introduction

How does a newborn agent learn about the world? When an animal (or robot) moves, its visual system is exposed to a shower of information. Usually, the speed with which

[4] Strictly speaking, this statement is true only after one has compensated for camera rotation, individual object motion, and image position. We address these issues in the paper.

(a) (b) (c) (d) (e)

Fig. 1: Sample frames from collected videos and their corresponding relative depth maps, where brightness encodes relative depth (the brighter the farther). From top to bottom: input image, relative depth image computed using Eq.(3), and predicted (relative) depth maps using our trained VGG16 FCN8s [1,2]. There is often a black blob around the center of the image, a singularity in depth estimation caused by the focus of expansion. (a)(b)(c): images from the CityDriving dataset, (d): images from the KITTI dataset, and (e): images from the CityScapes dataset.

something moves in the image is inversely proportional to its depth.[1] As an agent continues to experience visual stimuli under its own motion, it is natural for it to form associations between the appearance of objects and their relative motion in the image. For example, an agent may learn that objects that look like mountains typically don't move in the image (or change appearance much) as the agent moves. Objects like nearby buildings and bushes, however, appear to move rapidly in the image as the agent changes position relative to them. This continuous pairing of images with motion acts as a kind of automatic supervision that could eventually allow an agent both to understand the depth of objects and to group pixels into objects by this predicted depth. Thus, by moving through the world, an agent may learn to predict properties (such as depth) of *static* scenes.

A flurry of recent work has shown that *proxy tasks* (also known as *pretext* or *surrogate tasks*) such as colorization [3,4], jigsaw puzzles [5], and others [6,7,8,9,10,11,12,13], can induce features in a neural network that provide strong pre-training for subsequent tasks. In this paper, we introduce a new proxy task: estimation of relative depth from a single image. We show that a network that has been pre-trained, without human supervision, to predict relative scene depth provides a powerful starting point from which to fine-tune models for a variety of urban scene understanding tasks. Not only does this automatically supervised starting point outperform all other proxy task pre-training methods. For monocular depth understanding, it even performs better than the heavily supervised ImageNet pre-training, yielding results that are comparable with state-of-the-art methods.

To estimate relative scene depths without human supervision, we use a recent motion segmentation technique [14] to estimate relative depth from geometric constraints between a scene's motion field and the camera motion. We apply it to simple, publicly available YouTube videos taken from moving cars. Since this technique estimates depth *up to an unknown scale factor*, we compute *relative depth* of the scene during the

pre-training phase, where each pixel's value is in the range of $[0, 1]$ denoting its depth percentile over the entire image.[5]

Unlike work that analyzes video paired with additional information about direction of motion [15], our agent learns from "raw egomotion" video recorded from cars moving through the world. Unlike methods that require videos of moving objects [8], we neither depend on, nor are disrupted by, moving objects in the video. Once we have relative depth estimates for these video images, we train a deep network to predict the relative depth of each pixel from a *single image*, *i.e.*, to predict the relative depth *without* the benefit of motion. One might expect such a network to learn that an image patch that looks like a house and spans 20 pixels of an image (about 100 meters away) is significantly farther away than a pedestrian that spans 100 image pixels (perhaps 10 meters away). Figure 1 illustrates this prediction task and shows sample results obtained using a standard convolutional neural network (CNN) in this setting. For example, in the leftmost image of Fig. 1, an otherwise unremarkable traffic-light pole is clearly highlighted by its relative depth profile, which stands out from the background. Our hypothesis is that to excel at relative depth estimation, the CNN will benefit by learning to recognize such structures.

The goal of our work is to show that pre-training a network to do relative depth prediction is a powerful proxy task for learning visual representations. In particular, we show that a network pre-trained for relative depth prediction (from automatically generated training data) improves training for downstream tasks including semantic segmentation, joint semantic reasoning of road segmentation and car detection, and monocular (absolute) depth estimation. We obtain significant performance gains on urban scene understanding benchmarks such as KITTI [16,17] and CityScapes [18], compared to training a segmentation model from scratch. Compared to nine other proxy tasks for pre-training, our proxy task consistently provides the highest gains when used for pre-training. In fact, our performance on semantic segmentation and joint semantic reasoning tasks comes close to that of equivalent architectures pre-trained with ImageNet [19], a massive labeled dataset. Finally, for the monocular (absolute) depth estimation, *our pre-trained model achieves better performance than an ImageNet pre-trained model*, using both VGG16 [1] and ResNet50 [20] architectures.

As a final application, we show how our proxy task can be used for *domain adaptation*. One might assume that the more similar the domain of unlabeled visual data used for the proxy task (here, three urban scene understanding tasks) is to the domain in which the eventual semantic task is defined (here, semantic segmentation), the better the representation learned by pre-training. This observation allows us to go beyond simple pre-training, and effectively provide a domain adaptation mechanism. By adapting (fine-tuning) a relative depth prediction model to targets obtained from unlabeled data in a novel domain (say, driving in a new city) we can improve the underlying representation, priming it for better performance on a semantic task (*e.g.*, segmentation) trained with a small labeled dataset from this new domain. In experiments, we show that pre-training on unlabeled videos from a target city, absent any labeled data from that city, consistently improves all urban scene understanding tasks.

[5] Later, we will fine-tune networks to produce absolute depths.

