Towards Object Detection from Motion

Rico Jonschkowski & Austin Stone
Robotics at Google
{rjon,austinstone}@google.com

Abstract

We present a novel approach to weakly supervised object detection. Instead of annotated images, our method only requires two short videos to learn a new object: 1) a video of a moving object and 2) one or more "negative" videos of the scene without the object. The key idea of the algorithm is to train the object detector to produce physically plausible object motion when applied to the first video and to not detect anything in the second video. With this approach, our method learns to locate objects without any object location annotations. Video data is only required for training. Once the model is trained, it performs object detection on single images. We evaluate our method in three robotics settings that afford learning objects from motion: observing moving objects, watching demonstrations of object manipulation, and physically interacting with objects (see a video summary at `https://youtu.be/XVav0eG9iuQ`). An extended version of this paper can be found at `https://arxiv.org/abs/1909.12950`.

1 Introduction

A major bottleneck for object detection in robotics is the need for time-consuming image annotation. We take a step towards overcoming this problem by learning object detection from short videos with minimal supervision. To learn a new object, our approach only requires two short videos, one that shows the object in motion and one that shows the scene without the object. Such videos are easy and fast to generate – e.g. through human demonstrations or physical interaction of a robot – which makes this approach very promising for robotics.

The underlying assumption that our method is based on is that *an object is a collection of matter that moves as a unit*. We leverage this fact and use *motion* as a cue for learning object detection. Given a video of a moving object, our approach learns an object detector by optimizing its output to describe physically plausible motion. We additionally collect a *negative* video of the scene without the object and train the object detector to not respond to it, which allows the approach to ignore camera motion and other moving objects. Finally, we use the fact that *objects are spatially local* through a *spatial encoder* architecture that estimates the object's location based on the strongest local activations, which restricts the receptive field and induces robustness to non-local distractions.

Our contribution is a novel approach to weakly supervised learning of object detection that uses negative examples and motion (NEMO). Our method trains a spatial encoder network by optimizing consistency with object motion. NEMO only requires short videos of moving objects that are easy to collect and it does not rely on any pretraining or supervision beyond marking these videos as positive and negative. At inference, the learned model can detect objects regardless of whether they are moving or not because the model works on single images. Note that, although we are evaluating our model on video frames, it does not perform tracking but per frame detection.

NeurIPS 2019 Workshop on Robot Learning: Control and Interaction in the Real World, Vancouver, Canada

2 Related Work

This work is related to *physics-based representation learning*, where a latent representation is learned by optimizing consistency with physics, e.g. by optimizing consistency with a known dynamics model [34] or more general assumptions about physical interactions [17, 18], and by pairing such assumptions with image reconstruction [10, 35, 7]. Image embeddings have been learned based on spatio-temporal constistency [9], object permanence [15], equivariance to known ego-motion [16], and view point invariance [33]. While these approaches are similar to this paper in spirit, they learn image embeddings, whereas this paper learns to detect objects in the image coordinates. This more constrained object-based representation makes the presented approach particularly robust and efficient.

This paper is also inspired by *active perception* [3], using action to facilitate perception, e.g. using motion to identify and track objects [26], to segment them [8], to understand their articulation [21]. Combining this idea with learning enables generalization beyond the observed motion, e.g. to learn object segmentation from videos of moving objects [30, 31]. This paper follows the same direction and addresses object detection by introducing ideas from representation learning and by leveraging negative examples.

Labeling training videos as positive and negative examples can also be viewed as *weakly supervised learning*—learning from labels that are only partially informative. Weakly supervised object detection relies on image-wide labels to learn to localize the corresponding objects [29, 28]. This paper goes one step further by only using per-video labels. It compensates this reduction of supervision by adding motion as a cue for learning object detection.

