Challenge 1: Predictive Mask Annotation

Shabnam Ezatzadeh[1]

ABSTRACT

Labelling all the frames of a video with a great accuracy takes a lot of time and cost. In order to make annotation process faster, we need to use semiautomated or automated labelling methods. Therefore, we need to implement a method to annotate an object in a few frames and the tool keep detecting that object in next frames. Ultimately, we want the tool to annotate all video frames itself after annotating a few frames.

KEYWORDS

Annotations; Segmenting; Semi-supervised; Self-Supervised

1. Introduction

Semi-supervised video segmentation refers to the partitioning of objects in a given video sequence with available annotations in the first frame. A pixel-accurate, spatiotemporal bipartition of the video is an essential building block for a wide spectrum of applications, such as action recognition, object tracking, semantic labeling, to name a few. Semi-supervised techniques also provide proper initializations for further video editing and analysis tasks (e.g., interactive video cutout, dataset annotation) since they allow a tradeoff between accuracy and human interaction (Wang et al., 2018).

Also, weakly-supervised image recognition approaches have been extensively studied as they do not require expensive human effort. Among them, the most attractive one is learning to segment images from only image-level annotations. For such approaches, the arguably most critical challenge remaining unsolved is how to accurately and densely localize object regions to obtain high-quality object cues for initiating and improving the segmentation model training (Wei et al., 2018).

Recently, self-supervised learning has shown to be a promising replacement for manually labeled data. It aims to learn representations from the structure of unlabeled data, instead of relying on a supervised loss, which involves manual labels. The principle has been successfully applied in depth estimation for stereo pairs or image sequences. Additionally, semantic segmentation is known to be tightly coupled with depth (Hoyer et al., 2021).

[1] shabnamezatzadeh@gmail.com

The rest of this work is organized as follows: In Section 2, previous works are reviewed. In Section 3 recent works and publications which address this problem are presented. In Section 4 open-source repositories are sorted. Finally, In Section 5 the implementations of two public repositories are introduced.

2. Previous Works

(Fathi et al., 2011) addressed the problem of segmenting an object of interest out of a video. They proposed an incremental self-training approach by iteratively labeling the least uncertain frame and updating similarity metrics. Moreover, the usage of harmonic functions naturally supports interactive segmentation.

(Misra et al., 2015) introduced a semi-supervised learning technique for training object detectors from videos. Their technique addresses the detection of multiple objects without assuming exhaustive labeling of object instances in any input frame. They presented a scalable framework that discovers objects in video using SSL (See Fig.1).

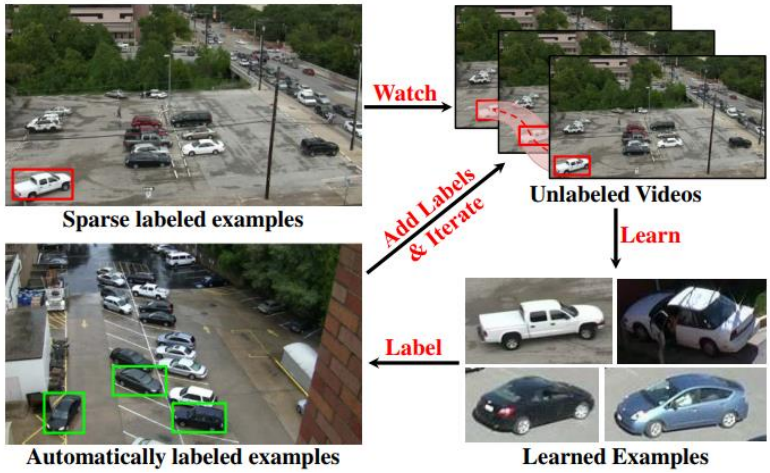

Fig. 1. Proposed method works with long video to automatically learn bounding box level annotations for multiple object instances. It does not assume exhaustive labeling of every object instance in the input videos, and from a handful of labeled instances can automatically label hundreds of thousands of instances.