In total, our work advances two pathways for integrating unlabeled data with visual learning.

- We propose a novel proxy task for self-supervised learning of visual representations; it is based on learning to predict relative depth, inferred from unlabeled videos. This unsupervised pre-training leads to better results over all other proxy tasks on the semantic segmentation task, and even outperforms supervised ImageNet pre-training for absolute depth estimation.
- We show that our task can be used to drive domain adaptation. Experiments demonstrate its utility in scene understanding tasks for street scenes in a novel city. Our adapted model achieves results that are competitive with state-of-the-art methods (including those that use *large supervised pre-training*) on the KITTI depth estimation benchmark.

Such methods of extracting knowledge from unlabeled data are likely to be increasingly important as computer vision scales to real-world applications; here massive dataset size can outpace herculean annotation efforts.

2 Related Work

Self-supervised learning. The idea of formulating supervised prediction tasks on unlabeled data has been leveraged for both images and videos. The idea, often called self-supervision, is most typically realized by removing part of the input and then training a network to predict it. This can take the form of deleting a spatial region and trying to inpaint it [10], draining an image of color and trying to colorize it [21,4,3], or removing the final frame in a sequence and trying to hallucinate it [22,23,24,25,26,27]. Generative Adversarial Networks, used for inpainting and several future frame prediction methods, can also be used to generate realistic-looking samples from scratch. This has found secondary utility for unsupervised representation learning [28,29,30]. Another strategy is to extract patches and try to predict their spatial or temporal relationship. In images, this has been done for pairs of patches [31] or for 3-by-3 jigsaw puzzles [5]. In videos, it can be done by predicting the temporal ordering of frames [9,13]. The correlation of frames in video is also a rich source of self-supervised learning signals. The assumption that close-by frames are more similar than far apart frames can be used to train embeddings on pairs [32,33,34] or triplets [6] of frames. A related idea that the representation of interesting objects should change slowly through time dates back to Slow Feature Analysis [35].

The works most closely related to ours may be [15,7,8], which aim to learn useful visual representations from unlabeled videos as well. Jayaraman & Grauman [15] learn a representation equivariant to ego-motion transformations, using ideas from metric learning. Agrawal *et al.* [7] concurrently developed a similar method that uses the ego-motion directly as the prediction target as opposed to as input to an equivariant transformation. Both of these works assume knowledge of the agent's own motor actions, which limits their evaluation in sample size due to lack of publicly available data. In our work, the ego-motion is inferred through optical flow, which means we can leverage large sources of crowd-sourced data, such as YouTube videos. Pathak *et al.* [8] use

optical flow and a graph-based algorithm to produce unsupervised segmentation maps. A network is then trained to approximate these maps, driving representation learning. The reliance on moving objects, as opposed to a moving agent, could make it harder to collect good data. Using a method based on ego-motion, the agent can promote its own representation, learning simply by moving, instead of having to find objects that move.

There is also work on using multi-modal sensory input as a source of supervision. Owens *et al.* [12] predict statistics of ambient sounds in videos. Beyond studying a single source of self-supervision, combining multiple self-supervision sources is increasingly popular. In [36], a set of self-supervision tasks are integrated via a multi-task setting. Wang *et al.* [37] propose to combine instance-level as well as category-level self-supervision. Both [36,37] achieve better performance than a single model.

Unsupervised learning of monocular depth estimation. A single-image depth predictor can be trained from raw stereo images, by warping the right image with a depth map predicted from the left and training it to reconstruct the left image [38,39]. This idea was extended in recent work to support fully self-supervised training on regular video, by predicting both depth and camera pose difference for pairs of nearby frames [40,41].

Although [40,41] are closely related to our work in the sense of unsupervised (or self-supervised) learning of depth and ego-motion from unlabeled videos, our work differs from them in two ways. First, neither of these two works emphasizes more general-purpose feature learning. Second, neither of them demonstrates their scalability to large-scale YouTube videos. [40] requires intrinsic camera parameters that are not available for most YouTube videos; our approach relies on optical flow only. [41] only reports experimental results on standard benchmark datasets, whose scale is an order of magnitude smaller than videos we use. It is unclear whether the heuristics (*e.g.*, the manually set camera intrinsic parameters, number of motion clusters) are robust to YouTube videos in the wild.

3 Inducing features by learning to estimate relative depth

As a proxy task, our goal is to induce a feature representation $f(I)$ of an RGB image $I(x, y)$ by predicting its depth image $z(x, y)$, where the representation $f(I)$ could be transferred to other downstream tasks (*e.g.*, semantic segmentation) with fine-tuning. In section 3.1, we introduce technical details of gathering images and corresponding depth maps. In section 3.2, we provide details of training CNNs to learn the feature representation $f(I)$.

3.1 Self-Supervised Relative Depth

As described above, we automatically produce depth images for video frames by analyzing the motion of pre-existing videos. In our experiments, we used three sets of videos: YouTube videos, videos from the KITTI database [16,17], and videos from the CityScapes database [18]. The YouTube videos consist of 135 videos taken from moving cars in major U.S. cities.[6] We call this dataset CityDriving. The stability of the

[6] They are crawled from a YouTube playlist, taking less than an hour.

camera in these videos makes them relatively easy for the depth estimation procedure. Some of the videos are extremely long, lasting several hours. The CityDriving dataset features a large number of man-made structures, pedestrians and cars. Following [42], we only keep two consecutive frames if they have moderate motion (*i.e.*, neither too slow nor too fast). To eliminate near duplicate frames, two consecutive depth maps must be at least 2 frames apart. We keep only the first one of two consecutive frames and the computed depth image. In total, we gathered 1.1M pairs of RGB images and their corresponding depth maps, where the typical resolution is 640×360. Similarly, we collect 30K and 24K pairs of RGB images and their relative depth maps for CityScapes and KITTI, respectively.