3 Object Detection from Negative Examples and Motion (NEMO)

The key idea of NEMO is to learn how to detect an object from two videos, a *positive video* that shows the target object in motion and a *negative video* of the same scene without that object. These videos are used to optimize two objectives: 1) Learn to detect "something that moves in a physically plausible way" in the positive video, such that its location varies over time without having instantaneous jumps, which is defined below as a combination of a *variation loss* and a *slowness loss*. 2) Learn to detect "something that is present in the positive video but not in the negative video", which is defined as a *presence loss*. These objectives are used to train a *spatial encoder* network, which estimates the object location based on the strongest activation after a stack of convolutions. Optimization is done by gradient descent. We will now look in detail into each of these components.

Network Architecture: Spatial Encoder NEMO's network architecture is an extension of the encoder component of a deep spatial autoencoder [7] and therefore called a *spatial encoder*. The spatial encoder is a stack of convolutional layers [23] without striding or pooling. It uses residual connections [12], batch normalization [14], and ReLU nonlinearities [27]. All experiments in this paper use 8 residual blocks with 32 channels and kernel size 3, which are applied to images scaled to 120×160 or 90×160. The output has a single channel, followed by a spatial softmax, which produces a probability distribution over the object's location in the image. We obtain a location estimate by taking the mean of that distribution (the spatial softargmax) and estimate the confidence in the network's prediction based on the maximum pre-softmax activation.

Losses: Variation, Slowness, & Presence The spatial encoder is trained by minimizing a combination of three losses—variation, slowness, and presence (see Fig. 1), which are defined here. Let us denote the input image at time t as $\boldsymbol{I}^{(t)} \in \mathbb{R}^{h \times w}$ where h and w are the height and width of the image. We will refer to the spatial encoder as f with parameters $\boldsymbol{\theta}$, and the output of f before the spatial softmax as $\boldsymbol{O}^{(t)} \in \mathbb{R}^{h \times w}$, such that $\boldsymbol{O}^{(t)} = f(\boldsymbol{I}^{(t)}; \boldsymbol{\theta})$. By applying the spatial softmax across image coordinates i and j, we get a probability image $\boldsymbol{P}^{(t)} \in \mathbb{R}^{h \times w}$ and its mean $\boldsymbol{z}^{(t)} \in \mathbb{R}^2$ normalized to $[-1, 1]$ as $P_{i,j}^{(t)} = \frac{e^{O_{i,j}^{(t)}}}{\sum_{i,j} e^{O_{i,j}^{(t)}}}$ and $\boldsymbol{z}^{(t)} = \begin{bmatrix} \sum_{i,j}(\frac{2i}{h} - 1)P_{i,j}^{(t)} \\ \sum_{i,j}(\frac{2j}{w} - 1)P_{i,j}^{(t)} \end{bmatrix}$.

The first two losses, variation and slowness, operate on the mean $\boldsymbol{z}$ in positive frames. Together, they measure whether the detected object location $\boldsymbol{z}^{(t)}$ moves in a physically plausible way by comparing pairs of $\boldsymbol{z}^{(t)}$ for different t.

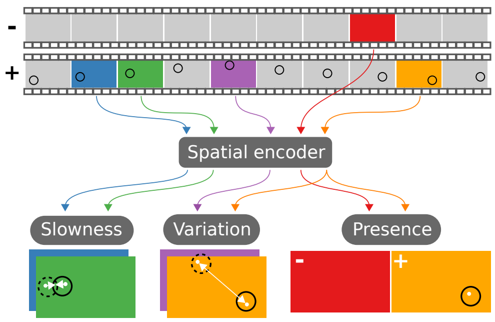

Figure 1: **NEMO overview.** Frames from a negative (-) and a positive video (+) with a moving object (black circle) are fed into the spatial encoder. Consecutive frames (blue and green) are optimized for slowness, which pulls location estimates together. Pairs of distant frames (purple and orange) are optimized for variation, which pushes location estimates apart. Combinations of positive and negative frames (orange and red) are optimized for detection in the positive frame, which increases/decreases activations in the positive/negative frame.