(Caelles et al., 2017) presented One-Shot Video Object Segmentation (OSVOS), based on a fully convolutional neural network architecture that is able to successively transfer generic semantic information, learned on ImageNet, to the task of foreground segmentation, and finally to learning the appearance of a single annotated object of the test sequence (hence one-shot). Fig. 2 shows an example result of OSVOS, where the input is the segmentation of the first frame (in red), and the output is the mask of the object in the 90 frames of the sequence (in green).

Fig. 2. Example results of the proposed technique

(Wei et al., 2018) utilized in a novel way to effectively overcome this critical limitation of weakly supervised segmentation approaches. They designed a generic classification network equipped with convolutional blocks of different dilated rates. It can produce dense and reliable object localization maps and effectively benefit both weakly- and semi-supervised semantic segmentation.

(Wang et al., 2018) introduced a semi-supervised video segmentation approach based on an efficient video representation, called as "super-trajectory". A super-trajectory corresponds to a group of compact point trajectories that exhibit consistent motion patterns, similar appearances, and close spatiotemporal relationships. (See Fig.3)

Fig. 3. Video segmentation method takes the first frame annotation as initialization (left). Leveraging on super-trajectories, the segmentation process achieves superior results even for challenging scenarios including heavy occlusions, complex appearance variations, and large shape deformations (middle, right).

3. Recent Works

(Wang et al., 2019) presented Label Diffusion Lidar Segmentation (LDLS), a novel approach for 3D point cloud segmentation which leverages 2D segmentation of an RGB image from an aligned camera to avoid the need for training on annotated 3D data.

(Lu et al., 2020) proposed a new method for video object segmentation (VOS) that addresses object pattern learning from unlabeled videos, unlike most existing methods which rely heavily on extensive annotated data. They introduced a unified unsupervised/weakly supervised learning framework, called MuG, that comprehensively captures intrinsic properties of VOS at multiple granularities. (See Fig.4)

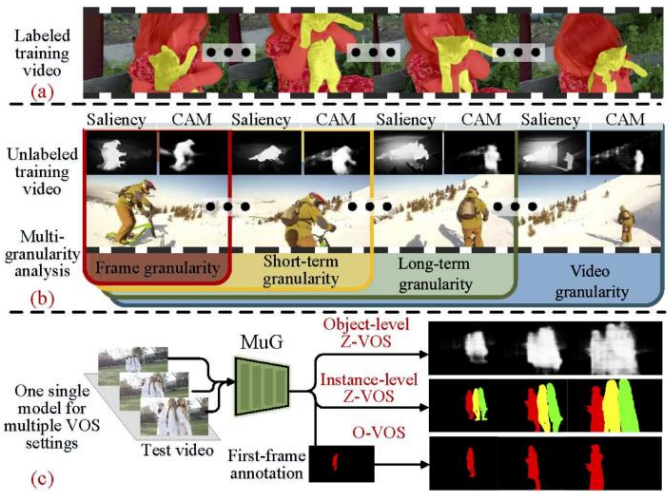

Fig. 4. (a) Current leading VOS methods are learned in a supervised manner, requiring large-scale elaborately labeled data. (b) Proposed model, MuG, provides an unsupervised/weakly-supervised framework that learns video object patterns from unlabeled videos. (c) Once trained, MuG can be applied to diverse VOS settings, with strong modeling ability and high generability.

(Zhou et al., 2020) proposed a novel Mask-guided Mean Teacher framework with Perturbation-sensitive Sample Mining (MMT-PSM), which consists of a teacher and a student network during training. Two networks are encouraged to be consistent both in feature and semantic level under small perturbations. The teacher's self-ensemble predictions from K-time augmented samples are used to construct the reliable pseudo labels for optimizing the student.