Denote the instantaneous coordinates of a point P in the environment by $(X, Y, Z)^T$, and the translational velocity of the camera in the environment by $(U, V, W)^T$. Let the motion field component (idealized optical flow) of the point P (in the image plane) be (u, v), corresponding to the horizontal and vertical image motion, respectively. The motion field can be written as the sum of translation and rotation components[7]

$$u = u_t + u_r, \quad v = v_t + v_r, \tag{1}$$

where the subscript t and r denote translation and rotation, respectively. According to the geometry of perspective projection [43], the following equations hold if the motion of the camera is purely translational,

$$u_t = \frac{-U + xW}{Z}, \quad v_t = \frac{-V + yW}{Z}, \tag{2}$$

where x and y are the coordinates of the point P in the image plane (the origin is at the image center).

Note that the depth Z can be estimated from either one of these equations. However, the estimate can be unstable if either u_t or v_t is small. To obtain a more robust estimate of Z, we square the two equations above and add them:

$$Z = \sqrt{\frac{(-U + xW)^2 + (-V + yW)^2}{u_t^2 + v_t^2}}. \tag{3}$$

Because we can only recover $(U, V, W)^T$ up to scale (see below), we can only compute the depth map of an image up to scale. To induce feature representations, we use depth orderings of pixels in an image. We compute the relative depth $z \in [0, 1]$ of the pixel P as its depth percentile (divided by 100) across all estimated depth values for the image. Since these percentiles are invariant to the velocity's unknown scale, we do not need to recover the absolute scale of velocity. Examples of these automatically obtained depth maps are given in Figure 1 and Figure 2.

To compute the optical flow, we use the state-of-the-art unsupervised method [44]. It first computes sparse pixel matchings between two video frames. It then interpolates to get dense pixelwise optical flow fields from sparse matchings, where we replace the

[7] Any motion in the image is due to the relative motion of a world point and the camera. This addresses motion of the object, the camera, or both.

Fig. 2: Samples of image pairs and computed translational optical flow that we use to recover the relative depth. From left to right: first images, second images, translational optical flow between input two images, and relative depth of the first images.

supervised edge detector [45] with its unsupervised version [42]. Based on the optical flow, we use the method proposed in [14] to recover the image motion of each pixel due to translational motion only (u_t, v_t), and also, the global camera motion $(U, V, W)^T$ up to an unknown scale factor. Specifically, the rotation of the camera can be estimated by finding the rotation such that the remaining motion, by removing the rotational component from the motion field, can be well-explained by angle fields, which are the angle part of the motion field. This procedure produces a translational optical flow field (u_t, v_t), and a set of regions in the image corresponding to the background and different object motions, along with the motion directions (U, V, W) of those regions. We refer readers to [14] for more technical details.

In summary, to obtain the depth map of each frame from a video, we:

- compute the optical flow (u, v) between a pair of frames [44];
- estimate the translational component (u_t, v_t) of the optical flow and the direction of camera translation (U, V, W) from the optical flow, using the method of [14];
- estimate the scene depth Z using Eq. 3, up to an unknown scale factor, from the translational component of the optical flow and the camera direction estimate and convert it to relative depth $z \in [0, 1]$.

3.2 Predicting Relative Depth From a Single Image

While a CNN for predicting depth from a single image is a core component of our system, we are primarily interested in relative depth prediction as a proxy task, rather than an end in itself. We therefore select standard CNN architectures and focus on quantifying the power of the depth task for pre-training and domain adaptation, compared to using the same networks with labeled data. Specifically, we work with variants of the standard AlexNet [46], VGG16 [1], and ResNet50 [20] architectures.

Given an RGB image I, we need pixelwise predictions in the form of a depth image z, so we modify both AlexNet, VGG16, and ResNet50 to produce outputs with the same

spatial resolution as the input image. In particular, we consider Fully Convolutional Networks (FCNs) [2] and an encoder-decoder with skip connections [39,40]. Detailed discussions can be found in the experiment section.

Since the relative depth (*i.e.*, the depth percentile) is estimated over the entire image, it is essential to feed the entire image to the CNN to make a prediction. For CityDriving and KITTI, we simply resize the input image to 224×416 and 352×1212. For CityScapes, we discard the bottom 20% portion or so of each video frame containing mainly the hood of a car, which remains static over all videos and makes the relative depth estimation inaccurate (recall our relative depth estimation is mainly based on motion information). The cropped input image is then resized to 384×992. During training, we employ horizontal flipping and color jittering for data augmentation. Since relative depth serves as a proxy, rather than an end task, even though the relative depth estimation is not always correct, the network is able to tolerate some degree of noise as shown in [8]; we can then repurpose the network's learned representation.

