Self-Supervised Human Detection and Segmentation via Background Inpainting

Isinsu Katircioglu, Helge Rhodin, Victor Constantin, Jörg Spörri, Mathieu Salzmann, and Pascal Fua, *Fellow, IEEE*

Abstract—While supervised object detection and segmentation methods achieve impressive accuracy, they generalize poorly to images whose appearance significantly differs from the data they have been trained on. To address this when annotating data is prohibitively expensive, we introduce a self-supervised detection and segmentation approach that can work with single images captured by a potentially moving camera. At the heart of our approach lies the observation that object segmentation and background reconstruction are linked tasks, and that, for structured scenes, background regions can be re-synthesized from their surroundings, whereas regions depicting the moving object cannot. We encode this intuition into a self-supervised loss function that we exploit to train a proposal-based segmentation network. To account for the discrete nature of the proposals, we develop a Monte Carlo-based training strategy that allows the algorithm to explore the large space of object proposals. We apply our method to human detection and segmentation in images that visually depart from those of standard benchmarks and outperform existing self-supervised methods.

Index Terms—Self-supervised training, importance sampling, proposal-based detection and segmentation, image inpainting

1 Introduction

Robust detection and segmentation of moving objects can now be achieved reliably in scenarios for which large amounts of annotated data are available [1]. However, for less common activities, such as skiing, it remains challenging, because the required training databases do not exist, as shown in Fig. 1. Self-supervised approaches [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13] promise to address this problem. However, some can only operate on video streams as opposed to single images [3], [8], [10], [13] while most others depend on strong constraints being satisfied, such as the target objects being seen against a static background.

To develop a more generic approach, we start from the observation that in most images the background forms a consistent, natural scene. Therefore, the appearance of any background patch can be predicted from its surroundings. By contrast, a moving person's appearance is unpredictable from the neighboring scene content and can be expected to be very different from what an inpainting algorithm would produce. We incorporate this insight into a proposal-generating deep network whose architecture is inspired by those of YOLO [14] and MaskRCNN [1] but does not require explicit supervision.

Specifically, for each proposal, we synthesize a background image by masking out the corresponding region and inpainting it from the rest of the image. The loss function we minimize favors the largest possible distance between this reconstructed background and the input image. This encourages the network to select regions that cannot be explained from their surrounding and are therefore salient. To handle the discrete nature of the proposals, we develop a Monte Carlo-based strategy to train our network. It operates on a discrete distribution, is unbiased, exhibits low variance, and is end-to-end trainable.

Our approach overcomes limitations in existing self-supervised human pose estimation methods requiring static cameras [9] or monochromatic background [6], [7]. We propose a self-supervised method that operates on single images and demonstrate its effectiveness on several human motion datasets captured with cameras that are static, pan-tilt-zoom, or hand-held. We can handle large camera motions and do not require *any* manual annotation. We focus on images acquired in realistic conditions such as Ski-PTZ dataset of [15], daily human motion Handheld190k in outdoor scene and figure skating FS-Singles as well as those of the standard H36M benchmark [16]. Fig. 1 depicts such a scenario in which our approach outperforms a state-of-the-art detection and instance segmentation method [1] trained on large annotated dataset [17]. It also outperforms existing self-supervised segmentation techniques [3], [5], [8], [10], [18]. Following standard practice in the self-supervision literature [3], [9], [10], we start from pre-trained network weights, which we fine-tune without any additional supervision in our target domain. However, we can also train from scratch with only a small performance loss. Finally, even though we focus on people,

- *Isinsu Katircioglu, Victor Constantin, Mathieu Salzmann, and Pascal Fua are with Computer Vision Laboratory, École Polytechnique Fédérale de Lausanne, 1015 Lausanne, Switzerland. E-mail: {isinsu.katircioglu, victor.constantin, mathieu.salzmann, pascal.fua}@epfl.ch.*
- *Helge Rhodin is with the Computer Vision Lab and the Imager Lab, University of British Columbia, Vancouver, BC V6T 1Z4, Canada. E-mail: rhodin@cs.ubc.ca.*
- *Jörg Spörri is with the Department of Orthopaedics, Balgrist University Hospital, University of Zurich, 8006 Zürich, Switzerland. E-mail: Joerg.Spoerri@balgrist.ch.*

Manuscript received 22 Sept. 2020; revised 17 Sept. 2021; accepted 24 Oct. 2021. Date of publication 29 Oct. 2021; date of current version 3 Nov. 2022.
This work was supported in part by the Swiss National Science Foundation (SNSF).
(Corresponding author: Isinsu Katircioglu.)
Recommended for acceptance by P. Favaro.
Digital Object Identifier no. 10.1109/TPAMI.2021.3123902

Fig. 1. *Domain specific detection and segmentation.* Our self-supervised method detects the skier well, while YOLO trained on a general dataset does not generalize to this challenging domain. Similarly, MaskRCNN trained on a general dataset sometimes misses body parts such as the upper body of the skier in (c).

we show that our approach also applies to other kinds of target objects. We will make our code and three new datasets we used for our experiments available upon acceptance of the paper.

2 Related Work

Most salient object detection and segmentation algorithms are fully-supervised [1], [14], [19], [20], [21], [22], [23], [24], [25], [26] and require large annotated datasets with paired images and labels. Our goal is to train a purely self-supervised method without either segmentation or object bounding box annotations. Note that this differs from the so-called *unsupervised object segmentation* methods [27], [28], [29], [30], [31], [32], [33], [34], [35], [36], [37], that require domain-specific annotations during training but not at test time, or the label of the first frame at inference time [38]. We focus our discussion on self- and weakly-supervised methods with regard to the type of training data used and refer to [3] for a complete discussion of methods using hand-crafted optimization.

Weakly-Supervised Methods. An early weakly-supervised method is the Hough Matching algorithm [39]. It uses an object classification dataset and identifies foreground as the image regions that have re-occurring Hough features within images of the same class. Similar principles have been followed to train deep networks for object detection [29], [40], optical flow estimation [41], [42], and object saliency [30]. These methods make the implicit assumption that the background varies across the examples and can therefore be excluded as noise. This assumption is violated when training on domain-specific images, where foreground and background are similar across the examples.

Motion-Based Methods. Conventional methods [3], [10], [18], [43], [44], [45], [46], [47], [48] explore the motion information mainly by resorting to hand-crafted features. [47] proposes a spatial-temporal energy function applied to optical flow field to obtain spatiotemporally consistent saliency maps that are further improved by using global appearance and location models. Similarly, [44] computes the optical flow to detect motion boundaries and refines them through ray-casting strategy. An alternative temporal solution [3] relies on the recurrence property of the primary object in a video. It finds the recurring candidate regions in the entire sequence by extracting color and motion cues through ultrametric contour maps. Identifying the matching segment tracks in different frames is done by minimizing a chi-square distance temporally in the feature space. Given video sequences, the temporal information can be exploited by assuming that the background changes slowly [49] or linearly [18]. However, even a static scene induces non-homogeneous deformations under camera translation, and it can be difficult to handle all types of camera motion within a single video, and to distinguish articulated human motion from background motion [50]. Some of the resulting errors can be corrected by iteratively refining the crude background subtraction results of [18] with an ensemble of student and teacher networks [8]. This, however, induces a strong dependence on the teacher used for bootstrapping. Recently, [13] showed that leveraging the temporal information at different granularities through forward-backward patch tracking and cross-frame semantic matching can be used to learn video object patterns from unlabeled videos. Note that these methods can only operate on video streams and exploit a strong temporal dependency, which our model does not require.

Our approach is conceptually related to VideoPCA [18], which models the background as the part of the scene that can be explained by a low-dimensional linear basis. This implicitly assumes that the foreground is harder to model than the background and can therefore be separated as the non-linear residual. Here, instead of using motion cues, we propose to rely on the predictability of image patches from their spatial neighborhood using deep neural networks. This gives us an advantage over VideoPCA, which only works with videos and comparably little background motion and complexity. Another closely related work [10] employs a similar inpainting network to ours on flow fields. It relies on an adversarial model that tries to hallucinate the optical flow from its surrounding while generating the mask of a supposedly moving object in the region where the inpainting network yields poor reconstruction. [10] is based on the PWC network [51] that is trained with supervision on a large object database to predict flow with clear object boundaries. In that sense, as the methods based on deep optical flow, it is not strictly self-supervised and can suffer from degenerate cases when applied to still images with no or little movement. We will nonetheless show that our approach can also benefit from such optical flow prediction if available, outperforming the other methods that use this information.

Self-Supervised Methods. Most similar to our approach are the self-supervised ones to object detection [2], [4], [7], [9] that complement auto-encoder networks by an attention mechanism. They first detect one or several bounding boxes, whose content is extracted using a spatial transformer [52]. This content is then passed through an auto-encoder and re-composited with a background. In [9], the background is assumed to be static and in [2], [7] even single colored, a severe restriction in practice. [7] uses a proposal-based network similar to ours, but resorts to approximating the proposal distribution with a continuous one to make the model differentiable. Here, we demonstrate that much simpler importance sampling is sufficient. In [53] a noisy segmentation masks is predicted by an unsupervised version [45] used as a pseudo label to train a ConvNet to segment moving objects from single images. [4] uses a generative model relying on the assumption that the image region strictly covering the salient object can be subject to random shifts without affecting the realism of the scene. Similarly, the method of [5] relies on an adversarial network whose generator extracts the object mask and redraws the object by assigning different color or texture features to that

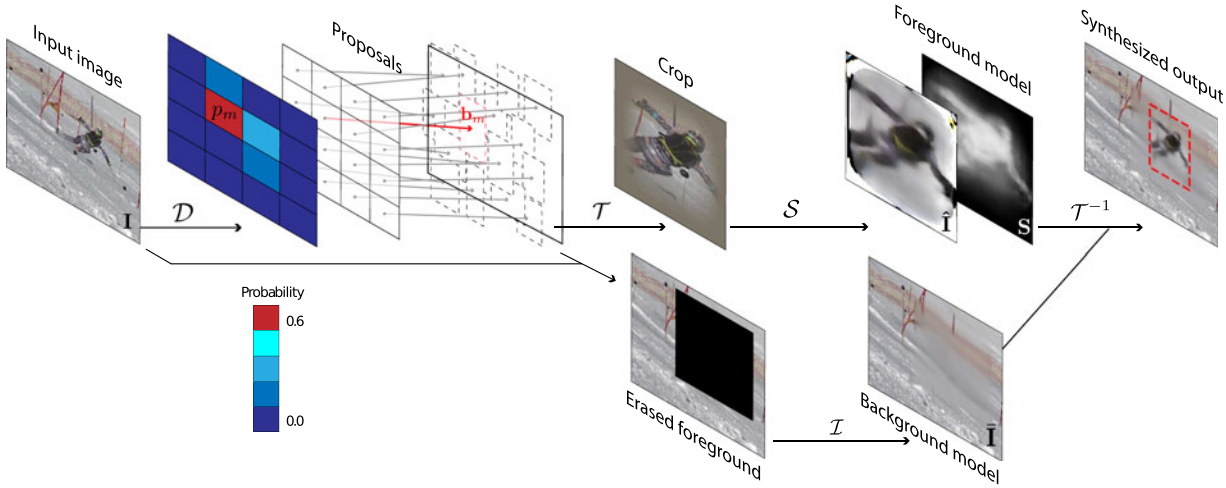


Fig. 2. *Architecture.* Our model $\mathcal{F}$ passes the input image $\mathbf{I}$ to a detector $\mathcal{D}$ that proposes potential bounding boxes. One of them is passed to a spatial transformer $\mathcal{T}$ that crops $\mathbf{I}$ and the result is fed to a segmentation network $\mathcal{S}$ that outputs a segmentation mask $\mathbf{S}$ and the corresponding foreground image $\hat{\mathbf{I}}$. In a separate branch, an inpainting network $\mathcal{I}$ fills the content of the bounding box to generate a background image $\bar{\mathbf{I}}$. Finally, the inverse transformer $\mathcal{T}^{-1}$ is used to combine $\hat{\mathbf{I}}$, masked by $\mathbf{S}$, and $\bar{\mathbf{I}}$ into an image that should be similar to the original one.

