HAL
open science

Learning to Better Segment Objects from Unseen Classes with Unlabeled Videos

Yuming Du, Yang Xiao, Vincent Lepetit

To cite this version:

Yuming Du, Yang Xiao, Vincent Lepetit. Learning to Better Segment Objects from Unseen Classes with Unlabeled Videos. International Conference on Computer Vision, Oct 2021, Online, Canada. hal-03482255

HAL Id: hal-03482255

https://hal.science/hal-03482255v1

Submitted on 15 Dec 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

arXiv:2104.12276v2 [cs.CV] 20 Aug 2021

Learning to Better Segment Objects from Unseen Classes with Unlabeled Videos

Yuming Du Yang Xiao Vincent Lepetit
LIGM, Ecole des Ponts, Univ Gustave Eiffel, CNRS, Marne-la-vallée, France
{yuming.du, yang.xiao, vincent.lepetit}@enpc.fr
https://dulucas.github.io/Homepage/gbopt/

Abstract

The ability to localize and segment objects from unseen classes would open the door to new applications, such as autonomous object learning in active vision. Nonetheless, improving the performance on unseen classes requires additional training data, while manually annotating the objects of the unseen classes can be labor-extensive and expensive. In this paper, we explore the use of unlabeled video sequences to automatically generate training data for objects of unseen classes. It is in principle possible to apply existing video segmentation methods to unlabeled videos and automatically obtain object masks, which can then be used as a training set even for classes with no manual labels available. However, our experiments show that these methods do not perform well enough for this purpose. We therefore introduce a Bayesian method that is specifically designed to automatically create such a training set: Our method starts from a set of object proposals and relies on (non-realistic) analysis-by-synthesis to select the correct ones by performing an efficient optimization over all the frames ***simultaneously****. Through extensive experiments, we show that our method can generate a high-quality training set which significantly boosts the performance of segmenting objects of unseen classes. We thus believe that our method could open the door for open-world instance segmentation using abundant Internet videos.*

1. Introduction

Instance segmentation models are now able to predict the masks of objects of known classes in query images [19, 50, 55], providing rich information for many downstream applications such as scene understanding [17, 47] and robot grasping [53, 59, 62]. Unfortunately, existing instance segmentation methods perform poorly on new classes [12]. This is an obstacle to the development of autonomous systems evolving in open worlds where there will always be objects that do not belong to known classes. Being able to detect and segment these objects would be the starting point of learning to grasp and manipulate them, for example.

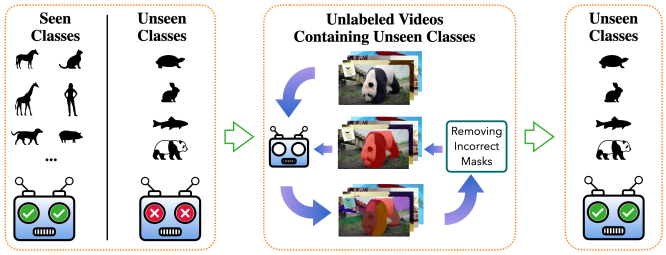

Figure 1. Starting from an instance segmentation model trained on some classes, we want to learn to localize and segment objects from new classes without any human label. We do this by using unlabeled videos, which are an abundant source of data. Our approach can automatically detect and select the object masks in the videos. We then use the selected masks to retrain the initial model that can then localize and segment objects from the new classes in still frames without losing performance on the old ones.

As Figure 1 illustrates, our goal is therefore to automatically improve the performance of instance segmentation models on static images containing objects from new classes without human intervention. This is in contrast with previous works that aiming at limiting the manual labeling burden for object segmentation by using bounding boxes only [21, 28, 65] or developing few-shot techniques [15, 60], but still require human intervention for new classes.

More specifically, we do *not* aim at predicting the categories of these new objects, but focus on robustly localizing and accurately segmenting them. In this sense, our work is therefore more related to recent object discovery methods, which attempt to segment objects without manual segmentation labels by grouping pixels according to some criterion [3, 18, 33, 45, 57]. However, these approaches are still very fragile as they can be easily affected by low-level perturbations in color, lighting, or texture.

Our strategy is therefore to rely on unlabeled video sequences, as unlabeled videos can be acquired without effort

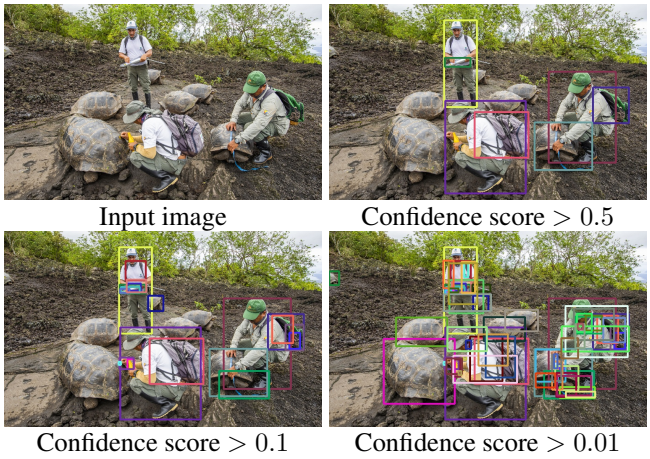

Figure 2. The class 'tortoise' does not belong to COCO classes. By decreasing the confidence score threshold of Mask R-CNN [19] trained on COCO, we can eventually localize and segment all the tortoises at the price of introducing many false positives. We filter these false positives using unlabeled video data.

while providing rich information. The general idea of using videos for self-supervision is not new [11, 23, 30, 34, 54, 56]. Here, we consider video sequences to automatically generate masks for the objects visible in their frames. Using these masks for training an instance segmentation model should then make it perform better on new objects visible in the videos, even if no human intervention was provided in the process. In our experiments, we still provide the videos, but it is possible to imagine a system that captures videos by itself.

Unfortunately, our early experiments showed that state-of-the-art video segmentation methods [14, 35, 42, 30, 51] were not sufficient for this purpose. We therefore developed our own method for automatically creating object masks from videos. Note that our task is subtly different from video segmentation methods, which aim at keeping track of the objects over consecutive frames. For our own goal, this is not needed and we only focus on correctly localizing and segmenting the objects in each frame. Like some video segmentation methods [35], we start from mask hypotheses for the objects visible in the videos obtained with a pre-trained class-agnostic instance segmentation. Such model generalizes to objects from unseen classes to some extent, but at the price of introducing many false positives [41]. Figure 2 shows that when the confidence threshold is lowered, both the detections of objects from unseen classes and incorrect detections get accepted.

