A SURVEY OF IMAGE AND VIDEO INSTANCE SEGMENTATION USING DEEP LEARNING

by

LEI KE

A Thesis Submitted to
The Hong Kong University of Science and Technology
for the PhD Qualifying Exam
in Computer Science and Engineering

August 2021, Hong Kong

A SURVEY OF IMAGE AND VIDEO INSTANCE SEGMENTATION USING DEEP LEARNING

by

LEI KE

Prof. Chi-Keung Tang, Thesis Supervisor

Prof. Dit-Yan YEUNG, Head of Department

Department of Computer Science and Engineering

26 August 2021

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

A SURVEY OF IMAGE AND VIDEO INSTANCE SEGMENTATION USING DEEP LEARNING

by

LEI KE

Department of Computer Science and Engineering

The Hong Kong University of Science and Technology

ABSTRACT

Instance segmentation is a fundamental research topic in computer vision with many real-world applications, including image/video editing, scene understanding and robotic perception. Various instance segmentation algorithms have been developed in the literature with remarkable progress. However, their performance is still not desirable when deploying in complex real-world environments, such as segmenting highly-overlapping instances or objects of novel categories. Besides, effectively and efficiently leveraging the rich temporal information in video segmentation remains a challenge.

In this survey, we give a comprehensive review of deep learning-based image/video instance segmentation methods. We first introduce how to advance segmentation performance with the bilayer decoupling structure and commonality-parsing techniques. We then present an effective and efficient prototypical cross-attention network for video instance segmentation. Next, we further show an application of video instance segmentation: object based video inpainting. Their limitations are also analyzed. In the end, we conclude this survey by summarizing several future research directions.

CHAPTER 1

INTRODUCTION

Instance segmentation is to detect and segment each distinct object of interest appearing in an image, which serves as a primary technique for many real-world applications, such as image/video editing [101, 112], scene understanding, autonomous driving [35, 84] and medical image analysis.

As a hybrid task of semantic segmentation and object detection, instance segmentation has achieved remarkable progress [66, 39, 77, 20, 12, 19] but still faces many challenges in complex real-world environments. First of all, segmenting highly-overlapping objects is difficult because typically no distinction is made between real object contours and occlusion boundaries. We observe a lot of segmentation errors are caused by occlusions which commonly occur in real images.

Another challenge comes from segmenting objects of novel categories since real world data often have a long-tailed and open-ended distribution. Typical instance segmentation methods rely on the fully supervised learning with precise mask-annotated data while this kind of pixel-level mask annotation is extremely labor-consuming. It is expensive to annotate a large number of rare categories for deep learning methods, so we study partially/semi supervised instance segmentation proposed for generalizable segmentation.

State-of-the-art approaches in instance segmentation often follow the Mask R-CNN [39] paradigm with the first stage detecting bounding boxes, followed by the second stage to segment instance masks. Mask R-CNN and its variants [77, 12, 20, 46, 19] have demonstrated notable performance, and most of the leading approaches in the COCO instance segmentation challenge [74] have adopted this pipeline. For more efficiency and flexibility, one-stage instance segmentation methods [126, 9, 119] remove the proposal generation and feature re-pooling steps, which has simpler procedures than their two-stage counterparts, but tend to be less accurate.

Nowadays, new challenges are exposed to instance segmentation task. With the popularity of new types of social media, especially shot video clip and live video streaming

applications, video instance segmentation (multiple object tracking and segmentation) attracts rapidly growing research interests. The temporal dimension carries rich information about the scene and the multiple temporal views of an object has the potential of improving the quality of predicted segmentation, localization, and categories. However, effectively and efficiently leveraging the rich temporal information remains a challenge.

In this survey, we give a comprehensive and systematical survey of deep learning-based image and video instance segmentation methods.

We first introduce how to promote image instance segmentation performance from bilayer structure decoupling. A method, called Bilayer Convolutional Network (BCNet), simultaneously predicting both occluding region (occluder) and partially occluded object (occludee) after ROI extraction is described in detail. Unlike previous occlusion-aware mask heads, which only regress both modal and amodal masks from the occludee, our BCNet has a bilayer GCN structure and considers the interactions between the top “occluder" and bottom “occludee" in the same ROI. The occlusion perception branch explicitly models the occluding object by performing joint mask and contour predictions, and distills essential occlusion information for the second graph layer to segment target object (occludee).

We then design a partially supervised commonality-parsing method, called CPMask, to generalize segmentation for novel object categories by supervised learning. In particular, we aim to learn two types of generalized commonalities: 1) the shape commonalities that can be generalized between different categories like similar instance contour or similar instance boundary features, captured by the Boundary-Parsing Module; 2) the appearance commonalities that shared among categories of instances owning similar appearance features such as similar texture or similar color distribution by the Non-local Affinity-Parsing Module.

Next, we present how to promote temporal segmentation performance when adapting image instance segmentation to video instance segmentation. A Prototypical Cross-Attention Network (PCAN) that efficiently exploits temporal information from long-term space-time memory is elaborated, which first distills a space-time memory into a set of prototypes and then employs cross-attention to retrieve rich information from the past frames. To segment each object, PCAN adopts a prototypical appearance module to learn a set of contrastive foreground and background prototypes, which are then propagated over

time. Besides, we further show a proposed application for video instance segmentation: Occlusion-Aware Video Object Inpainting, where we propose the VOIN (Video Object Inpainting Network), a unified multi-task framework for joint video object mask completion and object appearance recovery to infer invisible occluded object regions.

Finally, we conclude this survey by analyzing their limitations and also summarizing several future research directions for instance segmentation.

CHAPTER 2

PRELIMINARIES AND RELATED WORK

In this chapter, we first introduce works in image instance segmentation, especially for segmenting highly-overlapping occluded instances or objects of novel categories. Then we introduce video instance segmentation, also known as multiple object tracking and segmentation, especially on how to leverage the rich temporal information for improving segmentation performance. Finally, we introduce an application of video instance segmentation: video object inpainting.

2.1 Image Instance Segmentation

Two stage instance segmentation methods [66, 39, 77, 20, 12, 19, 22] achieve state-of-the-art performance by first detecting bounding boxes and then performing segmentation in each ROI region. FCIS [66] introduces the position-sensitive score maps within instance proposals for mask segmentation. Mask R-CNN [39] extends Faster R-CNN [98] with a FCN branch to segment objects in the detected box. PANet [77] further integrates multi-level feature of FPN to enhance feature representation. MS R-CNN [46] mitigates the misalignment between mask quality and score. CenterMask [61] is built upon the anchor free detector FCOS [107] with a SAG-Mask branch. In contrast, our BCNet is a bilayer mask prediction network for addressing the issues of heavy occlusion and overlapping objects in two-stage instance segmentation. Experiments validate that our approach leads to significant performance gain on overall instance segmentation performance not limited to heavily occluded cases.

One-stage instance segmentation methods remove the bounding box detection and feature re-pooling steps. AdaptIS [105] produces masks for objects located on point proposals. PolarMask [126] models instance masks in polar coordinates by instance center classification and dense distance regression. YOLOACT [9] introduces prototype

masks with per-instance coefficients. SOLO [119] applies the "instance categories" concept to directly output instance masks based on the location and size. Grouping-based approaches [55, 3, 76, 80, 6, 56] regard segmentation as a bottom-up grouping task by first producing pixel-wise predictions followed by grouping object instances in the post-processing stage. These one-stage methods, with simpler procedures than their two-stage counterparts, are more efficient but tend to be less accurate.

2.2 Occlusion-aware Instance Segmentation

Segmentation with Occlusion Handling Methods for occlusion handling have been proposed [106, 123, 34, 21]. A layout consistent random field is used in [123] to segment images of cars and faces by imposing asymmetric local spatial constraints. Ghiasi *et al*. [36] model occlusion by learning deformable models with local templates for human pose estimation while [48] reconstructs dense 3D shape for vehicle pose. Tighe *et al*. [108] build a histogram to predict occlusion overlap scores between two classes for inferring occlusion order in the scene parsing task. Chen *et al*. [23] handle occlusion by incorporating category specific reasoning and exemplar-based shape prediction for instance segmentation. For pedestrian detection with occlusion, bi-box regression is proposed in [147] for both full body and visible part estimation while repulsion loss [120] and aggregation loss [145] are designed to improve the detection accuracy. SeGAN [29] learns occlusion patterns by segmenting and generating the invisible part of an object. Recently, OCFusion [59] uses an additional branch to model instances fusion process for replacing detection confidence in panoptic segmentation. A self-supervised scene de-occlusion method is proposed in [141] by recovering the occlusion ordering and completing the mask and content for the invisible object parts.

Amodal Instance Segmentation Different from traditional segmentation which only focuses on visible regions, amodal instance segmentation can predict the occluded parts of object instances. Li and Malik [63] first propose a method by extending [62], which iteratively enlarges the modal bounding box following the direction of high heatmap values and synthetically adds occlusion. Zhu *et al*. [149] propose a COCO amodal dataset with 5000 images from the original COCO and use AmodalMask as a baseline, which is SharpMask [92] trained on amodal ground truth. COCOA *cls* [32] augments this dataset

by assigning class-labels to the objects while SAIL-VOS dataset in [44] is targeted for video object segmentation. In autonomous driving, Qi *et al.* [94] establish the large-scale KITTI [35] InStance segmentation dataset (KINS) and present ASN to improve amodal segmentation performance.

2.3 Instance Segmentation for Novel Objects

Generalizing instance segmentation model to novel categories with limited annotations is meaningful and challenging, which mainly has three different settings: **Weakly supervised** instance segmentation methods are developed to use weak labels to segment novel categories where the training samples are only annotated with bounding boxes [50, 97] or image-level labels [148, 1] without pixel-level annotations. **Few-shot supervised** instance segmentation [130] is proposed to solve this problem by imitating the human visual systems to learn new visual concepts with only a few well-annotated samples. **Partially supervised** instance segmentation is formulated in a mixture of strongly and weakly annotated scenario where only a small subset of base categories are well-annotated with both box and mask annotations while the novel categories only have box annotations. In Mask^X R-CNN [43], a parameterized weight transfer function is designed to transfer the visual information from detection to segmentation while ShapeMask [58] learns the intermediate concept of object shape as the prior knowledge.

2.4 Video Instance Segmentation

Video instance segmentation (VIS) Existing VIS methods [133, 7, 69] widely adapt the two-stage paradigm of Mask R-CNN [39] and its variants [46, 49] by adding an additional tracking branch. Thus, their typical pipelines first detect regions of interest (RoIs) and then use the instance features after RoIAlign to regress object mask and associate cross-frame instances. More recent works [14, 64, 75] employ a one-stage instance segmentation method, e.g. the anchor-free FCOS detector [107], which predicts a linear combination of mask bases [10] as its final segmentation. The aforementioned approaches make very limited use of temporal information to enhance the quality of the segmentation, instead relying on single image-based mask prediction, or only model short-term temporal correlation

between two consecutive frames [64, 93]. In the context of long-term temporal association, the offline method VisTr [121] adapts vision transformer [15] for VIS, but suffers from a huge computational burden and memory consumption due to the dense pixel-level attention operations over long sequences.

Multiple Object Tracking and Segmentation (MOTS) Similar to VIS, MOTS methods [114, 85, 90] mainly follow the tracking-by-detection paradigm. Objects are first detected and segmented, followed by association between frames. Track R-CNN [114] integrates temporal context feature from two neighboring frames using 3D convolutions. TrackFormer [83] performs joint object detection and tracking by recurrently using Transformers, while Stem-Seg [4] adopts a short 3D convolutional spatio-temporal volume to learn pixel embedding by treating segmentation as a bottom-up grouping. In contrast, our approach clusters appearance features in a long spatio-temporal volume with explicit foreground and background prototypes that are updates online. Besides, the mixture Gaussian components in instance appearance module equips PCAN a stronger modeling ability compared to instance-level average pooling [104, 135] or single Gaussian model [142, 47].

Temporal attention models Video understanding usually requires long-range sequential modeling of relations between spatio-temporal locations. Recently, attention-based approaches, such as non-local attention [118, 117, 88, 42] and transformers [28, 109], have been successfully adopted in video classification and action recognition. These tasks [81, 102, 127] involve dense pixel-level attention, leading to quadratic complexity in the sequence length, thus making them excessively expensive for long sequences. Improved temporal attention models mainly include double attention mechanism [24] on image recognition with global-local decomposition, and clustered attention Transformer [115] for language sequence modeling. Besides, recent prototypical methods [65, 132] use the EM algorithm for single-image semantic segmentation or few-shot learning [104]. Unlike these methods, our PCAN uses compact prototypical representation both for temporal feature aggregation and compact instance appearance feature propagation.

2.5 Occlusion-aware Video Object Inpainting

Video Inpainting. Previous works [122, 87, 37, 45, 82] on video inpainting fill arbitrary missing regions with visually pleasing content by learning spatial and temporal coherence, with deep learning based approaches [116, 129, 143, 51, 60, 89, 140, 16, 17] becoming mainstream in recent years. The first deep generative model applied in video inpainting [116] combines 3D and 2D convolutions to produce temporally consistent inpainting content. To achieve better temporal consistency, in [129, 143, 33] optical flows are used to guide propagation of information across frames. In [51] a temporal memory module is used with recurrent feedback. For modeling long-range dependencies, in [60] frame-wise attention is applied on frames aligned by global affine transformation, and in [89] the authors adopt pixel-wise attention to progressively fill the hole from its boundary. Temporal PatchGAN [16] based on SN-PatchGAN [139] and temporal shift modules [17] are proposed to further enhance inpainting quality. Most recently, a spatio-temporal transformer is proposed in [140] for video completion by using a multi-scale patch-based attention.

Amodal Object Completion. In amodal object completion, visible masks of objects are given and the task is to complete the modal into amodal masks, which is different from amodal instance segmentation [63, 149, 44, 94]. Previous amodal mask completion approaches make assumptions about the occluded regions, such as Euler Spiral [53], cubic Béziers [70] and simple curves (straight lines and parabolas) [103]. These unsupervised methods cannot deal with objects with complex shapes.

Among the prior amodal object completion works with appearance recovery, Ehsani *et al* [29] generate the occluded parts of objects using Unet [100] by leveraging about 5,000 synthetic images restricted to indoor scenes such as kitchen and living room. Yan *et al* [131] recover the appearance of occluded cars by synthesizing occluded vehicle dataset. Zhan *et al* [141] propose a self-supervised scene de-occlusion method PCNet which can complete the mask and content for invisible parts of more common objects without amodal annotations as supervisions. However, all of these methods are single image-based without considering temporal coherence and object motions. Extending them directly to complex video sequence may easily lead to unwanted temporal artifacts.

CHAPTER 3

OCCLUSION-AWARE IMAGE INSTANCE SEGMENTATION

In this chapter, we present how to improve instance segmentation performance under occlusions with the benefit of bilayer structure decoupling.

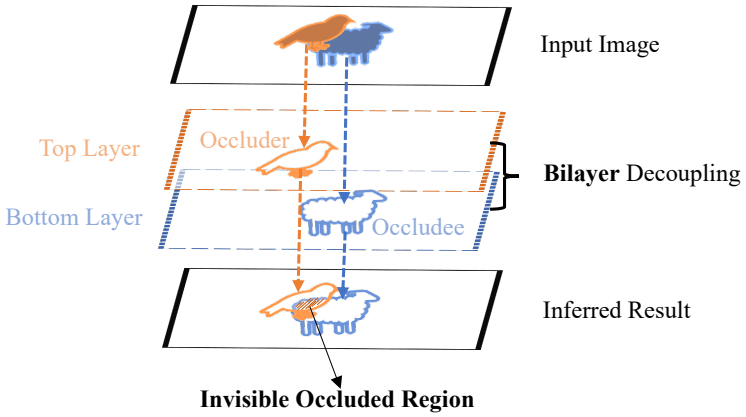
3.1 Overview

State-of-the-art approaches in instance segmentation often follow the Mask R-CNN [39] paradigm with the first stage detecting bounding boxes, followed by the second stage to segment instance masks. Mask R-CNN and its variants [77, 12, 20, 46, 19] have demonstrated notable performance, and most of the leading approaches in the COCO instance segmentation challenge [74] have adopted this pipeline. However, we note that most incremental improvement comes from better backbone architecture designs, with little attention paid in the instance mask regression after obtaining the ROI (Region-of-Interest) features from object detection. We observe that a lot of segmentation errors are caused by overlapping objects, especially for object instances belonging to the same class. This is because each instance mask is individually regressed, and the regression process implicitly assumes the object in an ROI has almost complete contour, since most objects in the training data in COCO do not exhibit significant occlusions.

We propose the Bilayer Convolutional Network (BCNet). As illustrated in Figure 3.1, BCNet simultaneously regresses both occluding region (occluder) and partially occluded object (occludee) after ROI extraction, which groups the pixels belonging to the occluding region and treat them equally as the pixels of the occluded object but in *two separate image layers*, and thus naturally decouples the boundaries for both objects and considers the interaction between them during the mask regression stage.

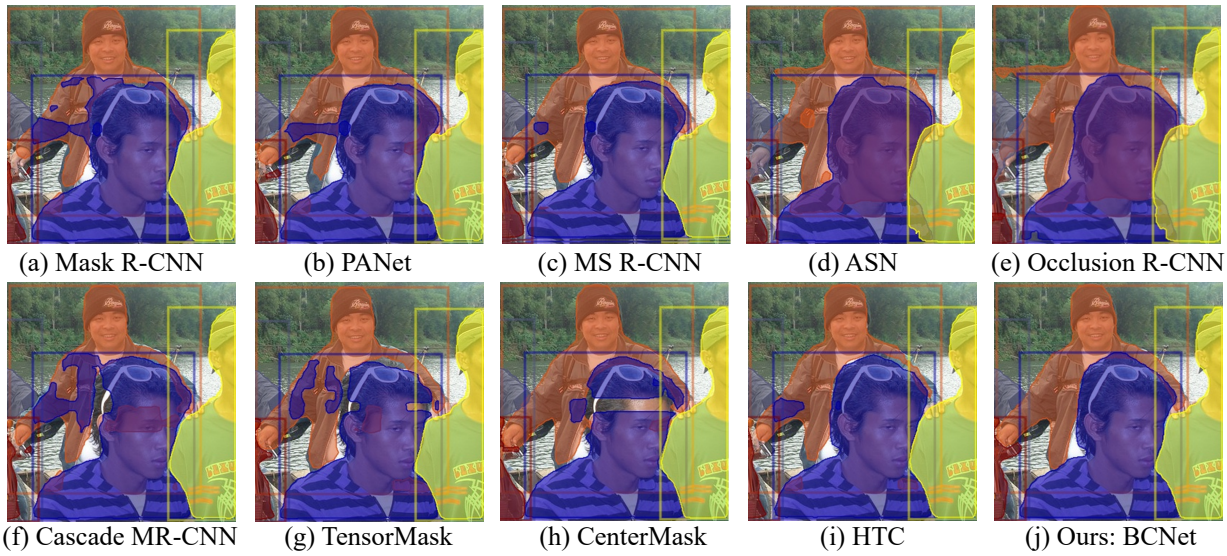
Previous approaches resolve the mask conflict between neighboring objects through non-maximum suppression or additional post-processing [78, 27, 62, 57, 38]. Consequently,

Figure 3.1: Simplified illustration of bilayer structure decoupling.