region. This is very different from our approach that aims to reconstruct the scene from its background. Along similar lines, the algorithm of [54] searches for the foreground object by compositing it into another image so that the discriminator fails to classify the resulting image as fake. These methods can be easily deceived by other background objects whose random displacement or texture change can still yield realistic images. In contrast to these GAN-based techniques, our approach works with images acquired using a moving camera and with an arbitrary background.

In addition to object detection, the algorithm of [9] also returns instance segmentation masks by reasoning about the extent and depth ordering of multiple people in a multi-camera scene. However, this requires multiple static cameras and a static background at training time, as does the approach of [55] that performs instance segmentation in crowded scenes.

3 Method

Our goal is to learn a salient person detector and segmentor from unlabeled videos acquired in as practical a setup as possible. We therefore only use raw videos or images as input and do not constrain the frame-to-frame camera motion.

3.1 Outline

Our basic intuition is that when people move with respect to the background, the area they occupy often looks quite different from the background. More specifically, we operate under the following two assumptions.

- **A1:** The foreground and background are distinguishable by color or texture as explained in detail by [56]. As discussed in Section 3.5, this can be relaxed by using optical flow.
- **A2:** Every part of the background must be uncovered more often than covered. This assumption is almost always valid in long videos depicting moving people, unlike the assumptions made in related approaches [3], [10], [28], [41], [42] that require people to move in every frame.

Hence, we cast the foreground segmentation task as one of finding an area that, when inpainted using information from the background, yields an image that is as different as possible from the true one. This makes sense under assumption A1 that people look different from the background. Assumption A2 is required to be able to train the inpainting network in a self-supervised manner. In the remainder of this section, we first present the architecture of the network we use for this purpose and then explain how we train it.

3.2 Network Architecture

We use the model $\mathcal{F}$ depicted by Fig. 2. It takes a single image $\mathbf{I} \in \mathbb{R}^{W \times H \times 3}$ as input. It then resynthesizes it by sampling a candidate bounding box, cropping the corresponding image patch, and, in parallel, predicting a foreground image $\hat{\mathbf{I}} \in \mathbb{R}^{128 \times 128 \times 3}$ and a segmentation mask $\mathbf{S} \in \mathbb{R}^{128 \times 128}$ from the crop, while inpainting the cropped region to generate a background image $\bar{\mathbf{I}} \in \mathbb{R}^{W \times H \times 3}$. Finally, the foreground crop and the background image are re-composed according to the segmentation mask. Formally, this can be written as

$$\mathcal{F}(\mathbf{I}) = \mathcal{T}^{-1}(\hat{\mathbf{I}} \circ \mathbf{S}) + \bar{\mathbf{I}} \circ (1 - \mathcal{T}^{-1}(\mathbf{S})), \tag{1}$$

where $\mathcal{T}$ is the spatial transformer corresponding to the selected bounding box, and $\circ$ is the element-wise multiplication.

To generate the segmentation mask $\mathbf{S}$, $\mathcal{F}$ relies on a detection network $\mathcal{D}$ inspired by the YOLO architecture [14]. It divides the image into a grid and computes for each cell c a probability p_c of a detection expressed in terms of a bounding box $\mathbf{b}_c \in \mathbb{R}^4$ that defines a center and offset from the grid center. Hence, it outputs a set of C candidate bounding boxes $\{\mathbf{b}_c\}_{c=1}^{C}$ and corresponding probabilities $\{p_c\}_{c=1}^{C}$ out of which one bounding box $\mathbf{b}_c$ is sampled according to its probability p_c. A segmentation network $\mathcal{S}$ then encodes and decodes the content of $\mathbf{b}_c$ into a segmentation mask $\mathbf{S}$ and the corresponding foreground image $\hat{\mathbf{I}}$.

In a separate branch, an inpainting network $\mathcal{I}$ generates the background image $\bar{\mathbf{I}}$. Since off-the-shelf inpainting networks [57], [58] trained on large and generic datasets tend to hallucinate objects, we rely instead on a U-Net architecture [59] to implement $\mathcal{I}$, which we pre-train without using any labels and for each dataset, as discussed below. When the background $\mathbf{B}$ is known *a priori*, for example because we use a static-camera, we can simplify our architecture by removing the inpainting branch and replacing $\bar{\mathbf{I}}$ by $\mathbf{B}$. This specific case has been addressed in [7], [9] but we will show that our approach yields better results.

Algorithm 1. Our Training and Test Procedures

input : $\mathbf{I} \in \mathbb{R}^{W \times H \times 3}$
output: $\mathbf{I}' \in \mathbb{R}^{W \times H \times 3}$ // Resynthesized image
for $\mathbf{I}_n$ in $\{\mathbf{I}_n\}_1^N$ **do**
  $\{(\mathbf{b}_c), (p_c)\}_{c=1}^C \leftarrow \mathcal{D}(\mathbf{I}_n)$ // Bounding box prediction
  if *training* **then**
    $m \leftarrow sample\_2D\_cell(p_c)$;
  else
    $m \leftarrow \arg\max_c p_c$;
  end
  if *exists*($\mathbf{B}$) **then**
    $\bar{\mathbf{I}}_n \leftarrow \mathbf{B}$
  else
    $\mathbf{I}_n^{bg} \leftarrow \mathbf{I}_n$
    $\mathbf{I}_n^{bg}[\mathbf{b}_m] \leftarrow 0$
    $\bar{\mathbf{I}}_n \leftarrow \mathcal{I}(\mathbf{I}_n^{bg})$ // Background inpainting
  end
  $\mathbf{I}_n^{crop} \leftarrow \mathcal{T}(\mathbf{I}_n, \mathbf{b}_m)$
  $\hat{\mathbf{I}}_n, \mathbf{S}_n \leftarrow \mathcal{S}(\mathbf{I}_n^{crop})$
  $\mathbf{I}_n' \leftarrow \mathcal{T}^{-1}(\hat{\mathbf{I}}_n \circ \mathbf{S}_n) + \bar{\mathbf{I}}_n \circ (1 - \mathcal{T}^{-1}(\mathbf{S}_n))$ // Resynthesis
end

At inference time, we simply run the trained model on the test image and pick the 2D grid cell with the highest occupancy probability $c^* = \arg\max_c p_c$. Its bounding box parameter estimates are fed into the spatial transformer $\mathcal{T}$ to crop the region of interest, which is then segmented by the segmentation network $\mathcal{S}$, as described above. The corresponding pseudo-code is given in Algorithm 1. Re-composing the image and background inpainting are only essential to train our model. They can be omitted at inference time for bounding box and segmentation mask prediction.

3.3 Training Losses

Given a set of unlabeled training images $\{\mathbf{I}_1, \ldots, \mathbf{I}_N\}$, we first train $\mathcal{I}$ and then $\mathcal{F}$, and therefore $\mathcal{D}$ and $\mathcal{S}$, in a self-supervised manner.

To train $\mathcal{I}$, we randomly remove image regions from the training images and inpaint them from their immediate surrounding. We compare the result to the original image using an L_2 pixel-wise loss augmented by a perceptual loss, which we minimize. This works well as long as assumption A2 introduced in Section 3.1 holds.

3.3.1 Foreground versus Background

To learn the weights of $\mathcal{F}$, we minimize a weighted sum of a foreground loss L_{fg} and a background loss L_{bg}. Given the probabilistic nature of the detections generated by the detector network $\mathcal{D}$, we take them to be expected values. We write

$$L_{\text{fg}}(\mathbf{I}) = \sum_{c=1}^{C} p_c L_2(\mathcal{F}_c(\mathbf{I}), \mathbf{I}) \ , \tag{2}$$

$$L_{\text{bg}}(\mathbf{I}) = -\sum_{c=1}^{C} p_c \frac{L_2(\bar{\mathbf{I}}_c, \mathbf{I})}{\texttt{area}(\mathbf{b}_c)}, \tag{3}$$

where L_2 is the pixel-wise mean square loss and p_c is the probability associated to bounding box $\mathbf{b}_c$ by the detector network. $\mathcal{F}_c(\mathbf{I})$ indicates the resynthesized image and $\bar{\mathbf{I}}_c$ is the background image generated by inpainting based on the sampled cell c, as discussed in Section 3.2. Minimizing L_{fg} encourages $\mathcal{F}_c(\mathbf{I})$ to be as similar as possible to $\mathbf{I}$, for all training images, but does not preclude the generation of bounding boxes on background objects. That is the role of L_{bg}. Because of the minus sign in front of the summation, minimizing it favors bounding boxes for which the inpainting generates an image that is different from the original one, which denotes an image location that cannot be reliably reconstructed from surrounding pixels by inpainting. Note that we normalize by dividing by $\texttt{area}(\mathbf{b}_c)$, which is the maximum number of pixels that may be different in $\bar{\mathbf{I}}_c$ and $\mathbf{I}$. This makes L_{bg} insensitive to the size of the bounding box. Without this division, L_{bg} would favor large regions, whether they contain an object or not. Nevertheless, minimizing L_{bg} by itself can favor bounding boxes with high-error density, whether or not they cover the whole person, as we will demonstrate in the ablation study of the results section.

3.3.2 Disentangled Training Strategy

In short, minimizing L_{bg} does not guarantee bounding boxes that fit to the person completely or precisely. By contrast, minimizing L_{fg} favors a tight fit of the segmentation mask $\mathbf{S}$ when the bounding box $\mathbf{b}_c$ is correctly located because the rest of $\mathcal{F}(\mathbf{I})$ is resynthesized using only background information, which is not relevant to the person's appearance. However, it can also yield meaningless solutions in which $\mathbf{b}_c$ is located in the background. To get the best of both world, we must therefore minimize L_{fg} and L_{bg} jointly.