In all experiments, we use L_1 loss for each pixel when training for depth prediction, *i.e.*, we train networks to regress the relative depth values. All AlexNet, VGG16, and ResNet50 variants are trained for 30 epochs using the Adam optimizer [47] with momentum of $\beta_1 = 0.9, \beta_2 = 0.999$, and weight decay of 0.0005. The learning rate is 0.0001 and is held constant during the pre-training stage.

4 Experiments

We consider three urban scene understanding tasks: semantic segmentation, joint semantic reasoning consisting of road segmentation and object detection [48], and monocular absolute depth estimation.

4.1 Semantic Segmentation

We consider three datasets commonly used for evaluating semantic segmentation. Their main characteristics are summarized below:

KITTI [49]: 100 training images, 46 testing images, spatial dimensions of 370×1226, 11 classes.

CamVid [50,51]: 367 training images, 101 validation images, 233 testing images, spatial dimensions of 720×960, 11 classes.

CityScapes [18]: 2975 training images, 500 validation images, 1525 testing images, spatial dimensions of 1024×2048, 19 classes. We conduct experiments on images at half resolution.

The first two datasets are much too small to provide sufficient data for "from scratch" training of a deep model; CityScapes is larger, but we show below that all three datasets benefit from pre-training. We use the curated annotations of the CamVid dataset released by [53]. As a classical CNN-based model for semantic segmentation, we report results of different variants of the Fully Convolutional Network (FCN) [2].

We compare our results to those obtained with other self-supervision strategies surveyed in Section 2. Since only AlexNet pre-trained models are available for most of the previous self-supervised methods, we also train an AlexNet. During training, the inputs

Table 1: Comparisons of mean IoU scores of AlexNet FCN32s for semantic segmentation using different self-supervised models. CS=CityScapes, K=KITTI, CV=CamVid.

pre-training method	supervision source	CS	K	CV
supervised	ImageNet labels	**48.1**	**46.2**	**57.4**
none	-	40.7	39.6	44.0
tracking [6]	motion	41.9	42.1	50.5
moving [7]	ego-motion	41.3	40.9	49.7
watch-move [8]	motion seg.	41.5	40.8	51.7
frame-order [9]	motion	41.5	39.7	49.6
context [10]	appearance	39.7	-	37.8
object-centric [11]	appearance	39.6	39.1	48.0
colorization [21,3]	appearance (color)	42.9	35.8	53.2
cross-channel [52]	misc.	36.8	40.8	46.3
audio [12]	video soundtrack	39.6	40.7	51.5
Ours	depth	**45.4**	**42.6**	**53.4**

are random crops of 352×352 for KITTI and 704×704 for CamVid. Each FCN32s using different pre-training models is trained for 600 epochs with a batch size of 16 using 4 GPUs. For CityScapes, the inputs to the network are random crops of 512×512. Each FCN32s is trained for 400 epochs with a batch size of 16. In addition to the random crops, random horizontal flips and color jittering are also performed. The CNNs at this stage (learning segmentation) are trained or fine-tuned using the Adam optimizer, where weight decay is 0.0005. For the learning rate, we use 0.0001 and decrease it by a factor of 10 at the 400th epoch (300th epoch for CityScapes).

Quantitative comparisons can be found in Table 1.[8] Our pre-trained model performs significantly better than the model learned from scratch on all three datasets, validating the effectiveness of our pre-training. Moreover, we obtain new state-of-the-art results on all three urban scene segmentation datasets among methods that use self-supervised pre-training. In particular, our model outperforms all other self-supervised models with motion cues (the first four self-supervised models in Table 1).

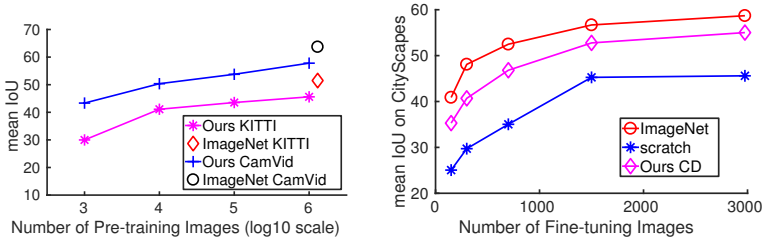
4.2 Ablation Studies

We perform ablation studies using VGG16 FCN32s on the semantic segmentation datasets. Specifically, we study the following aspects.

Number of pre-training images. Figure 3(a) demonstrates that the performance of our depth pre-trained model scales linearly with the log of the number of pre-training images on CamVid, which is similar to the conclusion of [8].

On KITTI, our pre-trained model initially has a big performance boost when the number of pre-training images increases from 1K to 10K. With enough data (more than

[8] We were unable to get meaningful results with [10] on KITTI and with [15] on all three segmentation datasets.



(a) (b)

Fig. 3: Ablation studies of performance by (a) varying number of pre-training images on KITTI and CamVid, (b) varying number of fine-tuning images of CityScapes.