(Valvano et al., 2020) learned to segment using scribble annotations in an adversarial game. With unpaired segmentation masks, they trained a multiscale GAN to generate realistic segmentation masks at multiple resolutions, while they used scribbles to learn their correct position in the image. Central to the model's success is a novel attention gating mechanism, which they conditioned with adversarial signals to act as a shape prior, resulting in better object localization at multiple scales. (See Fig.5)

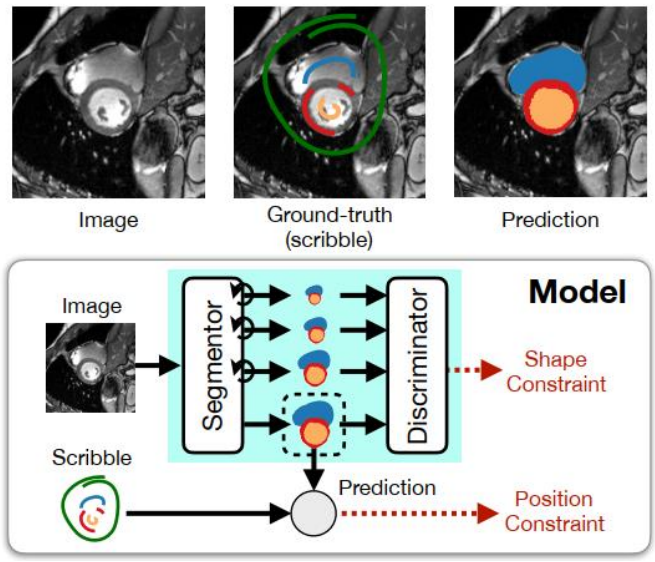

Fig. 5. In an adversarial game, the model learns to generate segmentation masks that look realistic at multiple scales and overlap with the available scribbled annotations. Loopy arrows in the figure, on the segmentor, represent the proposed attention gates, which under adversarial conditioning suppress irrelevant information in the extracted features maps.

(Hoyer et al., 2021) presented a framework for semi-supervised semantic segmentation, which is enhanced by self-supervised depth estimation(SDE) from unlabeled image sequences. They introduced three effective strategies capable of leveraging the knowledge learned from SDE. First, they show that the SDE feature representation can be transferred to semantic segmentation, by means of SDE pretraining and joint learning of segmentation and depth. Second, they demonstrate that the proposed DepthMix strategy outperforms related mixing strategies by avoiding inconsistent geometry of the generated images. Third, they present an automatic data selection for annotation algorithm based on SDE, which does not require human-in-the-loop annotations.

(Miao et.al., 2021) proposed a self-supervised spatio-temporal matching method coined Motion-Aware Mask Propagation (MAMP) for semi-supervised video object segmentation. During training, MAMP leverages the frame reconstruction task to train the model without the need for annotations. During inference, MAMP extracts high-resolution features from each frame to build a memory bank from the features as well as the predicted masks of selected past frames. Fig.6 shows an overview of our MAMP method for video object segmentation.