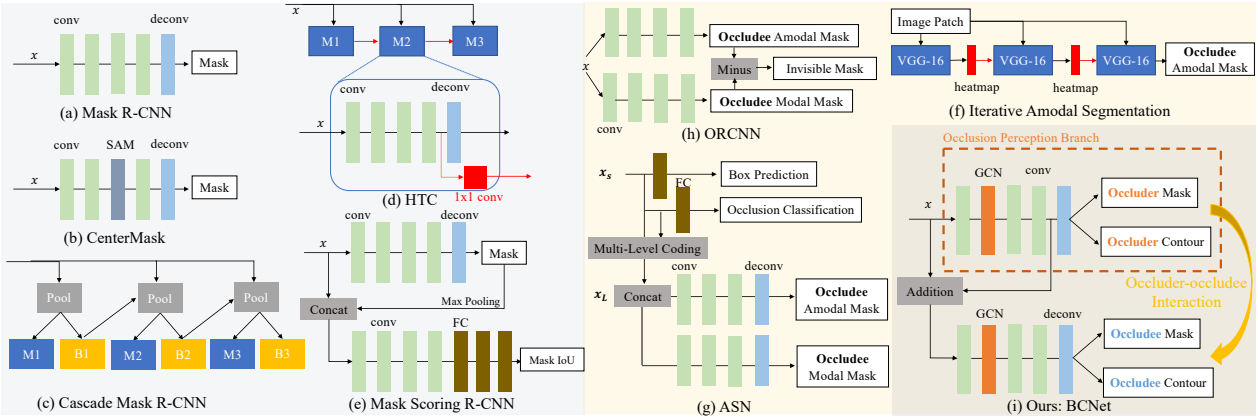
Unlike previous segmentation approaches operating on a single image layer (i.e., directly on the input image), we decouple overlapping objects into *two image layers*, where the top layer deals with the occluding objects (**occluder**) and the bottom layer for **occludee** (which is also referred to as target object in other methods as they do not explicitly consider the occluder). The overlapping parts of the two image layers indicate the invisible region of the occludee, which is explicitly modeled by our occlusion-aware BCNet framework.

Figure 3.2: Instance Segmentation with overlapping objects.



Instance Segmentation on **COCO** [74] validation set by a) Mask R-CNN [39], b) PANet [77], c) Mask Scoring R-CNN [46], d) ASN [94], e) Occlusion R-CNN (ORCNN) [32], f) Cascade Mask R-CNN [12], g) TensorMask [22], h) CenterMask [61], i) HTC [19] and j) Our BCNet. Note that d) and e) are specially designed for amodal mask prediction. In this example, the bounding box is given to compare the quality of different regressed instance masks.

Figure 3.3: A brief comparison of mask head architectures.



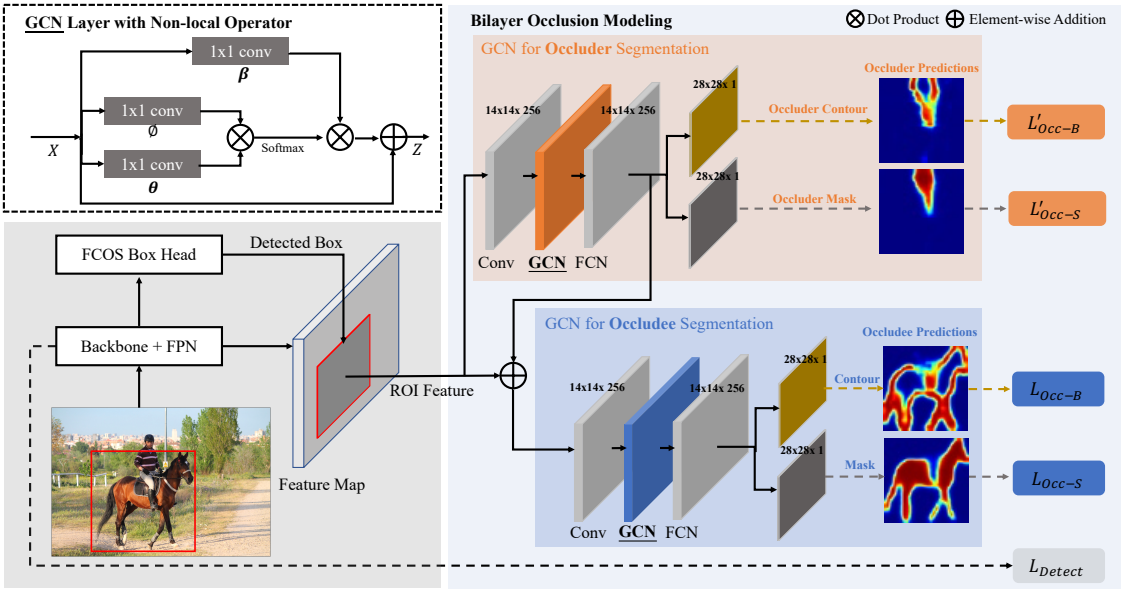
a) Mask R-CNN [39], b) CenterMask [61], c) Cascade Mask R-CNN [12], d) HTC [19], e) Mask Scoring R-CNN [46], f) Iterative Amodal Segmentation [63], g) ASN [94], h) ORCNN [32], where f), g) and h) are specially designed for amodal/occlusion mask prediction, i) Ours: BCNet.

their results are over-smooth along boundaries or exhibit small gaps between neighboring objects. Furthermore, since the receptive field in the ROI observes multiple objects that belong to the same class, when the occluding regions were included as part of the occluded object, traditional mask head design falls short of resolving such conflict, leaving a large portion of error as shown in Figure 3.2. We compare BCNet with recent amodal segmentation methods [94, 32], which predict complete object masks, including the occluded region. However, these amodal methods only regress single occluded target in the ROI, thus lacking occluder-occludee interaction reasoning, making their specially designed decoupling structure suffer when handling mask conflict between highly-overlapping objects. Correspondingly, Figure 3.3 compares the architecture of our BCNet with previous mask head designs [39, 77, 46, 19, 61, 12, 94, 32]

3.2 Method

We first give an overview to the overall instance segmentation framework, and then describe the proposed Bilayer Graph Convolutional Network (BCNet) with explicit occluder-occludee modeling. Finally, we specify the objective functions for the whole network optimization, and provide details of training and inference process.

Figure 3.4: Architecture of BCNet with bilayer occluder-occludee relational modeling.



BCNet consists of three modules; (1) Backbone [40] with FPN for feature extraction from input image; (2) Detection branch [107] for predicting instance proposals; (3) BCNet with bilayer GCN structure for mask prediction. For cropped ROI feature, the first GCN explicitly models occluding regions (occluder) by simultaneously detecting occlusion contours and masks, which distills essential shape and position information to guide the second GCN in mask prediction for the occludee. We utilize the non-local operator [117, 118] detailed in section 3.2.2 to implement the GCN layer. Visualization results are resized to square size.

3.2.1 Overall Architecture

Motivation For images with heavy occlusion, multiple overlapping objects in the same bounding box may result in confusing instance contours from both real objects and occlusion boundaries. The mask head design of Mask R-CNN and its variants [46, 19, 12, 94, 32] in Figure 3.3 directly regress the occludee with a fully convolutional network, which neglects both the occluding instances and the overlapping relations between objects. To mitigate this limitation, BCNet extends existing two stage instance segmentation methods, by adding an occlusion perception branch parallel to the traditional target prediction pipeline. Thus, the interactions between objects within the ROI region can be well considered during the mask regression stage.

Figure 3.4 gives the overall architecture of BCNet for addressing occlusion in instance segmentation. Following typical models [39, 61] for instance segmentation, our model has

three parts: (1) Backbone [40] with FPN [72] for ROI feature extraction; (2) Object detection head in charge of predicting bounding boxes as instance proposals. We employ FCOS [107] as the object detector owing to its anchor-free efficiency though our method is flexible and can deploy any existing fully supervised object detectors [98, 95, 73]; (3) The occlusion-aware mask head, BCNet, uses bilayer GCN structure for decoupling overlapping relations and segments the instance proposals obtained from the object detection branch. BCNet reformulates the traditional class-agnostic segmentation as two complementary tasks: occluder modeling using the first GCN and occludee prediction with the second GCN, where the auxiliary predictions from the first GCN provide rich occlusion cues, such as shape and positions of occluding regions, to guide target (occludee) object segmentation.

3.2.2 Bilayer Occluder-Occludee Modeling

Bilayer GCN Structure for Instance Segmentation Recently, Graph Convolutional Network (GCN) [54] has been adopted to model long-range relationships in images [25, 144, 67] and videos [118]. Given highly-overlapping objects, pixels belonging to the same partially occluded object may be separated into disjoint subregions by the occluder. Thus, we adopt GCN as our basic block due to its non-local property [117], where each graph node represents a single pixel on the feature map. To explicitly model the occluding region, we further extend the single GCN block to the bilayer GCN structure as shown in Figure 3.4, which constructs two orthogonal graphs in a single general framework.

Following [118], given an adjacency graph $\mathcal{G} = \langle \mathcal{V}, \mathcal{E} \rangle$ with edges $\mathcal{E}$ among nodes $\mathcal{V}$, we represent the graph convolution operation as,

$$\mathbf{Z} = \boldsymbol{\omega}(\mathbf{A}\mathbf{X}\mathbf{W}_g) + \mathbf{X}, \tag{3.1}$$

where $\mathbf{X} \in \mathbb{R}^{N \times K}$ is the input feature, $N = H \times W$ is the number of pixel grids within the ROI region and K is the feature dimension for each node, $\mathbf{A} \in \mathbb{R}^{N \times N}$ is the adjacency matrix for defining neighboring relations of graph nodes by feature similarities, and $\mathbf{W}_g \in \mathbb{R}^{K \times K'}$ is the learnable weight matrix for the output transform, where $K' = K$ in our case. The output feature $\mathbf{Z} \in \mathbb{R}^{N \times K'}$ consists of the updated node feature by global information propagation within the whole graph layer, which is obtained after non-linear functions $\boldsymbol{\omega}(\cdot)$ including

layer normalization [5] and ReLU functions. We add a residual connection after the GCN layer.

To construct the adjacency matrix $\mathbf{A}$, we define the pairwise similarity between every two graph nodes $\mathbf{x}_i, \mathbf{x}_j$ by dot product similarity as,

$$\mathbf{A}_{ij} = softmax(F(\mathbf{x}_i, \mathbf{x}_j)), \tag{3.2}$$

$$F(\mathbf{x}_i, \mathbf{x}_j) = \theta(\mathbf{x}_i)^T \phi(\mathbf{x}_j), \tag{3.3}$$

where θ and ϕ are two trainable transformation function implemented by 1×1 convolution as shown in the non-local operator part of Figure 3.4, so that high confidence edge between two nodes corresponds to larger feature similarity.

In our bilayer GCN structure, we further define $\mathcal{G}^i$ to indicate the ith graph, X_{roi} for the input ROI feature and $\mathbf{W}_f$ for weights in FCN layers, then the complete formulae are:

$$\mathbf{Z}^1 = \mathbf{\alpha}(\mathbf{A}^1 \mathbf{X}_f \mathbf{W}_g^1) + \mathbf{X}_f, \tag{3.4}$$

$$\mathbf{X}_f = \mathbf{Z}^0 \mathbf{W}_f^0 + \mathbf{X}_{roi}, \tag{3.5}$$

$$\mathbf{Z}^0 = \mathbf{\alpha}(\mathbf{A}^0 \mathbf{X}_{roi} \mathbf{W}_g^0) + \mathbf{X}_{roi}. \tag{3.6}$$

For connecting the two GCN blocks, the output feature $\mathbf{Z}^0$ of the occluder from the first GCN is directly added to $\mathbf{X}_{roi}$ to obtain the fused *occlusion-aware* feature $\mathbf{X}_f$, which is the input for the second GCN layer to output $\mathbf{Z}^1$ for occludee mask prediction.

Compared to previous class-agnostic mask head with single layer structure, where there is only binary label (foreground/background) per pixel, the bilayer GCN additionally constructs a new semantic graph space for *occluding region*. Thus a pixel node in overlapping areas in ROI can concurrently correspond to two different states in bilayer graph. While other choices may exist, we believe modeling GCN as a dual-layered structure as shown in Figure 3.4 is a natural choice for handling occlusion.

Occluder-occludee Modeling We explicitly model occlusion patterns by detecting both contours and masks for the occluders using the first GCN layer. Since the second GCN layer jointly predicts contours for the occludee, the overlap between the two layers can be directly identified as occlusion boundary which can thus be distinguished from real

object contour (e.g., the occluder and occludee prediction on the rightmost of Figure 3.4). The rationale behind this design is that such irregular occlusion boundary unrelated to the occludee is confusing, which in turn provides essential cues for decoupling occlusion relations. Besides, accurate boundary localization explicitly contributes to segmentation mask prediction.

The module for occluder modeling is designed in a simple yet effective way: one 3×3 convolutional layer followed by one GCN layer and one FCN layer. Then we feed the output to the up-sampling layer and one 1×1 convolutional layer to obtain one channel feature map for joint boundary and mask predictions. The boundary detection for occluder is trained with loss $\mathcal{L}'_{\text{Occ-B}}$:

$$\mathcal{L}'_{\text{Occ-B}} = \mathcal{L}_{\text{BCE}}(W_{\text{B}}\mathcal{F}_{\text{occ}}(\mathbf{X}_{\text{roi}}), \mathcal{GT}_{\text{B}}), \tag{3.7}$$

where $\mathcal{L}_{\text{BCE}}$ denotes the binary cross-entropy loss, $\mathcal{F}_{\text{occ}}$ denotes the nonlinear transformation function of the occlusion modeling module, W_{B} is the boundary predictor weight, $\mathbf{X}_{\text{roi}}$ is the cropped FPN feature map given by RoIAlign operation for the target region, and $\mathcal{GT}_{\text{B}}$ is the off-the-shelf occluder boundary that can be readily computed from mask annotations.

For occluder mask prediction, it utilizes the shared feature $\mathcal{F}_{\text{occ}}(\mathbf{X}_{\text{roi}})$, which is jointly optimized by boundary prediction. The segmentation loss $\mathcal{L}'_{\text{Occ-S}}$ for occluder modeling is designed as

$$\mathcal{L}'_{\text{Occ-S}} = \mathcal{L}_{\text{BCE}}(W_{\text{S}}\mathcal{F}_{\text{occ}}(\mathbf{X}_{\text{roi}}), \mathcal{GT}_{\text{S}}), \tag{3.8}$$

where W_{S} denotes the trainable weight of segmentation mask predictor by 1×1 convolutional layer, and $\mathcal{GT}_{\text{S}}$ is the mask annotations for the occluder.

3.3 Experiments

COCO and COCO-OCC We conduct experiments on COCO dataset [74], where we train on 2017*train* (115k images) and evaluate results on both 2017*val* and 2017*test-dev* using the standard metrics. For further investigating segmentation performance with occlusion handling, we propose a subset split, called COCO-OCC, which contains 1,005 images extracted from the validation set (5k images) where the overlapping ratio between the

bounding boxes of objects is at least 0.2. Segmenting COCO-OCC with highly overlapping objects is much more difficult than 2017*val*, where we observe a performance gap around 3.0AP for the same model in the experiment section.

KINS and COCOA We also evaluate BCNet on two amodal instance segmentation benchmarks: (1) **KINS** [94], built on the original KITTI [35], is the largest amodal segmentation benchmark for traffic scenes with both annotated amodal and modal masks for instances. BCNet is trained on the training split (7,474 images and 95,311 instances) and tested on the testing split (7,517 images and 92,492 instances) following the setting in [94]. (2) **COCOA** [149] is a subpart of COCO [74], where we train BCNet on the official training split (2,500 images) and test on the validation split (1,323 images). Note that each instance has no class label and we only use the modal and amodal mask labels for the COCOA dataset.

3.3.1 Ablation Study

Effect of Explicit Occlusion Modeling We validate the efficacy of different components proposed for explicit occlusion modeling on the first GCN layer. Table 3.1 tabulates the quantitative comparison: 1) Baseline: BCNet with no explicit occlusion modeling targets; 2) modeling segmentation masks for occluding regions (**occluder**); 3) modeling contours of the occluding regions; 4) **joint** occlusion modeling on both masks and contours. Compared to the baseline, joint occlusion modeling produces the most obvious improvement especially for the heavy occlusion cases, which promotes mask AP on the standard validation set from 32.65 to 33.43, and the AP on the proposed COCO-OCC split is increased from 29.04 to 30.37.

Table 3.1: Effect of the first GCN for occlusion modeling by predicting contours and masks on COCO with ResNet-50-FPN model.

Occlusion (Occluder) Modeling		COCO-OCC		COCO	
Contour	Mask	AP	AP_{50}	AP	AP_{50}
		29.04	49.22	32.65	52.39
	✓	29.65	49.42	33.25	52.82
✓		30.18	49.94	33.41	53.02
✓	✓	**30.37**	**50.40**	**33.43**	**53.12**

Effect of Bilayer Occluder-occludee Modeling Built on the first GCN layer with explicit occlusion modeling, we further validate the second GCN layer in Table 3.2, which demonstrates the importance of *occlusion-aware* feature *guidance* for the second GCN layer to segment target object (**occludee**) by boosting 1.23 AP on COCO-OCC, and 1.06 AP on COCO respectively. Table 3.3 shows the results comparison on adopting the proposed *bilayer structure* and existing direct regression model with single layer. On the COCO-OCC split, bilayer GCN improves AP from 29.63 to 30.68 compared to single GCN, and bilayer FCN boosts the performance of single FCN from 28.43 to 30.12.