Table 2: Mean IoU scores of semantic segmentation using different architectures on different datasets. CD=CityDriving, CS=CityScapes, CV=CamVid, and K=KITTI.

pre-training	FCN32s			FCN16s			FCN8s		
	CS	CV	K	CS	CV	K	CS	CV	K
ImageNet	**58.7**	**63.7**	**51.5**	**62.9**	**65.9**	**55.3**	**63.4**	**67.0**	**56.4**
scratch	45.4	41.0	32.4	51.3	44.1	33.1	51.6	44.3	34.2
Ours CD	55.0	57.8	45.6	57.6	**59.0**	47.7	59.8	**60.3**	48.6
Ours CD+K	56.0	**58.5**	46.0	56.9	58.8	**48.2**	58.9	60.1	49.0
Ours CD+CS	**56.2**	**58.5**	**47.4**	**58.5**	58.8	47.8	**60.5**	59.9	**49.6**

10K), the performance also scales linearly with the log of the number of pre-training images.

Number of fine-tuning images. Figure 3(b) shows that every model (ImageNet, scratch, our depth pre-trained model) benefits from more fine-tuning data on the CityScapes dataset. For both ImageNet and our depth pre-trained models, it suggests that more fine-tuning data is also beneficial for transferring the previously learned representations to a new task.

4.3 Domain Adaptation by Pre-Training

In the experiments described above, the two stages (pre-training on self-supervised depth prediction, followed by supervised training for segmentation) rely on data that come from significantly different domains. The self-supervised learning uses videos obtained from moving through North American cities. In contrast, none of the target dataset images were collected in the same geographic locations. For instance, CityScapes includes data from driving in German cities. Thus, in addition to a shift in task, the fine-tuning of the network for segmentation must also deal with a *domain shift* in the input.

CityScapes [18] and KITTI [16] make available video sequences that give temporal context to every image in the dataset. None of these extra frames are labeled, but we can leverage them in the following way. Before training the network on segmentation, we fine-tune it, using the same self-supervised relative depth prediction task described

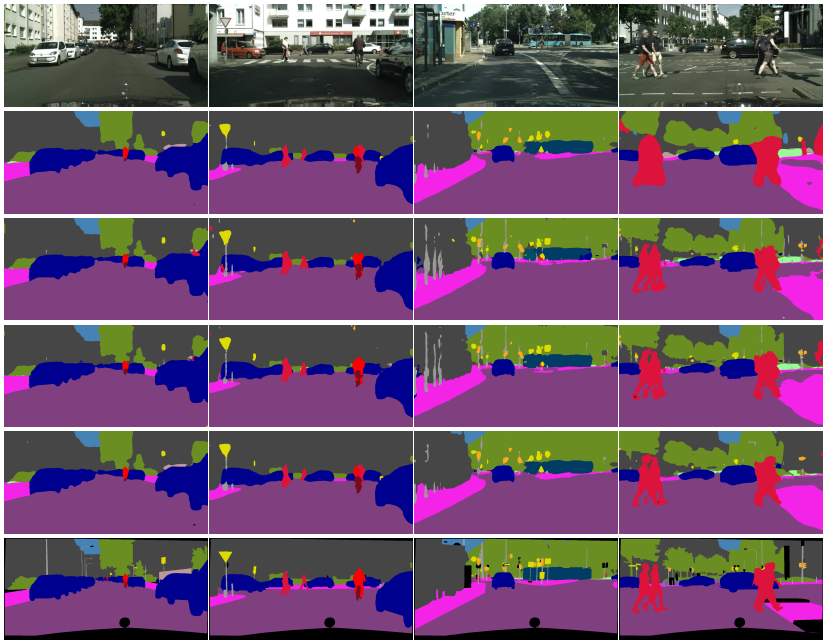


Fig. 4: Qualitative semantic segmentation results on CityScapes. From top to bottom: input images, predictions of FCN8s with no pre-training, our FCN8s pre-trained on CityDriving, our FCN8s pre-trained on CityDriving adapted to CityScapes, ImageNet FCN8s, and ground-truth annotations. The difference between the 2nd and 3rd rows shows a clear benefit of pre-training with relative depth prediction. The difference between 3rd rows and 4th rows shows the benefit our unsupervised domain adaptation using pre-training.

in Section 3.1, on these videos. Our intuition is that this may induce some of the modifications in the network that reflect the changing distribution of the input. Then, we proceed as before to train the fine-tuned representation on the semantic segmentation data. Specifically, we fine-tune different FCNs variants based on VGG16 sequentially, *i.e.*, from FCN32s to FCN16s, and finally to FCN8s. For FCN32s, the training procedure is identical to AlexNet FCN32s described earlier. FCN16s and FCN8s are trained for the same number of epochs as FCN32s, where the learning rate is set to 0.00002 and 0.00001, respectively, and kept constant during training.

The effectiveness of our unsupervised domain adaptation for semantic segmentation can be found in Table 2. The last two rows of demonstrate that such fine-tuning can consistently improve the performance of a self-supervised model over all FCN variants on both CityScapes and KITTI, validating its effectiveness as a domain adaptation approach. Interestingly, we can see that while fine-tuning is helpful for FCN32s on CamVid initially, it does not help much for FCN16s and FCN8s. Perhaps this is due to the domain gap between CamVid and CityScapes/KITTI. Qualitative semantic segmentation results can be found in Fig. 4.