Unfortunately, finding a balance between these two competing objectives by relative weighting alone has proved difficult, if not impossible. Instead, we designed a *disentangled training strategy* in which we isolate their conflicting influence on the individual network components to stabilize the training when their contributions are weighted.

Specifically, the probabilities p_c are only optimized according to L_{bg} so that L_{fg} cannot bias them towards the background regions, where it has a trivial solution. Conversely, $\mathbf{b}_c$ is optimized only according to L_{fg} to favor a tight fit without the opposite bias from L_{bg} towards high error density $\mathbf{b}_c$ with only partial coverage of the person. Similarly, $\mathcal{S}$ is optimized solely according to L_{fg} to yield the best possible reconstruction, instead of the largest distance to the background as induced by L_{bg}. This can all be computed in a single forward-backward pass by treating the excluded variables as constants in the respective objectives, that is, by cutting their gradient flow.

3.3.3 Full Training Loss

To speed up the convergence and to make the segmentation crisper, we introduce a perceptual loss and regularization terms in addition to L_{fg} and L_{bg}.

*Perceptual Loss.*We take it to be

$$L_{\phi} = \sum_{c=1}^{C} p_c L_2(\phi(\mathcal{F}_c(\mathbf{I})), \phi(\mathbf{I})) \ , \tag{4}$$

where $\phi(.)$ denotes the low level features obtained by passing its input to a pre-trained ResNet18 network.

*Probability Regularizer.*We take it to be the L_1 loss

$$L_p = \sum_{c=1}^{C} |p_c|, \tag{5}$$

that promotes sparsity of the non-zero probabilities.

*Segmentation Mask Regularizer.*We take it to be a v-shaped prior that operates on $\mathbf{S}$ and stabilizes the early training iterations by encouraging the average value of the segmentation mask to be larger than a threshold value λ yet sparse and less noisy when exceeding this threshold. We write

$$L_v = \left| \left(\frac{1}{WH} \sum_{x}^{W} \sum_{y}^{H} \mathcal{T}^{-1}(\mathbf{S})_{xy} \right) - \lambda \right| + \lambda \ , \tag{6}$$

where W and H are the image width and height, respectively, and λ is set to 0.005. Note that this threshold does not control the size of the segmentation. The small value is exceeded quickly and makes L_v an L_1 prior for subsequent training iterations.

Joint Loss. In practice, we use a weighted combination of these losses, given by

$$L_{joint} = \alpha L_{\text{bg}} + \beta L_{\text{fg}} + \gamma L_{\phi} + \eta L_v + \zeta L_p, \tag{7}$$

applied to N unlabeled images within a batch, where $\alpha = 0.1, \beta = 1, \gamma = 2, \eta = 0.25$ and $\zeta = 0.1$.

3.4 Monte Carlo and Importance Sampling

Computing the losses of Eqs. (2), (3), and (4) involves summing over the C bounding boxes proposed by the detection network $\mathcal{D}$ and their corresponding probabilities. In practice, we use $C = 64$ and back-propagating through all 64 possibilities at each training iteration makes the computation expensive. Hence, for practical purposes, it has proved necessary to reduce this cost.

Since all three losses are of the form $L = \sum_{c=1}^{C} p_c f(\mathbf{I}, \mathbf{b}_c)$, where f is a differentiable function, the simplest way to speed up the computation would be to randomly sample a small subset of the C bounding boxes and write

$$L \approx \mathbf{E}_c[f(\mathbf{I}, \mathbf{b}_c)] \text{ with } c \sim p \ , \tag{8}$$

where $\mathbf{E}_c$ denotes the expectation over c drawn from the categorical proposal distribution $p = \{p_1, \ldots, p_C\}$ output by the network $\mathcal{D}$. Unfortunately, the resulting loss estimate would then not be differentiable with respect to the network weights, thus precluding end-to-end gradient-based optimization.

Instead, we use Monte Carlo sampling to evaluate all three losses and introduce an auxiliary distribution q to rewrite Eq. (8) as

$$L \approx \mathbf{E}_c \left[\frac{p_c}{q_c} f(\mathbf{I}, \mathbf{b}_c) \right] \text{ with } c \sim q \ . \tag{9}$$

This approximation holds for any two probability distributions and drawing the samples according to q instead of p does not depend on the network weights, thus provides differentiability [60]. However, this Monte Carlo sampling comes at the cost of a potentially high approximation error when using only a few samples. For instance, by choosing q to be the uniform sampling distribution $\mathcal{U}$, most of the uniformly drawn samples will have a low probability p and, therefore, negligible influence. To reduce this error, we rely on importance sampling [61], [62] to provide a low-variance unbiased estimator by taking the sampling distribution q to be similar to p. Then $p_c/q_c \approx 1$ and the fraction does not influence the result much. However, the derivatives can still be computed because q_c is a constant and the gradient of Eq. (9) is the same as in the likelihood ratio method [63] used in the REINFORCE algorithm [64].

In practice, to prevent division by very small values that could lead to numerical instability, we take the q probabilities to be

$$q_c = p_c(1 - C\epsilon) + \epsilon \ . \tag{10}$$

As a side effect, ϵ controls the probability that an unlikely case is chosen, which induces a form of exploration that is helpful in the early training stages of the network.

When approximating the expectation with a single sample, we can rewrite the losses introduced in Sections 3.3.1 and 3.3.3 as

$$L_{\text{bg}}(\mathbf{I}) = -\frac{p_c}{q_c} \frac{L_2(\bar{\mathbf{I}}_c, \mathbf{I})}{\texttt{area}(\mathbf{b}_c)} \ , \tag{11}$$

$$L_{\text{fg}}(\mathbf{I}) = \frac{p_c}{q_c} L_2(\mathcal{F}_c(\mathbf{I}), \mathbf{I}) \ , \tag{12}$$

$$L_{\phi}(\mathbf{I}) = \frac{p_c}{q_c} L_2(\phi(\mathcal{F}_c(\mathbf{I})), \phi(\mathbf{I})) \ , \tag{13}$$

with $c \sim q$ and inject these new definitions into that of the joint loss L_{joint} of Eq. (7).

3.5 Exploiting Optical Flow for Training Purposes

When video sequences are available at training time, we can exploit optical flow to help detect the foreground subject. To this end, we use optical flow images obtained by running FlowNet 2.0 [65] on pairs of consecutive frames stabilized by computing a homography using SIFT keypoints to warp one onto the other. We use the resulting optical flow image $\mathbf{I}_f$ as an intermediate supervision to our model. To this end, we train a second inpainting network $\mathcal{I}_f(\mathbf{I}_f, \mathbf{b}_c)$ to reconstruct flow images instead of regular ones. We then introduce an additional flow background objective $L_{\text{bg}}(\mathbf{I}_f)$, with the same weight as $L_{\text{bg}}(\mathbf{I})$, into L_{joint} of Eq. (7) that favors the $\mathbf{b}_c$s with higher inpainting loss on the flow images. This

Fig. 3. *Optical flow image generation on Ski-PTZ and Handheld190k.* We use a homography based on SIFT keypoints to compute rectified images that are provided as input to FlowNet 2.0. (a) Source image warped to the target scene; (b) Target image; (c) Optical flow image highlighting the moving foreground region between the source image and the target image after the background motion is eliminated. In Ski-PTZ, the optical flow images provide strong cues about the foreground object as the scene was captured by rotating cameras, making homography estimation effective. In Handheld190k, because the camera undergoes translations, the homography and optical flow estimates are less accurate, but can nonetheless improve our segmentation performance.

objective regulates bounding box detection by assigning higher confidence to foreground regions where the motion is clearly different from that of the background. As shown in Fig. 3, this lets us ignore the background motion due to a moving camera. Because we only use flow images for intermediate supervision, our model still operates at test time with single images. FlowNet is pretrained on the synthetic MPI Sintel Flow Dataset [66] and, when included, makes our approach superior to other approaches using this level of supervision.

3.6 Implementation Details

Overall Training. All training stages are performed on a single NVIDIA V100 32GB GPU using Adam with a learning rate of 1e-3 and batch size 16. First, the inpainting network is optimized for 200k iterations and subsequently the complete network for an additional 100k iterations. The decoding part of the synthesis network $\mathcal{S}$ uses a reduced learning rate of 1e-4, to prevent occasional diverging behavior. We use an input image resolution of 640px×360px for the Ski-PTZ, Handheld190k and FS-Singles datasets, and 500px×500px for H36M.

We typically use ImageNet-trained weights to initialize our encoder components but can also train them from scratch. We rely on the Focal Spatial Transformers (FST) of [9] to speed up convergence, and expand the erased region in $\mathcal{I}$ in both dimensions by 15% of the size of that predicted by $\mathcal{D}$ to increase the chances of covering the object. Moreover, we discard location offsets outside the image and limit the offset to 1.5 times the bounding box width, as larger ones are already fully covered by the neighboring bounding boxes. We performed a grid search on the relative weights of the loss terms, the offset limits, and λ.

*Detection Network.*We predict one candidate bounding box relative to each grid cell in a regular grid using a fully-convolutional architecture similar to that of YOLO [14]. We use a ResNet-18 backbone [67], which reduces the input dimensionality by a factor 16, from 128×128 to 8×8. The feature size is set to five, two for the bounding box location offset, two for scale, and one for the probability. Each feature output represents the bounding box parameters predicted by one grid cell and the offset is relative to the cell center, as shown in Fig. 2. The estimated probabilities p_c are forced to be positive and to sum to one by using a soft-max activation unit. To prevent this network from constantly predicting bounding boxes at the borders of the image, where the inpainting error would be high, we zero out the outer cell probabilities.

Fig. 4. *Off-the-shelf inpainting results on Ski-PTZ.* (a) Input image with the middle part hidden. We show the inpainting results of (b) [57], (c) [58] trained on ImageNet and (d) [58] on Places2.

Synthesis Network. $\mathcal{S}$ is a bottleneck auto-encoder based on the publicly available implementation of [68]. The encoding part is a 50-layer residual network, and the weights are initialized with ones trained on ImageNet classification. The hidden layer is 856 dimensional, split into a 600 dimensional space and a 256 dimensional space that is replicated spatially to a $512 \times 8 \times 8$ feature map to encode spatially invariant features. The decoding is done with the second half of a U-Net [59] architecture with 64, 128, 256, 512 feature channels in each stage, respectively. The final network layer outputs four feature maps, three to predict the color image $\hat{\mathbf{I}}$ and one for the segmentation mask $\mathbf{S}$.