The *variation loss* encodes the assumption that the target object does not stay still in the video by enforcing that $\boldsymbol{z}_{t+d}$ is different from $\boldsymbol{z}_t$ for d in some range $[d_{\text{min}}, d_{\text{max}}]$. The variation loss measures proximity using $e^{-\text{distance}}$, which is 1 if $\boldsymbol{z}_t = \boldsymbol{z}_{t+d}$ and goes to 0 with increasing distance [18].

$$\mathcal{L}_{\text{variation}}(\boldsymbol{\theta}) = \mathbb{E}_{t,d\in[d_{\text{min}},d_{\text{max}}]}[e^{-\beta||\boldsymbol{z}_{t+d}-\boldsymbol{z}_t||}],$$

where β scales how far $\boldsymbol{z}_t$ and $\boldsymbol{z}_{t+d}$ need to be apart and d_{min} and d_{max} define for which time differences variation is enforced. All experiments use $\beta = 10$, $d_{\text{min}} = 50$, and $d_{\text{max}} = 100$.

The *slowness loss* encodes the assumption that objects move with relatively low velocities, i.e., that their locations at time t and $t+1$ are typically close to each other. Consequently, this loss measures the squared distance between $\boldsymbol{z}$ in consecutive time steps t and $t+1$, which favors smooth over erratic object trajectories [36, 17].

$$\mathcal{L}_{\text{slowness}}(\boldsymbol{\theta}) = \mathbb{E}_t[||\boldsymbol{z}_{t+1} - \boldsymbol{z}_t||^2].$$

The *presence loss* encodes the assumption that the object is present in the positive video but not in the negative one. Taking a positive frame t and a negative frame t^-, we can compute the probability $q^{(t,t^-)}$ of the object being in the positive frame by computing the spatial softmax jointly over both frames and summing over all pixels. The loss is then defined as negative log probability.

$$\mathcal{L}_{\text{presence}}(\boldsymbol{\theta}) = \mathbb{E}_{t,t^-}[-\log(q^{(t,t^-)})], \text{where } \; q^{(t,t^-)} = \frac{\sum_{i,j} e^{O_{i,j}^{(t)}}}{\sum_{i,j} e^{O_{i,j}^{(t)}} + e^{O_{i,j}^{(t^-)}}}.$$

These losses are combined in a weighted sum, $\mathcal{L}(\boldsymbol{\theta}) = w_{\text{v}}\mathcal{L}_{\text{var.}}(\boldsymbol{\theta}) + w_{\text{s}}\mathcal{L}_{\text{slown.}}(\boldsymbol{\theta}) + w_{\text{p}}\mathcal{L}_{\text{pres.}}(\boldsymbol{\theta})$, where the weights were chosen such that all gradients have the same order of magnitude. All experiments use $w_{\text{v}} = 2$, $w_{\text{s}} = 10$, and $w_{\text{p}} = 1$. The losses are optimized from minibatches of size 10. For numerical stability of the gradient computation, Gaussian noise $\boldsymbol{\epsilon} \sim \mathcal{N}(\mu = 0, \sigma = 10^{-5})$ is added to $\boldsymbol{z}_t$. The loss $\mathcal{L}(\boldsymbol{\theta})$ is optimized using Adam [22] with default parameters and $m = 50$ random restarts. The method is implemented based on TensorFlow [1] and Keras [6].

4 Experiments

We evaluate NEMO in three settings that afford object detection from motion:

- 1. Learning to detect moving objects by observing them (Fig. 2 top)
- 2. Learning to detect static objects from human demonstrations (Fig. 2 middle)
- 3. Learning to detect static objects by physically interacting with them (Fig. 2 bottom)

In all settings, our method was trained on short (less than five minutes) positive and negative videos and then tested on individual frames from a different video. Note that NEMO does not perform tracking. All results show per frame detection. Since settings 2 and 3 feature multiple objects, a separate detector was trained per object, using videos of the other objects as negative examples. The robot in setting 3 executed a pre-defined movement to produce object motion.