Table 3: Results of joint semantic reasoning, including road segmentation and car detection.

pre-training	Road Segmentation		Car Detection (AP)		
	F_1	AP	Easy	Medium	Hard
ImageNet	**96.33**	**92.26**	**95.59**	**86.43**	**72.28**
scratch	93.78	91.37	89.37	79.93	66.02
Ours CD	94.74	92.13	92.84	84.73	69.47
Ours CD+K	**95.66**	**92.14**	**94.31**	**85.72**	**70.50**

4.4 Joint Semantic Reasoning

Joint semantic reasoning is important for urban scene understanding, especially with respect to tasks such as autonomous driving [48]. We investigate the effectiveness of our pre-trained model and the unsupervised strategy of domain adaptation using the MultiNet architecture [48] for joint road segmentation and car detection.[9] MultiNet consists of a single encoder, using the VGG16 as backbone, and two sibling decoders for each task. For road segmentation, the decoder contains three upsampling layers, forming an FCN8s. The car detection decoder directly regresses the coordinates of objects. Following [48], the entire network is jointly trained using the Adam optimizer, using a learning rate of 0.00005 and weight decay of 0.0005 for 200K steps. We refer readers to [48] for more technical details.

We replace the ImageNet-trained VGG16 network with a randomly initialized one and our own VGG16 pre-trained on CityDriving using relative depth. For the road segmentation task, there are 241 training and 48 validation images. For car detection, there are 7K training images 481 validation images. Detailed comparisons on the validation set can be found in Table 3. We use the F_1 measure and Average Precision (AP) scores for road segmentation evaluation and AP scores for car detection. AP scores for different car categories are reported separately. We can clearly see that our pre-trained model (Ours CD) consistently outperforms the randomly initialized model (scratch in Table 3). Furthermore, by using the domain adaptation strategy via fine-tuning on the KITTI raw videos (Ours CD+K), we can further close the gap between an ImageNet pre-trained model. Remarkably, after fine-tuning, the F_1 score of road segmentation and AP scores for easy and medium categories of our pre-trained model are pretty close to the ImageNet counterpart's. (See last row of Table 3.)

4.5 Monocular Absolute Depth Estimation

For the monocular absolute depth estimation, we adopt the U-Net architecture [54] similar to [39,40], which consists of a fully convolutional encoder and another fully convolutional decoder with skip connections. In order to use an ImageNet pre-trained

[9] We use the author's released code https://github.com/MarvinTeichmann/MultiNet. As the scene classification data is not publicly available, we only study road segmentation and car detection here.

Table 4: Monocular depth estimation on the KITTI dataset using the split of Eigen *et al.* [55] (range of 0-80m). For model details, Arch.=Architecture, A=AlexNet, V=VGG16, and R=ResNet50. For training data, Class.=classification, I=ImageNet, CD=CityDriving, K=KITTI, CS=CityScapes. **pp** indicates test-time augmentation by horizontally flipping the input image.

Method	Arch.	Training Data				Error Metrics				Accuracy Metrics		
		Class.	Stereo	Video	GT	Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
[55]	A	I	-	-	K	0.203	1.548	6.307	0.282	0.702	0.890	0.958
[56]	A	I	-	-	K	0.202	1.614	6.523	0.275	0.678	0.895	0.965
[39]+**pp**	R	-	CS+K	-	-	0.114	0.898	4.935	0.206	0.861	0.949	0.976
[40]	V	-	-	CS+K	-	0.198	1.836	6.565	0.275	0.718	0.901	0.960
[57]	R	I	K	-	K	**0.113**	**0.741**	**4.621**	**0.189**	**0.862**	**0.960**	**0.986**
Ours	V	I	-	-	K	0.157	1.115	5.546	0.233	0.768	0.922	0.974
Ours	V	-	-	-	K	0.163	1.241	5.649	0.238	0.765	0.918	0.970
Ours	V	-	-	CD	K	0.154	1.117	5.499	0.228	0.775	0.928	0.976
Ours	V	-	-	CD+K	K	**0.148**	**1.056**	**5.317**	**0.221**	**0.791**	**0.932**	**0.977**
Ours	R	I	-	-	K	0.128	0.933	5.073	0.203	0.827	0.945	0.980
Ours	R	-	-	-	K	0.131	0.937	5.032	0.203	0.827	0.946	0.981
Ours	R	-	-	CD	K	0.128	0.901	**4.898**	0.198	0.834	0.948	0.983
Ours	R	-	-	CD+K	K	**0.125**	**0.881**	4.903	**0.195**	**0.840**	**0.951**	**0.983**

model, we replace the encoder with the VGG16 and ResNet50 architectures. We use the training and validation set of [39], containing 22.6K and 888 images, respectively. We evaluate our model on the Eigen split [55,39], consisting of 697 images, where ground-truth absolute depth values are captured using LiDAR at sparse pixels. Unlike [39], which uses stereo image pairs as supervision to train the network, or [40], which uses neighboring video frames as supervision to train the network (*yet camera intrinsic parameters are required*), we use the absolute sparse LiDAR depth values to fine-tune our network. The entire network (either VGG16 or ResNet50 version) is trained for 300 epochs using the Adam optimizer with a weight decay of 0.0005. The initial learning rate is 0.0001 and decreased by factor of 10 at the 200th epoch.

Detailed comparisons can be found in Table 4. We can observe that our pre-trained models consistently outperforms ImageNet counterparts, as well as randomly initialized models, using either VGG16 or ResNet50 architectures. It is worth noting, however, that converting relative depth to absolute depth is non-trivial. Computing relative depth (*i.e.*, percentile from absolute depth) is a non-linear mapping. The inverse transformation from relative depth to absolute depth is not unique. Following [40], we multiply our relative depth by a factor as the ratio between relative depth and absolute depth, we get pretty bad results (RMSE of 11.08 vs 4.903), showing this task is non-trivial.