Inpainting Network. In principle, any off-the-shelf inpainting network trained on large and generic background datasets could be used. For instance, those of [57], [58] can produce very plausible results. However, in domain-specific images, they tend to hallucinate objects, as shown in Fig. 4, and are therefore ill-suited for our purpose. Instead, we train $\mathcal{I}$ from scratch, by reconstructing randomly removed rectangular image regions. Note that it is acceptable for $\mathcal{I}$ not to generalize well to new scenes as it is not needed at test time. We implement it using a 6 layer U-Net model [59] with 8, 16, 32, 64, 128, 256 feature channels in each stage. It takes as input an image from which a selected bounding box region is removed and outputs the entire image with the initially removed patch re-synthesized. It is trained independently from the rest of the pipeline and separately for each dataset by feeding images with randomly occluded regions of varying sizes. In our full pipeline, the weights of the inpainting network are frozen and to remove the image evidence corresponding to the foreground person, the hidden patch in the input image to the inpainting network is selected to be the predicted bounding box expanded by 15% in both dimensions.

Importance Sampling. For the importance sampling function q, we use $\epsilon = 0.001$, which makes the method numerically stable while the probability of choosing a random bounding box stays low, i.e., 6.4% for 64 cells.

Following common practice in the self-supervised segmentation literature [10], [30], [31], the final segmentation masks are generated by a CRF [69] post-processing step that uses both unary and pairwise bilateral potential

TABLE 1
Segmentation Results on the Ski-PTZ, Handheld190k and FS-Singles Datasets

Method	Uses Optical Flow	Ski-PTZ		Handheld190k		FS-Singles	
		J Measure	F Measure	J measure	F measure	J measure	F measure
ReDO [5]	✗	0.43	0.49	0.33	0.38	0.68	0.77
VideoPCA [18]	✓	0.54	0.61	0.47	0.49	0.55	0.69
ARP [3]	✓	0.72	0.82	0.60	0.68	0.56	0.69
Unsup-DilateU-Net [8]	✓	0.63	0.73	0.67	0.75	0.53	0.53
Unsup-Mov-Obj w/o CRF [10]	✓	0.61	0.71	0.60	0.68	0.53	0.73
Unsup-Mov-Obj [10]	✓	0.66	0.76	0.75	0.83	0.68	0.85
Ours w/o optical flow	✗	0.62	0.69	0.75	0.87	0.66	0.72
Ours w/ optical flow	✓	0.70	0.77	0.70	0.79	0.69	0.80
Ours w/ optical flow + CRF	✓	**0.73**	**0.83**	**0.76**	**0.85**	**0.71**	**0.86**

Our method with optical flow consistently outperforms the other self-supervised methods, and ours without flow exceeds or is on par with the other baselines on all three datasets. The best results in each column are shown in bold.

terms. This CRF post-processing does not involve any training; the unary potentials are taken to be the thresholded segmentation masks predicted by our method, and we use the default values of [69] for the pairwise potentials.

4 Experiments

In this section, we first demonstrate the effectiveness of our approach at dealing with unusual motions acquired with PTZ cameras using the Ski-PTZ dataset of [15]. We then introduce a novel Handheld190k dataset depicting people performing 14 everyday activities and a figure skating FS-Singles dataset with different step, spin and jump combinations to demonstrate that our method can handle general moving cameras. For evaluation purposes, we provide ground-truth segmentations for both. Finally, we present the experiments with different loss functions and hyper-parameter study on the Ski-PTZ dataset and analyze the influence of different aspects of our approach on the well-known H36M dataset [16]. Altogether our results show that our approach outperforms the existing self-supervised segmentation techniques, including the ones that exploit temporal cues at inference time [3], [10], approaches the accuracy of supervised methods on objects they have been trained for but seen in different conditions, and outperforms them on previously-unseen objects.

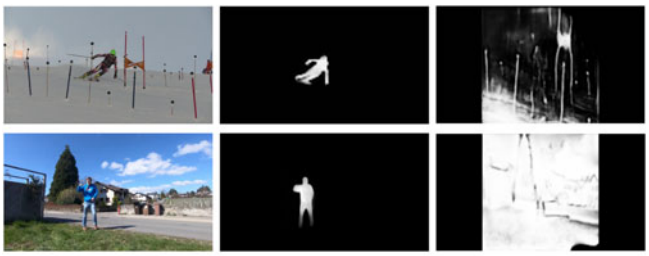


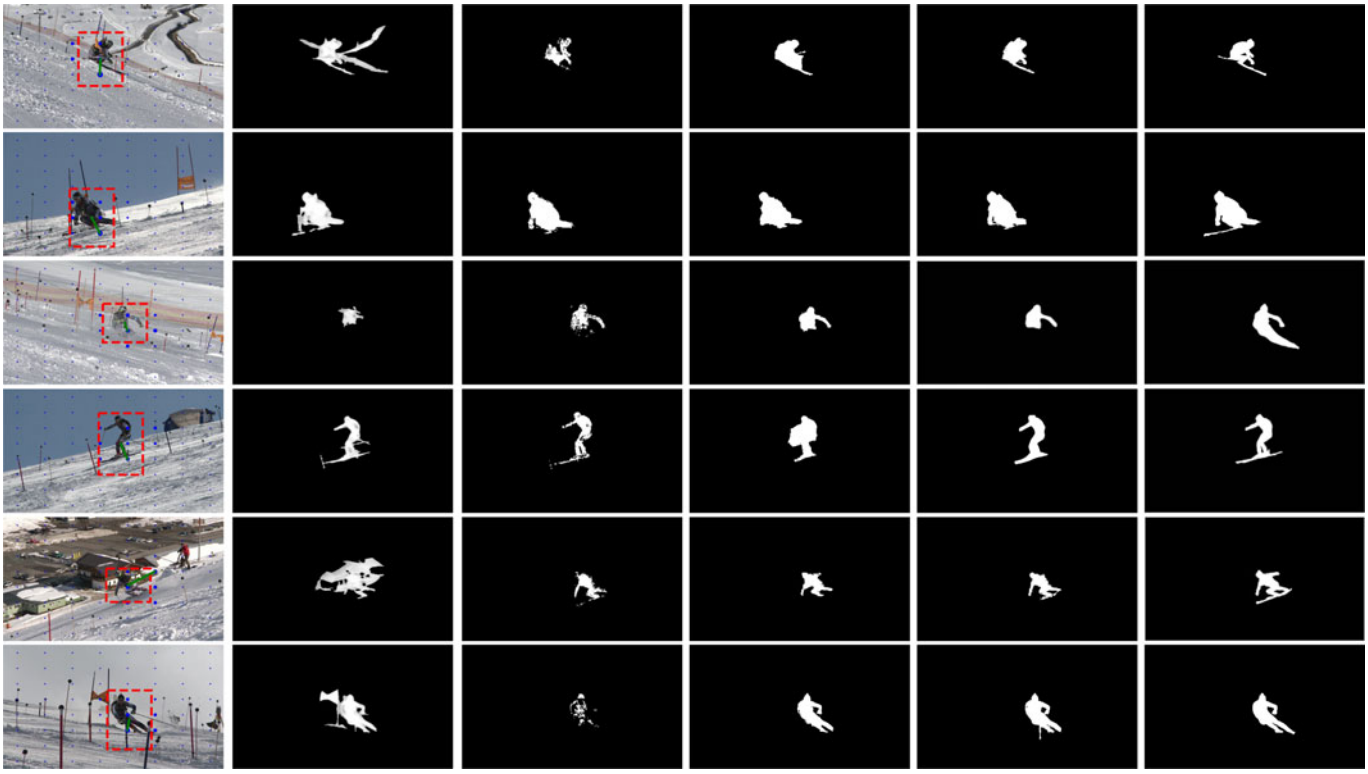
Fig. 5. *Soft segmentation masks generated by our method and PerturbedGAN (P-GAN) [4] on training examples.* Top row: P-GAN mask generated on the Ski-PTZ dataset, the poles and snow patches are segmented as foreground. Bottom row: P-GAN mask generated on the Handheld190k dataset contains the foreground subject together with the ground they are standing on.

4.1 Unusual Activity Filmed Using PTZ-Cameras

Let us first consider the Ski-PTZ dataset of [15] featuring six skiers on a slalom course. We split the videos of six skiers as four/one/one to form training, validation, and test sets, with, respectively, 7800, 1818 and 1908 frames. The intrinsic and extrinsic parameters of the pan-tilt-zoom cameras are constantly adjusted to follow the skier. As a result, nothing is static in the images, the background changes quickly, and there are additional people standing as part of the background. We use the full image as input, evaluate detection accuracy using the available 2D pose annotations and segmentation accuracy by manually segmenting 16 frames from each of the six cameras, which add up to 192 frames in two test sequences. To determine the hyperparameter values, we use 3 manually segmented frames from each of the six cameras, for a total 36 frames in two validation sequences.

In Table 1(left), we compare our approach to several state-of-the-art self-supervised segmentation baselines in terms of the J- and F-measures of [70]. The former is defined as the intersection-over-union between the ground truth segmentation mask and the prediction, while the latter is the harmonic average between the precision and the recall at the mask boundaries. To be fair, we compensate for different segmentation masks quantification levels by a grid search (at 0.05 intervals) to select the best J-measure threshold for each method. Our approach with optical flow outperforms all the baselines in terms of both J- and F-measure. When not using optical flow for training purposes, our approach remains on par with other self-supervised methods despite their use of explicit temporal dependencies. In particular, the comparison to [10] without CRF post-processing shows that our method can achieve the same performance against an optical flow based method without needing a flow-based intermediate supervision. Note that all the baselines are trained on our datasets from scratch using same amount of data, except for [8] that additionally uses a segmentation mask discriminator trained on the combination of the ImageNet VID and YouTube Objects datasets. In other words, while this method is trained in a self-supervised fashion, it relies on a significantly larger amount of data than ours.

In Fig. 5, we compare our method qualitatively to a recent self-supervised method [4]. Note that their generative



(a) Input/Ours detection (b) Unsup-DilateU-Net [8] (c) Unsup-Mov-Obj [10] (d) ARP [3] (e) Ours w/ optical flow (f) Ground truth

Fig. 6. *Qualitative results on the Ski-PTZ.* Example results on the test images. (a) The detection results show the predicted bounding box with red dashed lines, the relative confidence of the grid cells with blue dots and the bounding box center offset with green lines (better viewed on screen). (b) Segmentation mask prediction of [8]. (c)Segmentation mask prediction of [10]. (d) Segmentation mask prediction of [3]. (e) Our segmentation mask prediction. (f) Ground truth segmentation mask. Note that in the third row even though the skier is mostly occluded by snow, our method can detect and segment the visible part of the body. Our method is more accurate than [8] in terms of background removal and outperforms [10] in terms of correctness of the object boundary. Note that in contrast to our method, [3] uses explicit temporal cues at inference time.

model fails to segment the foreground object alone and instead segments background objects and sometimes even the ground. Therefore, we couldn't obtain any reasonable quantitative results for [4]. This method relies on the property that foreground regions can undergo random perturbations without altering the realism of the scene. However, in the Ski-PTZ dataset, some background objects, such as poles, also satisfy this property, and the generator can choose to keep these regions. We also trained [54], another recent self-supervised method that discovers object masks by copying the selected region of the image onto another image with the goal of obtaining a realistic scene, on the Ski-PTZ dataset and obtained implausible masks for the same reason. Since these methods performed poorly on the training samples, we do not provide their quantitative results on the test data.