Table 3.2: Effect of the second GCN for detecting occludee contours for final mask prediction guided by the output of first GCN.

Target (Occludee) Modeling			COCO-OCC		COCO	
Guidance	Contour	Mask	AP	AP_{50}	AP	AP_{50}
		✓	29.45	49.73	32.56	52.21
✓		✓	30.37	50.40	33.43	53.12
✓	✓	✓	**30.68**	**50.62**	**33.62**	**53.26**

Using FCN or GCN? Table 3.3 also reveals the advantage of GCN over FCN, where GCN achieves consistent superior performance both in the singe layer and bilayer structure. We also compute the number of parameters of each model and find that although GCN has more trainable parameters, the increased model size is acceptable compared to performance gain, because the feature size of input ROI has been down-sampled to only 14×14 (spatial size) with 256 channels.

Table 3.3: Effect of bilayer structure using GCN vs. FCN implementation.

Structure	FCN	GCN	COCO-OCC		COCO		Params
			AP	AP_{50}	AP	AP_{50}	
Single Layer	✓		28.43	48.24	33.01	52.62	51.0M
		✓	29.63	49.59	33.14	52.81	51.4M
Bilayer	✓		30.12	49.04	33.16	52.80	53.4M
		✓	**30.68**	**50.62**	**33.62**	**53.26**	54.0M

Influence of Object Detector To investigate the influence of object detectors to BCNet, besides using one-stage detector FCOS [107], we also use representative two-stage detector Faster R-CNN [98] to perform experiments. As shown in Table 3.4, the performance gain brought by BCNet is consistent, with an improvement of 2.23 (for FCOS) and 2.04 (for

Faster R-CNN) mask *AP* on COCO-OCC respectively. Here, baseline denotes mask head design in Mask R-CNN.

Table 3.4: Influence of the object detector (FCOS vs. Faster R-CNN) on BCNet.

Model	COCO-OCC		COCO		Params
	AP	AP_{50}	AP	AP_{50}	
FCOS [61] + Baseline	28.43	48.24	33.01	52.62	51.0M
FCOS [107] + Ours	**30.68**	**50.62**	**33.62**	**53.26**	54.0M
Faster R-CNN [39] + Baseline	29.67	49.95	33.45	53.70	60.0M
Faster R-CNN [98] + Ours	**31.71**	**51.15**	**34.61**	**54.41**	63.2M

3.3.2 Performance Comparison and Analysis

Comparison with SOTA Methods Table 3.8 compares BCNet with state-of-the-art instance segmentation methods on COCO dataset. BCGN achieves consistent improvement on different backbones and object detectors, demonstrating its effectiveness by outperforming both PANet [77] and Mask Scoring R-CNN [46] by 1.5 *AP* using Faster R-CNN, and exceeding CenterMask [61] by 1.3 *AP* using FCOS. Our single model achieves comparable result with HTC [19], which uses a 3-stage cascade refinement with multiple object detectors and mask heads, and far more parameters.

Table 3.5: Results on the COCOA dataset.

Model	AP_{all}	AP_t	AP_s
AmodalMask [149]	5.7	5.9	0.8
AmodalMRCNN [32]	21.51	21.09	9.0
ORCNN [32]	20.32	20.63	7.8
BCNet	**23.09**	**22.72**	**9.53**

Table 3.6: Results on the KINS dataset.

Model	AP_{Det}	AP_{Seg}
Mask R-CNN [32]	26.97	24.93
Mask R-CNN + ASN [94]	27.86	25.62
PANet [77]	27.39	25.99
PANet + ASN [94]	28.41	26.81
BCNet	**28.87**	**27.30**

Table 3.7: Results on COCO-OCC split.

Model	AP	AP_{50}
Mask R-CNN [40]	29.67	49.95
CenterMask [61]	29.05	49.07
MS R-CNN [46]	30.32	50.01
Ours	31.71	51.15

Comparison with Amodal Segmentation Methods Table 3.5 and Table 3.6 compare BCNet with other SOTA amodal segmentation methods on both the COCOA [149] and KINS [94] datasets, where: 1) AmodalMask [149] directly predicts amodal masks from image patches; 2) Occlusion RCNN (ORCNN) [32] is an extension of Mask R-CNN with both amodal and modal mask heads; 3) ASN module [94] contains additional occlusion classification branch and multi-level coding. Compared to these occlusion handling approaches, our bilayer GCN with cascaded structure still performs favorably against the

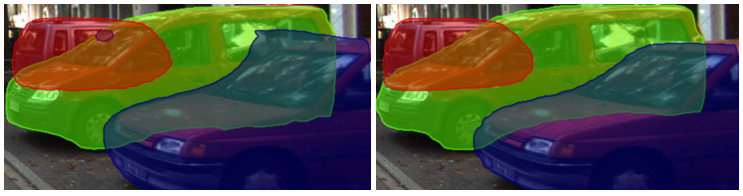
Table 3.8: Comparison with SOTA methods on COCO *test-dev* set.

Method	Backbone	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Mask R-CNN [39]	ResNet-50	35.6	57.6	38.1	18.7	38.3	46.6
PANet [77]	ResNet-50	36.6	58.0	39.3	16.3	38.1	**52.4**
BCNet + Faster R-CNN [98]	ResNet-50	**38.4**	**59.6**	**41.5**	**21.9**	**40.9**	49.3
Mask R-CNN [39]	ResNet-101	37.0	59.2	39.5	17.1	39.3	52.9
MaskLab [20]	ResNet-101	37.3	59.8	39.6	19.1	40.5	50.6
Mask Scoring R-CNN [46]	ResNet-101	38.3	58.8	41.5	17.8	40.4	54.4
BMask R-CNN [26]	ResNet-101	37.7	59.3	40.6	16.8	39.9	**54.6**
HTC [19]	ResNet-101	39.7	**61.8**	43.1	21.0	42.2	53.5
BCNet + Faster R-CNN [98]	ResNet-101	**39.8**	61.5	**43.1**	**22.7**	**42.4**	51.1
YOLACT [9]	ResNet-101	31.2	50.6	32.8	12.1	33.3	47.1
TensorMask [22]	ResNet-101	37.1	59.3	39.4	17.4	39.1	51.6
ShapeMask [58]	ResNet-101	37.4	58.1	40.0	16.1	40.1	53.8
CenterMask [61]	ResNet-101	38.3	-	-	17.7	40.8	**54.5**
BlendMask [18]	ResNet-101	38.4	60.7	41.3	18.2	41.5	53.3
BCNet + FCOS [107]	ResNet-101	**39.6**	**61.2**	**42.7**	**22.3**	**42.3**	51.0

Figure 3.5: Qualitative results of the amodal mask predictions on COCOA.

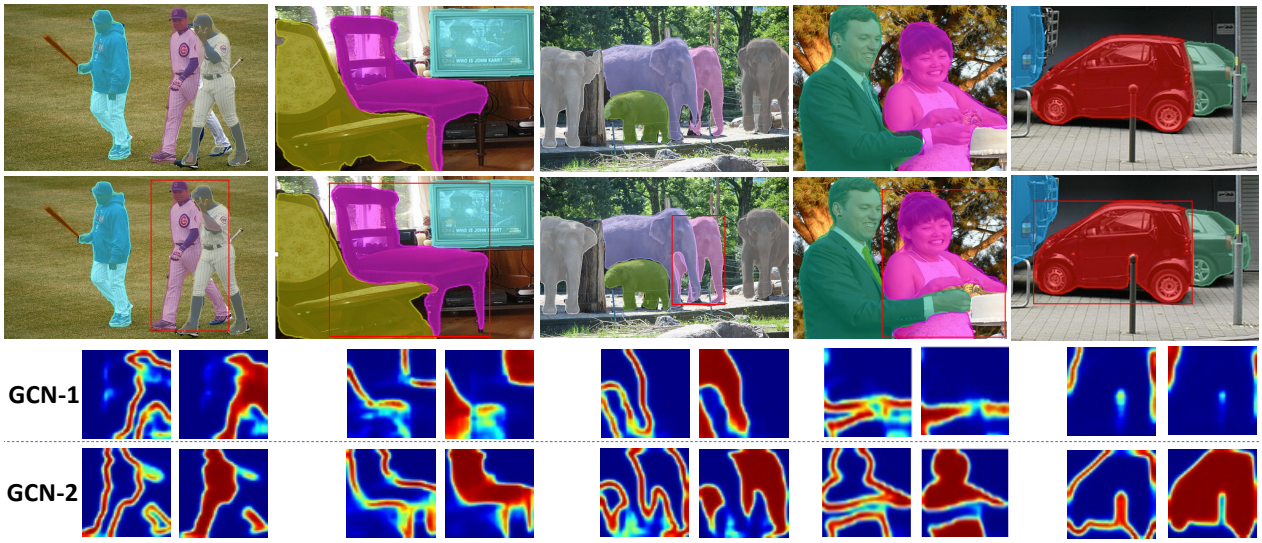
BCNet hallucinates a more reasonable shape for the baby carriage without producing a large portion of segmentation error. We remove the "stuff" background for more clarity.

Figure 3.6: Qualitative results comparison of the amodal mask predictions on KINS.



state-of-the-art methods, which shows the effectiveness of BCNet in decoupling overlapping objects and mask completion under the amodal segmentation setting. Figure 3.5 and Figure 3.6 show the qualitative comparison on COCOA and KINS respectively.

Figure 3.7: Qualitative instance segmentation results on COCO.



Qualitative instance segmentation results of CenterMask [61] (top row) and our BCNet (middle row) on **COCO** [74], both using ResNet-101-FPN and FCOS detector [107]. The bottom row visualizes squared heatmap of contour and mask predictions by the two GCN layers for the occluder and occludee in the same **ROI region** specified by the red bounding box, which also makes the final segmentation result of BCNet more explainable than previous methods.

Evaluation on Occluded Images We adopt COCO-OCC split to compare the occlusion handling ability of BCNet with other methods on images with highly overlapping objects. As shown in Table 3.7, our BCNet with Faster R-CNN detector has 31.71 *AP vs.* 30.32 for the Mask Scoring R-CNN [46]. By further training BCNet on the synthetic occlusion dataset, the performance of *AP* and AP_{50} is significantly promoted to 32.89 and 53.25 respectively, which shows the advantage brought by this new dataset.

Qualitative Evaluation. Figure 3.7 shows qualitative comparison of CenterMask [61] and BCNet on images with overlapping objects. In each ROI region, GCN-1 detects occluding regions while GCN-2 models the partially occluded instance by directly regressing the contours and masks. For example, BCNet decouples the occluding and occluded baseball players in similar clothes into GCN-1 and GCN-2 respectively, and detects the left leg missed by CenterMask.

CHAPTER 4

GENERALIZABLE IMAGE INSTANCE SEGMENTATION

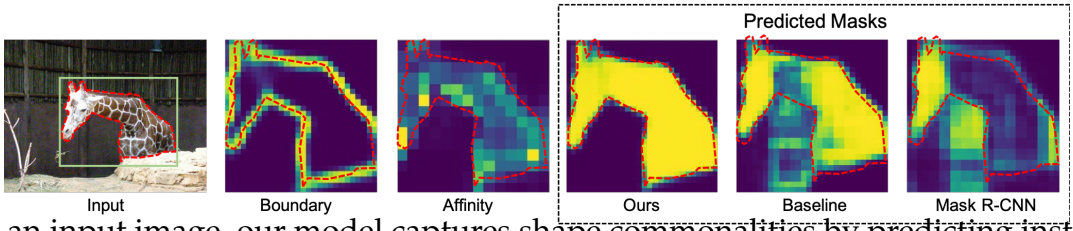
Partially supervised instance segmentation aims to perform learning on limited mask-annotated categories of data thus eliminating expensive and exhaustive mask annotation. The learned models are expected to be generaliable to novel categories. Existing methods either learn a transfer function from detection to segmentation, or cluster shape priors for segmenting novel categories.

4.1 Overview

Pixel-level mask annotation is extremely labor-consuming and thus expensive to be performed on large amount of data which is typically required for deep learning methods. On the other hand, it is less expensive and more feasible to perform annotation of bounding box for instances, which motivates the newly proposed task: *partially supervised instance segmentation* [43, 58]. It aims to learn instance segmentation models on limited mask-annotated categories of data, which can be generalized to new (novel) categories with only bounding-box annotations available. The partially supervised instance segmentation is much more challenging than the typical instance segmentation in full supervision. The major difficulty lies in how to learn the class-agnostic features for instance segmentation that can be generalized from the mask-annotated categories to novel categories.

A straightforward way for partially supervised instance segmentation is to directly extend existing fully supervised algorithms to segmentation of novel categories by class-agnostic training [91, 92]. It treats all mask-annotated categories of instances involved in training as one foreground category and forces the model to learn to distinguish between foreground and background regions for segmentation. This brute-force way of class-agnostic training expects the model to learn all the generalized features between annotated and novel categories by itself, which is hardly achieved. As the initiator of the partially

Figure 4.1: The shape and appearance commonalities between objects.



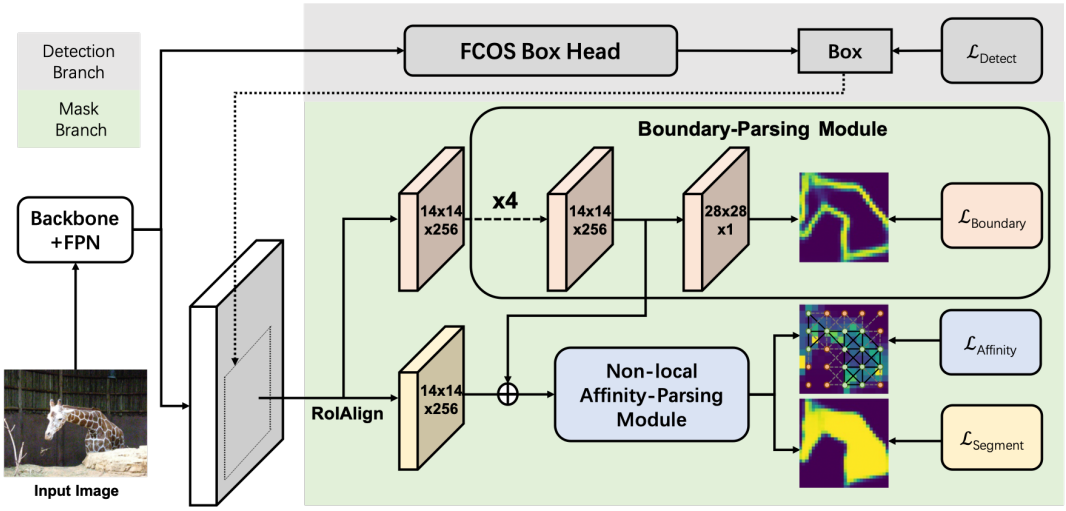
Given an input image, our model captures shape commonalities by predicting instance boundaries and learns the appearance commonalities by modeling pairwise affinities among all pixels. The learned class-agnostic commonalities in both shape and appearance enable our model to segment more accurate mask than other models.

supervised instance segmentation, Mask^X R-CNN [43] transfers the visual information from the modeling of bounding box to the mask head through a parameterized transfer function. Subsequently, ShapeMask [58] seeks to extract the generic class-agnostic shape features across different categories by summarizing a collection of shape priors as reference for segmenting new categories.

Whilst both Mask^X R-CNN and ShapeMask have distinctly advanced the performance of partially supervised instance segmentation, there are two potential limitations. First, the generalized **appearance** features for segmentation that shared across different categories, e.g., similar hairy body surface between dogs and cats or similar textures on the furniture surface, are not explicitly explored. These class-agnostic appearance features can be potentially generalized from mask-annotated categories of data to novel categories for segmentation. Second, the common **shape** features that can be generalized across different categories are not explicitly learned in a supervised way, though ShapeMask refines the shape priors by simply clustering the annotated masks and adapts them to a given novel object. In this work we intend to tackle the partially supervised instance segmentation by addressing these two issues.

We propose to capture the underlying commonalities which can be generalized across different categories by supervised learning for partially supervised instance segmentation. In particular, we aim to learn two types of generalized commonalities: 1) the shape commonalities that can be generalized between different categories like similar instance contour or similar instance boundary features; 2) the appearance commonalities that shared among categories of instances owning similar appearance features such as similar texture or similar color distribution. The resulting model, which is referred to as Commonality-

Figure 4.2: Architecture of our Commonality-Parsing Network.



CPMask consists of a detection branch and a mask branch. The cropped RoI feature based on predicted bounding boxes is first processed by Boundary-Parsing Module of the mask branch for predicting instance boundaries to guide the learning of shape commonalities in intermediate feature maps. Then the feature maps are fed into Non-local Affinity-Parsing Module (presented in Fig. 4.3) to learn the appearance commonalities by modeling the pairwise affinities among all pixels in feature maps. Finally, the feature maps incorporating both shape and appearance commonalities are used for mask prediction.

Parsing Network, can be trained in an end-to-end manner.

4.2 Method

4.2.1 Class-Agnostic Learning Framework

Figure 4.2 presents the architecture of our Commonality-Parsing Network. Following typical models [77, 39, 20] for instance segmentation, our model contains two branches: 1) the object detection branch in charge of predicting bounding boxes as instance proposals, and 2) the mask branch for predicting segmented masks for the instance proposals obtained from the object detection branch.

We adopt FCOS [107], which is an excellent one-stage detection model, as our object detection backbone. Note that it is readily replaceable by any other object detection frameworks [73, 79, 96]. As illustrated in Figure 4.2, a backbone network equipped with FPN [72] is first employed to extract intermediate convolutional features for downstream processing. The object detection branch is then utilized to predict bounding boxes with positions as well as categories for potential instances.

The mask branch is responsible for segmenting each of target instances predicted by the object detection branch. It is composed of two core modules designed specifically for class-agnostic learning by parsing the commonalities across both the shape and appearance features: Boundary-Parsing Module and Non-local Affinity-Parsing Module. These two modules are trained on a small set of mask-annotated categories of data (termed as base categories) and the learned inter-category commonality of both shape and appearance information enables our model to perform instance segmentation on novel categories of image data.

4.2.2 Boundary-Parsing Module for Learning Shape Commonality

Boundary-Parsing Module is designed to learn the underlying commonalities with respect to the shape information that can be generalized from the mask-annotated categories to mask-unseen novel categories of data.