Figure 2 shows object detection on individual frames of test videos. These results show that our method is able to discover objects without any image level annotations from a few short videos of

Figure 2: Qualitative results on test images. Settings 1.-3. top to bottom. Colored dots and image crops visualize detected object locations. For more details, see `https://youtu.be/XVav0eG9iuQ`.

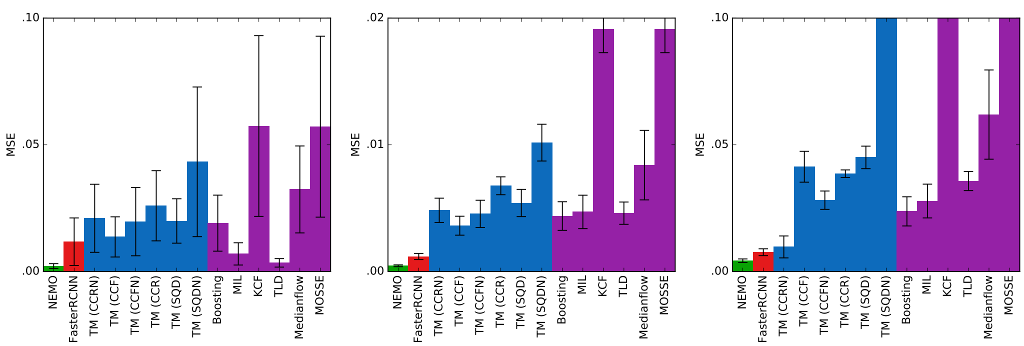

Figure 3: Test error comparison. Settings 1.-3. left to right, bars denote standard errors.

moving objects and that it is robust to distracting motion of the camera, the arm, and other moving objects as well as to substantial occlusions during training and testing.

To evaluate detection accuracy, Figure 3 compares NEMO (green) to FasterRCNN [32] trained on COCO [25] (red), template matching with different metrics [24] (blue), and tracking [11, 2, 13, 20, 19, 4] (purple) using OpenCV [5]. Note that none of the methods we compare to can solve the problem NEMO is addressing because each requires some amount of ground truth object location annotations. For template matching and tracking methods, we provide annotated bounding boxes in the first frame to initialize tracking and to extract templates. For FasterRCNN, we use ground truth locations throughout the test video to match predicted bounding boxes to target objects, which is needed because the evaluated object classes are not present in COCO. Although NEMO does not need any information about object locations during training or testing, it outperforms the other methods in all three settings. These results show the advantage of adapting to the given set of objects using unsupervised learning and hint at the potential of future work on object detection from motion.

References

[1] Martín Abadi et al. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[2] Boris Babenko, Ming-Hsuan Yang, and Serge Belongie. Visual tracking with online multiple instance learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 983–990, 2009.

[3] Ruzena Bajcsy. Active perception. In *IEEE Proceedings*, volume 76, pages 996–1006, 1988.

[4] David S Bolme, J Ross Beveridge, Bruce A Draper, and Yui Man Lui. Visual object tracking using adaptive correlation filters. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2544–2550, 2010.

[5] Gary Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

[6] François Chollet et al. Keras. `https://keras.io`, 2015.

[7] Chelsea Finn, Xin Yu Tan, Yan Duan, Trevor Darrell, Sergey Levine, and Pieter Abbeel. Deep spatial autoencoders for visuomotor learning. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 512–519, 2016.

[8] Paul Fitzpatrick. First contact: an active vision approach to segmentation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, volume 3, pages 2161–2166. IEEE, 2003.

[9] Ruohan Gao, Dinesh Jayaraman, and Kristen Grauman. Object-centric representation learning from unlabeled videos. In *Asian Conference on Computer Vision (ACCV)*, November 2016.

[10] Ross Goroshin, Michael F Mathieu, and Yann LeCun. Learning to linearize under uncertainty. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1234–1242, 2015.