It is however possible to filter the incorrect masks using information provided by the videos. Some methods [8, 29, 35, 37, 44] rely on tracklets to track and filter the masks. We claim that such strategy is not optimal, especially for our goal: Image background is underexploited, while it can be very useful by indicating the *absence* of objects; The optical flow is not used or underexploited, while it gives strong cues about moving objects in videos. To fully exploit the background and motion information across unlabeled video frames, we therefore developed a (non-realistic) analysis-by-synthesis approach. Using a Bayesian framework, we derive an objective function with an additional non-overlapping constraint. The objective function, which consists of three loss terms, is designed to explore the background and motion information to remove incorrect masks and select masks that are temporally consistent over the entire video. The non-overlapping constraint comes from the fact that one pixel can at most belong to one object in the image and helps to reject some false positives. Moreover, we provide a two-stage optimization algorithm to optimize this objective function efficiently.

To evaluate our approach, we created a novel dataset named Unseen-VIS based on the YouTube Video Instance Segmentation (YouTube-VIS) dataset [61], which contains objects that do *not* belong to COCO classes. Starting from the raw masks generated on the training part of Unseen-VIS using a class-agnostic Mask R-CNN pretrained on COCO dataset, we apply our method to automatically select the correct masks. We demonstrate that using these masks can boost the performance of the Mask R-CNN on the test set of Unseen-VIS without losing performance on COCO classes.

To summarize our contributions:

- We propose a Bayesian method to generate high-quality masks on unlabeled videos containing unseen classes;
- We create a benchmark to evaluate the quality of generated masks on unlabeled videos;
- We demonstrate on our benchmark that our proposed method can be used to improve the performance of an instance segmentation model on unseen classes.

2. Related Work

In this section, we first review recent works on instance segmentation from color images, especially those targeting new classes. We also review self-learning methods on unlabeled images. Finally, we review several works for video object segmentation as this topic is closely related to our approach.

Image-level Instance Segmentation. While state-of-the-art methods for object detection and segmentation [19, 50, 55] rely on large amounts of manually labeled images, a few works aim at reducing the annotation burden for learning to detect and segment object classes. They however still require manual annotations: Weakly-supervised methods require no mask annotations but bounding boxes annotations [21, 28, 65] and few-shot methods require a (small) number of manual object masks [15, 60].

Self-Learning on Unlabeled Data. Recently, several methods have been proposed to explore self-supervision using unlabeled static images. They use data distillation [46], unlabeled images from the web [31], consistency across image flipping [24], or an estimate of the uncertainty of prediction [38]. While such approaches are very interesting, unlabelled videos are easily available and have the potential to make the results much more reliable. We compare our method against the most representative methods in our experiments and show we achieve much better performance.

Another type of approach proceeds in a bottom-up fashion by grouping pixels with similar colors or image features to generate masks [3, 18, 33, 45, 57]. However, this can be easily affected by local textures or colors, and some of these methods have been demonstrated on synthetic images only. Because it starts from a pre-trained instance segmentation model, our approach is much more robust.

Like us, some methods leverage unlabeled videos for on urban scene segmentation [6] and face and human detection[25]. However, these works only focus on how to enhance the model performance on the existing classes and do not consider novel classes. [40] makes use of stereo video data together with depth information to reconstruct a static background, then object proposals are generated from the foreground regions by subtraction. While this is an interesting approach, they require depth data and static backgrounds.

Video Object Segmentation. Our work also relates to One-Shot, Zero-Shot video object segmentation (VOS) and saliency-based video object segmentation. One-Shot VOS aims at segmenting the objects in the video when the ground truth segmentation is given for a frame. One-Shot VOS methods typically warp the provided segmentation to other frames [4, 30, 54]. They thus require manual annotations and cannot generate new predictions if new objects appear. Some Zero-Shot methods [9, 39, 51, 52] are trained with video labels on seen classes and are able to generalize to unseen classes, but video labeling is very labor-extensive. Some methods look for salient regions in videos [7, 10, 11, 14, 22, 27, 34, 42], as salient regions tend to correspond to objects. However, saliency prediction has two major limitations for our purpose: (a) It can be fooled by non-salient camouflaged objects. (b) Two adjacent objects would be merged into a single salient region, while we want to identify them individually.

Like us, a few methods already adopt a proposal-based approach [2, 35, 37], but rely on classical tracking algorithms to track the proposals such as tracklets. By contrast, our approach relies on (non-realistic) analysis-by-synthesis. Analysis-by-synthesis is an old concept in computer vision, but has been recently increasingly popular. By aiming at explaining the whole image, it can exploit more information. Moreover, it is conceptually simple and requires few easy-to-fix hyper-parameters. We show in our experiments that our method performs better than the state-of-the-art video object segmentation method UnOVOST [35] for the purpose of generating object masks.

3. Method

As discussed in Section 1, our goal is to improve the performance of a pre-trained class-agnostic instance segmentation on unseen classes. Our pipeline consists of three steps:

- Mask Generation: We use our baseline instance segmentation network on unlabeled videos containing unseen classes for mask generation;
- Mask Selection: We apply our method to automatically select the correct masks on unlabeled videos;
- Model Refinement: We use our generated masks to fine-tune or retrain our baseline network to boost its performance on unseen classes.

In this section, we present our baseline instance segmentation network and our approach for automatically selecting high-quality masks by exploring the video information. As we will show in the following section, compared to exhaustive search, our approach is highly efficient and requires few easy-to-fix hyper-parameters.

3.1. Baseline Network for Mask Generation

To generate masks from unlabeled videos, we use a class-agnostic Mask R-CNN [19] with a ResNet-50-FPN [32] backbone as our baseline network. Following previous work [41], we refer to this class-agnostic Mask R-CNN as 'MP R-CNN' for Mask Proposal R-CNN, as it aims only at generating mask proposals regardless of object classes. Note that in practice, Mask R-CNN could be replaced by any other trainable instance segmentation methods. As we mentioned in Section 1, the instance segmentation network may assign low confidence scores for some correct detections of unseen classes. Therefore, during the mask generation stage, we set the confidence score threshold to 0 to get as many detections as possible.

3.2. Mask Selection

Given a video of T frames, we start from a set of mask candidates $\mathcal{M}_t = \{M_{t,1}..M_{t,N}\}$ for each frame I_t obtained using our baseline network, with N the number of masks candidates in I_t. To select the mask candidates that actually correspond to objects, we exploit the following cues and constraint:

- The **"Background cue"**: Segmenting typical backgrounds such as sky or grass gives us a cue about where the objects are, whether they move or not.
- The **"Flow cue"**: The optical flow between consecutive frames gives us a cue about the moving objects.

• The **"Consistency cue"**: The selected masks should be consistent not only between consecutive frames, but also over long sequences.
• The **"Non-overlapping constraint"**: An additional constraint that is usually overlooked is that the masks should not overlap: Ideally, one pixel in the image can belong to at most one mask.

As we will show in the following sections, each cue corresponds to a loss term in the final objective function. Each of them and the non-overlapping constraint contribute to removing the false positives, as will be demonstrated by our ablation study in Section 4.3.