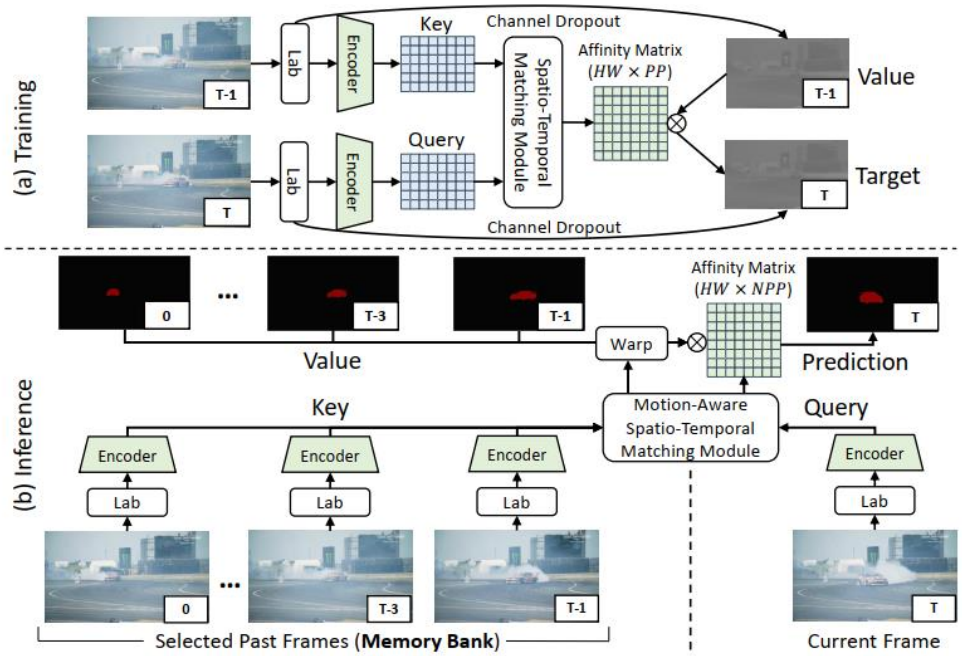

Fig. 6. Framework of the proposed MAMP. (a) A random pair of neighboring frames in the same video is sampled to train the model. The frames are converted to Lab color space and channel dropout is used only on the ab channels to generate the reconstruction target for self-supervision. During training, a vanilla spatio-temporal matching module is used because the selected frames are adjacent. Only the encoder weights are learned during training. (b) The trained encoder is used to encode the selected past frames into Key and store them in the memory bank along with their object masks as Value for mask propagation. The query frame (T) is encoded into Query to retrieve the spatio-temporal correspondences from Key and Value. The affinity matrix is computed in a local manner, where HW is the area of Query, N is the number of feature maps in Key, and P P is the area of region of interest (ROI) in one feature map of Key.

4. Open-source repositories

In this section, some public repositories are mentioned. They are sorted with their ranks on GitHub and bolded items are implemented in section 5.

Paper	Title	Method	Link
Waleed, 2017 (GitHub)	**Mask R-CNN for object detection and instance segmentation on Keras and TensorFlow**	Mask-RCNN	TensorFlow
Caelles et al., 2017 (CVPR)	One-Shot Video Object Segmentation	fully-convolutional neural network	PyTorch TensorFlow
Luiten et al., 2018	PReMVOS: Proposal-generation, Refinement and Merging for Video Object Segmentation	Mask-RCNN	TensorFlow
Hoyer et al., 2021 (CVPR)	**Three Ways to Improve Semantic Segmentation with Self-Supervised Depth Estimation**	self-supervised depth estimation (SDE)	PyTorch
Mondal et al., 2019	Revisiting CycleGAN for semi-supervised segmentation	CycleGAN	PyTorch
Zhou et al., 2020 (CVPR)	Learning Saliency Propagation for Semi-Supervised Instance Segmentation	Mask-RCNN	PyTorch
Zhou et al., 2020	Deep Semi-supervised Knowledge Distillation for Overlapping Cervical Cell Instance Segmentation	Mask-guided Mean Teacher framework with Perturbation-sensitive Sample Mining (MMT-PSM)	PyTorch
Valvano et al., 2020	Learning to Segment from Scribbles using Multi-scale Adversarial Attention Gates	multiscale GAN	TensorFlow
Wang et al., 2019	LDLS: 3-D Object Segmentation Through Label Diffusion From 2-D Images	Mask-RCNN	TensorFlow
Miao et.al., 2021	Self-Supervised Video Object Segmentation by Motion-Aware Mask Propagation	self-supervised spatio-temporal matching	PyTorch

5. Implementation

In this section, the implementations of two repositories are mentioned:

5.1) Mask R-CNN for object detection and instance segmentation on Keras and TensorFlow[2]

I chose this item because of its rank on GitHub (20.4 k star). This is an implementation of Mask R-CNN on Python 3, Keras, and TensorFlow. The model generates bounding boxes and segmentation masks for each instance of an object in the image. It's based on Feature Pyramid Network (FPN) and a ResNet101 backbone.

DanaXa_Mask R-CNN_Demo.ipynb (Attached):

It shows an example of using a model pre-trained on MS COCO to segment objects in your own images. It includes code to run object detection and instance segmentation on arbitrary images. (See Fig.7)

[2] https://github.com/matterport/Mask_RCNN

Fig. 7. Results of the Mask R-CNN-Demo

DanaXa_Mask R-CNN_Train_shapes.ipynb (Attached):