There are several ways to design the structure of Boundary-Parsing Module and we just investigate a straightforward yet proved effective way: four 3×3 convolutional layers with ReLU as the activation functions, followed by one upsampling layer and one 1×1 convolutional layer to output one channel of feature map as boundary predictions. The Boundary-Parsing Module is trained with the boundary loss:

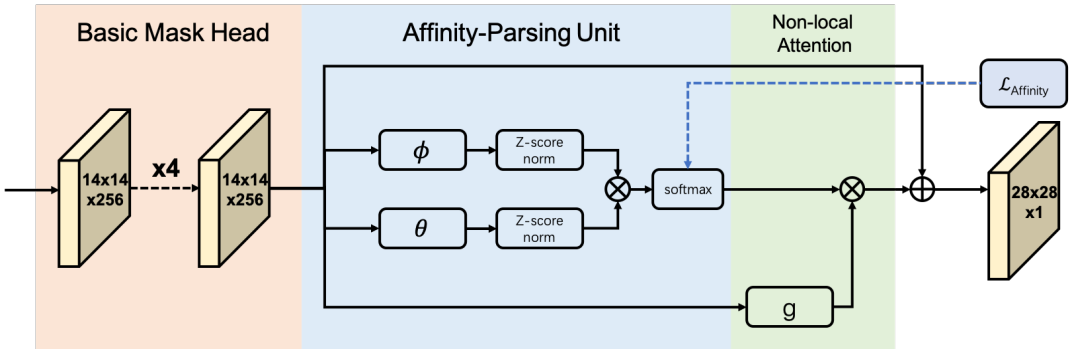
$$\mathcal{L}_{\text{boundary}} = \mathcal{L}_{\text{BCE}}(\mathcal{F}_{\text{B}}(\mathbf{X}), \mathcal{GT}_{\text{B}}), \tag{4.1}$$

where $\mathcal{L}_{\text{BCE}}$ denotes the binary cross-entropy loss, $\mathcal{F}_{\text{B}}$ denotes the nonlinear transformation functions by Boundary-Parsing Module, $\mathbf{X}$ is the output cropped FPN feature maps after *RoIAlign* operation corresponding to a target instance predicted by the object detection branch and $\mathcal{GT}_{\text{B}}$ is the off-the-shelf boundary ground-truth that can be readily obtained from mask annotations.

4.2.3 Non-local Affinity-Parsing Module for Appearance Commonality

Similar categories tend to share similar appearance commonality, e.g., similar hairy body surface between dogs and cats, or similar texture on the furniture surface. This kind of appearance commonalities can be leveraged for class-agnostic learning to generalize

Figure 4.3: Architecture of our Non-local Affinity-Parsing Module.



The Basic Mask Head processes the input feature with four convolutional layers with 3×3 kernel and ReLU. Subsequently, the Affinity-Parsing unit performs supervised learning to model the pairwise affinities among pixels in feature maps. Finally, the non-local attention is employed to coordinate feature maps based on affinities to enable our model perceive more context information and increase the appearance separation between the instance and the background.

the instance segmentation to novel categories. Therefore, we propose Non-local Affinity-Parsing Module to learn the appearance commonalities across different categories by parsing the affinities among pixels of feature maps in a non-local way. The pixels belonging to an instance (in the foreground region) are expected to have much closer affinities than the affinities between foreground and background pixels.

Formally, given the output cropped FPN feature maps $\mathbf{X}$ after *RoIAlign* operation for an instance proposal, we first fuse it with the output feature maps $\mathcal{F}_B(\mathbf{X})$ from Boundary-Parsing Module which incorporates the shape commonality information by element-wise additions. Then the nonlinear transformation $\mathcal{G}$ by four convolutional layers is performed on the fused features as a basic mask head of operations:

$$\mathbf{C} = \mathcal{G}(\mathbf{X} \oplus \mathcal{F}_B(\mathbf{X})). \tag{4.2}$$

The obtained feature maps $\mathbf{C} \in R^{c \times h \times w}$, with c feature maps of size $h \times w$, is then fed into the non-local affinity-parsing unit for modeling affinity. Specifically, we model the affinity between the pixel at (i, j) and the pixel at (m, n) in a latent embedding space by:

$$\mathbf{A}_{(<i,j>,<m,n>)} = f\Big[\frac{(\theta(\mathbf{C}_{i,j}) - \mu_{i,j})}{\sigma_{i,j}}, \frac{(\phi(\mathbf{C}_{m,n}) - \mu_{m,n})}{\sigma_{m,n}}\Big], \tag{4.3}$$

where $\mathbf{C}_{i,j} \in R^c$ corresponds to the vectorial representation (in channel dimension) for the pixel at (i, j) and the same goes for $\mathbf{C}_{m,n}$. Herein, θ, ϕ are linear embedding functions and

f is a kernel function for encoding affinity. In practice, we opt for the dot-product operator for f, which is a typical way of modeling similarity. μ and σ are the mean value and the standard deviation respectively. Note that here we apply the *z-score* normalization for both $\theta(\mathbf{C}_{i,j})$ and $\phi(\mathbf{C}_{m,n})$ to ease the convergence during optimization.

Larger affinity value indicates closer relationship while smaller affinity value implies larger difference. We expect that the affinities between pixels belonging to an instance (foreground) region are much higher than that between foreground and background pixels. To this end, we introduce a supervision signal to guide the optimization to achieve the desired affinity distribution. In particular, we impose an affinity constraint to maximize the affinities among pixels in the foreground region Fg and minimize the affinities between foreground Fg and background Bg pixels:

$$\begin{aligned}
&\mathbf{A} = \text{softmax}(\mathbf{A}),\\
&\mathcal{L}_{\text{Affinity}} = \mathcal{L}_{\text{1-norm}}(1, \sum_{\substack{<i,j>\in Fg\\<m,n>\in Fg}} \mathbf{A}_{<i,j>,<m,n>}) + \mathcal{L}_{\text{1-norm}}(0, \sum_{\substack{<i,j>\in Fg\\<m,n>\in Bg}} \mathbf{A}_{<i,j>,<m,n>}).
\end{aligned} \tag{4.4}$$

Here we first normalize $\mathbf{A}$ using a *softmax* operator and then impose the loss function that encourages the sum of affinities among foreground pixels to be close to 1 for more appearance affinities while pushing the affinities between foreground and background pixels to be 0 for larger appearance separation.

The supervised learning on the affinity distribution enables our model to perceive the appearance separability between the foreground (instance) and background regions. To further increase this appearance separation, we propose to coordinate feature maps by explicitly incorporating the learned affinities in a non-local attention manner [117, 146]:

$$\widetilde{\mathbf{C}}_{i,j} = \sum_{\forall <m,n>} \mathbf{A}_{<i,j>,<m,n>} \cdot g(\mathbf{C}_{m,n}), \tag{4.5}$$

where g is a linear embedding function. Here we coordinate the vectorial representation for the pixel at (i, j) in the feature maps by attending each pixel with the corresponding affinity. Such coordination on feature maps enables our model to perceive the context of whole image region with affinity-based attention, thus resulting in more separation of appearance between foreground and background and closer affinities among pixels in foreground (instance) region, which is beneficial for learning appearance commonalities and instance segmentation.

Together with original feature maps $\mathbf{C}$, the output coordinated feature maps $\tilde{\mathbf{C}}$ from the Non-local Affinity-Parsing Module is subsequently fed into one upsampling layer and one 1×1 convolutional layer for the final prediction of segmented mask:

$$\mathcal{L}_{\text{Segment}} = \mathcal{L}_{\text{BCE}}(\mathcal{F}_{1\times 1conv}(\tilde{\mathbf{C}} \oplus \mathbf{C}), \mathcal{GT}_S), \tag{4.6}$$

where $\mathcal{F}_{1\times 1conv}$ denotes the nonlinear transformation functions by 1×1 convolutional layer and $\mathcal{GT}_S$ is the ground-truth mask annotations.

4.3 Experiments

We conduct experiments on MS COCO dataset [74] to evaluate our model. We first perform ablation study to investigate the effect of Boundary-Parsing Module and Non-local Affinity-Parsing Module, then we compare our model with state-of-the-art methods in three different settings for instance segmentation: 1) partially-supervised setting, 2) few-shot setting and 3) fully-supervised setting.

4.3.1 Experimental Setup

We follow the typical data split on COCO: *train2017* for training and *val2017* for test. Both of them contain 80 categories of samples. In both of our experiments on partially-supervised setting and few-shot setting, we split the 80 COCO categories into “voc" and “non-voc" subsets where the *voc* subset contains the categories in PASCAL VOC [30] dataset while the remaining categories are included in the *non-voc* subset. For few-shot setting, each novel category in the training data only contains a small amount of samples with annotations of both bounding box and mask.

4.3.2 Partially-supervised Instance Segmentation

In this section we compare our model to other state-of-the-art methods for partially-supervised instance segmentation. Table 4.1 presents the quantitative results on COCO dataset with two sets of experiments: use *voc* and *non-voc* as the base categories respectively and the other one as novel categories. Our model outperforms the state-of-the-art

method	voc → non-voc						non-voc → voc					
	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Mask R-CNN [39]	18.5	34.8	18.1	11.3	23.4	21.7	24.7	43.5	24.9	11.4	25.7	35.1
Mask GrabCut [43]	19.7	39.7	17.0	6.4	21.2	35.8	19.6	46.1	14.3	5.1	16.0	32.4
$Mask^X$ R-CNN [43]	23.8	42.9	23.5	12.7	28.1	33.5	29.5	52.4	29.7	13.4	30.2	41.0
ShapeMask [58]	30.2	49.3	31.5	16.1	38.2	38.4	33.3	56.9	34.3	17.1	**38.1**	45.4
Ours	**34.4**	**53.8**	**36.8**	**18.8**	**39.1**	**47.7**	**35.9**	**59.5**	**37.8**	**18.1**	37.1	**50.8**
Oracle Mask R-CNN [43]	34.4	55.2	36.3	15.5	39.0	52.6	39.1	64.5	41.4	16.3	38.1	55.1
Oracle ShapeMask [58]	35.0	53.9	37.5	17.3	41.0	49.0	40.9	65.1	43.4	18.5	41.9	56.6
Oracle Ours	37.9	58.5	40.5	19.7	42.5	54.4	41.9	66.5	45.7	22.0	42.1	58.1

Table 4.1: Experiment results of partially supervised instance segmentation on COCO *val* set.

Figure 4.4: Qualitative results on novel COCO categories. We use *voc* subset as the base (mask-annotated) categories for training.

ShapeMask by a large margin: 4.2 AP on the *non-voc* novel categories and 2.6 AP on the *voc* novel categories respectively. Besides, we also provide the *oracle* performance which corresponds to the performance under fully-supervision and can be considered as the performance upper bound for partially-supervised learning. We observe that the performance gap between our model and its oracle version is narrowed to 3.5/6.0 AP compared to 4.8/7.6 AP by ShapeMask and 10.6/9.6 AP by $Mask^X$ R-CNN, indicating the advantages of agnostic learning by our specifically designed modules. Note that our model even obtains competitive performance with the oracle models of ShapeMask and $Mask^X$ R-CNN under the same ResNet-101 backbone.

Fig. 4.4 shows qualitative results on multiple samples that randomly selected from

method	10-shot						20-shot					
	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Mask R-CNN-ft [130]	1.9	4.7	1.3	0.2	1.4	3.2	3.7	8.5	2.9	0.3	2.5	5.8
Meta R-CNN [130]	4.4	10.6	3.3	**0.5**	3.6	7.2	6.4	14.8	4.4	**0.7**	4.9	9.3
Ours	**7.1**	**12.0**	**7.2**	0.3	**5.5**	**12.2**	**10.3**	**16.6**	**10.7**	**0.7**	**8.0**	**17.5**
Ours (no base)	4.2	9.7	3.0	0.1	3.0	8.0	6.6	13.5	5.5	0.2	5.3	12.3

Table 4.2: Experimental results of few-shot instance segmentation on COCO *val* set.

COCO dataset including various scenes, which shows that our model is able to segment all different kinds of objects precisely, even for quite small ones.

4.3.3 Few-shot Instance Segmentation

In this section, we directly apply our model to the challenging few-shot instance segmentation without any network adaption. Few-shot instance segmentation is another challenging task for novel categories. In this task, the model is first trained on base categories with numerous training samples and then generalizes to novel categories with only a few (10 or 20 shots) training samples by directly fine-tuning. Following Meta R-CNN [130], the *non-voc* subset is used as base categories with full samples per category and the *voc* subset as the novel categories with only 10/20 training samples per category. For fair comparison, we follow Meta R-CNN [130] and use ResNet-50 as backbone and input image size is resized to (600, 1000). Note that the annotations of both bounding box and mask are provided for training samples in novel categories in the few-shot setting.

As shown in Table 4.2, our model outperforms Meta R-CNN (the state-of-the-art method) by 2.7/3.9 AP in the 10/20-shot settings. Although our model is not specifically designed for few-shot learning, it still obtains the state-of-the-art performance, demonstrating that our proposed model is not limited to the partially supervised learning, and is general for other novel instance segmentation tasks. To validate the effectiveness of our model in the extreme annotation scenarios, we further propose a more challenging setting: “no base", in which the model is only trained on the novel categories with limited training samples, i.e., no pre-training on base categories. Table 4.2 shows that our model is still comparable with or better than other methods in the “no base" setting. These results in the few-shot setting manifest the genralization and effectiveness of our model in the extreme annotation scenarios.

CHAPTER 5

TEMPORAL COHERENT VIDEO INSTANCE SEGMENTATION AND ITS APPLICATION

In this chapter, we present an effective and efficient prototypical cross-attention network for video instance segmentation and show its downstream application.

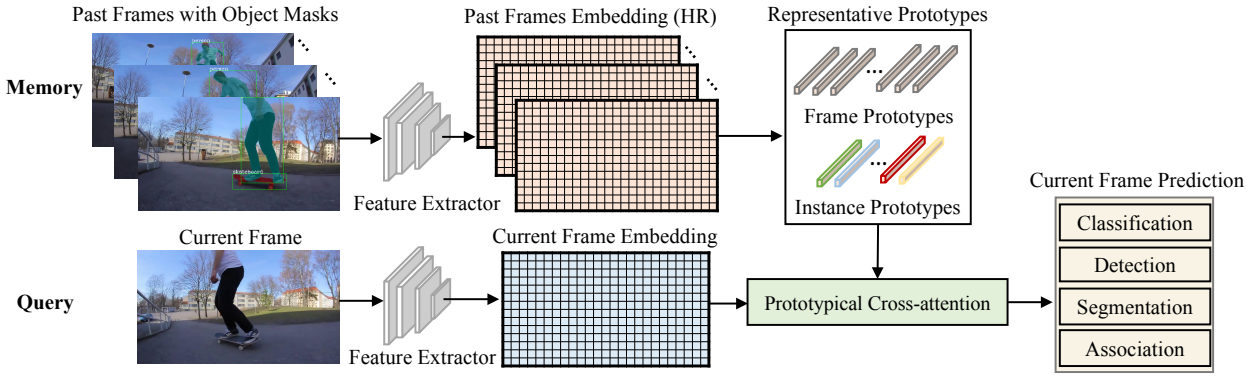
5.1 Overview

Multiple object tracking and segmentation (MOTS), also known as Video Instance Segmentation (VIS), is an important problem with many real-world applications, including autonomous driving [35, 84] and video analysis [11, 133]. The task involves tracking and segmenting all objects within a video from a given set of semantic classes. We are witnessing rapidly growing research interest on MOTS thanks to the introduction of large scale benchmarks [133, 137, 114]. State-of-the-art methods [133, 14, 114, 90] for MOTS mainly follow the tracking-by-detection paradigm, where objects are first detected and segmented in individual frames and then associated over time.

Although methods based on the popular tracking-by-detection philosophy have shown promising results, temporal modeling is limited to the object association phase [133, 14, 75] and only between two adjacent frames [114, 64]. On the other hand, the temporal dimension carries rich information about the scene. The information encoded in multiple temporal views of an object has the potential of improving the quality of predicted segmentation, localization, and categories. However, effectively and efficiently leveraging the rich temporal information remains a challenge. While sequential modeling has been applied for video processing [118, 121, 31, 88, 42], these methods generally operate directly on the high-resolution deep features, requiring large computational and memory consumption, which greatly limits their use.

We propose a **P**rototypical **C**ross-**A**ttention **M**odule, termed **PCAM**, to leverage temporal information for multiple object tracking and segmentation. As illustrated in Figure 5.1,

Figure 5.1: The Prototypical Cross-attention Network for MOTS/VIS.



PCAN first condenses the space-time memory and high-resolution frame embeddings into frame-level and instance-level prototypes. These are then employed to retrieve rich temporal information from past frames by our efficient prototypical cross-attention operation.

the module first distills spatio-temporal information into condensed prototypes using clustering based on Expectation Maximization. The resulting prototypes, composed of Gaussian Components, yield a rich and generalizable yet compact representation of the past visual features. Given a deep feature embedding of the current frame, PCAM then employs prototypical cross-attention to read relevant information from prior frames.

Based on the noise-reduced clustered video features information, we further develop a **P**rototypical **C**ross-**A**ttention **N**etwork (**PCAN**) for MOTS, that integrates the general PCAM at two stages in the network: on the frame-level and instance-level. The former reconstructs and aligns temporal past frame features with current frame, while the instance level integrates specific information about each object in the video. For robustness to object appearance change, PCAN represents each object instance by learning sets of contrastive foreground and background prototypes, which are propagated in an online manner. With a limited number of prototypes for each instance or frame, PCAN efficiently performs long-range feature aggregation and propagation in a video with linear complexity. Consequently, our PCAN outperforms standard non-local attention [118] and video transformer [121] on both the large-scale Youtube-VIS and BDD-MOTS benchmarks.