To combine these cues to select the correct masks in a given video sequence, we rely on a Bayesian framework. This selection problem can be formalized as maximizing the probability of the detected masks given the frames of the video:

$$P\big(\mathbf{C}_1, .., \mathbf{C}_T | I_1, .., I_T\big) \ , \tag{1}$$

where $\mathbf{C}_t$ is a set of binary random variables, with $\mathbf{C}_{t,i} = 1$ corresponding to the event that mask $M_{t,i}$ is selected and 0 that it is not. We show in the supplementary material that maximizing this probability is equivalent to minimizing the following objective function:

$$\begin{aligned} \arg\min_{\{\Delta_1,..,\Delta_T\}} \quad & \textstyle\sum_t \big(\lambda_I \mathcal{L}_I\big(I_t, \Delta_t\big) + \\ & \lambda_F \mathcal{L}_F\big(F_t, I_t, I_{t+1}, \Delta_t, \Delta_{t+1}\big) + \\ & \lambda_p \mathcal{L}_p(\Delta_t, \Delta_{t+1})\big) \ , \end{aligned} \tag{2}$$

under the non-overlapping constraint that will be detailed below. λ_I, λ_F, and λ_p are constant weights. F_t represents the optical flow for the pair of frames (I_t, I_{t+1}). $\Delta_t = \{\delta_{t,1}, .., \delta_{t,N}\}$ denotes the realization of $\mathbf{C}_t$ where $\delta_{t,i}$ is the realization of the random binary variable $\mathbf{C}_{t,i}$. $\delta_{t,i} = 1$ when $M_{t,i}$ is selected, otherwise $\delta_{t,i} = 0$.

We call $\mathcal{L}_I$ the Background loss and $\mathcal{L}_F$ the Flow loss, as they exploit the Background cue and the Flow cue respectively. $\mathcal{L}_p$ enforces consistent selections between consecutive frames. We detail these three losses below.

3.2.1 Background Loss $\mathcal{L}_I$

We use $\mathcal{L}_I$ to exploit the Background cue that hints at where the objects are. As shown in Figure 3, to evaluate it, we compare a binary image generated for the selected masks and the foreground/background probability map predicted by a binary segmentation network f by calculating their cross entropy, as the image background should match the background of the selected masks. By doing this comparison over all the image locations, we can exploit information from the whole image to guide the mask selection—we will rely on the same strategy for the other terms. Formally, we take

$$\mathcal{L}_I\big(I_t, \Delta_t\big) = \mathrm{CE}\big(\mathrm{Bg}(I_t), 1 - \overline{\mathrm{Fg}}(\Delta_t)\big) \ , \tag{3}$$

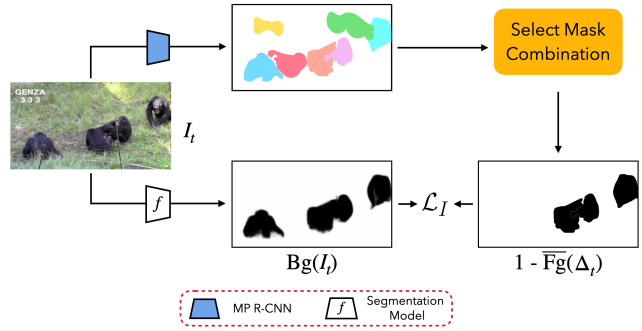

Figure 3. To evaluate the background loss $\mathcal{L}_I$ of Eq. (3), we compare the background predicted for the image and the background of the selected masks.

where CE denotes the cross-entropy, $\mathrm{Bg}(I_t)$ is a probability map for each pixel to belong to the background as predicted by the network f, and $\overline{\mathrm{Fg}}(\Delta_t)$ is the binary image of masks in $\mathcal{M}_t$ such that $\delta_{t,i} = 1$. For f, we use the network architecture proposed in [26] trained on the same training data as our baseline network for mask generation. The details about the segmentation network f can be found in the supplementary material.

3.2.2 Flow Loss $\mathcal{L}_F$

We use $\mathcal{L}_F$ to exploit the Flow cue: The optical flow is the result of the object motions and the camera motion, and thus also hints at where the objects are. Even if an object is static but the camera is in motion, when the distance between the camera and the background is large enough, relative motion will make the optical flow of the object regions stand out from the background optical flow.

Figure 4 shows how we evaluate this term. We compare the flow predicted by an optical flow estimator g and a "synthetic optical flow" generated using the masks selected in $\mathcal{M}_t$ and $\mathcal{M}_{t+1}$. Similar to the $\mathcal{L}_I$ term, this comparison allows us to exploit information from all the image locations. In practice, we use the method of [49] for g. To generate the synthetic optical flow, we use the colors of the pixels in the selected masks to compute their optical flow. We average the flow in F_t on the pixels that do not belong to any mask to assign these pixels the resulting value. The detailed procedure can be found in the supplementary material.

Using this procedure, the measured flow $F_t = g(I_t, I_{t+1})$ and the synthetic flow are similar when all moving objects are correctly selected in both frames, even when the camera is in motion. More formally, we take:

$$\mathcal{L}_F\big(F_t, I_t, I_{t+1}, \Delta_t, \Delta_{t+1}\big) = \big\|F_t - \overline{F}_t\big\|_1 \ , \tag{4}$$

where $\overline{F}_t = \overline{F}_t(I_t, I_{t+1}, \Delta_t, \Delta_{t+1})$ is the synthetic flow generated for the selected masks in $\mathcal{M}_t$ and $\mathcal{M}_{t+1}$. We use the L1-norm to compare the two flows to be robust to outlier

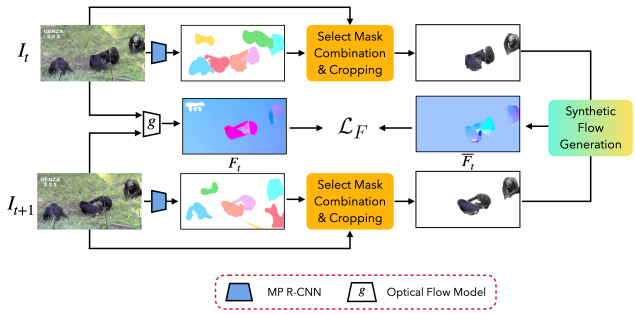

Figure 4. To evaluate the flow loss $\mathcal{L}_F$ of Eq. (4), we compare the optical flow estimated between two consecutive images and the optical flow computed for the masks selected in the two images.