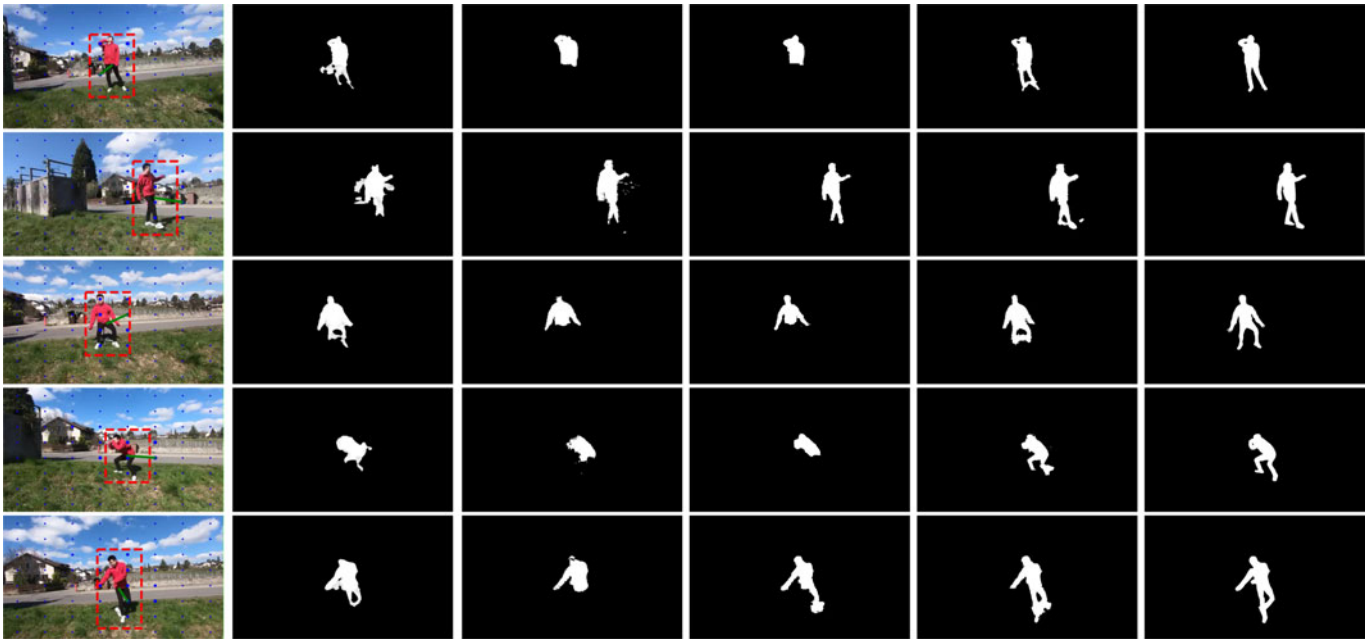
We provide qualitative results in Fig. 6. The probability distribution, visualized as blue dots whose magnitude reflect the predicted likelihood, shows clear peaks on the persons. The limitations include occasional false positives, such as the gates on the slope in close proximity to the skier, reducing precision.

4.2 Activities Captured Using Moving Cameras

To demonstrate the effectiveness of our approach in the presence of general moving cameras, we introduce a new Handheld190k dataset captured by hand-held cameras. It features three training, one validation and one test sequences, comprising 120 855, 23 076 and 46 326 images, respectively, with a single actor performing actions mimicking those in H36M. We manually annotated 112 frames in the validation and 240 frames in the test sequence to provide ground truth segmentation masks, which we believe will be useful for evaluating other self- and weakly-supervised methods. The camera operators moved laterally, to test robustness to camera translation and hand-held rotation. We provide examples of our detection and segmentation results in Fig. 7. Our method is robust to the undirected camera motion and to dynamic background motion, such as branches swinging in the wind and clouds moving, and to salient textures in the background, such as that of the house facade.

To perform a quantitative comparison, we use the 240 manually-segmented test images taken from different motion classes with the subject in many different poses. In Table 1(middle), we compare the results of our approach with those of the same methods as for the ski dataset. Our approach, both with and without optical flow, outperforms all the self-supervised baselines. This is even true for [8] despite its use of a much larger dataset to train a discriminator in an unsupervised fashion and also for [3] that exploits strong temporal dependencies.

We also evaluate our method on a new FS-Singles dataset composed of single men's figure skating videos collected from YouTube. The videos are captured by general moving



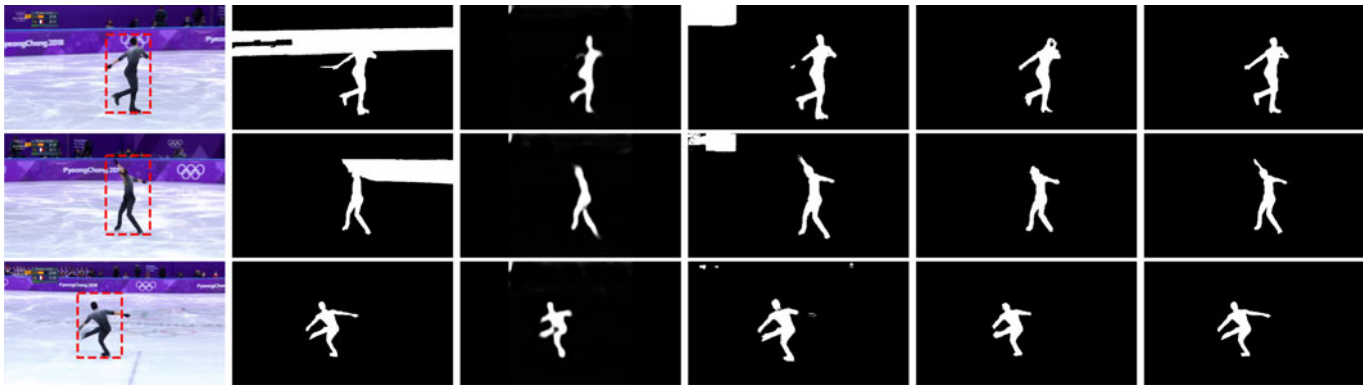
(a) Input/Ours detection (b) Unsup-DilateU-Net [8] (c) Unsup-Mov-Obj [10] (d) ARP [3] (e) Ours w/ optical flow (f) Ground truth

Fig. 7. *Qualitative results on the Handheld190k.* (a) Our detection result. The blue dots coincide with the grid cell centers and their size indicates the confidence of the bounding box proposals. The selected bounding box is illustrated with a red dashed line and the center of the grid cell yielding this proposal is connected to the center of the red box through the green line. (b) Segmentation mask prediction of [8]. (c)Segmentation mask prediction of [10]. (d)Segmentation mask prediction of [3]. (e) Our segmentation mask prediction. (f) Ground truth segmentation mask. Our method can segment the full body of the actor more accurately than [3], [8], [10] despite the other moving objects in the scene such as the clouds and occasionally appearing cars and pedestrians. In some frames, the shadow is also segmented since it moves with the primary object.

cameras and these cameras are usually adjusted fast enough to follow the movements of the skater to keep the subject in the footage. The FS-Singles dataset contains 18 training, 2 validation and 3 test sequences with 10 613, 684 and 1656 frames and 6, 2 and 1 skaters, respectively.

The quantitative experiments on this dataset are conducted using 50 manually-segmented test images including diverse and extreme figure skating motions such as axel jump, sit spin and camel spin. In Table 1(right), we compare our approach to the self-supervised baselines. Our approach with optical flow outperforms all of them. The overall lower scores of the self-supervised methods on this dataset are due to the motion blur caused by the fast movements of the skaters, the low contrast between the ice and certain body parts and the audience in the background. In Fig. 8, we compare the segmentation results of our method to those of the second, third and fourth best-performing methods. Note that our method can accurately detect the skater, even when the scene is cluttered with the audience in the background. The failure cases of our method are mainly due to the low contrast between the ice and the hands and feet of the skater, particularly in extreme spinning poses. Furthermore, the appearance of the skater occasionally matches that of the background people, making it difficult to detect the foreground subject precisely.

Overall, our method that relies on a single image at test time consistently yields the highest scores on all three datasets against other self-supervised methods that operate on



(a) Input/Ours detection (b) ARP [3] (c) ReDO [5] (d) Unsup-Mov-Obj [10] (e) Ours w/ optical flow (f) Ground truth

Fig. 8. *Qualitative results on the FS-Singles.* (a) Our detection result. (b) Segmentation mask prediction of [3]. (c) Segmentation mask prediction of [5]. (d) Segmentation mask prediction of [10]. (e) Our segmentation result. (f) Ground truth segmentation mask. Our method is more accurate than [3] and [10] in terms of removing the background regions.

TABLE 2
MaskRCNN Segmentation Results on the Ski-PTZ, Handheld190k and FS-Singles Datasets

	Ski-PTZ		Handheld190k		FS-Singles	
Method	J Measure	F Measure	J measure	F measure	J measure	F measure
MaskRCNN [1]	**0.73**	0.77	**0.83**	**0.95**	**0.87**	**0.96**
ARP [3]	0.72	0.82	0.60	0.68	0.56	0.69
Unsup-Mov-Obj [10]	0.66	0.76	0.75	0.83	0.68	0.85
Ours w/ flow + CRF	**0.73**	**0.83**	0.76	0.85	0.71	0.86

The direct application of off-the-shelf MaskRCNN on Handheld190k and FS-Singles datasets outperforms the self-supervised methods in Table 1 whereas on Ski-PTZ dataset with unusual motions, our method reaches the maximum F score and is on par with MaskRCNN in J score. This outcome is expected since MaskRCNN is trained on MS-COCO dataset that includes person class as one of the training categories.

single images [4], [5], [8], [54] as well as the ones that require video and use temporal cues at inference time [3], [10], [18].

4.3 Comparison to Supervised Models

In this section we compare our method to MaskRCNN applied in an off-the-shelf manner. Table 2 reports the results of MaskRCNN trained on the MS-COCO dataset [17], which contains the person class in various sports and daily life scenarios, including skiing and skating. On the Ski-PTZ dataset, our method outperforms MaskRCNN. This demonstrates the benefits of self-supervised learning to handle unusual scenarios, where the data differs significantly from that in the publicly-available datasets. On the Handheld190k and FS-Singles datasets, MaskRCNN yields the highest scores, which is not surprising as the test sequences look similar to those in the MS-COCO training set. However, many other object categories are not present in the MS-COCO dataset. In those cases, simply exploiting MaskRCNN becomes non-trivial, because it provides class-specific segmentations, and thus cannot directly handle unknown objects.