5.2 Method

5.2.1 Traditional Cross-Attention

To utilize the rich temporal information to improve the segmentation prediction, recent approaches [88, 42] have employed cross-attention. We consider past spatio-temporal information encoded in a memory $\mathbf{M}$, consisting of deep features of size $H \times W \times T \times C$. The memory encapsulates valuable information about the past appearances and predictions of objects and background in a scene. To attend to the memory, the information is first separately embedded into key $\mathbf{k}^M$ and value $\mathbf{v}^M$ feature vectors. The keys are used to address relevant memories whose corresponding values are returned. The standard memory reading process is a non-local operation computed as the weighted sum,

$$y_i = \frac{1}{Z_i} \sum_{j=1}^{H \times W \times T} \exp(\mathbf{k}_i^Q \cdot \mathbf{k}_j^M)\mathbf{v}_j^M \,, \tag{5.1}$$

where $\mathbf{k}^Q$ denotes query key map, which is predicted from the current frame. Further, i and j are the index of each query and the memory location, and $Z_i = \sum_j \exp(\mathbf{k}_i^Q \cdot \mathbf{k}_j^M)$ is the normalizing factor.

Although proven effective, the standard attention operation (5.1) is known to suffer from poor computational and memory scaling properties [68]. In particular, since all queries are matched to all keys, it experiences a quadratic scaling $\mathcal{O}((HW)^2)$ of computations in the spatial size HW of the feature map. This is particularly problematic for segmentation tasks, where fine-grained high-resolution information is desired to improve the quality of the predictions.

5.2.2 Prototypical Cross-Attention

To address the aforementioned limitations of the standard cross-attention, we introduce the prototypical cross-attention. Our approach is based on a clustered memory $\mathbf{M}_c$. We call these clusters prototypes, since they correspond to representative items in the memory. While clustering effectively reduces the number of items in the memory, it also serves to

deprecate noisy information, leading to a more generalizable and robust representation of the memory.

To employ an attention mechanism, similar to (5.1), we require a clustering of the memory that generates a principled continuous and differentiable clustering assignment function. We therefore cluster the keys in the memory by fitting a Gaussian Mixture Model (GMM),

$$p(\mathbf{k}) = \frac{1}{N}\sum_{j=1}^{N} p(\mathbf{k}|z=j)\,, \qquad p(\mathbf{k}|z=j) = \frac{1}{(2\pi\sigma^2)^{\frac{D}{2}}}\exp\left(-\frac{1}{2\sigma^2}\|\mathbf{k}-\mathbf{k}_j^{\mu}\|^2\right) \tag{5.2}$$

Here, N denotes the number of Gaussian mixtures, D is the feature dimension of the keys. We use a constant variance parameter σ^2 and uniform cluster priors $p(z=j)=\frac{1}{N}$, where z denotes the latent cluster assignment variable. The component means $\mathbf{k}^{\mu}$ represent the prototype keys in the memory. We generate the clustering (5.2) using the standard Expectation-Maximization algorithm.

The GMM allows us to compute a soft cluster assignment by evaluating the posterior probability of the latent assignment variable z. Using Bayes rule, the probability of a key value $\mathbf{k}$ to be assigned to the jth prototype is derived as,

$$p(z=j|\mathbf{k}) = \frac{p(\mathbf{k}|z=j)p(z=j)}{\sum_{l=1}^{N} p(\mathbf{k}|z=l)p(z=l)} = \frac{\exp\left(-\frac{1}{2\sigma^2}\|\mathbf{k}-\mathbf{k}_j^{\mu}\|^2\right)}{\sum_{l=1}^{N}\exp\left(-\frac{1}{2\sigma^2}\|\mathbf{k}-\mathbf{k}_l^{\mu}\|^2\right)}\,. \tag{5.3}$$

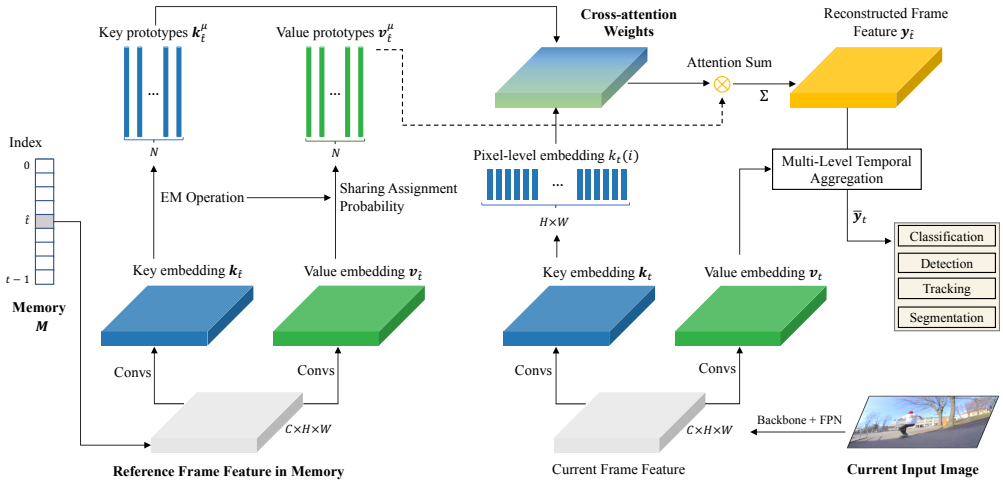
The resulting cluster assignment can thus be written as a SoftMax operation, where the corresponding logits are provided by the negative cluster distance $\|\mathbf{k}-\mathbf{k}_j^{\mu}\|^2$ scaled with a temperature of $2\sigma^2$.

Since the clustering is performed in the key space of the memory, we next retrieve the corresponding value prototypes. To this end, we employ the key cluster assignment probabilities in (5.3) to compute the values for each memory prototype,

$$\mathbf{v}_j^{\mu} = \sum_{l=1}^{N} p(z=j|\mathbf{k}_l^{M})\mathbf{v}_l^{M}\,. \tag{5.4}$$

For attending to our clustered memory, we first predict the key encodings $\mathbf{k}_i^{Q}$ of the query image. We then read from the clustered memory by computing the average over the

Figure 5.2: Overview of our frame-level prototypical cross-attention.



For a frame $\hat{t}$ in the memory we first perform GMM-based clustering to achieve the key $\mathbf{k}_{\hat{t}j}^{\mu}$ and value $\mathbf{v}_{\hat{t}j}^{\mu}$ prototypes. Given the key encoding $\mathbf{k}_t$ of the current frame, we attend to the prototypes to generate the reconstructed feature $\mathbf{y}_{\hat{t}}$, which are then aggregated temporally and fused with the current value encoding $\mathbf{v}_t$.

value prototypes $\mathbf{v}_j^{\mu}$, weighted with the cluster assignment probabilities,

$$\mathbf{y}_i = \sum_{j=1}^{N} p(z=j|\mathbf{k}_i^Q)\mathbf{v}_j^{\mu} = \frac{1}{Z_i}\sum_{j=1}^{N} \exp\left(-\frac{1}{2\sigma^2}\|\mathbf{k}_i^Q - \mathbf{k}_j^{\mu}\|^2\right)\mathbf{v}_j^{\mu}. \tag{5.5}$$

The final attention operation has much similarity with the original dot-product cross attention (5.1). Note that the key-query similarity in our approach is measured by Euclidian distance instead of a dot-product. Importantly, our formulation (5.5) attends to a reduced set of N prototypes, while the original attention (5.1) requires attending to the full spatio-temporal memory of size $H \times W \times T$.

5.2.3 Prototypical Cross-attention Network

Here, we propose the Prototypical Cross-attention Network (PCAN) for MOTS by integrating our prototypical cross-attention module into both the frame-level and instance-level. The former aims to align and aggregate temporal frame features stored in memory, while the latter is for propagating the instance appearance features over time and produce instance cross-attention maps to help segmentation. Besides, we also design a prototypical

instance appearance module to represent each video tracklet with contrastive mixture foreground and background prototypes.

Frame-level Prototypical Cross-attention In Figure 5.2, prototypical cross-attention first produces prototypes by fitting a Gaussian mixtures model (5.2) to the feature in the memory. To provide further flexibility when dynamically updating the memory $\mathbf{M}$, we first perform frame-wise clustering for each reference frame feature at index $\hat{t}$ to compute the N key prototypes $\{\mathbf{k}^{\mu}_{\hat{t}i}\}_{j=1}^{N}$, and retrieve the corresponding value embeddings $\{\mathbf{v}^{\mu}_{\hat{t}j}\}_{j=1}^{N}$ using (5.4) for each memory frame $\hat{t}$ independently. The key and value features are predicted using two parallel convolutional layers.

Frame-wise prototypical memory attention Given the query key encoding $\mathbf{k}^{Q}_{ti}$ of the current frame t, we perform prototypical cross-attention to each memory frame $\hat{t}$ independently using our formulation (5.3) as,

$$\mathbf{y}_{\hat{t}i} = \frac{1}{Z_{\hat{t}i}} \sum_{j=1}^{N} \exp\left(-\frac{1}{2\sigma^2}\|\mathbf{k}_{ti} - \mathbf{k}^{\mu}_{\hat{t}j}\|^2\right) \mathbf{v}^{\mu}_{\hat{t}j}\,, \qquad Z_{\hat{t}i} = \sum_{l=1}^{N} \exp\left(-\frac{1}{2\sigma^2}\|\mathbf{k}^{Q}_{ti} - \mathbf{k}^{\mu}_{\hat{t}l}\|^2\right)\,. \quad (5.6)$$

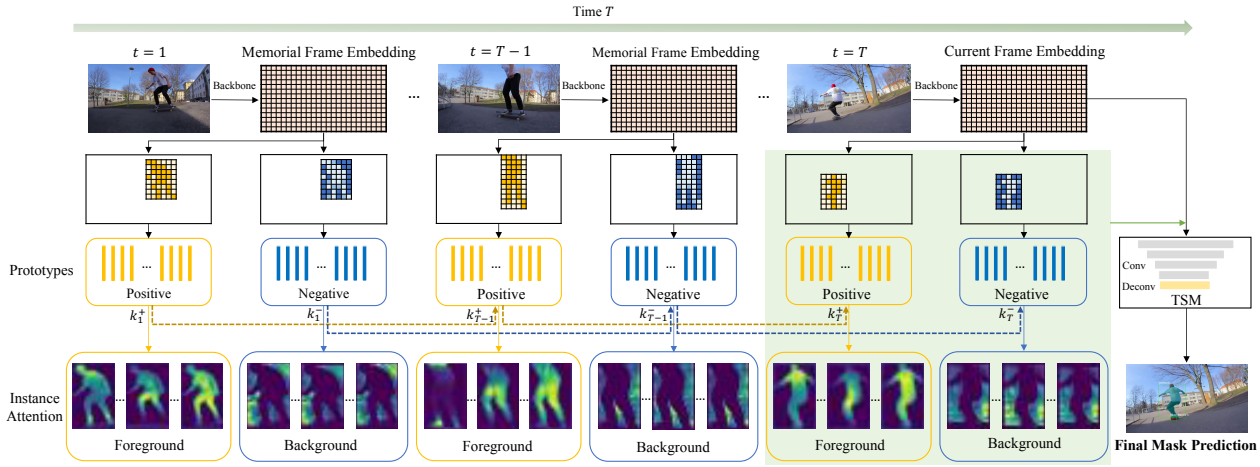
Note that the index i refers to a spatial coordinate in the current frame. The resulting feature map $\mathbf{y}_{\hat{t}}$ can intuitively be seen as a projection of features from frame $\hat{t}$ to the current frame. This projection essentially aligns the condensed feature information in frame $\hat{t}$ with the current frame.

Temporal feature aggregation Since frame-wise attention does not fuse temporal information, we perform a temporal aggregation. The temporal information $\mathbf{y}_{\hat{t}}$ in (5.6) from different frames $\hat{t}$ are fused as a linear combination, weighted by the feature similarity with the current frame. Specifically, the temporally aggregated representation is obtained as

$$\bar{\mathbf{y}}_{ti} = \sum_{\hat{t}=1}^{t} w_{\hat{t}i}\mathbf{y}_{\hat{t}i}\,, \qquad w_{\hat{t}i} = \frac{\exp(\mathbf{y}_{ti}\cdot\mathbf{y}_{\hat{t}i})}{\sum_{s=1}^{t}\exp(\mathbf{y}_{ti}\cdot\mathbf{y}_{si})}\,. \quad (5.7)$$

Note that $\hat{t} = t$ in the sum refers to the value embedding $\mathbf{y}_{ti} = \mathbf{v}^{Q}_{ti}$ extracted from the current frame. The contribution of each frame $\hat{t}$ is thus weighted by the similarity to

Figure 5.3: Our instance-level prototypical attention with foreground and background prototypes and temporal propagation.



The foreground/background attention maps from (bottom) demonstrate the localized and discriminative appearance representation. Temporal Segmentation Module (TSM) takes the current frame, initial mask, and instance attention maps as input and generates the final mask.

this current frame prediction using the attention weights $w_{\hat{t}i}$. This strategy ensures that incorrect or dissimilar regions are suppressed when computing the final aggregated feature embedding $\bar{\mathbf{y}}_t$. To handle object with large-scale variation and produce more fine-grained instance mask prediction, we further extend temporal aggregation to multi-level using different levels of the extracted FPN features, as detailed in the supplementary material.

Instance-level Prototypical Cross-attention In additional to the condensed frame-level representation, for more accurate segmentation results, we further encode each tracked object with compact and robust appearance prototypes. To further empower our proposed attention mechanism, we utilize the initially detected object mask to identify each foreground instance. We then separately model the extracted foreground and background features using a GMM (5.2). We denote the resulting foreground prototypes as $\mathbf{k}_{tj}^+$ and background prototypes as $\mathbf{k}_{tj}^-$. The former thus focuses on the appearance of the specific object, creating a rich and dynamic appearance model. When employed in our prototypical cross-attention framework (Section 5.2.2), it provides fine-grained attention from localized prototypes that naturally learn to focus specific parts of views of the object, as visualized in Fig. 5.3. Furthermore, the background prototypes $\mathbf{k}_{tj}^-$ capture valuable information about

the background appearance, which can greatly alleviate the segmentation process. For each object instance we attend to the foreground and background prototypes separately using (5.3). The results are concatenated together with the initial mask detection to the Temporal Segmentation Head (TSM) for final prediction, as illustrated in Figure 5.3.

Tracklet feature propagation and updating To effectively model the object appearance change and preserve the most relevant information, we design a recurrent instance appearance updating scheme. From the first video frame where object appears, the accumulated prototypes $\bar{\mathbf{k}}^{+}_{tj}$, $\bar{\mathbf{k}}^{-}_{tj}$ for the instance are propagated to the subsequent frames and updated with new appearance prototypes $\mathbf{k}^{+}_{tj}$, $\mathbf{k}^{-}_{tj}$ using an update rate λ as,

$$\bar{\mathbf{k}}^{+}_{tj} = (1-\lambda)\bar{\mathbf{k}}^{+}_{t-1,j} + \lambda \mathbf{k}^{+}_{tj}, \qquad \bar{\mathbf{k}}^{-}_{tj} = (1-\lambda)\bar{\mathbf{k}}^{-}_{t-1,j} + \lambda \mathbf{k}^{-}_{tj}. \tag{5.8}$$

Figure 5.3 also reveals the consistency of the attended region of a specific prototype j.

5.3 Experiments

Here, we present comprehensive evaluation and analysis of our approach. Experiments are performed on two large scale datasets, namely YouTube-VIS [133] and BDD100K [137].

5.3.1 Experiment setup

Youtube-VIS YouTube-VIS [133] dataset contains 2,883 high quality videos with 131k annotated object instances belonging to 40 diverse categories. The task is to simultaneously classifying, segment and track object instances belonging to these categories. The evaluation metrics for this task are an adaptation of the Average Precision (AP) and Average Recall (AR) of image instance segmentation.

BDD100K We also evaluate on the large-scale tracking and segmentation dataset of BDD100K [137], which is a challenging self-driving dataset with 154 videos (30,817 images) for training, 32 videos (6,475 images) for validation, and 37 videos (7,484 images) for testing. The dataset provides 8 annotated categories for evaluation, where the images in the tracking set are annotated per 5 FPS with 30 FPS frame rate. We adopt the well-established MOTS metrics [114] to our task.

Table 5.1: Comparison with state-of-the-art on the YouTube-VIS validation set. Results are reported in terms of mask accuracy (AP) and recall (AR). Asterisks * denote concurrent works on arXiv.

Method	Backbone	Type	Online	AP	AP_{50}	AP_{75}	AR_1	AR_{10}
VisTr* [121]	ResNet-50	Transformer	×	34.4	55.7	36.5	33.5	38.9
OSMN [134]	ResNet-50	Two-stage	✓	23.4	36.5	25.7	28.9	31.1
FEELVOS [113]	ResNet-50	Two-stage	✓	26.9	42.0	29.7	29.9	33.4
DeepSORT [124]	ResNet-50	Two-stage	✓	26.1	42.9	26.1	27.8	31.3
MaskTrack R-CNN [133]	ResNet-50	Two-stage	✓	30.3	51.1	32.6	31.0	35.5
STEm-Seg [4]	ResNet-50	One-stage	×	30.6	50.7	33.5	31.6	37.1
SipMask [14]	ResNet-50	One-stage	✓	32.5	53.0	33.3	33.5	38.9
STMask* [64]	ResNet-50	One-stage	✓	33.5	52.1	36.9	31.1	39.2
SG-Net* [75]	ResNet-50	One-stage	✓	34.8	**56.1**	36.8	35.8	40.8
PCAN (Ours)	ResNet-50	One-stage	✓	**36.1**	54.9	**39.4**	**36.3**	**41.6**
STMask* [64]	ResNet-101	One-stage	✓	36.3	55.2	39.9	33.7	42.0
SG-Net* [75]	ResNet-101	One-stage	✓	36.3	57.1	39.6	35.9	43.0
PCAN (Ours)	ResNet-101	One-stage	✓	**37.6**	**57.2**	**41.3**	**37.2**	**43.9**

5.3.2 State-of-the-Art Comparison

We compare our approach with the state-of-the-art methods on the aforementioned large-scale MOTS/VIS benchmarks Youtube-VIS and BDD100K, where PCAN outperforms all existing methods without bells and whistles, and shows efficacy to both one-stage and two-stage segmentation frameworks. We follow the official metrics of each benchmark to evaluate our model.

Youtube-VIS The results of Youtube-VIS benchmark is in Table 5.1, where PCAN achieves the best mask AP of 36.1% using ResNet-50 and 37.6% using ResNet-101 respectively, while being an online method. Our approach consistently surpasses most recent SOTA methods, including STMask [64] and SG-Net [75] by a significant margin. These methods only conduct temporal modeling between two adjacent frames for feature correlation. Compared to our baseline SipMask [14], a single-image based segmentation with object centerness association, PCAN improves the mask AP from 32.5% to 36.1%, which shows the effectiveness of long-term temporal modeling in helping object tracking and segmentation.