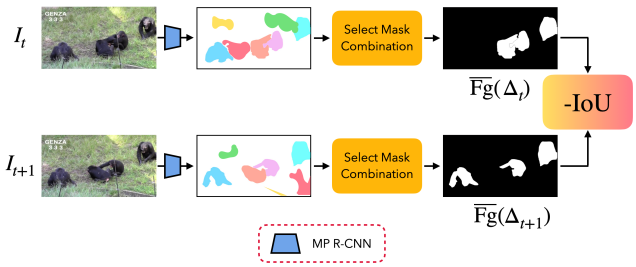

Figure 5. To evaluate the regularization loss $\mathcal{L}_p$ of Eq. (5), we compare the binary images of the selected masks in two consecutive images.

values that are very common in the predicted flow. Figure 4 shows examples for F_t and $\overline{F}_t$.

3.2.3 Regularization Loss $\mathcal{L}_p$ and Constraint

As discussed above, the optimization in Eq. (2) should be done under the constraint that no masks selected for the same frame overlap each other.

$\mathcal{L}_p(\Delta_t, \Delta_{t+1})$ is usually interpreted in tracking problems as a motion model. We use it to enforce a consistent selection of masks between consecutive frames. Figure 5 shows how we compute it: We use a very simple motion model and assume that the objects move slowly, in other words, the areas segmented as objects do not change abruptly between two consecutive frames. Formally, we take :

$$\mathcal{L}_p(\Delta_t, \Delta_{t+1}) = -\text{IoU}\big(\overline{\text{Fg}}(\Delta_t), \overline{\text{Fg}}(\Delta_{t+1})\big) \ , \qquad (5)$$

i.e. the negative Intersection-over-Union between the binary images of the masks selected for frames I_t and I_{t+1}. It is set to 0 when no masks are selected for none of the two images.

3.3. Two-Stage Optimization

In this section, we introduce an efficient way to minimize Eq. (2). Note that minimizing this function requires to optimize on all the frames simultaneously. A naive approach is to apply exhaustive search for the solution of the problem, where the number of evaluations of the objective function would be $O(2^{NT})$, with N the number of mask candidates per frame and T the number of frames (typical values $N = 15$ and $T = 180$ would require $\sim 10^{810}$ evaluations). This is clearly computationally prohibitive.

We provide here an efficient two-stage algorithm, depicted in Figure 6. In the first stage, based on the background loss $\mathcal{L}_I$ and the non-overlapping constraint, we select the top-K most promising combinations of masks for each frame independently. Note that the combinations violating the non-overlapping constraint are simply discarded. Then, in the second stage, we optimize the complete objective function over all frames simultaneously to find the best combinations for each frame. For both stages, we can use Dijkstra's algorithm [13] to significantly decrease the complexity of the computations. In the worst case, the number of evaluations of the objective function becomes $O(KTN^3 + K^2T^2)$. We use $K = 10$ in practice, which reduces the required evaluations from $\sim 10^{810}$ to $\sim 10^7$ for the numerical example above.

This optimization problem is related to many previous works on multiple object tracking [1, 36, 44, 58, 63, 64], which typically use graph-based methods to solve related problems efficiently. One of the main differences with these works is that, in our video-level optimization, each node corresponds to a combination of masks instead of a single bounding box or mask. Besides, we do not have access to the object classes, and our optimization is under the constraint that the masks do not overlap, while these works typically rely on bounding boxes that can overlap when the objects are close to each other.

1. Image-Level Optimization. At this stage, for each frame I_t, we look for the top-K combinations of masks in the power set $\mathcal{P}(\mathcal{M}_t)$ that minimize the Eq.(3) under the non-overlapping constraint. An exhaustive search would take 2^N evaluations of the objective function.

However, we note that this problem can be formulated as a K-shortest path search problem in a binary tree, where each pair of branches of a node corresponds to the selection or not of a mask, and each branch has an associated weight: This weight is set to infinity if the branch corresponds to the selection of a mask that overlaps with one of its ancestors, otherwise it depends on the value of $\mathcal{L}_I$ computed only on the mask. By iteratively applying Dijkstra's algorithm to find the top-K combinations of masks, we reduce the number of evaluations to $O(KN^3)$. Note that our proposed algorithm is agnostic to the order of the mask candidates in $\mathcal{M}_t$. More details can be found in the supplementary material.

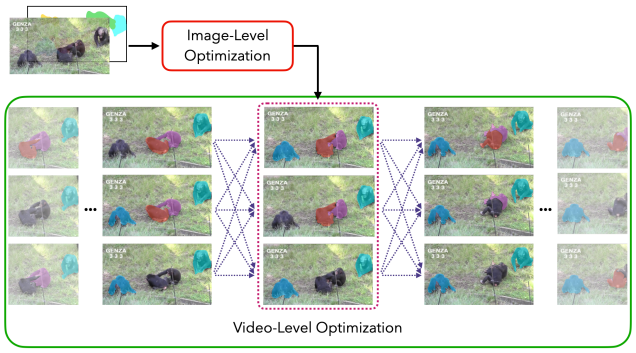

Figure 6. Given a video, we first run the image-level optimisation on each frame and get the top-K combinations of masks for each frame. The video-level optimisation selects the best combination for each frame efficiently by solving a shortest path problem. For this figure, K is set to 3. More details are given in Section 3.3.

2. Video-Level Optimization. As shown in Figure 6, we generate a graph with the remaining top-K combinations of masks for each frame: Each node corresponds to one combination and each edge is labeled with the loss given in Eq.(2) for the two combinations it links. Finding the best combination for each frame becomes the problem of finding the shortest path in this graph. Instead of doing 2^{KT} evaluations of the objective function to find the shortest path, we can use Dijkstra's algorithm [13] here as well to significantly accelerate the speed of our algorithm. The number of evaluations is reduced to $O(K^2T^2)$. We explain in more detail how we build the graph in the supplementary material.

4. Experiments

In this section, we first present our benchmark for evaluating our approach. Then we compare our method with several previous methods for the purpose of generating masks on unlabeled videos and analyze the results. We also conduct a thorough ablation study to show the influence of different components of our method. Due to space limits, implementation details were moved to the supplementary material.

4.1. Experimental Setup

Implementation Details. As described in Section 3.1, we use a class-agnostic Mask R-CNN [19] with a ResNet-50-FPN [32] backbone as our baseline for mask generation. Our baseline is pre-trained on the COCO dataset, which contains 80 classes and 115k training images. We follow the training strategy as described in [19]. We take $\lambda_I = \lambda_F = 1$ and $\lambda_p = 0.5$ for the weights in Eq. (2) in all our experiments.

Evaluation. To benchmark our method, we created a dataset we call "Unseen-VIS". The training part of "Unseen-VIS" consists of videos collected from the YouTube Video Instance Segmentation (YouTube VIS) [61] and is used for mask generation. The test part of "Unseen-VIS" contains static images extracted from YouTube VIS for evaluation.