To nonetheless evaluate the performance of MaskRCNN in this challenging scenario, we captured an indoor scene featuring many static objects and a moving robot that we aim to segment with a hand-held camera. Fig. 9 compares the detections and segmentation masks output by MaskRCNN for all MS-COCO classes with those obtained with our method. Because the custom robot cannot be associated with any existing MS-COCO category, MaskRCNN tends to split it into multiple objects. Obtaining a consistent mask of the robot would then require parsing these multiple detections. By contrast, our self-supervised approach naturally generalizes to such a previously-unseen object.

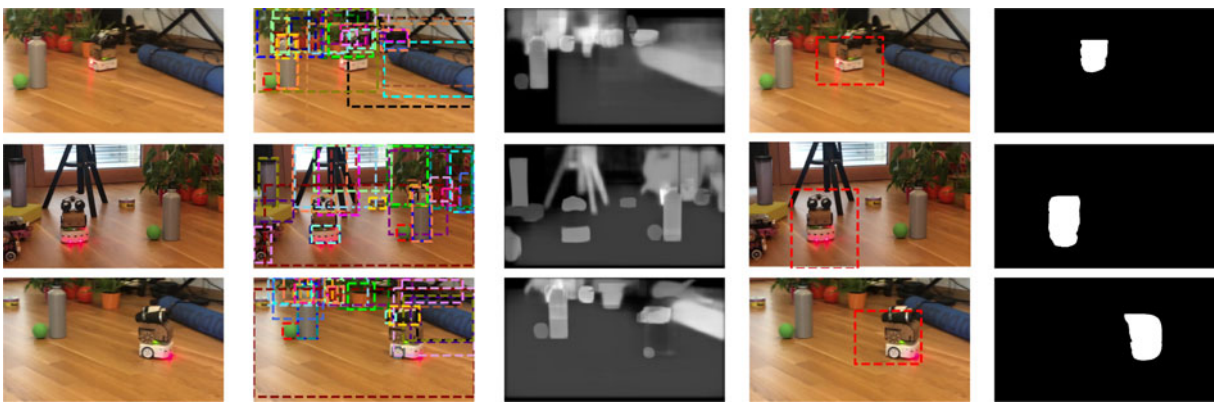
4.4 Ablation Study

In Table 3, we investigate the effectiveness of different mask priors introduced in Section 3.3.3 and ImageNet pre-training on the validation part of the Ski-PTZ dataset. Although L_1 yields better segmentation masks than L_2, it tends to suppress the mask values too strictly, which causes convergence problems. This is mitigated by our L_v prior, which achieves the highest scores in all measures, with consistently reliable results. This demonstrates that imposing regularization on the segmentation masks allows us to obtain sharper masks, removing the noise around the foreground object. We repeated the Ski-PTZ experiment without optical flow extension four times with the best-performing configuration and computed the mean and std on the validation sequences; the J- and F-measure are consistent, respectively, $0.67 \pm 0.004, 0.73 \pm 0.006$.

Table 3 also shows the comparison of using ImageNet or self-supervised weights for network initialization, with only a small performance drop for the latter.

Furthermore, Table 4 compares the performance of our method for different values of hyper-parameters, where the subscript of **b** corresponds to the minimum and maximum size of the bounding box and λ used in our L_v prior is the percentage of the pixels that should be activated in the segmentation mask.

Fig. 10 depicts the influence of the segmentation mask regularizer of Eq. (6). It shows the percentage of segmentation masks that have lower and higher mean values than λ at different training stages. At convergence, the mean segmentation mask value is always higher than λ. Without this regularizer, the mean value of the segmentation mask would grow even larger, causing the mask to incorporate noise and fuzzy regions around the person. Since λ doesn't have to match the exact size of the object, setting it to a small



(a)Input (b)MaskRCNN detection (c)MaskRCNN segmentation (d)Ours detection (e)Ours segmentation

Fig. 9. *Qualitative results of MaskRCNN on a moving robot sequence captured with a handheld camera.* MaskRCNN generally fails to detect the moving robot as a single object and does not yield a segmentation mask with high confidence.

TABLE 3
Analysis of the Mask Prior Effect and ImageNet Pre-Training on the Ski-PTZ Validation Sequences

Setting	Ski-PTZ J Measure	Ski-PTZ F Measure
Ours w/o optical flow w/o prior	0.51	0.53
Ours w/o optical flow w/ L_2 prior	0.61	0.69
Ours w/o optical flow w/ L_1 prior	0.62	0.69
Ours w/o optical flow w/ L_v prior	**0.67**	**0.73**
No ImageNet pre-training, L_v prior	0.60	0.63
Unsupervised pre-training [71], L_v prior	0.62	0.68

We demonstrate the influence of using mask priors to supress the noise surrounding the foreground object and have clear-cut masks. At the bottom part of the table we show the results of using random weights and features from [71] instead of using weights from ImageNet pre-training.

value suffices to trigger the generation of masks early on. We use the same value $\lambda = 0.005$ in all our experiments.

People in a Controlled Environment. We evaluate different aspects of our approach using the H36M dataset [16] that comprises 3.6 million frames and 15 motion classes. It features 5 subjects for training and 2 for validation, seen from different viewpoints against a static background and with good illumination.

On this dataset, we first study the importance of our model choices for training and probabilistic inference. As shown in Fig. 11a, using uniform sampling instead of importance sampling does not converge. Fig. 11b illustrates that joint training of $\mathcal{D}$ with L_{fg} and L_{bg}, instead of our disentangled one, produces bounding boxes that are too large. Fig. 11c shows that using only the background objective leads to small detections that miss the subject and (d) that direct regression without multiple candidates diverges. These failure cases are representative of the behavior on the whole dataset. To explore an alternative

TABLE 4
Hyper-Parameter Study on the Ski-PTZ Validation Sequences

Setting	Ski-PTZ J Measure	Ski-PTZ F Measure
$\mathbf{b}_{[0.1,0.5]}$, $L_{v,\lambda=0.0005}$	0.55	0.55
$\mathbf{b}_{[0.1,0.5]}$, $L_{v,\lambda=0.001}$	0.57	0.62
$\mathbf{b}_{[0.1,0.5]}$, $L_{v,\lambda=0.005}$	0.54	0.56
$\mathbf{b}_{[0.20,0.5]}$, $L_{v,\lambda=0.0005}$	0.61	0.70
$\mathbf{b}_{[0.20,0.5]}$, $L_{v,\lambda=0.001}$	0.60	0.65
$\mathbf{b}_{[0.20,0.5]}$, $L_{v,\lambda=0.005}$	0.61	0.67
$\mathbf{b}_{[0.30,0.5]}$, $L_{v,\lambda=0.0005}$	0.57	0.64
$\mathbf{b}_{[0.30,0.5]}$, $L_{v,\lambda=0.001}$	0.57	0.64
$\mathbf{b}_{[0.30,0.5]}$, $L_{v,\lambda=0.005}$	0.57	0.63
$\mathbf{b}_{[0.20,0.60]}$, $L_{v,\lambda=0.0005}$	0.61	0.69
$\mathbf{b}_{[0.20,0.60]}$, $L_{v,\lambda=0.001}$	0.61	0.68
$\mathbf{b}_{[0.20,0.60]}$, $L_{v,\lambda=0.005}$	0.62	0.68
$\mathbf{b}_{[0.20,0.70]}$, $L_{v,\lambda=0.0005}$	0.62	0.67
$\mathbf{b}_{[0.20,0.70]}$, $L_{v,\lambda=0.001}$	0.59	0.66
$\mathbf{b}_{[0.20,0.70]}$, $L_{v,\lambda=0.005}$	0.62	0.65
$\mathbf{b}_{[0.20,0.80]}$, $L_{v,\lambda=0.0005}$	0.61	0.68
$\mathbf{b}_{[0.20,0.80]}$, $L_{v,\lambda=0.001}$	**0.67**	**0.73**
$\mathbf{b}_{[0.20,0.80]}$, $L_{v,\lambda=0.005}$	0.61	0.66

In this table we analyze the effectiveness of our hyper-parameter choice for the minimum and maximum bounding box sizes (given in square brackets as $\mathbf{b}_{[scale_{min},scale_{max}]}$) as well as the threshold λ for the L_v loss. We conduct these experiments using our approach without optical flow.

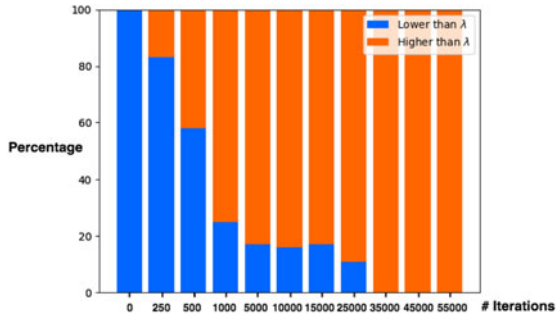


Fig. 10. *Impact of the segmentation mask regularizer of Eq. (6).* Early in the training, a high percentage of masks have a mean value lower than λ. When the model converges, all masks have a mean value above this threshold.

strategy to Monte Carlo-based sampling, we replaced the importance sampling in our method with the categorical reparameterization used in [7]. Since both strategies approximate the same objective, they had similar outcomes with a difference in the convergence speed and detection performance. To this end, we tried Gumbel-Softmax distribution [72]. We found out that setting the temperature to 0.1 yielded the best results. Increasing this value has a similar effect as increasing the ϵ in Eq. (10) and approaches uniform sampling. Our experiments show that Gumbel-Softmax based categorical reparameterization did not lead to faster convergence and in fact degraded the detection performance as shown in Fig. 11e. Our method delivers a $\text{mAP}_{0.5}$ score of 0.58 which is significantly higher than the $\text{mAP}_{0.5}$ score of 0.30 obtained by using Gumbel-Softmax as our sampling strategy. Furthermore, our importance sampling approach is simpler than [7] and is an unbiased estimator. It does not need custom layers that behave differently in the forward and backwards passes during optimization, which is the case for the Gumbel-Softmax categorical reparameterization. Please note that direct comparison to [7] is not possible since it requires monochromatic backgrounds. Therefore, it does not apply to the Ski-PTZ, Handheld190k and FS-Singles datasets and was demonstrated only on simple synthetic cases, such as MNIST and Atari games, with multiple objects that go beyond the scope of our approach. Finally, Fig. 11f demonstrates that our full model using the disentangled training strategy and importance sampling can accurately detect the person and estimate tighter bounding boxes.

In Table 5, we evaluate detection accuracy on H36M and Ski-PTZ. Note that our method delivers an $\text{mAP}_{0.5}$ score that is significantly better than that of the general YOLO [14] detector trained on MS-COCO dataset. On the left side of Table 5, we compare our detection accuracy to that of a very recent self-supervised deep learning method [9]. Our slightly lower accuracy stems from not explicitly assuming a static background, which [9] does. While valid in a lab, this assumption results in total failure in outdoor scenes with moving backgrounds. Notably, our method is robust to undirected camera motion and to dynamic background motion, and works equally well for the very different domains of skiing and every-day activities.