It shows how to train Mask R-CNN on your own dataset. This notebook introduces a toy dataset (Shapes) to demonstrate training on a new dataset. Below (Fig.8-10), you can see the training, detection, and evaluation results of the proposed model.

```
# which layers to train by name pattern.
model.train(dataset_train, dataset_val,
            learning_rate=config.LEARNING_RATE,
            epochs=1,
            layers='heads')
```

```
Starting at epoch 0. LR=0.001

Checkpoint Path: /content/Mask_RCNN/logs/shapes20210818T1509/mask_rcnn_shapes_{epoch:04d}.h5
Selecting layers to train
fpn_c5p5               (Conv2D)
fpn_c4p4               (Conv2D)
fpn_c3p3               (Conv2D)
fpn_c2p2               (Conv2D)
fpn_p5                 (Conv2D)
fpn_p2                 (Conv2D)
fpn_p3                 (Conv2D)
fpn_p4                 (Conv2D)
In model:  rpn_model
    rpn_conv_shared        (Conv2D)
    rpn_class_raw          (Conv2D)
    rpn_bbox_pred          (Conv2D)
mrcnn_mask_conv1       (TimeDistributed)
mrcnn_mask_bn1         (TimeDistributed)
mrcnn_mask_conv2       (TimeDistributed)
mrcnn_mask_bn2         (TimeDistributed)
mrcnn_class_conv1      (TimeDistributed)
mrcnn_class_bn1        (TimeDistributed)
mrcnn_mask_conv3       (TimeDistributed)
mrcnn_mask_bn3         (TimeDistributed)
mrcnn_class_conv2      (TimeDistributed)
mrcnn_class_bn2        (TimeDistributed)
mrcnn_mask_conv4       (TimeDistributed)
mrcnn_mask_bn4         (TimeDistributed)
mrcnn_bbox_fc          (TimeDistributed)
mrcnn_mask_deconv      (TimeDistributed)
mrcnn_class_logits     (TimeDistributed)
mrcnn_mask             (TimeDistributed)
WARNING:tensorflow:From /usr/local/lib/python3.7/dist-packages/tensorflow/python/ops/math_ops.py:3066: to_int32 (from t
Instructions for updating:
Use tf.cast instead.
/usr/local/lib/python3.7/dist-packages/tensorflow/python/ops/gradients_impl.py:110: UserWarning: Converting sparse Inde
  "Converting sparse IndexedSlices to a dense Tensor of unknown shape. "
/usr/local/lib/python3.7/dist-packages/keras/engine/training.py:1987: UserWarning: Using a generator with `use_multiprc
  UserWarning('Using a generator with `use_multiprocessing=True`'
Epoch 1/1
 20/100 [=====>.....................] - ETA: 3334s - loss: 3.9794 - rpn_class_loss: 0.0743 - rpn_bbox_loss: 1.2477 -
```

Fig. 8. Results of the Mask R-CNN-training

```
[15] results = model.detect([original_image], verbose=1)

     r = results[0]
     visualize.display_instances(original_image, r['rois'], r['masks'], r['class_ids'],
                                 dataset_val.class_names, r['scores'], ax=get_ax())

     Processing 1 images
     image                    shape: (128, 128, 3)         min:   28.00000  max:  204.00000  uint8
     molded_images            shape: (1, 128, 128, 3)      min:  -88.70000  max:  100.10000  float64
     image_metas              shape: (1, 16)               min:    0.00000  max:  128.00000  int64
     anchors                  shape: (1, 4092, 4)          min:   -0.71267  max:    1.20874  float32
```