The original YouTube VIS dataset contains 2,883 videos with 131k object instances spanning 40 classes, among which 24 coincide with COCO. We thus consider the remaining 16 classes [1] as *unseen* classes which results in 795 videos in total. We randomly selected 595 videos as the training set, which we refer as Unseen-VIS-train. The labeled static images in the remaining 200 videos are used for evaluation, which we refer as Unseen-VIS-test. All the videos of Unseen-VIS-train are used as unlabeled videos, and their ground truth masks are ignored.

For quantitative evaluation, we rely on the standard COCO metrics: AP, AP_{50}, AP_{75}, and AR_1, AR_3 and AR_5 as the maximum number of objects per image in our testing set is 4. We do not use AP_S, AP_M, and AP_L as the object scales in COCO differ largely from YouTube VIS.

4.2. Results on Unseen-VIS-test

4.2.1 Video-Annotation-Free Mask Generation

We first compare our method to other approaches that can also generate masks given video sequences without using any video annotations. Each method is first applied on the Unseen-VIS-train dataset for mask generation, then we compare the performance of MP R-CNN on the Unseen-VIS-test set after fine-tuning on these masks. As only one over five frames is annotated in Unseen-VIS-train (19352 annotated frames in total), we thus use only the masks of these frames for training for fair comparison among different methods.

Saliency/Flow-based methods. FST [42] and NLC [14] can generate masks from videos by estimating the saliency and motion of the objects in videos. IOA [10] trains a deep neural network on the output of an unsupervised soft foreground segmentation algorithm [48] to segment objects in videos. These methods can identify moving regions in the videos but can not separate adjacent objects in the images, thus a proposal generated by these methods may actually correspond to several objects.

Tracking-based methods. TWB [2] and UnOVOST [35] rely on a frame-by-frame tracking pipeline applied to mask proposals. These methods are the closest methods to ours as we all rely on an instance segmentation model for proposal generation. However, as we mentioned before, for our final goal (training a better object detector), we do not need to keep track of the detected objects.

Similarity Propagation. Given masks for a frame,

[1] panda, lizard, seal, shark, mouse, frog, tiger, leopard, fox, deer, ape, snake, monkey, rabbit, fish, turtle.

Method used for mask generation	Unseen-VIS-test					
	AP	AP_{50}	AP_{75}	AR_1	AR_3	AR_5
(bef. fine-tuning)	35.8	61.2	38.1	33.3	47.3	50.3
NLC [14]	1.2	3.8	1.0	2.4	5.3	6.9
IOA [10]	2.4	8.5	0.9	6.9	8.7	9.5
FST [42]	17.0	41.8	11.3	22.0	30.6	33.1
UnOVOST [35]	31.1	55.6	32.2	29.9	44.5	48.2
TWB [2]	31.2	53.4	32.8	31.5	46.7	50.0
DD [46]	36.6	63.8	38.5	32.5	46.2	49.3
ZS-UVC [30]	21.2	42.6	19.9	26.3	40.0	43.2
Ours	**39.0**	**67.9**	**41.3**	**35.2**	**48.9**	**51.4**

Table 1. **Mask Generation without Video Annotation.** Performance of MP R-CNN on Unseen-VIS-test after fine-tuning on masks generated by various methods applied to Unseen-VIS-train without using any video annotation.

UVC [30] warps these masks to consecutive frames based on the estimated correspondences between consecutive frames. For this experiment, we use it in a zero-shot setting ("ZS-UVC"), where the masks of the first frame are instead generated by thresholding the confidence score on the first frame prediction of MP R-CNN (we use a threshold of 0.1 in practice).

Self-training methods. We also compare our method with the self-learning Data Distillation (DD) method [46]. We follow their proposed test time augmentation to generate masks on each frame of the Unseen-VIS-train videos independently, as this method performs on single images.

We report the results in Table 1. Compared with the Saliency/Flow-based methods and Tracking-based methods, the masks generated by our method are of high quality and can largely improve the performance of the baseline network in all metrics. Our method also outperforms the state-of-the-art self-training method (DD) [46] for two-stage object detection by a large margin.

4.2.2 Video-Annotation-Dependent Mask Generation

In addition to the aforementioned methods, we further compare with two methods that adopt different settings and explore the upper bound of our method.

RVOS. RVOS [51] is an end-to-end video object segmentation framework that directly runs on videos, which requires labeled videos for training. We adopt the Zero-Shot setting for RVOS, where it is trained with ResNet50 [20] as backbone on 1089 videos (25869 annotated frames in total) of seen classes of YouTube VIS and directly applied to the Unseen-VIS-train videos for mask generation.

OS-UVC. Here, we consider the One-Shot setting for UVC, where the ground truth masks of the first frame are given for all Unseen-VIS-train videos.

Selected using GT. We use the ground truth mask labels of Unseen-VIS-train to select the masks predicted by MP R-CNN. The similarity among masks is evaluated based on their Intersection-over-Union, and the Hungarian algorithm is used to select the masks that best match with the ground truth. This can be regarded as an upper bound that we can achieve given the predictions of MP R-CNN.

Method used for mask generation	Video Annotations		Unseen-VIS-test					
	Seen	Unseen	AP	AP_{50}	AP_{75}	AR_1	AR_3	AR_5
(bef. fine-tuning)			35.8	61.2	38.1	33.3	47.3	50.3
RVOS [51]	✓		38.5	68.9	38.0	35.4	49.5	52.8
Ours	-	-	39.0	67.9	41.3	35.2	48.9	51.4
OS-UVC [30]		✓	41.5	73.7	42.9	39.1	52.7	54.9
Selected using GT		✓	42.7	75.1	45.3	37.3	53.4	53.6
Trained with GT		✓	50.8	80.9	54.6	43.6	58.6	60.6

Table 2. **Mask Generation with Video Annotations.** Performance of MP R-CNN on Unseen-VIS-test after fine-tuning on the masks generated by various methods that require manual video annotations except ours. RVOS uses labeled videos of Seen classes for training. OS-UVC uses the ground truth masks for the first frame for mask generation. "Selected using GT" represents the masks generated by MP R-CNN selected using ground truth masks and can be regarded as an upper-bound.

Trained with GT. We report the performance of MP R-CNN fine-tuned with ground truth mask labels of Unseen-VIS-train. This can also be regarded as an upper bound, where all the classes have already been Seen.

We report the results in Table 2. Compared to RVOS [51] trained with labeled videos, our method can still achieve comparable results on recall while surpassing their results by a large margin on AP_{75}, which means that the masks selected by our method are of better quality. Importantly for practical applications, our method was able to deal with the large domain gap between the images in COCO on which we pre-train MP R-CNN and the frames in YouTube VIS on which we apply and evaluate our method. RVOS was trained and applied on videos from YouTube VIS, and therefore was not confronted to a domain gap.

While OS-UVC [30] achieves higher results than our approach, it relies mainly on a high-quality first frame mask: We observe a large performance drop when we replace the ground truth masks ("OS-UVC" in Table 2) by the predicted masks ("ZS-UVC" in Table 1). Besides, it can only track objects visible in the first frame as it does not handle the emergence of new objects.