4.5 Discussion

Optical Flow. As noted in [10], the motion-based segmentation methods that require computing the optical flow

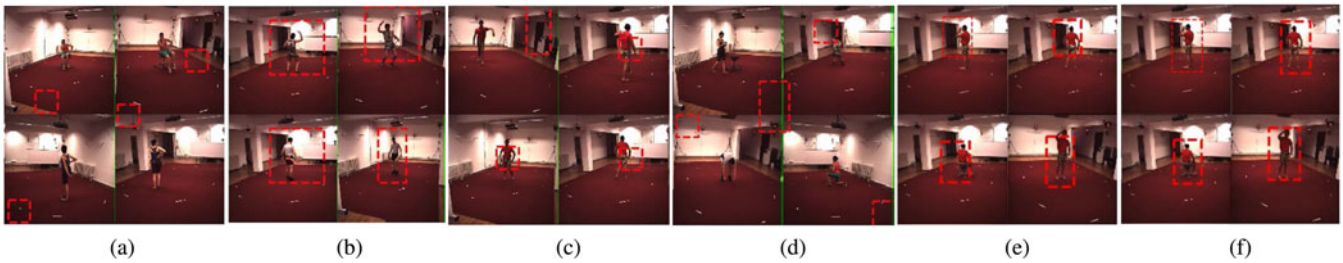


Fig. 11. *Ablation study on H36M.* (a) Uniform sampling does not converge. (b) Joint training of L_{fg} and L_{bg} (c) only L_{bg} (d) direct regression of a single bounding box using L_{fg} and L_{bg} (e) Gumbel-Softmax (f) Ours.

between consecutive images can be error-prone due to the irregular or insufficient movement of the object. This gives us leverage against approaches that rely only on optical flow since our method can reliably detect the foreground object from single RGB images and uses optical flow only as an extension during training time. In Fig. 12, we present possible failure cases that can occur when the optical flow partially covers the object due to its static parts. Since the inpainting module in [10] tries to reconstruct the masked optical flow, it is prone to errors whenever the optical flow image is unreliable. It can be seen that our method can accurately segment the object in this case. Hence, based on our experimental evidence in Table 1, optical flow should always be used if available and in combination with the RGB image.

Multiple People. Although our focus is on handling single objects or persons, our probabilistic framework can handle several at test time by sampling more than once. Fig. 13 shows the predicted cell probability as blue dots whose size is proportional to the probability. The fully-convolutional architecture operates locally and thereby predicts a high person probability close to both subjects. As a result, both the detection and segmentation results remain accurate as long as the individuals are sufficiently separated. Note that the model used for this experiment was still trained on single subjects. In future work, we will attempt self-supervised training of multiple interacting people, which has so far only been established in controlled environments.

Other Object Categories. In this section, we investigate the applicability of our method to standard benchmarks with other object categories. The existing object detection datasets SegTrackV2 and FBMS59 comprise multiple objects, which we do not support. Therefore, we demonstrate the qualitative performance of our method on the standard DAVIS2016 [70] benchmark that consists of various object categories such as car, cow and goat. DAVIS contains 30 training and 20 testing sequences, which are very short compared to other benchmarks suitable for deep-learning based methods. We follow the standard procedure and use the validation sequences for evaluation. Since our method does not require any annotations, we train and test on the validation sequences with an average of 70 frames per video. So far, we have evaluated our method on datasets with human subjects. Therefore we pick non-human object categories in the DAVIS2016 validation dataset to show that our method is not specific to a particular object type. As shown in Fig. 14, our performance on DAVIS2016 varies, depending on the length and footage of the sequence. However, we do not expect our method to compete with approaches tuned for short video snippets. Many of these short sequences include objects that move slowly, remaining mostly in the same image region. This makes them easy to inpaint, thus violating our assumptions A1 and A2 (Section 3.1). Fig. 14(top) shows a successful segmentation result on a longer sequence with a moving object. Fig. 14(middle) illustrates a partially successful case that occurs when the location of the object changes with the background elements and the content of the scene in a short video clip provides significant clue about the reconstruction of the foreground object. In Fig. 14(bottom), we present a failure case that occurs when our method is applied to very short videos with negligible object displacement. In this case, our inpainting network can reconstruct the foreground object together with the background region, which causes holes in

TABLE 5
Detection Results on the H36M and Ski-PTZ Datasets

H36M dataset	
Method	$\text{mAP}_{0.5}$
NSD [9]	**0.710**
Ours	0.580

Ski-PTZ	
Method	$\text{mAP}_{0.5}$
YOLOv3 [14]	0.155
Ours	**0.520**

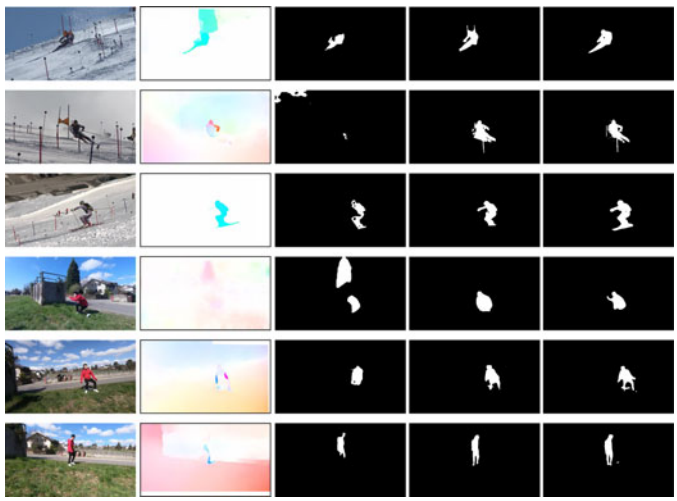


Fig. 12. *Optical flow failure.* When the optical flow image cannot be used to find the complete outline of the subject, for example because some part of their body is static, our method can still segment the moving object from a single RGB image, whereas [10] tends to yield poor results. To highlight the effect of using optical flow, we present the raw segmentation predictions of [10] and ours, before the post-processing step.

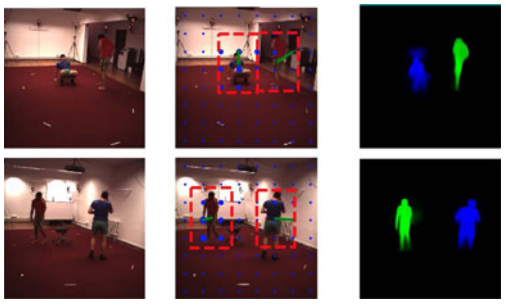


Fig. 13. *Multi-person detection and segmentation results*, generated by sampling our model multiple times. As the model is trained on single persons this only works for non-intersecting cases.

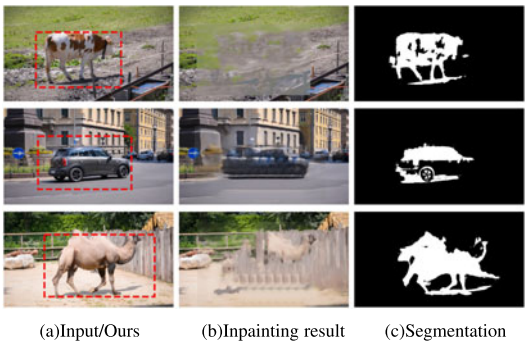


Fig. 14. *Examples of qualitative results on DAVIS2016 [70] validation sequences.* Top row: successful segmentation of the moving object. Middle row: partially successful case in which the background scene carries information about the moving object's location. Bottom row: poor segmentation result that occurs when our inpainting network can resonstruct certain parts of the moving object due to its slow motion.

the regions of the segmentation mask that are already reconstructed by the inpainting network.

In short, DAVIS2016 features only few videos per category, with each video being short, making them ill-suited to deep-learning based self-supervised approaches that exploit large unlabeled video collections. By contrast, we contribute new benchmarks with manual annotations for quantitative evaluation and three very different settings with significantly more and longer training videos that can be used to evaluate future self-supervised deep learning-based segmentation methods.

5 Conclusion

We have proposed a self-supervised method for object detection and segmentation that lends itself for application in domains where general purpose detectors fail. Our core contributions are the Monte Carlo-based optimization of proposal-based detection, new foreground and background objectives, and their joint training on unlabeled videos captured by static, rotating and handheld cameras. Our experiments demonstrate that, even if trained only on single persons, our approach generalizes to multi-person detection, as long as the persons are sufficiently separated. In contrast to many existing solutions [3], [18], [49], [50], our approach does not exploit temporal cues at test time. In the future, we will integrate temporal dependencies explicitly, which will facilitate addressing the scenario where multiple people interact closely, by incorporating physics-inspired constraints enforcing plausible motion.

References

[1] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2980–2988.
[2] S. Eslami *et al.*, "Attend, infer, repeat: Fast scene understanding with generative models," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2016, pp. 3233–3241.
[3] Y. J. Koh and C.-S. Kim, "Primary object segmentation in videos based on region augmentation and reduction," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 7417–7425.
[4] A. Bielski and P. Favaro, "Emergence of object segmentation in perturbed generative models," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2019, pp. 7254–7264.
[5] M. Chen, T. Artieres, and L. Denoyer, "Unsupervised object segmentation by redrawing," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2019, pp. 12705–12716.
[6] A. Kosiorek, H. Kim, Y. W. Teh, and I. Posner, "Sequential attend, infer, repeat: Generative modelling of moving objects," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2018, pp. 8615–8625.
[7] E. Crawford and J. Pineau, "Spatially invariant unsupervised object detection with convolutional neural networks," in *Proc. 33rd AAAI Conf. Artif. Intell.*, 2019, pp. 3412–3420.
[8] I. Croitoru, S. V. Bogolin, and M. Leordeanu, "Unsupervised learning of foreground object segmentation," *Int. J. Comput. Vis.*, vol. 127, pp. 1279–1302, 2019.
[9] H. Rhodin, V. Constantin, I. Katircioglu, M. Salzmann, and P. Fua, "Neural scene decomposition for human motion capture," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 7695–7705.
[10] Y. Yang, A. Loquercio, D. Scaramuzza, and S. Soatto, "Unsupervised moving object detection via contextual information separation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 879–888.
[11] Z. Lin *et al.*, "SPACE: Unsupervised object-oriented scene representation via spatial attention and decomposition," in *Proc. Int. Conf. Learn. Representations*, 2020. [Online]. Available: https://openreview.net/forum?id=rkl03ySYDH
[12] Y. Benny and L. Wolf, "OneGAN: Simultaneous unsupervised learning of conditional image generation, foreground segmentation, and fine-grained clustering," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 514–530.
[13] X. Lu, W. Wang, J. Shen, Y.-W. Tai, D. J. Crandall, and S. C. H. Hoi, "Learning video object segmentation from unlabeled videos," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 8957–8967.
[14] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 779–788.
[15] H. Rhodin *et al.*, "Learning monocular 3D human pose estimation from multi-view images," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 8437–8446.
[16] C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu, "Human3.6M: Large scale datasets and predictive methods for 3D human sensing in natural environments," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 7, pp. 1325–1339, Jul. 2014.
[17] T.-Y. Lin *et al.*, "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 740–755.
[18] O. Stretcu and M. Leordeanu, "Multiple frames matching for object discovery in video," in *Proc. Brit. Mach. Vis. Conf.*, 2015, pp. 186.1–186.12.
[19] J. Cheng, Y. H. Tsai, S. Wang, and M. H. Yang, "SegFlow: Joint learning for video object segmentation and optical flow," in *Proc. Int. Conf. Comput. Vis.*, 2017, pp. 686–695.
[20] H. Song, W. Wang, S. Zhao, J. Shen, and K.-M. Lam, "Pyramid dilated deeper ConvLSTM for video salient object detection," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 744–760.
[21] G. Bhat *et al.*, "Learning what to learn for video object segmentation," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 777–794.
[22] Y. Li, Z. Shen, and Y. Shan, "Fast video object segmentation using the global context module," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 735–750.
[23] X. Lu, W. Wang, M. Danelljan, T. Zhou, J. Shen, and L. V. Gool, "Video object segmentation with episodic graph memory networks," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 661–679.
[24] S. Seo, J.-Y. Lee, and B. Han, "URVOS: Unified referring video object segmentation network with a large-scale benchmark," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 208–223.