BDD100K Table 5.2 shows our results on BDD100K tracking and segmentation benchmark, where PCAN outperforms the strong baseline methods MaskTrackRCNN [133] and QDTrack-mots [90]. Our approach achieves a large advantage in mMOTSA, with over

Table 5.2: State-of-the-art comparison on the BDD100K segmentation tracking validation set. I: ImageNet. C: COCO. S: Cityscapes. B: BDD100K.

Method	Pretrained	Online	mMOTSA↑	mMOTSP↑	mIDF↑	ID sw.↓	mAP↑
SortIoU	I, C, S	✓	10.3	59.9	21.8	15951	22.2
MaskTrackRCNN [113]	I, C, S	✓	12.3	59.9	26.2	9116	22.0
STEm-Seg [4]	I, C, S	×	12.2	58.2	25.4	8732	21.8
QDTrack-mots [90]	I, C, S	✓	22.5	59.6	40.8	1340	22.4
QDTrack-mots-fix [90]	I, B	✓	23.5	66.3	44.5	973	25.5
PCAN (Ours)	I, B	✓	**27.4**	**66.7**	**45.1**	**876**	**26.6**

3 points gain and around 10% ID switches decrease. MOTSA measures segmentation as well as tracking quality, while ID Switches can measure the performance of identity consistency. The significant advancements demonstrate that our method with prototypical cross-attention enables more accurate pixel-wise object tracking by effectively exploiting temporal information.

5.3.3 Ablation study and analysis

We conduct detailed ablation studies on Youtube-VIS validation set, where we investigate the effect of our proposed prototypical cross-attention components for MOTS during training and testing.

Effect of frame-level prototypical cross-attention module To study the importance of temporal information amount, we conduct an ablation study on models with different input temporal window lengths in Table 5.3. A temporal length of 1 thus means that no prior temporal information guidance is used during video instance segmentation. By varying the frame length from 1 to 32, the mask AP increases from 32.5% to 35.4%, which reveals that richer temporal information with multiple views of a segmented object indeed brings more gain to model performance.

Effect of multi-layer temporal aggregation Since we perform temporal feature aggregation on the extracted FPN features, to help deal with objects with partial occlusion and large-scale variation, we also study the effect of using different levels of the extracted FPN features. In Table 5.4, we select the FPN feature map from P3-P5 layers for (excluding P6 and P7 due to impractical computation cost), and perform prototypical temporal aggregation on each FPN layer. We find that multi-layer information is also important to final

Table 5.3: Results of varying temporal memory length in our PCAN on YouTube-VIS.

Length	AP	AP_{50}	AP_{75}	AR_1	AR_{10}
1	32.5	53.0	33.3	33.5	38.9
2	33.7	53.8	35.3	33.9	39.5
4	33.9	**54.0**	36.8	34.1	40.0
8	34.2	53.7	37.6	34.4	40.3
16	34.6	53.7	38.3	35.4	40.5
32	**35.4**	53.8	**39.1**	**35.9**	**41.0**

Table 5.4: Effect of multi-layer prototypical feature fusion with tube length 4 on YouTube-VIS.

FPN Layer	AP	AP_{50}	AP_{75}	AR_1	AR_{10}
P3	30.8	51.7	32.0	32.6	37.0
P4	32.0	51.5	34.1	32.6	37.2
P5	32.9	52.1	35.9	33.2	38.6
P3-P4	33.1	52.3	35.6	33.6	38.5
P3-P5	**33.9**	**54.0**	**36.8**	**34.1**	**40.0**

Table 5.5: Comparison with non-local attention [117] and transformer [15] on YouTube-VIS.

Length	Prototypical Cross-attention			Non-local Attention			Transformer (Multi-Head Self-Attention)		
	AP	FLOPs(B)	Memory(M)	AP	FLOPs(B)	Memory(M)	AP	FLOPs(B)	Memory(M)
2	33.7	5.8	323	33.2	24.3	2497	24.6	103.8	5321
4	33.9	12.0	652	33.3	49.1	4763	25.8	387.2	9844
8	**34.2**	23.7	1419	33.6	99.6	9631	28.3	1413.3	18762

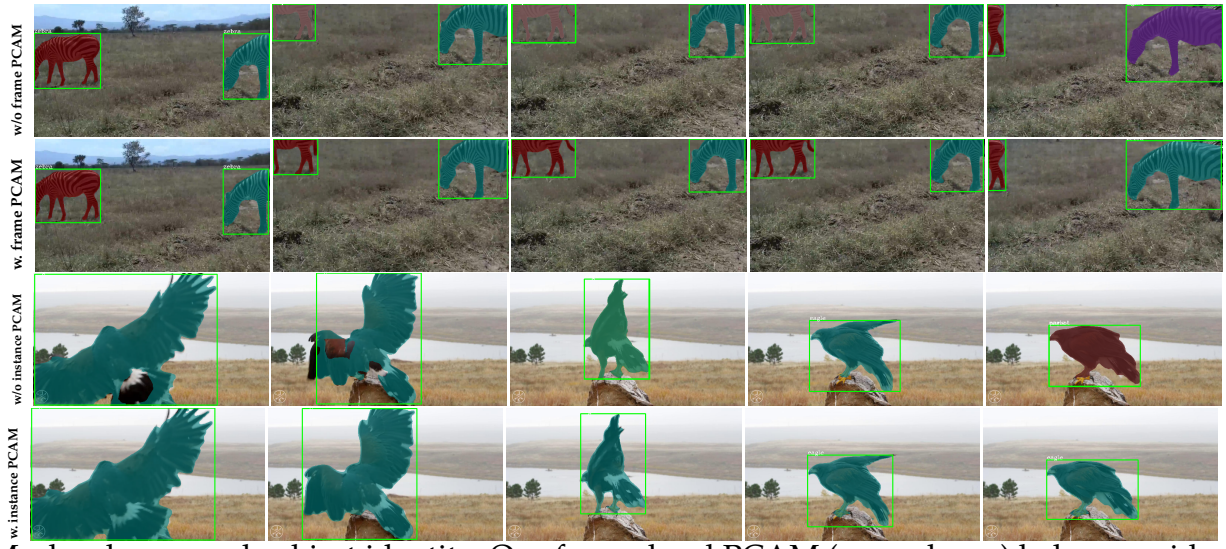
model performance.

Computation and memory efficiency In Table 5.5 we analyze different attention mechanisms. Compared to standard space-time memory reading using non-local attention [117, 88] or recent popular transformer [121, 15] with multi-head self-attention layer, the prototypical cross-attention with condensed prototypes not only enjoys high accuracy advantage, but also largely reduces the memory consumption and computation amount. For input tube length 8, the prototypical memory consumption is less than 10% of the transformer with negligible FLOPs computation due to the small number of representative prototypes in (5.5).

Effect of instance-level prototypical appearance module We analyze the instance-level prototypical cross-attention module, which represents each video tracklet using the contrastive prototypes. In Table 5.6, we study the influence of instance prototype number and the effect of foreground-background contrasting. Using both positive and negative prototypes improves AP from 32.5% to 33.9%. Compared to the single prototype representation, the GMM demonstrate a stronger appearance modeling ability. We further find that the performance saturates when the number is larger than 60. In the supplementary file we provide additional instance cross-attention maps visualization to highlight the various attended regions, which also reveal the implicit unsupervised temporal consistency for prototype focusing specific object region over time.

In Table 5.7, we investigate the effectiveness of instance prototype (including the both

Figure 5.4: Qualitative impact of our PCAM on YouTube-VIS.



Mask colors encode object identity. Our frame-level PCAM (second row) helps provide consistent detections and preserve identities compared to the baseline (first row). The instance-level PCAM (fourth row) provides more accurate masks, while further improving identity consistency compared to not employing our module (third row).

Table 5.6: Ablation study on instance-level prototypes on YouTube-VIS.

Pos. Proto. Number	Neg. Proto. Number	AP	AP_{50}
0	0	32.5	53.0
1	0	32.4	52.3
0	1	32.1	52.4
1	1	32.7	52.8
5	5	33.1	53.6
30	30	**33.9**	**54.1**
50	50	33.6	53.8

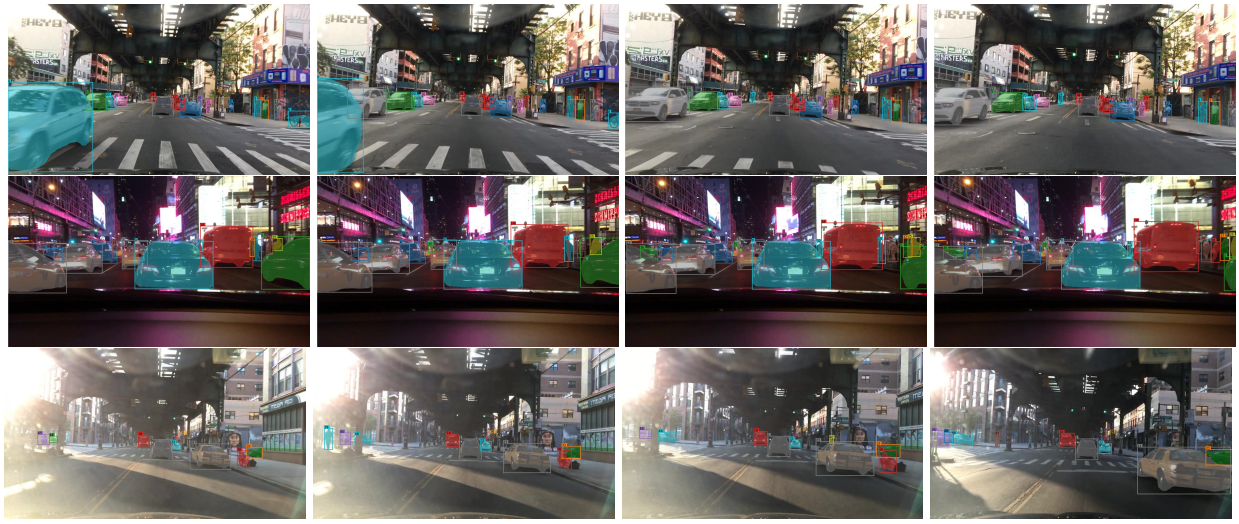
Table 5.7: Ablation on instance-level EM feature propagation on YouTube-VIS.

version	AP	AP_{50}
No instance prototype propagation	33.5	53.2
Using initial instance prototype	33.0	52.8
Update momentum = 0.2	**34.3**	**53.8**
Update momentum = 0.5	34.0	53.6

positive and negative ones) propagation in an online manner, and compared it with using the instance prototype in the initial frame or current frame. We find that updating object prototypes recurrently with a momentum of 0.2 improves video segmentation AP of 1.3%.

Qualitative analysis In Figure 5.4, we showcase qualitative ablation results of PCAN on Youtube-VIS. Compared to the baseline, we see that our model results in more consistent segmentation and better tracking using prototypical cross-attention module. We also provide visual results on BDD100K in Figure 5.5, where PCAN produces robust tracking and segmentation results even under large object appearance change (first row) or low illumination (second row). In the 3rd row, PCAN has limitations in handling missing detections (the person in the first frame) with limited appearance information under extreme lighting, and produce tracking errors in the second frame when visible parts of the same car is totally different across frame and with low appearance similarity.

Figure 5.5: Qualitative results of our method on BDD100K.



PCAN produces robust tracking and segmentation results under large motion and appearance changes (1st row) and heavy traffic in low-light conditions (2nd row). In the 3rd row, PCAN misses a detection (the person to the left in 1st frame), and produces tracking errors (2nd frame) when it covers totally different regions of the car with low appearance similarity. Zoom for better view.

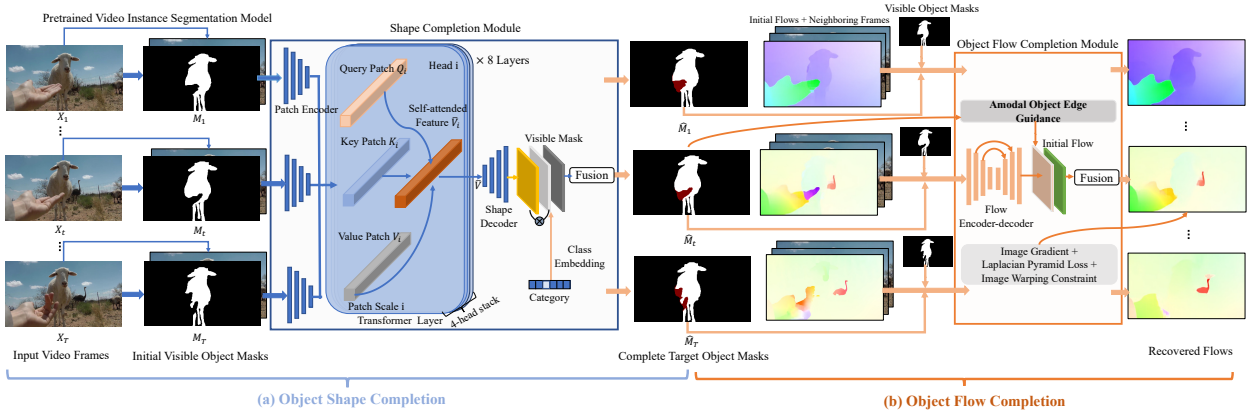


Figure 5.6: **(a)** Object shape completion, which associates transformed temporal patches and object semantics; **(b)** Object flow completion, which recovers complete object flow subject to the amodal object contours.

5.4 Application of Video Instance Segmentation: Video object inpainting

Given $_1^{\mathsf{T}} = \{X_1, X_2, ..., X_T\}$ as an input video sequence with frame length T, frame resolution $H \times W$, and $_1^{\mathsf{T}} = \{M_1, M_2, ..., M_T\}$ denotes the corresponding frame-wise binary masks for the visible regions of the target occluded object, we formulate the video object inpainting problem as self-supervised learning to infer complete object masks $\hat{}_1^{\mathsf{T}} = \{\hat{M}_1, \hat{M}_2, ..., \hat{M}_T\}$

and produce completed video frames $\hat{Y}_1^T = \{\hat{Y}_1, \hat{Y}_2, ..., \hat{Y}_T\}$ with realistic amodal object content.

Figures 5.6 and 5.7 together depict the whole pipeline of our proposed video object inpainting approach VOIN, which consists of the following the three stages: a) *object shape completion*: we compute the amodal object shapes based on its visible object content (Section 5.4.1); b) *object flow completion*: complete object flow is estimated with sharp motion boundary under the guidance of amodal object contour (Section 5.4.2); c) *flow-guided video object inpainting*: with the completed object and flow within its contour, motion trajectories are utilized to warp pertinent pixels to inpaint the corrupted frames. To generate highly plausible video content, we improve temporal shift module by making it occlusion-aware and use multi-class discriminator with spatio-temporal attention, instead of only using single-image completion techniques as in [33, 129] (Section 5.4.3).

5.4.1 Occlusion-Aware Shape Completion

Given an input video sequence, it is easy to obtain the modal masks for the target occluded object using the existing video object/instance segmentation [111, 99, 133]. However, learning the full completion video masks of occluded instances is very difficult due to diverse object shapes and occlusion patterns.

To address this problem, we propose a novel object shape completion module (see Figure 5.6(a)), which recovers amodal segmentation masks for the occluded video object in a self-supervised training scheme. Our shape module with 8 transformer layers is inspired from the recent spatio-temporal transformers in video understanding [121, 136, 2, 140, 52] for capturing long-range spatio-temporal coherence. Each transformer layer has a multi-head structure to deal with the multi-scaled embeded image patches transformed from the whole input video sequence, followed by the scaled dot-product attention mechanism [110], which models temporal shape associations of the same occluded object among both the neighboring and distant encoded spatial feature patches in parallel.

Specifically, suppose there are k heads in the transformer layer, then we compute self-attended feature $\bar{V}$:

$$\bar{V} = \text{Multihead}(Q, K, V) = f_c([\bar{V}_i]_{i=1}^k), \tag{5.9}$$

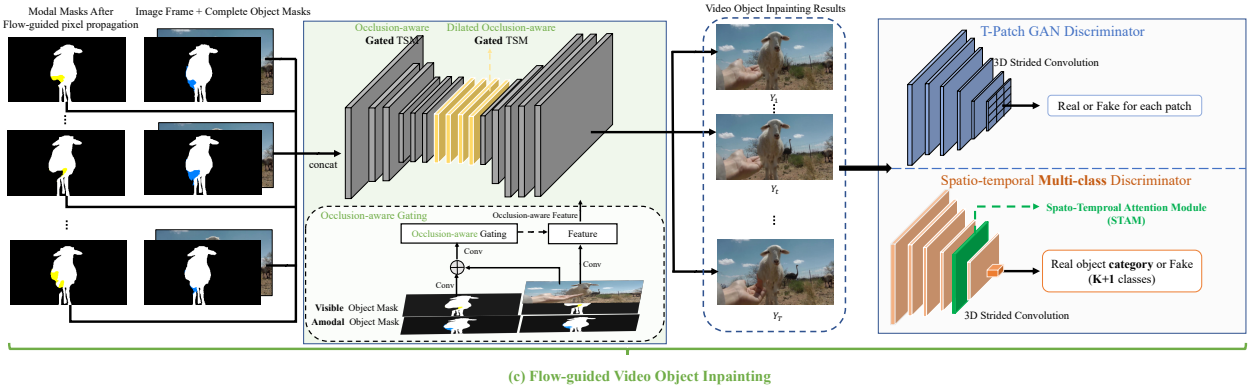


Figure 5.7: **(c)** Flow-guided video object inpainting with occlusion-aware gating.

$$\bar{V}_i = \text{SelfAtt}(Q_i, K_i, V_i) = \frac{softmax(Q_i K_i^T)}{\sqrt{d_k}} V_i, \tag{5.10}$$

where $\bar{V}_i$ is the self-attended feature on the i-th head, Q_i, K_i, V_i are respectively the query, key and value embedding matrices for these spatial feature patches with total size $T \times H/r_1 \times W/r_2$, frame resolution is $H \times W$, r_1 and r_2 are patch size, d_k is the dimension for query patch features, and f_c are convolutional layers merging the outputs from the k heads. $\bar{V}$ is then passed through the frame-level shape decoder for up-sampling, which is combined with the class embedding feature multiplied by the visible object masks for incorporating semantics and spatial shape prior. Finally, the merged features are refined by the fusion convolution layers to produce the amodal object shape masks.