After simply retraining from scratch both on the 80 seen classes of COCO and the masks generated by our approach on Unseen-VIS-train, MP R-CNN achieves 35.2 mask AP on COCO minival and 38.9 mask AP on Unseen-VIS-test. Compared with the MP R-CNN pre-trained only with COCO dataset, which achieves 35.3 mask AP on COCO minival, we achieve better performance on Unseen-VIS-test while maintaining the performance on COCO. More details can be found in the supplementary material.

Application to Zero-Shot Video Object Segmentation. As one of the state-of-the-art zero-shot video object seg-

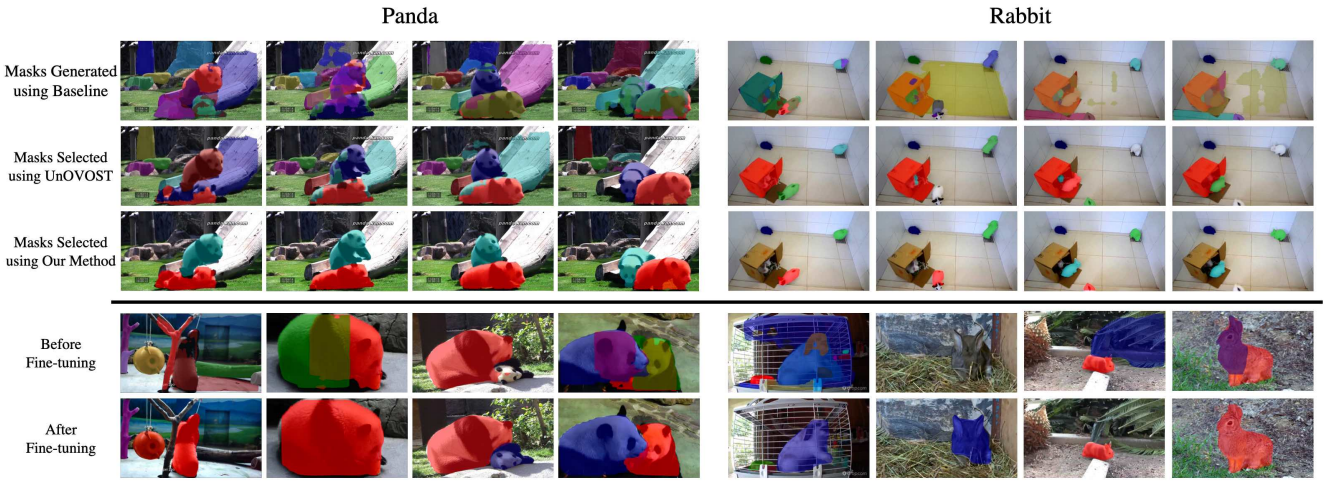

Figure 7. **Qualitative results of selected masks on Unseen-VIS-train and detections of new classes on Unseen-VIS-test after fine-tuning on these selected masks. Top:** First row: Masks detected by our baseline network MP R-CNN on two sequences from Unseen-VIS-train; Second row: Masks selected by UnOVOST [35]; Third row: Masks selected by our approach. Note that we keep the masks for the pandas and rabbits, and reject the masks that do not correspond to real objects. **Bottom:** Masks detected in still images from Unseen-VIS-test. Fourth row: Masks detected by MP R-CNN before we fine-tuned it on the masks selected by our approach on Unseen-VIS-train; Fifth row: Masks detected by MP R-CNN *after* fine-tuning. The masks generated by our method results in a significantly better model for the new classes: We can now correctly segment pandas and rabbits in new videos, even if no manual segmentations for pandas and rabbits were provided. *For more examples, please refer to the video in the supplementary material.*

Method	*Use Unlabeled Data*	$\mathcal{J}$ & $\mathcal{F}$	$\mathcal{J}$			$\mathcal{F}$		
		Mean	**Mean**	**Recall**	**Decay**	**Mean**	**Recall**	**Decay**
UnOVOST [35]		56.2	54.4	63.7	-0.01	57.9	65.0	0.00
UnOVOST+	✓	**59.9**	**59.1**	**70.0**	**-0.06**	**60.8**	**70.8**	**-0.03**

Table 3. **Zero-Shot Video Object Segmentation evaluation on the DAVIS dataset [43].** UnOVOST [35] relies on an instance segmentation network for mask generation. UnOVOST+ row: After fine-tuning its mask generation network on the DAVIS training dataset using the masks generated by our approach, it achieves higher results on all metrics.

mentation methods [2], UnOVOST [35] segments the objects in the videos by linking the masks predicted by an instance segmentation network on each frame. As show in Table 3, by fine-tuning the original mask generation model on the masks generated on the DAVIS training dataset [43] using our method, we achieve much better results. This demonstrates that downstream tasks can benefit from the performance boost brought by our method.

4.3. Ablation Study

Table 4 shows the positive impact made by each loss term and constraint in Eq. (2). The masks obtained by applying only the background loss $\mathcal{L}_I$ can already improve the performance of baseline on unseen classes. Similarly, adding the constraint that the masks should not overlap, the flow loss $\mathcal{L}_F$, or the regularization loss $\mathcal{L}_p$ has a positive impact. In particular, this shows that both the flow loss $\mathcal{L}_F$ and the regularization loss $\mathcal{L}_p$ help reranking the combinations of masks given by the background loss.

[2] https://davischallenge.org/challenge2019/leaderboards.html

$\mathcal{L}_I$	Overl.Constr.	$\mathcal{L}_F$	$\mathcal{L}_p$	AP	AP_{50}	AP_{75}	AR_1	AR_3	AR_5
				35.8	61.2	38.1	33.3	47.3	50.3
✓				36.6	65.2	37.8	33.3	47.7	50.8
✓	✓			38.1	64.6	40.2	34.2	48.5	51.2
✓	✓	✓		38.7	67.0	40.5	34.7	48.7	51.3
✓	✓	✓	✓	39.0	67.9	41.3	35.2	48.9	51.4

Table 4. **Ablation study** on the different components of our method.

5. Conclusion

In this paper, we attacked the problem of localizing and segmenting objects from unseen classes without any manual mask labels. We showed that, based on an instance segmentation model pretrained on some seen classes, our method provides high-quality masks for unseen classes after analysing unlabeled videos, without requiring difficult-to-tune hyper-parameters. Moreover, we provided an efficient implementation by breaking down the computationally prohibitive optimization into a two-stage optimization.

It should nevertheless be noted that, in the unsupervised case, the concept of objects is quite ill-defined. The boundary between "things" and "stuff" [5, 16] is sometimes fuzzy: For example, should we consider a stone on the background as an object? What if a person pushes this stone? The granularity of the problem is also not clear: Should we consider a person as one object, or each of their clothes as individual objects? This ambiguity does not arise in the supervised case, as the annotators decide what is an object, but makes the evaluation of unsupervised object detection difficult. It may be important to rethink the definition of object, either

by its shape or its function, to make the evaluation of unsupervised object segmentation more meaningful.