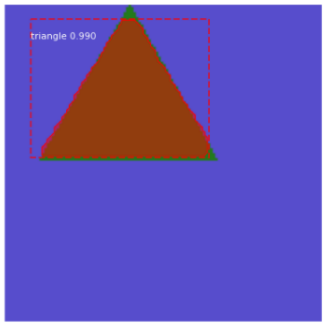

Fig. 9. Results of the Mask R-CNN-detection

Evaluation

```
[16] # Compute VOC-Style mAP @ IoU=0.5
     # Running on 10 images. Increase for better accuracy.
     image_ids = np.random.choice(dataset_val.image_ids, 10)
     APs = []
     for image_id in image_ids:
         # Load image and ground truth data
         image, image_meta, gt_class_id, gt_bbox, gt_mask =\
             modellib.load_image_gt(dataset_val, inference_config,
                                    image_id, use_mini_mask=False)
         molded_images = np.expand_dims(modellib.mold_image(image, inference_config), 0)
         # Run object detection
         results = model.detect([image], verbose=0)
         r = results[0]
         # Compute AP
         AP, precisions, recalls, overlaps =\
             utils.compute_ap(gt_bbox, gt_class_id, gt_mask,
                              r["rois"], r["class_ids"], r["scores"], r['masks'])
         APs.append(AP)

     print("mAP: ", np.mean(APs))

     mAP:  0.966666667163372
```

Fig. 10. Results of the Mask R-CNN-evaluation

5.2) Three Ways to Improve Semantic Segmentation with Self-Supervised Depth Estimation[3]

I chose this item because it is new work. Training deep networks for semantic segmentation requires large amounts of labeled training data, which presents a major challenge in practice, as labeling segmentation masks is a highly labor-intensive process. To address this issue, Hoyer et al., 2021 presented a framework for semi-supervised semantic segmentation, which is enhanced by self-supervised monocular depth estimation from unlabeled images. In particular, they proposed three key contributions:

[3] https://github.com/lhoyer/improving_segmentation_with_selfsupervised_depth

1) Transferring knowledge from features learned during self-supervised depth estimation to semantic segmentation.
2) implementation of a strong data augmentation by blending images and labels using the structure of the scene.
3) Utilization of the depth feature diversity as well as the level of difficulty of learning depth in a student-teacher framework to select the most useful samples to be annotated for semantic segmentation.

They validate the proposed model on the Cityscapes dataset, where all three modules demonstrate significant performance gains, and they achieve state-of-the-art results for semi-supervised semantic segmentation. Below, you can see the qualitative results of the proposed model trained with only 100 annotated semantic segmentation samples.

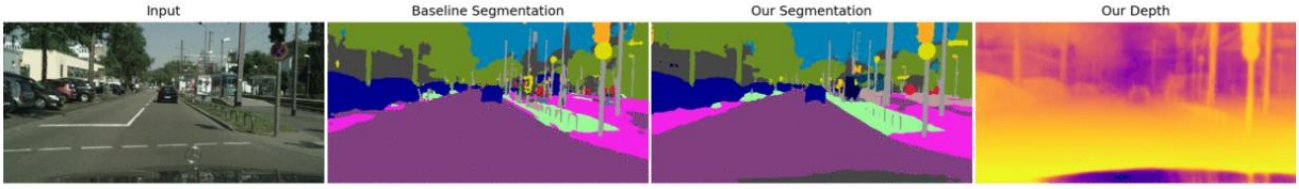

Fig. 11. Qualitative results of the proposed model

Also, I implement Inference with a Pretrained Model in

DanaXa_Self-Supervised Depth Estimation.ipynb (Attached):

I tested their pretrained model (trained on 372 Cityscapes images) on some of my own images (See Fig.12), you can download the checkpoint here, unzip it, and run it using:

```
!python inference.py --machine ws --model /path/to/checkpoint/dir/ --data /path/to/data/dir/
```