Moreover, pre-training as domain adaptation also improves the performance of our pre-trained model. After fine-tuning our pre-trained model using KITTI's raw videos (Ours CD+K), our ResNet50 model achieves better results than most of the previous methods [55,56,39,40]. The results are also on par with the state-of-the-art method [57].

5 Conclusions and Discussions

We have proposed a new proxy task for self-supervised learning of visual representations. It requires only access to unlabeled videos taken by a moving camera. Representations are learned by optimizing prediction of relative depth, recovered from estimated motion flow, from individual (single) frames. We show this task to be a powerful proxy task, which is competitive with recently proposed alternatives as a means of pre-training representations on unlabeled data. We also demonstrate a novel application of such pre-training, aimed at domain adaptation. When given videos taken by cars driven in cities, self-supervised pre-training primes the downstream urban scene understanding networks, leading to improved accuracy after fine-tuning on a small amount of manually labeled data.

Our work offers novel insights about one of the most important questions in vision today: how can we leverage unlabeled data, and in particular massive amounts of unlabeled video, to improve recognition systems. While a comprehensive picture of self-supervision methods and the role they play in this pursuit is yet to emerge, our results suggest that learning to predict relative depth is an important piece of this picture.

While the gap of the performance between self-supervised methods and their ImageNet counterparts is quickly shrinking, none of current self-supervised methods performs better than ImageNet pre-trained models on tasks involving semantics (*e.g.*, semantic segmentation and object detection). This makes pre-training on ImageNet still practically critical for many computer vision tasks. Despite this fact, this does not mean self-supervised methods are unimportant or unnecessary. The value of self-supervised methods lies in the fact that the training data can easily be scaled up without tedious and expensive human effort.

On other tasks, better performance of self-supervised methods than ImageNet counterparts has been achieved, including our monocular depth estimation and surface normal prediction [37]. Moreover, it has been shown that combining different self-supervised methods can lead to better performance [36,37]. All of these make it very promising that representations learned using self-supervised methods may surpass what ImageNet provides us today.

Acknowledgement

The experiments were performed using equipment obtained under a grant from the Collaborative R&D Fund managed by the Massachusetts Tech Collaborative. GS was also supported by AFOSR award FA9550-18-1-0166. This material is based on research sponsored by the Air Force Research Laboratory and DARPA under agreement number FA8750-18-2-0126. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Air Force Research Laboratory and DARPA or the U.S. Government.

References

1. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. CoRR **abs/1409.1556** (2014)
2. Shelhamer, E., Long, J., Darrell, T.: Fully convolutional networks for semantic segmentation. TPAMI (2017)
3. Larsson, G., Maire, M., Shakhnarovich, G.: Colorization as a proxy task for visual understanding. In: CVPR. (2017)
4. Zhang, R., Isola, P., Efros, A.A.: Colorful image colorization. In: ECCV. (2016)
5. Noroozi, M., Favaro, P.: Unsupervised learning of visual representations by solving jigsaw puzzles. In: ECCV. (2016)
6. Wang, X., Gupta, A.: Unsupervised learning of visual representations using videos. In: ICCV. (2015)
7. Agrawal, P., Carreira, J., Malik, J.: Learning to see by moving. In: CVPR. (2015)
8. Pathak, D., Girshick, R.B., Dollár, P., Darrell, T., Hariharan, B.: Learning features by watching objects move. In: CVPR. (2017)
9. Misra, I., Zitnick, C.L., Hebert, M.: Unsupervised learning using sequential verification for action recognition. In: ECCV. (2016)
10. Pathak, D., Krähenbühl, P., Donahue, J., Darrell, T., Efros, A.: Context encoders: Feature learning by inpainting. In: CVPR. (2016)
11. Gao, R., Jayaraman, D., Grauman, K.: Object-centric representation learning from unlabeled videos. In: ACCV. (2016) 248–263
12. Owens, A., Wu, J., McDermott, J.H., Freeman, W.T., Torralba, A.: Ambient sound provides supervision for visual learning. In: ECCV. (2016)
13. Lee, H., Huang, J., Singh, M., Yang, M.: Unsupervised representation learning by sorting sequences. In: ICCV. (2017)
14. Bideau, P., Learned-Miller, E.: It's moving! A probabilistic model for causal motion segmentation in moving camera videos. In: ECCV. (2016)
15. Jayaraman, D., Grauman, K.: Learning image representations tied to ego-motion. In: ICCV. (2015) 1413–1421
16. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the KITTI vision benchmark suite. In: CVPR. (2012)
17. Geiger, A., Lenz, P., Stiller, C., Urtasun, R.: Vision meets robotics: The KITTI dataset. International Journal of Robotics Research (IJRR) (2013)
18. Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The cityscapes dataset for semantic urban scene understanding. In: CVPR. (2016)
19. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: A large-scale hierarchical image database. CVPR (2009)
20. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR. (2016)
21. Larsson, G., Maire, M., Shakhnarovich, G.: Learning representations for automatic colorization. In: ECCV. (2016)
22. Ranzato, M., Szlam, A., Bruna, J., Mathieu, M., Collobert, R., Chopra, S.: Video (language) modeling: a baseline for generative models of natural videos. arXiv preprint arXiv:1412.6604 (2014)
23. Srivastava, N., Mansimov, E., Salakhutdinov, R.: Unsupervised learning of video representations using lstms. In: ICML. (2015)
24. Mathieu, M., Couprie, C., LeCun, Y.: Deep multi-scale video prediction beyond mean square error. arXiv preprint arXiv:1511.05440 (2015)