Acknowledgement We thank Ze Chen, Zerui Chen, Changqian Yu, Enze Xie, Tianze Xiao, Yinda Xu, Xi Shen, Mathis Petrovich, and Philippe Chiberre for valuable feedback. This project has received funding from the CHIST-ERA IPALM project.

References

[1] Jérôme Berclaz, Francois Fleuret, Engin Turetken, and Pascal Fua. Multiple Object Tracking Using K-Shortest Paths Optimization. *IEEE TPAMI*, 33(9), 2011.

[2] Philipp Bergmann, Tim Meinhardt, and Laura Leal-Taixé. Tracking Without Bells and Whistles. In *CVPR*, 2019.

[3] Christopher P. Burgess, Loic Matthey, Nicholas Watters, Rishabh Kabra, Irina Higgins, Matt Botvinick, and Alexander Lerchner. Monet: Unsupervised Scene Decomposition and Representation. In *arXiv*, 2019.

[4] Sergi Caelles, Kevis-Kokitsi Maninis, Jordi Pont-Tuset, Laura Leal-Taixé, Daniel Cremers, and Luc Van Gool. One-Shot Video Object Segmentation. In *CVPR*, 2017.

[5] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. COCO-Stuff: Thing and Stuff Classes in Context. In *CVPR*, 2018.

[6] Liang-Chieh Chen, Raphael Gontijo Lopes, Bowen Cheng, Maxwell D. Collins, Ekin D. Cubuk, Barret Zoph, Hartwig Adam, and Jonathon Shlens. Naive-Student: Leveraging Semi-Supervised Learning in Video Sequences for Urban Scene Segmentation. In *ECCV*, 2020.

[7] Yi-Wen Chen, Yi-Hsuan Tsai, Yen-Yu Lin, and Ming-Hsuan Yang. VOSTR: Video Object Segmentation via Transferable Representations. *IJCV*, 128(4), 2020.

[8] Jingchun Cheng, Yi-Hsuan Tsai, Wei-Chih Hung, Shengjin Wang, and Ming-Hsuan Yang. Fast and Accurate Online Video Object Segmentation via Tracking Parts. In *CVPR*, 2018.

[9] Jingchun Cheng, Yi-Hsuan Tsai, Shengjin Wang, and Ming-Hsuan Yang. SegFlow: Joint Learning for Video Object Segmentation and Optical Flow. In *ICCV*, 2017.

[10] Ioana Croitoru, Simion-Vlad Bogolin, and Marius Leordeanu. Unsupervised Learning from Video to Detect Foreground Objects in Single Images. In *ICCV*, 2017.

[11] Achal Dave, Pavel Tokmakov, and Deva Ramanan. Towards Segmenting Anything That Moves. In *ICCV*, 2019.

[12] Akshay Dhamija, Manuel Gunther, Jonathan Ventura, and Terrance Boult. The Overlooked Elephant of Object Detection: Open Set. In *WACV*, 2020.

[13] Edsger W. Dijkstra. A Note on Two Problems in Connexion with Graphs. *Numerische mathematik*, 1(1), 1959.

[14] Alon Faktor and Michal Irani. Video Segmentation by Non-Local Consensus Voting. In *BMVC*, 2014.

[15] Zhibo Fan, Jin-Gang Yu, Zhihao Liang, Jiarong Ou, Changxin Gao, G. Xia, and Y. Li. FGN: Fully Guided Network for Few-Shot Instance Segmentation. In *CVPR*, 2020.

[16] D. A. Forsyth, J. Malik, M. M. Fleck, H. Greenspan, T. K. Leung, S. Belongie, C. Carson, and C. Bregler. Finding Pictures of Objects in Large Collections of Images. In *Object Representation in Computer Vision*, 1996.

[17] Georgia Gkioxari, Jitendra Malik, and Justin Johnson. Mesh R-CNN. In *ICCV*, 2019.

[18] Klaus Greff, Raphaël Lopez Kaufman, Rishabh Kabra, Nick Watters, Christopher Burgess, Daniel Zoran, Loic Matthey, Matthew Botvinick, and Alexander Lerchner. Multi-Object Representation Learning with Iterative Variational Inference. In *ICML*, 2019.

[19] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *ICCV*, 2017.

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *CVPR*, 2016.

[21] Ronghang Hu, Piotr Dollár, Kaiming He, Trevor Darrell, and Ross Girshick. Learning to Segment Every Thing. In *CVPR*, 2018.

[22] Suyog Dutt Jain, Bo Xiong, and Kristen Grauman. Fusionseg: Learning to Combine Motion and Appearance for Fully Automatic Segmentation of Generic Objects in Videos. In *CVPR*, 2017.

[23] Tomas Jakab, Ankush Gupta, Hakan Bilen, and Andrea Vedaldi. Self-Supervised Learning of Interpretable Keypoints from Unlabelled Videos. In *CVPR*, 2020.

[24] Jisoo Jeong, Seungeui Lee, Jeesoo Kim, and Nojun Kwak. Consistency-Based Semi-Supervised Learning for Object Detection. In *NeurIPS*, 2019.

[25] SouYoung Jin, Aruni Roychowdhury, Huaizu Jiang, Ashish Singh, Aditya Prasad, Deep Chakraborty, and Erik Learned-Miller. Unsupervised Hard Example Mining from Videos for Improved Object Detection. In *ECCV*, 2018.

[26] Alexander Kirillov, Kaiming He, Ross Girshick, Carsten Rother, and Piotr Dollár. Panoptic Segmentation. In *CVPR*, 2019.

[27] Yeong Jun Koh and Chang-Su Kim. Primary Object Segmentation in Videos Based on Region Augmentation and Reduction. In *CVPR*, 2017.

[28] Weicheng Kuo, A. Angelova, Jitendra Malik, and Tsung-Yi Lin. ShapeMask: Learning to Segment Novel Objects by Refining Shape Priors. In *ICCV*, 2019.

[29] Fuxin Li, Taeyoung Kim, Ahmad Humayun, David Tsai, and James M. Rehg. Video Segmentation by Tracking Many Figure-Ground Segments. In *ICCV*, 2013.

[30] Xueting Li, Sifei Liu, Shalini De mello, Xiaolong Wang, Jan Kautz, and Ming-Hsuan Yang. Joint-Task Self-Supervised Learning for Temporal Correspondence. In *NeurIPS*, 2019.

[31] Yandong Li, Di Huang, Danfeng Qin, Liqiang Wang, and Boqing Gong. Improving Object Detection with Selective Self-Supervised Self-Training. In *ECCV*, 2020.