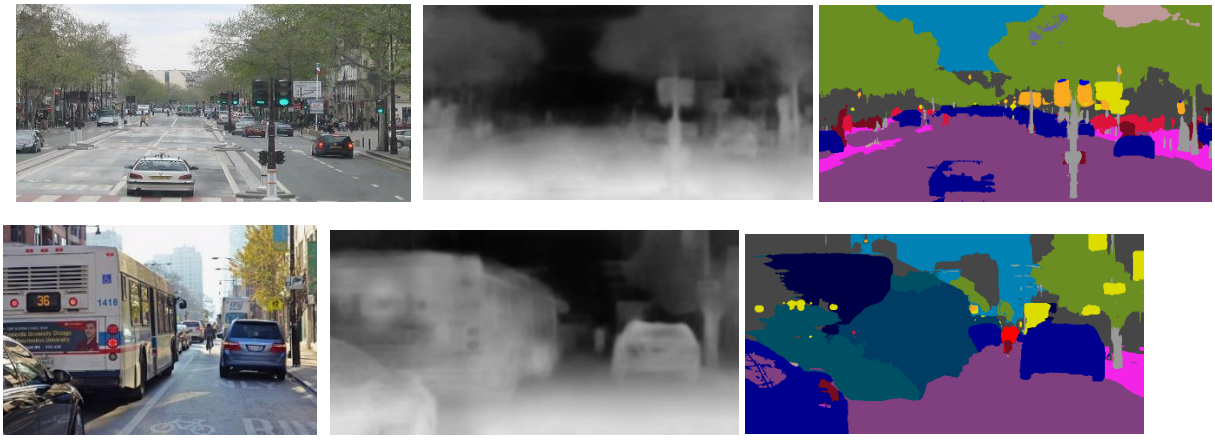

Fig. 12. Input, depth, and label outputs in my implementation of Semantic Segmentation with Self-Supervised Depth Estimation

References

Caelles, S., Maninis, K. K., Pont-Tuset, J., Leal-Taixé, L., Cremers, D., & Van Gool, L. (2017). One-shot video object segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 221-230).

Fathi, A., Balcan, M. F., Ren, X., & Rehg, J. M. (2011). Combining self training and active learning for video segmentation. Georgia Institute of Technology.

Hoyer, L., Dai, D., Chen, Y., Koring, A., Saha, S., & Van Gool, L. (2021). Three ways to improve semantic segmentation with self-supervised depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 11130-11140).

Lu, X., Wang, W., Shen, J., Tai, Y. W., Crandall, D. J., & Hoi, S. C. (2020). Learning video object segmentation from unlabeled videos. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 8960-8970).

Luiten, J., Voigtlaender, P., & Leibe, B. (2018, December). Premvos: Proposal-generation, refinement and merging for video object segmentation. In *Asian Conference on Computer Vision* (pp. 565-580). Springer, Cham.

Miao, B., Bennamoun, M., Gao, Y., & Mian, A. (2021). Self-Supervised Video Object Segmentation by Motion-Aware Mask Propagation. *arXiv preprint arXiv:2107.12569*.

Misra, I., Shrivastava, A., & Hebert, M. (2015). Watch and learn: Semi-supervised learning for object detectors from video. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 3593-3602).

Mittal, S., Tatarchenko, M., & Brox, T. (2019). Semi-supervised semantic segmentation with high-and low-level consistency. IEEE transactions on pattern analysis and machine intelligence.

Mondal, A. K., Agarwal, A., Dolz, J., & Desrosiers, C. (2019). Revisiting cyclegan for semi-supervised segmentation. *arXiv preprint arXiv:1908.11569*.

Valvano, G., Leo, A., & Tsaftaris, S. A. (2021). Learning to Segment from Scribbles using Multi-scale Adversarial Attention Gates. *IEEE Transactions on Medical Imaging*.

Waleed, A. (2017). Mask R-CNN for object detection and instance segmentation on Keras and TensorFlow. https://github.com/matterport/Mask_RCNN

Wang, B. H., Chao, W. L., Wang, Y., Hariharan, B., Weinberger, K. Q., & Campbell, M. (2019). LDLS: 3-D object segmentation through label diffusion from 2-D images. *IEEE Robotics and Automation Letters*, *4*(3), 2902-2909.

Wang, W., Shen, J., Porikli, F., & Yang, R. (2018). Semi-supervised video object segmentation with super-trajectories. IEEE transactions on pattern analysis and machine intelligence, 41(4), 985-998.

Wei, Y., Xiao, H., Shi, H., Jie, Z., Feng, J., & Huang, T. S. (2018). Revisiting dilated convolution: A simple approach for weakly-and semi-supervised semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 7268-7277).

Zhou, Y., Chen, H., Lin, H., & Heng, P. A. (2020, October). Deep semi-supervised knowledge distillation for overlapping cervical cell instance segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention* (pp. 521-531). Springer, Cham.

Zhou, Y., Wang, X., Jiao, J., Darrell, T., & Yu, F. (2020). Learning saliency propagation for semi-supervised instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 10307-10316).