[11] Helmut Grabner, Michael Grabner, and Horst Bischof. Real-time tracking via on-line boosting. In *British Machine Vision Conference (BMVC)*, volume 1, page 6, 2006.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[13] João F Henriques, Rui Caseiro, Pedro Martins, and Jorge Batista. High-speed tracking with kernelized correlation filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(3):583–596, 2015.

[14] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning (ICML)*, pages 448–456, 2015.

[15] Eric Jang, Coline Devin, Vincent Vanhoucke, and Sergey Levine. Grasp2vec: Learning object representations from self-supervised grasping. *Proceedings of Machine Learning Research*, 2018.

[16] Dinesh Jayaraman and Kristen Grauman. Learning image representations tied to ego-motion. In *IEEE International Conference on Computer Vision (ICCV)*, pages 1413–1421, 2015.

[17] Rico Jonschkowski and Oliver Brock. Learning state representations with robotic priors. *Autonomous Robots*, 39(3):407–428, 2015.

[18] Rico Jonschkowski, Roland Hafner, Jonathan Scholz, and Martin Riedmiller. Pves: Position-velocity encoders for unsupervised learning of structured state representations. In *New Frontiers for Deep Learning in Robotics Workshop at RSS*, 2017.

[19] Zdenek Kalal, Krystian Mikolajczyk, and Jiri Matas. Forward-backward error: Automatic detection of tracking failures. In *International Conference on Pattern Recognition (ICPR)*, pages 2756–2759, 2010.

[20] Zdenek Kalal, Krystian Mikolajczyk, Jiri Matas, et al. Tracking-learning-detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(7):1409, 2012.

[21] Dov Katz and Oliver Brock. Manipulating articulated objects with interactive perception. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 272–277. IEEE, 2008.

[22] Diederik P Kingma and Jimmy Lei Ba. Adam: Amethod for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015.

[23] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[24] John P Lewis. Fast template matching. In *Vision interface*, volume 95, pages 15–19, 1995.

[25] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision (ECCV)*, pages 740–755. Springer, 2014.

[26] Alan J Lipton, Hironobu Fujiyoshi, and Raju S Patil. Moving target classification and tracking from real-time video. In *Fourth IEEE Workshop on Applications of Computer Vision (WACV)*, pages 8–14. IEEE, 1998.

[27] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *International Conference on Machine Learning (ICML)*, pages 807–814, 2010.

[28] Maxime Oquab, Léon Bottou, Ivan Laptev, and Josef Sivic. Is object localization for free?-weakly-supervised learning with convolutional neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 685–694, 2015.

[29] Megha Pandey and Svetlana Lazebnik. Scene recognition and weakly supervised object localization with deformable part-based models. In *International Conference on Computer Vision (ICCV)*, pages 1307–1314. IEEE Computer Society, 2011.

[30] Deepak Pathak, Ross B Girshick, Piotr Dollár, Trevor Darrell, and Bharath Hariharan. Learning features by watching objects move. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, page 7, 2017.

[31] Deepak Pathak, Yide Shentu, Dian Chen, Pulkit Agrawal, Trevor Darrell, Sergey Levine, and Jitendra Malik. Learning instance segmentation by interaction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 2042–2045, 2018.

[32] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems (NIPS)*, pages 91–99, 2015.

[33] Pierre Sermanet, Corey Lynch, Yevgen Chebotar, Jasmine Hsu, Eric Jang, Stefan Schaal, and Sergey Levine. Time-contrastive networks: Self-supervised learning from video. *arXiv preprint arXiv:1704.06888*, 2017.

[34] Russell Stewart and Stefano Ermon. Label-free supervision of neural networks with physics and domain knowledge. In *AAAI Conference on Artificial Intelligence*, pages 1–7, 2017.

[35] Manuel Watter, Jost Springenberg, Joschka Boedecker, and Martin Riedmiller. Embed to control: A locally linear latent dynamics model for control from raw images. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2746–2754, 2015.

[36] Laurenz Wiskott and Terrence J Sejnowski. Slow feature analysis: Unsupervised learning of invariances. *Neural Computation*, 14(4):715–770, 2002.