[25] H. Seong, J. Hyun, and E. Kim, "Kernelized memory network for video object segmentation," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 629–645.
[26] Z. Yang, Y. Wei, and Y. Yang, "Collaborative video object segmentation by foreground-background integration," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 332–348.
[27] F. Perazzi, O. Wang, M. Gross, and A. Sorkine-Hornung , "Fully connected object proposals for video segmentation," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 3227–3234.
[28] Y. T. Hu, J. B. Huang, and A. G. Schwing, "Unsupervised video object segmentation using motion saliency-guided spatio-temporal propagation," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 813–830.
[29] S. D. Jain, B. Xiong, and K. Grauman, "FusionSeg: Learning to combine motion and appearance for fully automatic segmentation of generic objects in videos," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2117–2126.
[30] S. Li, B. Seybold, A. Vorobyov, A. Fathi, Q. Huang, and C.-C. J. Kuo, "Instance embedding transfer to unsupervised video object segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 6526–6535.
[31] S. Li, B. Seybold, A. Vorobyov, X. Lei, and C.-C. J. Kuo, "Unsupervised video object segmentation with motion-based bilateral networks," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 215–231.
[32] X. Lu, W. Wang, C. Ma, J. Shen, L. Shao, and F. Porikli, "See more, know more: Unsupervised video object segmentation with co-attention siamese networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 3618–3627.
[33] C. Wang *et al.*, "Densefusion: 6D object pose estimation by iterative dense fusion," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 3338–3347.
[34] Z. Yang, Q. Wang, L. Bertinetto, W. Hu, S. Bai, and P. Torr, "Anchor diffusion for unsupervised video object segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 931–940.
[35] W. Wang *et al.*, "Learning unsupervised video object segmentation through visual attention," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 3059–3069.
[36] L. Zhang, J. Zhang, Z. Lin, R. Měch, H. Lu, and Y. He, "Unsupervised video object segmentation with joint hotspot tracking," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 490–506.
[37] M. Zhen *et al.*, "Learning discriminative feature with CRF for unsupervised video object segmentation," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 445–462.
[38] X. Wang, A. Jabri, and A. A. Efros, "Learning correspondence from the cycle-consistency of time," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 2561–2571.
[39] M. Cho, S. Kwak, C. Schmid, and J. Ponce, "Unsupervised object discovery and localization in the wild: Part-based matching with bottom-up region proposals," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 1201–1210.
[40] X.-S. Wei, C.-L. Zhang, J. Wu, C. Shen, and Z.-H. Zhou, "Unsupervised object discovery and co-localization by deep descriptor transforming," *Pattern Recognit.*, vol. 88, pp. 113–126, 2019.
[41] P. Tokmakov, K. Alahari, and C. Schmid, "Learning motion patterns in videos," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 531–539.
[42] P. Tokmakov, K. Alahari, and C. Schmid, "Learning video object segmentation with visual memory," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 4491–4500.
[43] Y. J. Lee, J. Kim, and K. Grauman, "Key-segments for video object segmentation," in *Proc. Int. Conf. Comput. Vis.*, 2011, pp. 1995–2002.
[44] A. Papazoglou and V. Ferrari, "Fast object segmentation in unconstrained video," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2013, pp. 1777–1784.
[45] A. Faktor and M. Irani, "Video segmentation by non-local consensus voting," in *Proc. Brit. Mach. Vis. Conf.*, 2014. [Online]. Available: http://dx.doi.org/10.5244/C.28.21
[46] M. Keuper, B. Andres, and T. Brox, "Motion trajectory segmentation via minimum cost multicuts," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 3271–3279.
[47] W. Wang, J. Shen, and F. Porikli, "Saliency-aware geodesic video object segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 3395–3402.
[48] E. Haller and M. Leordeanu, "Unsupervised object segmentation in video by efficient selection of highly probable positive features," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 5095–5103.
[49] O. Barnich and M. V. Droogenbroeck, "ViBe: A universal background subtraction algorithm for video sequences," *IEEE Trans. Image Process.*, vol. 20, no. 6, pp. 1709–1724, Jun. 2011.
[50] C. Russell, R. Yu, and L. Agapito, "Video pop-up: Monocular 3D reconstruction of dynamic scenes," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 583–598 .
[51] D. Sun, X. Yang, M.-Y. Liu, and J. Kautz, "PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 8934–8943.
[52] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu, "Spatial transformer networks," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2015, pp. 2017–2025.
[53] D. Pathak, R. Girshick, P. Dollár, T. Darrell, and B. Hariharan, "Learning features by watching objects move," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 6024–6033.
[54] R. Arandjelovic and A. Zisserman, "Object discovery with a copy-pasting GAN," vol. abs/1905.11369, 2019.
[55] P. Baqué, F. Fleuret, and P. Fua, "Deep occlusion reasoning for multi-camera multi-target detection," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 271–279.
[56] M. Leordeanu, *Unsupervised Learning in Space and Time*. Berlin, Germany: Springer, 2020.
[57] D. Pathak, P. Krähenbühl, J. Donahue, T. Darrell, and A. A. Efros, "Context encoders: Feature learning by inpainting," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 2536–2544.
[58] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. S. Huang, "Generative image inpainting with contextual attention," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 5505–5514.
[59] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Conf. Med. Image Comput. Comput. Assisted Interv.*, 2015, pp. 234–241.
[60] R. Y. Rubinstein and D. P. Kroese, *Simulation and the Monte Carlo Method*. Hoboken, NJ, USA: Wiley, 2016.
[61] H. Kahn and A. W. Marshall, "Methods of reducing sample size in Monte Carlo computations," *J. Operations Res. Soc. America*, vol. 1, no. 5, pp. 263–278, 1953.
[62] D. Koller and N. Friedman, *Probabilistc Graphical Models*. Cambridge, MA, USA: MIT Press, 2009.
[63] P. M. Glynn, "Likelihood ratio gradient estimation for stochastic systems," *Commun. ACM*, vol. 33, no. 10, pp. 75–84, 1990.
[64] R. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Mach. Learn.*, vol. 8, pp. 229–256, 1992.
[65] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox, "FlowNet 2.0: Evolution of optical flow estimation with deep networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1647–1655.
[66] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black, "A naturalistic open source movie for optical flow evaluation," in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 611–625.
[67] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.
[68] H. Rhodin, M. Salzmann, and P. Fua, "Unsupervised geometry-aware representation for 3D human pose estimation," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 765–782.
[69] P. Krähenbühl and V. Koltun, "Efficient inference in fully connected CRFs with Gaussian edge potentials," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2011, pp. 109–117.
[70] 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," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 724–732.
[71] Z. Wu, Y. Xiong, S. X. Yu, and D. Lin, "Unsupervised feature learning via non-parametric instance discrimination," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 3733–3742.
[72] E. Jang, S. Gu, and B. Poole, "Categorical reparameterization with Gumbel-Softmax," in *Proc. 5th Int. Conf. Learn. Representations*, 2017. [Online]. Available: https://openreview.net/forum?id=rkE3y85ee

Isinsu Katircioglu received the BSc degree in computer science from Middle East Technical University, Ankara, Turkey, in 2014, and the MSc degree from EPFL, Lausanne, Switzerland, in 2016. Currently, she is working toward the PhD degree in Computer Vision Laboratory, EPFL, Lausanne, Switzerland. Her research interests include deep learning, 3D pose estimation, and motion prediction.

Helge Rhodin received the PhD degree in computer science from Max-Plack-Institute for Informatics, Saarbrücken, Germany, in 2016. He later joined the Computer Vision Lab, EPFL as postdoctoral researcher in 2017. He is currently an assistant professor at UBC in the Computer Vision Lab and the Imager Lab. His research interests range from 3D computer vision, over machine learning, to computer graphics and augmented reality.

Victor Constantin received the MSc degree in computer science from EPFL, Lausanne, Switzerland, in 2016. He is currently working as a research engineer in Computer Vision Laboratory, EPFL. His research interests include deep learning, 3D pose estimation, and garment draping.

Jörg Spörri received the MSc degree from ETH Zurich, Zürich, Switzerland, in 2009, and the PhD degree from the University of Salzburg, Salzburg, Austria, in 2012. He currently works with the Balgrist University Hospital (University of Zurich) as the head of Sports Medical Research. His research interests include prevention and rehabilitation of sports injuries with special emphasis on epidemiology, load management, movement biomechanics, musculoskeletal imaging, and exercise physiology.

Mathieu Salzmann received the MSc and PhD degrees from EPFL, Lausanne, Switzerland, in 2004 and 2009, respectively. He then joined the International Computer Science Institute and the EECS Department with the University of California at Berkeley as a postdoctoral fellow, and later the Toyota Technical Institute at Chicago as a research assistant professor. He was a senior researcher and research leader with NICTA's Computer Vision Research Group in Canberra. He is currently a senior researcher with Computer Vision Lab, EPFL and an artificial intelligence engineer with ClearSpace.

Pascal Fua (Fellow, IEEE) received the engineering degree from Ecole Polytechnique, Paris, France, in 1984, and the PhD degree in computer science from the University of Orsay, Orsay, France, in 1989. He joined EPFL in 1996 as a professor with the School of Computer and Communication Science. He is the head of the Computer Vision Lab. He has (co) authored more than 300 publications in refereed journals and conferences and received several ERC grants. He has been an associate editor of the *IEEE Transactions for Pattern Analysis and Machine Intelligence*.