5.4.2 Occlusion-Aware Flow Completion

Our flow completion algorithm first computes the initial optical flows and then focuses on recovering the flow fields within the completed occluded object region subject to the amodal object contour.

In Figure 5.6(b), the flow generator adopts Unet [100] encoder-decoder structure with skip connections from encoders to the corresponding layers in the decoder, which takes neighboring image frames, initial flow, visible and amodal object masks as inputs x. Instead of computing the recovered flow directly, we formulate flow completion as a residual learning problem [40], where $\phi(x) := O - \bar{O}$, O is the desired flow output, $\bar{O}$ is the initial corrupted flow, and $\phi(x)$ represents the flow residue learned by the encoder-decoder generator. This formulation effectively reduces the training difficulty for dense pixel

regression.

To recover accurate object flow with sharp motion boundary, especially for the occluded region, we incorporate amodal object contour to guide the flow prediction process by enforcing flow smoothness within the complete object region, as flow fields are typically piecewise smooth, where gradients are small except along the distinct object motion boundaries. To effectively regularize the flow completion network, instead of simply adopting L1 regression loss between predictions and ground-truth flows as in [129], we additionally utilize the image gradient loss, Laplacian Pyramid loss [8] and image warping loss of hallucinated content for joint optimization, which further promote the precision of flow prediction.

5.4.3 Flow-Guided Video Object Inpainting

The resulting completed object flow above is employed to build dense pixels correspondences across frames, which is essential since previously occluded regions in the current frame may be disoccluded and become visible in a distant frame, especially for objects in slow motion, which is very difficult for a generative model to handle such long-range temporal dependencies.

We follow [129, 33] using forward-backward cycle consistency threshold (5 pixels) to filter out unreliable flow estimations and warp pixels bidirectionally to fill the missing regions based on the valid flow. The main difference is that we only warp pixels within the foreground object region, which guarantees the occluded areas are not filled by any background colors while reducing the overall computational burden. Figure 5.7 highlights the regions in yellow within a completed mask tracked by optical flow.

To fill in remaining pixels after the above propagation (i.e., the blue regions in Figure 5.7), which can be in large numbers for previously heavily occluded objects, we propose to train an occlusion-aware gated generator to inpaint the occluded regions of videos objects, where the gating feature is learned under the guidance of both amodal object masks and occlusion masks, and two spatio-temporal discriminators with multi-class adversarial losses. As usual, the discriminators will be discarded during testing.

Occlusion-Aware TSM We adopt the residual Temporal Shift Module (TSM) [71, 17] as our building blocks here, which shift partial channels along the temporal dimension to perform joint spatio-temporal feature learning, and achieve the performance of 3D convolution at 2D CNN's complexity.

To make TSM occlusion-aware and learn a dynamic feature selection mechanism for different spatial locations, we guide the gated feature learning process [139] with both amodal object masks and occlusion masks, thus making our improved or occlusion-aware model capable of reasoning the occluded regions from other visible parts along the spatio-temporal dimension as illustrated in Figure 5.8.

Specifically, the generator in Figure 5.7 has the encoder-decoder structure with Occlusion-aware TSM replacing all vanilla convolutions layers, which have a larger temporal receptive field n than original setting [71] and can be formulated as

$$Gate^{x,y}_{occ}(t) = \sum_{x,y} W_g \cdot I^{x,y}_t + \bar{f}^{x,y}_t(\hat{M}^{occ}_t, \hat{M}_t), \tag{5.11}$$

$$S^{x,y}_t = \sum_{x,y} W_f \cdot TSM(I^{x,y}_{t-n}, \cdots, I^{x,y}_t, \cdots, I^{x,y}_{t+n}), \tag{5.12}$$

$$Out^{x,y}_t = \sigma(Gate^{x,y}_{occ}(t)) \odot \phi(S^{x,y}(t)), \tag{5.13}$$

where $Gate^{x,y}_{occ}$ serves as a soft attention map (for identifying occluded/visible/background areas) on the feature volume $S^{x,y}_t$ output by the TSM, $\bar{f}$ are convolutional layers fusing the occlusion mask $\hat{M}^{occ}_t$ and complete object mask $\hat{M}_t$, W_g and W_f are respectively the kernel weights for gating convolution and shift module, and $I^{x,y}_t$ and $Out^{x,y}_t$ respectively denote the input and final output activation at (t, x, y), σ is sigmoid function, and ϕ is the ReLU function.

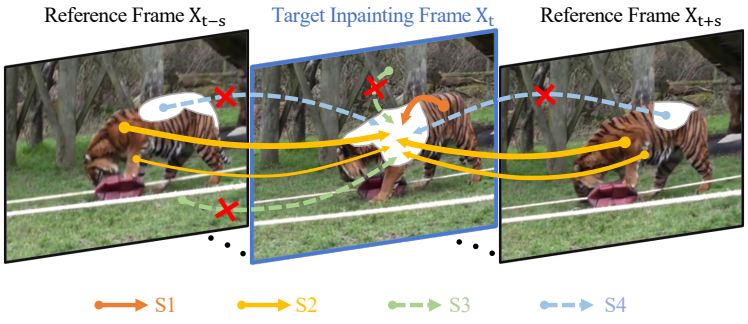
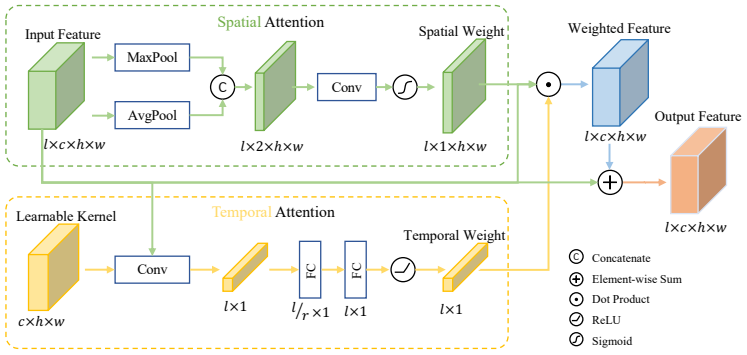


Figure 5.8: Illustration of our occlusion-aware gating scheme.

Multi-Class Discriminator with STAM To make object inpainting results more realistic, we adopt two discriminators to simultaneously regularize the GAN training process. The first discriminator considers video perceptual quality and temporal consistency, while the second considers object semantics based on both the global and local features since the occlusion holes may appear anywhere in the video with irregular shape.

We adopt T-PatchGAN as the first discriminator [16, 140]. For the second discriminator, we propose a new spatio-temporal attention-based multi-class discriminator, which classifies the category of the inpainting object into one of the K real classes and an additional fake class, by picking the most relevant frames from an input video while focusing on their discriminative spatial regions. Figure 5.7 shows that the multi-class discriminator is composed of six 3D convolution layers (kernel size 3×5×5) with a spatio-temporal attention module (STAM) embedded above the 4th layer. This STAM design is inspired by [41, 125]. Figure 5.9 shows the parallel branches for spatial and temporal attention.

Figure 5.9: The design of spatio-temporal attention module (STAM).



Two parallel branches are respectively used for computing spatial and temporal attention weights. The weighted feature forms a residual connection with the original input for final output.

5.4.4 Experiments on Video Object Inpainting

Occlusion Inpainting Setting Since this paper focuses on inpainting occluded regions of video objects, we propose a new inpainting setting which is different from previous ones for undesired object removal or arbitrary mask region inpainting. The fill-up regions are restricted to occluded regions of the target object, which can be given by user or our object shape completion module using the visible object content. This setting is in line with

real-world applications such as video scene de-occlusion.

YouTube-VOI Benchmark. To support training and evaluation of our new video object inpainting task, we use the YouTube-VOS [128] dataset as our video source to construct our large-scale YouTube-VOI benchmark, which contains 5,305 videos (4,774 for training and 531 for evaluation) with resolution higher than 640×480, a 65-category label set including common objects such as people, animals and vehicles, and over 2 million occluded and visible masks.

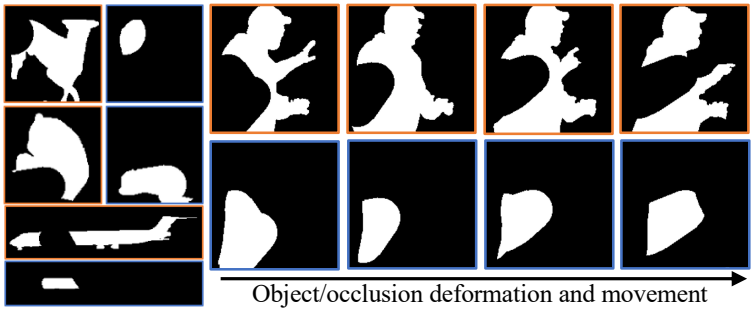


Figure 5.10: Sample visible masks (orange boxes) and occluded masks (blue boxes) for moving video objects generated by our algorithm with different object categories and occlusion patterns.

Our YouTube-VOI is a very challenging dataset for video object inpainting, which is representative of complex real-world scenarios in high diversity, including different realistic occlusions caused by vehicles, animals and human activities. Although amodal object masks are not annotated in Youtube-VOS, we show that the proposed VOIN model can still perform amodal object shape completion for video with only modal annotations, by incorporating the self-supervised training scheme in [141] to learn object shape associations between frames, utilizing the annotated object semantics, and training on huge number of occlusions masks in various degrees and patterns of occlusion.

Comparison with State-of-the-arts Using the Youtube-VOI benchmark, we compare VOIN with the most recent and relevant state-of-the-art video inpainting approaches and adapt their original input by additionally concatenating visible object masks: 1) DFVI [129], which fills corrupted regions using pixel propagation based on the predicted complete flow; 2) LGTSM [17], where learnable shift module is designed for inpainting generator and T-PatchGAN discriminator [16] is utilized; 3) FGVC [33], which uses a flow completion module guided by Canny edge extraction [13] and connection [86]; 4) STTN [140], which

completes missing regions using multi-scale patch-based attention module. Note that both DFVI and FGVC conduct flow completion for inpainting, and fill the remaining unseen video regions using only image inpainting method [138].

Table 5.8: Quantitative comparison on flow completion (EPE) and inpainting quality (PSNR, SSIM and LPIPS) on Youtube-VOI benchmark with the state-of-the-art methods.

Model	Use flow?	EPE ↓	PSNR ↑	SSIM ↑	LPIPS ↓
DFVI [129]	✓	4.79	44.91	0.952	0.099
LGTSM [17]		-	45.19	0.979	0.024
FGVC [33]	✓	3.69	43.90	0.924	0.065
STTN [140]		-	45.97	0.986	0.020
Ours		-	46.33	0.989	0.013
Ours	✓	**3.11**	**48.99**	**0.994**	**0.008**

Quantitative Results Comparison. Table 5.8 also reports quantitative comparison on inpainting quality under complex occlusion scenarios on the Youtube-VOI test set. Compared to existing models, our VOIN substantially improves video reconstruction quality with both per-pixel and overall perceptual measurements, where our model outperforms the most recent STTN [140] and FGVC [33] by a large margin, especially in terms of PSNR and LPIPS. Our improved results show the effectiveness of our proposed occlusion-aware gating scheme and the multi-class discriminator with STAM. On the other hand, the results produced by DFVI and FGVC are not on par with ours, especially for inpainting foreground object with large occlusion due to their incorrect flow completion and the lack of temporal consistency for generating video content (limited in using single image-based inpainting model DeepFill [138]).

Qualitative Results Comparison. Figure 5.12 shows sample video completion results for inpainting occluded video objects, where our occlusion-aware VOIN produces temporally coherent and visually plausible content than previous methods [17, 33, 140]. Figure 5.11 shows its application in video scene de-occlusion. Refer to the supplementary video results for extensive qualitative comparison.

Figure 5.11: Video scene de-occlusion results comparison

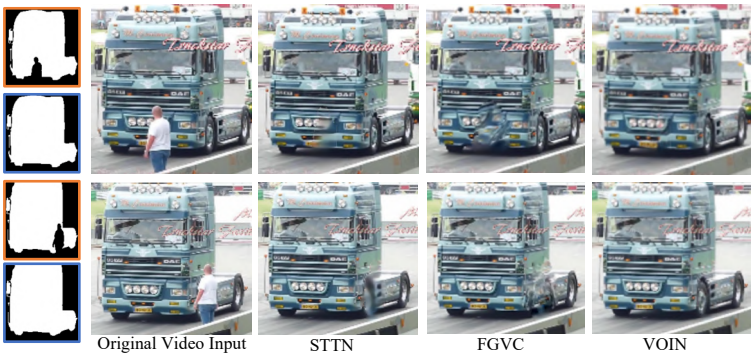
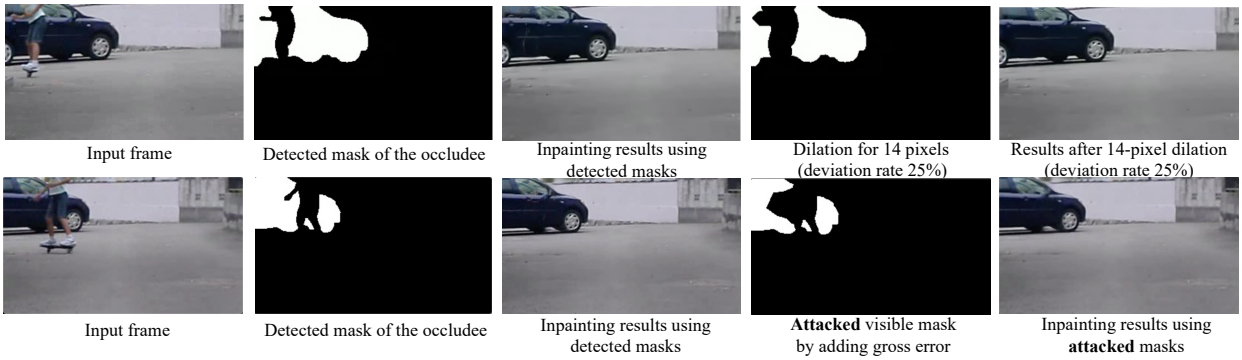


Figure 5.12: Qualitative comparison with state-of-the-art video inpainting methods.

The results are reported on Youtube-VOI. In particular, FGVC also adopts completed flow to guide the video inpainting process, but their results suffer from unnatural pixel transition due to the incorrect flow estimation. Zoom in for better view.

Figure 5.13: Sample visual results of VOIN given inaccurate mask segmentation (dilation and gross segmentation errors), which show the robustness of VOIN.



CHAPTER 6

CONCLUSION

This survey gives a comprehensive review of deep learning-based image/video instance segmentation methods, which includes its background, issues, techniques, evaluation datasets and related works up to the state of the art. In this section, we also discuss their corresponding limitations and some future research directions on instance segmentation.

6.1 Limitation

We first introduce how to improve the accuracy of image instance segmentation with occlusion handling and detail a framework called bilayer convolutional network. Although this method is able to accurately decouple the occluder and occludee in the same RoI, the requirement for detecting occluders also restricts them being contained in the predefined training classes and limits its application.

Then, we study how to promote the generalizability of image instance segmentation on novel object categories. A deep instance segmentation method with commonality-parsing is presented. Despite of its captured class-agnostic features with great generalization ability, we still observe an obvious improvement space to the oracle upper-bound, and some failure cases in open-world settings.

Besides, we introduce how to adapt image instance segmentation to video instance segmentation by exploiting rich spatio-temporal information. A deep video instance segmentation framework with both effectiveness and efficiency is described in detail. In spite of its good performance on several well-known benchmarks, it is still limited in processing some videos with crowded scenes and complex/fast background motions, which becomes the bottleneck in the practical application.

Finally, we further present a new application based on video instance segmentation: occlusion-aware video object inpainting. We design the **VOIN** (Video Object Inpainting

Network), a unified multi-task framework for joint video object mask completion and object appearance recovery. A potential challenge for VOIN is the applicable scope on the inpainted video objects. Presently Youtube-VOI contains 65 categories, including common objects (e.g., people, animals and vehicles), which is quite general as the first attempt on this new task. But more classes could readily be extended, by considering shape/appearance commonality across categories.

6.2 Future Work

Recently, transformer-based instance segmentation have become a hot topic. With the advancements in computing power and the adaptation of the models from NLP tasks, such as DETR [15], a new transformer encoder-decoder architecture is proposed for object detection with high accuracy. It could be further utilized in instance segmentation for simplifying the existing segmentation paradigm by removing the need for many hand-designed components. How to produce high-quality segmentation prediction results with practical memory/computation burden and real-time speed with transformer paradigm is a new challenge.

In addition, current state-of-the-art instance segmentation method is designed in the closed-world setting while real-world applications require segmenting all objects that appear in the videos, even unseen objects. A fully class-agnostic open-world object segmentation method without semi-supervision also deserves further exploration. Also, as a fundamental computer vision technique, instance segmentation could be used in assisting many downstream applications besides inpainting, such as object pose/shape reconstruction and image captioning generation.

REFERENCES

[1] Jiwoon Ahn, Sunghyun Cho, and Suha Kwak. Weakly supervised learning of instance segmentation with inter-pixel relations. In *CVPR*, 2019.

[2] Emre Aksan, Peng Cao, Manuel Kaufmann, and Otmar Hilliges. Attention, please: A spatio-temporal transformer for 3d human motion prediction. *arXiv preprint arXiv:2004.08692*, 2020.

[3] Anurag Arnab and Philip HS Torr. Pixelwise instance segmentation with a dynamically instantiated network. In *CVPR*, 2017.

[4] Ali Athar, Sabarinath Mahadevan, Aljoša Ošep, Laura Leal-Taixé, and Bastian Leibe. Stem-seg: Spatio-temporal embeddings for instance segmentation in videos. In *ECCV*, 2020.

[5] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[6] Min Bai and Raquel Urtasun. Deep watershed transform for instance segmentation. In *CVPR*, 2017.

[7] Gedas Bertasius and Lorenzo Torresani. Classifying, segmenting, and tracking object instances in video with mask propagation. In *CVPR*, 2020.

[8] Piotr Bojanowski, Armand Joulin, David Lopez-Paz, and Arthur Szlam. Optimizing the latent space of generative networks. *arXiv preprint arXiv:1707.05776*, 2017.

[9] Daniel Bolya, Chong Zhou, Fanyi Xiao, and Yong Jae Lee. Yolact: real-time instance segmentation. In *ICCV*, 2019.