25. Vondrick, C., Pirsiavash, H., Torralba, A.: Generating videos with scene dynamics. In: NIPS. (2016)
26. Xue, T., Wu, J., Bouman, K., Freeman, B.: Visual dynamics: Probabilistic future frame synthesis via cross convolutional networks. In: NIPS. (2016)
27. Lotter, W., Kreiman, G., Cox, D.: Deep predictive coding networks for video prediction and unsupervised learning. In: ICLR. (2017)
28. Radford, A., Metz, L., Chintala, S.: Unsupervised representation learning with deep convolutional generative adversarial networks. (2016)
29. Springenberg, J.T.: Unsupervised and semi-supervised learning with categorical generative adversarial networks. In: ICLR. (2016)
30. Donahue, J., Krähenbühl, P., Darrell, T.: Adversarial feature learning. In: ICLR. (2017)
31. Doersch, C., Gupta, A., Efros, A.A.: Unsupervised visual representation learning by context prediction. In: ICCV. (2015)
32. Mobahi, H., Collobert, R., Weston, J.: Deep learning from temporal coherence in video. In: ICML. (2009)
33. Isola, P., Zoran, D., Krishnan, D., Adelson, E.H.: Learning visual groups from co-occurrences in space and time. arXiv preprint arXiv:1511.06811 (2015)
34. Jayaraman, D., Grauman, K.: Slow and steady feature analysis: higher order temporal coherence in video. In: CVPR. (2016)
35. Wiskott, L., Sejnowski, T.J.: Slow feature analysis: Unsupervised learning of invariances. Neural computation **14**(4) (2002) 715–770
36. Doersch, C., Zisserman, A.: Multi-task self-supervised visual learning. In: ICCV. (2017)
37. Wang, X., He, K., Gupta, A.: Transitive invariance for self-supervised visual representation learning. In: ICCV. (2017)
38. Garg, R., Kumar, B.G.V., Carneiro, G., Reid, I.D.: Unsupervised CNN for single view depth estimation: Geometry to the rescue. In: ECCV. (2016) 740–756
39. Godard, C., Mac Aodha, O., Brostow, G.J.: Unsupervised monocular depth estimation with left-right consistency. In: CVPR. (2017)
40. Zhou, T., Brown, M., Snavely, N., Lowe, D.G.: Unsupervised learning of depth and ego-motion from video. In: CVPR. (2017)
41. Vijayanarasimhan, S., Ricco, S., Schmid, C., Sukthankar, R., Fragkiadaki, K.: Sfm-net: Learning of structure and motion from video. In: arXiv. (2017)
42. Li, Y., Paluri, M., Rehg, J.M., Dollár, P.: Unsupervised learning of edges. In: CVPR. (2016) 1619–1627
43. Horn, B.K.P.: Robot Vision. MIT Press, Cambridge, MA, USA (1986)
44. Hu, Y., Li, Y., Song, R.: Robust interpolation of correspondences for large displacement optical flow. In: CVPR. (2017)
45. Dollár, P., Zitnick, C.L.: Fast edge detection using structured forests. IEEE Trans. Pattern Anal. Mach. Intell. **37**(8) (2015) 1558–1570
46. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: NIPS. (2012) 1106–1114
47. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: ICLR. (2015)
48. Teichmann, M., Weber, M., Zöllner, J.M., Cipolla, R., Urtasun, R.: Multinet: Real-time joint semantic reasoning for autonomous driving. In: CVPR. (2016)
49. Ros, G., Ramos, S., Granados, M., Bakhtiary, A., Vázquez, D., López, A.M.: Vision-based offline-online perception paradigm for autonomous driving. In: WACV. (2015)
50. Brostow, G.J., Shotton, J., Fauqueur, J., Cipolla, R.: Segmentation and recognition using structure from motion point clouds. In: ECCV. (2008) 44–57
51. Brostow, G.J., Fauqueur, J., Cipolla, R.: Semantic object classes in video: A high-definition ground truth database. Pattern Recognition Letters (2008)

52. Zhang, R., Isola, P., Efros, A.A.: Split-brain autoencoders: Unsupervised learning by cross-channel prediction. In: CVPR. (2017)
53. Kundu, A., Vineet, V., Koltun, V.: Feature space optimization for semantic video segmentation. In: CVPR. (2016)
54. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: MICCAI. (2015)
55. Eigen, D., Puhrsch, C., Fergus, R.: Depth map prediction from a single image using a multi-scale deep network. In: NIPS. (2014)
56. Liu, F., Shen, C., Lin, G., Reid, I.D.: Learning depth from single monocular images using deep convolutional neural fields. IEEE TPAMI. **38**(10) (2016) 2024–2039
57. Kuznietsov, Y., Stückler, J., Leibe, B.: Semi-supervised deep learning for monocular depth map prediction. In: CVPR. (2017)