[32] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature Pyramid Networks for Object Detection. In *CVPR*, 2017.

[33] Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-Centric Learning with Slot Attention. In *NeurIPS*, 2020.

[34] Xiankai Lu, Wenguan Wang, Jianbing Shen, Yu-Wing Tai, David J. Crandall, and Steven C.H. Hoi. Learning Video Object Segmentation from Unlabeled Videos. In *CVPR*, 2020.

[35] Jonathon Luiten, Idil Esen Zulfikar, and Bastian Leibe. UnOVOST: Unsupervised Offline Video Object Segmentation and Tracking. In *WACV*, 2020.

[36] Tianyang Ma and Longin Jan Latecki. Maximum Weight Cliques with Mutex Constraints for Video Object Segmentation. In *CVPR*, 2012.

[37] Anton Milan, Laura Leal-Taixé, Konrad Schindler, and Ian Reid. Joint Tracking and Segmentation of Multiple Targets. In *CVPR*, 2015.

[38] Natalia Neverova, David Novotny, and Andrea Vedaldi. Correlated Uncertainty for Learning Dense Correspondences from Noisy Labels. In *NeurIPS*, 2019.

[39] Seoung Wug Oh, Joon-Young Lee, Ning Xu, and Seon Joo Kim. Video Object Segmentation Using Space-Time Memory Networks. In *ICCV*, 2019.

[40] Aljoša Ošep, Paul Voigtlaender, Jonathon Luiten, Stefan Breuers, and Bastian Leibe. Large-Scale Object Mining for Object Discovery from Unlabeled Video. In *ICRA*, 2019.

[41] Aljoša Ošep, Paul Voigtlaender, Mark Weber, Jonathon Luiten, and Bastian Leibe. 4D Generic Video Object Proposals. In *ICRA*, 2020.

[42] Anestis Papazoglou and Vittorio Ferrari. Fast Object Segmentation in Unconstrained Video. In *ICCV*, 2013.

[43] Federico Perazzi, Jordi Pont-Tuset, Brian Mcwilliams, Luc Van Gool, Markus Gross, and Alexander Sorkine-Hornung. A Benchmark Dataset and Evaluation Methodology for Video Object Segmentation. In *CVPR*, 2016.

[44] Federico Perazzi, Oliver Wang, Markus Gross, and Alexander Sorkine-Hornung. Fully Connected Object Proposals for Video Segmentation. In *ICCV*, 2015.

[45] Trung Pham, Vijay BG Kumar, Thanh-Toan Do, Gustavo Carneiro, and Ian Reid. Bayesian Semantic Instance Segmentation in Open Set World. In *ECCV*, 2018.

[46] Ilija Radosavovic, P. Dollár, Ross B. Girshick, Georgia Gkioxari, and Kaiming He. Data Distillation: Towards Omni-Supervised Learning. In *CVPR*, 2018.

[47] Martin Runz, Kejie Li, Meng Tang, Lingni Ma, Chen Kong, Tanner Schmidt, Ian Reid, Lourdes Agapito, Julian Straub, Steven Lovegrove, and Richard Newcombe. FroDO: From Detections to 3D Objects. In *CVPR*, 2020.

[48] Otilia Stretcu and Marius Leordeanu. Multiple Frames Matching for Object Discovery in Video. In *BMVC*, 2015.

[49] Zachary Teed and Jia Deng. RAFT: Recurrent All-Pairs Field Transforms for Optical Flow. In *ECCV*, 2020.

[50] Zhi Tian, Chunhua Shen, and Hao Chen. Conditional Convolutions for Instance Segmentation. In *ECCV*, 2020.

[51] Carles Ventura, Miriam Bellver, Andreu Girbau, Amaia Salvador, Ferran Marques, and Xavier Giro-I-Nieto. RVOS: End-To-End Recurrent Network for Video Object Segmentation. In *CVPR*, 2019.

[52] Paul Voigtlaender, Yuning Chai, Florian Schroff, Hartwig Adam, Bastian Leibe, and Liang-Chieh Chen. FEELVOS: Fast End-To-End Embedding Learning for Video Object Segmentation. In *CVPR*, 2019.

[53] Kentaro Wada, Kei Okada, and Masayuki Inaba. Joint Learning of Instance and Semantic Segmentation for Robotic Pick-and-Place with Heavy Occlusions in Clutter. In *ICRA*, 2019.

[54] Xiaolong Wang, Allan Jabri, and Alexei A. Efros. Learning Correspondence from the Cycle-Consistency of Time. In *CVPR*, 2019.

[55] Xinlong Wang, Tao Kong, Chunhua Shen, Yuning Jiang, and Lei Li. SOLO: Segmenting Objects by Locations. In *ECCV*, 2020.

[56] Nicholas Watters, Daniel Zoran, Theophane Weber, Peter Battaglia, Razvan Pascanu, and Andrea Tacchetti. Visual Interaction Networks: Learning a Physics Simulator from Video. In *NeurIPS*, 2017.

[57] Kelvin Wong, Shenlong Wang, Mengye Ren, Ming Liang, and Raquel Urtasun. Identifying Unknown Instances for Autonomous Driving. In *Conference on Robot Learning*, 2020.

[58] Zheng Wu, Thomas H. Kunz, and Margrit Betke. Efficient Track Linking Methods for Track Graphs Using Network-Flow and Set-Cover Techniques. In *CVPR*, 2011.

[59] Christopher Xie, Yu Xiang, Arsalan Mousavian, and Dieter Fox. The Best of Both Modes: Separately Leveraging RGB and Depth for Unseen Object Instance Segmentation. In *Conference on robot learning*, 2020.

[60] Xiaopeng Yan, Ziliang Chen, Anni Xu, Xiaoxi Wang, Xiaodan Liang, and L. Lin. Meta R-CNN: Towards General Solver for Instance-Level Low-Shot Learning. In *ICCV*, 2019.

[61] Linjie Yang, Yuchen Fan, and Ning Xu. Video Instance Segmentation. In *ICCV*, 2019.

[62] Lin Yen-Chen, Andy Zeng, Shuran Song, Phillip Isola, and Tsung-Yi Lin. Learning to See Before Learning to Act: Visual Pre-Training for Manipulation. In *ICRA*, 2020.

[63] Amir Roshan Zamir, Afshin Dehghan, and Mubarak Shah. GMCP-Tracker: Global Multi-Object Tracking Using Generalized Minimum Clique Graphs. In *ECCV*, 2012.

[64] Dong Zhang, Omar Javed, and Mubarak Shah. Video Object Segmentation through Spatially Accurate and Temporally Dense Extraction of Primary Object Regions. In *CVPR*, 2013.

[65] Yanzhao Zhou, Xin Wang, Jianbin Jiao, and Trevor Darrell. Learning Saliency Propagation for Semi-Supervised Instance Segmentation. In *CVPR*, 2020.