[10] Daniel Bolya, Chong Zhou, Fanyi Xiao, and Yong Jae Lee. Yolact: Real-time instance segmentation. In *ICCV*, 2019.

[11] 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.

[12] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *CVPR*, 2018.

[13] John Canny. A computational approach to edge detection. *TPAMI*, (6):679–698, 1986.

[14] Jiale Cao, Rao Muhammad Anwer, Hisham Cholakkal, Fahad Shahbaz Khan, Yanwei Pang, and Ling Shao. Sipmask: Spatial information preservation for fast image and video instance segmentation. In *ECCV*, 2020.

[15] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020.

[16] Ya-Liang Chang, Zhe Yu Liu, Kuan-Ying Lee, and Winston Hsu. Free-form video inpainting with 3d gated convolution and temporal patchgan. In *ICCV*, 2019.

[17] Ya-Liang Chang, Zhe Yu Liu, Kuan-Ying Lee, and Winston Hsu. Learnable gated temporal shift module for deep video inpainting". In *BMVC*, 2019.

[18] Hao Chen, Kunyang Sun, Zhi Tian, Chunhua Shen, Yongming Huang, and Youliang Yan. BlendMask: Top-down meets bottom-up for instance segmentation. In *CVPR*, 2020.

[19] Kai Chen, Jiangmiao Pang, Jiaqi Wang, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jianping Shi, Wanli Ouyang, et al. Hybrid task cascade for instance segmentation. In *CVPR*, 2019.

[20] Liang-Chieh Chen, Alexander Hermans, George Papandreou, Florian Schroff, Peng Wang, and Hartwig Adam. Masklab: Instance segmentation by refining object detection with semantic and direction features. In *CVPR*, 2018.

[21] Xianjie Chen and Alan L Yuille. Parsing occluded people by flexible compositions. In *CVPR*, 2015.

[22] Xinlei Chen, Ross Girshick, Kaiming He, and Piotr Dollár. Tensormask: A foundation for dense object segmentation. In *ICCV*, 2019.

[23] Yi-Ting Chen, Xiaokai Liu, and Ming-Hsuan Yang. Multi-instance object segmentation with occlusion handling. In *CVPR*, 2015.

[24] Yunpeng Chen, Yannis Kalantidis, Jianshu Li, Shuicheng Yan, and Jiashi Feng. a^2-nets: Double attention networks. In *NeurIPS*, 2018.

[25] Yunpeng Chen, Marcus Rohrbach, Zhicheng Yan, Yan Shuicheng, Jiashi Feng, and Yannis Kalantidis. Graph-based global reasoning networks. In *CVPR*, 2019.

[26] Tianheng Cheng, Xinggang Wang, Lichao Huang, and Wenyu Liu. Boundary-preserving mask r-cnn. In *ECCV*, 2020.

[27] Jifeng Dai, Kaiming He, and Jian Sun. Instance-aware semantic segmentation via multi-task network cascades. In *CVPR*, 2016.

[28] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2020.

[29] Kiana Ehsani, Roozbeh Mottaghi, and Ali Farhadi. Segan: Segmenting and generating the invisible. In *CVPR*, 2018.

[30] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010.

[31] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *ICCV*, 2019.

[32] Patrick Follmann, Rebecca Kö Nig, Philipp Hä Rtinger, Michael Klostermann, and Tobias Bö Ttger. Learning to see the invisible: End-to-end trainable amodal instance segmentation. In *WACV*, 2019.

[33] Chen Gao, Ayush Saraf, Jia-Bin Huang, and Johannes Kopf. Flow-edge guided video completion. In *ECCV*, 2020.

[34] Tianshi Gao, Benjamin Packer, and Daphne Koller. A segmentation-aware object detection model with occlusion handling. In *CVPR*, 2011.

[35] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*, 2012.

[36] Golnaz Ghiasi, Yi Yang, Deva Ramanan, and Charless C Fowlkes. Parsing occluded people. In *CVPR*, 2014.

[37] Miguel Granados, James Tompkin, K Kim, Oliver Grau, Jan Kautz, and Christian Theobalt. How not to be seen—object removal from videos of crowded scenes. In *Computer Graphics Forum*, 2012.

[38] Bharath Hariharan, Pablo Arbeláez, Ross Girshick, and Jitendra Malik. Hypercolumns for object segmentation and fine-grained localization. In *CVPR*, 2015.

[39] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, 2017.

[40] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[41] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *CVPR*, 2018.

[42] Ping Hu, Fabian Caba, Oliver Wang, Zhe Lin, Stan Sclaroff, and Federico Perazzi. Temporally distributed networks for fast video semantic segmentation. In *CVPR*, 2020.

[43] Ronghang Hu, Piotr Dollár, Kaiming He, Trevor Darrell, and Ross Girshick. Learning to segment every thing. In *CVPR*, 2018.

[44] Yuan-Ting Hu, Hong-Shuo Chen, Kexin Hui, Jia-Bin Huang, and Alexander G Schwing. Sail-vos: Semantic amodal instance level video object segmentation-a synthetic dataset and baselines. In *CVPR*, 2019.

[45] Jia-Bin Huang, Sing Bing Kang, Narendra Ahuja, and Johannes Kopf. Temporally coherent completion of dynamic video. *ACM Transactions on Graphics (TOG)*, 35(6):1–11, 2016.

[46] Zhaojin Huang, Lichao Huang, Yongchao Gong, Chang Huang, and Xinggang Wang. Mask scoring r-cnn. In *CVPR*, 2019.

[47] Joakim Johnander, Martin Danelljan, Emil Brissman, Fahad Shahbaz Khan, and Michael Felsberg. A generative appearance model for end-to-end video object segmentation. In *CVPR*, 2019.

[48] Lei Ke, Shichao Li, Yanan Sun, Yu-Wing Tai, and Chi-Keung Tang. Gsnet: Joint vehicle pose and shape reconstruction with geometrical and scene-aware supervision. In *ECCV*, 2020.

[49] Lei Ke, Yu-Wing Tai, and Chi-Keung Tang. Deep occlusion-aware instance segmentation with overlapping bilayers. In *CVPR*, 2021.

[50] Anna Khoreva, Rodrigo Benenson, Jan Hosang, Matthias Hein, and Bernt Schiele. Simple does it: Weakly supervised instance and semantic segmentation. In *CVPR*, 2017.

[51] Dahun Kim, Sanghyun Woo, Joon-Young Lee, and In So Kweon. Deep video inpainting. In *CVPR*, 2019.

[52] Tae Hyun Kim, Mehdi SM Sajjadi, Michael Hirsch, and Bernhard Schölkopf. Spatio-temporal transformer network for video restoration. In *ECCV*, 2018.

[53] Benjamin B Kimia, Ilana Frankel, and Ana-Maria Popescu. Euler spiral for shape completion. *IJCV*, 54(1-3):159–182, 2003.

[54] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *ICLR*, 2017.

[55] Alexander Kirillov, Evgeny Levinkov, Bjoern Andres, Bogdan Savchynskyy, and Carsten Rother. Instancecut: from edges to instances with multicut. In *CVPR*, 2017.

[56] Shu Kong and Charless C Fowlkes. Recurrent pixel embedding for instance grouping. In *CVPR*, 2018.

[57] Philipp Krähenbühl and Vladlen Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In *NeurIPS*, 2011.

[58] Weicheng Kuo, Anelia Angelova, Jitendra Malik, and Tsung-Yi Lin. Shapemask: Learning to segment novel objects by refining shape priors. In *ICCV*, 2019.

[59] Justin Lazarow, Kwonjoon Lee, and Zhuowen Tu. Learning instance occlusion for panoptic segmentation. In *CVPR*, 2020.

[60] Sungho Lee, Seoung Wug Oh, DaeYeun Won, and Seon Joo Kim. Copy-and-paste networks for deep video inpainting. In *ICCV*, 2019.

[61] Youngwan Lee and Jongyoul Park. Centermask: Real-time anchor-free instance segmentation. In *CVPR*, 2020.

[62] Ke Li, Bharath Hariharan, and Jitendra Malik. Iterative instance segmentation. In *CVPR*, 2016.

[63] Ke Li and Jitendra Malik. Amodal instance segmentation. In *ECCV*, 2016.

[64] Minghan Li, Shuai Li, Lida Li, and Lei Zhang. Spatial feature calibration and temporal fusion for effective one-stage video instance segmentation. In *CVPR*, 2021.

[65] Xia Li, Zhisheng Zhong, Jianlong Wu, Yibo Yang, Zhouchen Lin, and Hong Liu. Expectation-maximization attention networks for semantic segmentation. In *ICCV*, 2019.

[66] Yi Li, Haozhi Qi, Jifeng Dai, Xiangyang Ji, and Yichen Wei. Fully convolutional instance-aware semantic segmentation. In *CVPR*, 2017.

[67] Yin Li and Abhinav Gupta. Beyond grids: Learning graph representations for visual recognition. In *NeurIPS*, 2018.

[68] Yingwei Li, Xiaojie Jin, Jieru Mei, Xiaochen Lian, Linjie Yang, Cihang Xie, Qihang Yu, Yuyin Zhou, Song Bai, and Alan L Yuille. Neural architecture search for lightweight non-local networks. In *CVPR*, 2020.

[69] Chung-Ching Lin, Ying Hung, Rogerio Feris, and Linglin He. Video instance segmentation tracking with a modified vae architecture. In *CVPR*, 2020.

[70] Hongwei Lin, Zihao Wang, Panpan Feng, Xingjiang Lu, and Jinhui Yu. A computational model of topological and geometric recovery for visual curve completion. *Computational Visual Media*, 2(4):329–342, 2016.

[71] Ji Lin, Chuang Gan, and Song Han. Tsm: Temporal shift module for efficient video understanding. In *ICCV*, 2019.

[72] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017.

[73] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, 2017.

[74] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014.

[75] Dongfang Liu, Yiming Cui, Wenbo Tan, and Yingjie Chen. Sg-net: Spatial granularity network for one-stage video instance segmentation. In *CVPR*, 2021.

[76] Shu Liu, Jiaya Jia, Sanja Fidler, and Raquel Urtasun. Sgn: Sequential grouping networks for instance segmentation. In *ICCV*, 2017.

[77] Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia. Path aggregation network for instance segmentation. In *CVPR*, 2018.

[78] Shu Liu, Xiaojuan Qi, Jianping Shi, Hong Zhang, and Jiaya Jia. Multi-scale patch aggregation (mpa) for simultaneous detection and segmentation. In *CVPR*, 2016.

[79] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. SSD: Single shot multibox detector. In *ECCV*, 2016.

[80] Yiding Liu, Siyu Yang, Bin Li, Wengang Zhou, Jizheng Xu, Houqiang Li, and Yan Lu. Affinity derivation and graph merge for instance segmentation. In *ECCV*, 2018.

[81] Xiankai Lu, Wenguan Wang, Danelljan Martin, Tianfei Zhou, Jianbing Shen, and Van Gool Luc. Video object segmentation with episodic graph memory networks. In *ECCV*, 2020.

[82] Yasuyuki Matsushita, Eyal Ofek, Weina Ge, Xiaoou Tang, and Heung-Yeung Shum. Full-frame video stabilization with motion inpainting. *TPAMI*, 28(7):1150–1163, 2006.

[83] Tim Meinhardt, Alexander Kirillov, Laura Leal-Taixe, and Christoph Feichtenhofer. Trackformer: Multi-object tracking with transformers. *arXiv preprint arXiv:2101.02702*, 2021.

[84] Anton Milan, Laura Leal-Taixé, Ian Reid, Stefan Roth, and Konrad Schindler. Mot16: A benchmark for multi-object tracking. *arXiv preprint arXiv:1603.00831*, 2016.

[85] Anton Milan, Laura Leal-Taixé, Konrad Schindler, and Ian Reid. Joint tracking and segmentation of multiple targets. In *CVPR*, 2015.

[86] Kamyar Nazeri, Eric Ng, Tony Joseph, Faisal Qureshi, and Mehran Ebrahimi. Edgeconnect: Structure guided image inpainting using edge prediction. In *ICCVW*, Oct 2019.

[87] Alasdair Newson, Andrés Almansa, Matthieu Fradet, Yann Gousseau, and Patrick Pérez. Video inpainting of complex scenes. *Siam journal on imaging sciences*, 7(4):1993–2019, 2014.

[88] Seoung Wug Oh, Joon-Young Lee, Ning Xu, and Seon Joo Kim. Video object segmentation using space-time memory networks. In *ICCV*, 2019.

[89] Seoung Wug Oh, Sungho Lee, Joon-Young Lee, and Seon Joo Kim. Onion-peel networks for deep video completion. In *ICCV*, 2019.

[90] Jiangmiao Pang, Linlu Qiu, Xia Li, Haofeng Chen, Qi Li, Trevor Darrell, and Fisher Yu. Quasi-dense similarity learning for multiple object tracking. In *CVPR*, 2021.

[91] Pedro O Pinheiro, Ronan Collobert, and Piotr Dollár. Learning to segment object candidates. In *NeurIPS*, 2015.

[92] Pedro O Pinheiro, Tsung-Yi Lin, Ronan Collobert, and Piotr Dollár. Learning to refine object segments. In *ECCV*, 2016.

[93] Jiyang Qi, Yan Gao, Yao Hu, Xinggang Wang, Xiaoyu Liu, Xiang Bai, Serge Belongie, Alan Yuille, Philip Torr, and Song Bai. Occluded video instance segmentation. *arXiv preprint arXiv:2102.01558*, 2021.

[94] Lu Qi, Li Jiang, Shu Liu, Xiaoyong Shen, and Jiaya Jia. Amodal instance segmentation with kins dataset. In *CVPR*, 2019.

[95] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *CVPR*, 2016.

[96] Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. In *CVPR*, 2017.

[97] Tal Remez, Jonathan Huang, and Matthew Brown. Learning to segment via cut-and-paste. In *ECCV*, 2018.

[98] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NeurIPS*, 2015.

[99] Andreas Robinson, Felix Jaremo Lawin, Martin Danelljan, Fahad Shahbaz Khan, and Michael Felsberg. Learning fast and robust target models for video object segmentation. In *CVPR*, 2020.

[100] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, 2015.

[101] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. Grabcut: Interactive foreground extraction using iterated graph cuts. 23(3):309–314, 2004.

[102] Hongje Seong, Junhyuk Hyun, and Euntai Kim. Kernelized memory network for video object segmentation. In *ECCV*, 2020.

[103] Nathan Silberman, Lior Shapira, Ran Gal, and Pushmeet Kohli. A contour completion model for augmenting surface reconstructions. In *ECCV*, 2014.

[104] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *NeurIPS*, 2017.

[105] Konstantin Sofiiuk, Olga Barinova, and Anton Konushin. Adaptis: Adaptive instance selection network. In *ICCV*, 2019.

[106] Jian Sun, Yin Li, Sing Bing Kang, and Heung-Yeung Shum. Symmetric stereo matching for occlusion handling. In *CVPR*, 2005.

[107] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *ICCV*, 2019.

[108] Joseph Tighe, Marc Niethammer, and Svetlana Lazebnik. Scene parsing with object instances and occlusion ordering. In *CVPR*, 2014.

[109] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017.

[110] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017.

[111] 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.

[112] Vladimir Vezhnevets and Vadim Konouchine. Growcut: Interactive multi-label nd image segmentation by cellular automata. 1:150–156, 2005.

[113] 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.

[114] Paul Voigtlaender, Michael Krause, Aljosa Osep, Jonathon Luiten, Berin Balachandar Gnana Sekar, Andreas Geiger, and Bastian Leibe. Mots: Multi-object tracking and segmentation. In *CVPR*, 2019.

[115] Apoorv Vyas, Angelos Katharopoulos, and François Fleuret. Fast transformers with clustered attention. *NeurIPS*, 2020.

[116] Chuan Wang, Haibin Huang, Xiaoguang Han, and Jue Wang. Video inpainting by jointly learning temporal structure and spatial details. In *AAAI*, 2019.

[117] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *CVPR*, 2018.

[118] Xiaolong Wang and Abhinav Gupta. Videos as space-time region graphs. In *ECCV*, 2018.

[119] Xinlong Wang, Tao Kong, Chunhua Shen, Yuning Jiang, and Lei Li. Solo: Segmenting objects by locations. *arXiv preprint arXiv:1912.04488*, 2019.

[120] Xinlong Wang, Tete Xiao, Yuning Jiang, Shuai Shao, Jian Sun, and Chunhua Shen. Repulsion loss: Detecting pedestrians in a crowd. In *CVPR*, 2018.

[121] Yuqing Wang, Zhaoliang Xu, Xinlong Wang, Chunhua Shen, Baoshan Cheng, Hao Shen, and Huaxia Xia. End-to-end video instance segmentation with transformers. *arXiv preprint arXiv:2011.14503v1*, 2020.

[122] Yonatan Wexler, Eli Shechtman, and Michal Irani. Space-time completion of video. *TPAMI*, 29(3):463–476, 2007.

[123] John Winn and Jamie Shotton. The layout consistent random field for recognizing and segmenting partially occluded objects. In *CVPR*, 2006.

[124] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. Simple online and realtime tracking with a deep association metric. In *IEEE international conference on image processing (ICIP)*, 2017.

[125] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. Cbam: Convolutional block attention module. In *ECCV*, 2018.

[126] Enze Xie, Peize Sun, Xiaoge Song, Wenhai Wang, Xuebo Liu, Ding Liang, Chunhua Shen, and Ping Luo. Polarmask: Single shot instance segmentation with polar representation. In *CVPR*, 2020.

[127] Haozhe Xie, Hongxun Yao, Shangchen Zhou, Shengping Zhang, and Wenxiu Sun. Efficient regional memory network for video object segmentation. In *CVPR*, 2021.

[128] Ning Xu, Linjie Yang, Yuchen Fan, Dingcheng Yue, Yuchen Liang, Jianchao Yang, and Thomas Huang. Youtube-vos: A large-scale video object segmentation benchmark. *arXiv preprint arXiv:1809.03327*, 2018.

[129] Rui Xu, Xiaoxiao Li, Bolei Zhou, and Chen Change Loy. Deep flow-guided video inpainting. In *CVPR*, 2019.

[130] Xiaopeng Yan, Ziliang Chen, Anni Xu, Xiaoxi Wang, Xiaodan Liang, and Liang Lin. Meta r-cnn : Towards general solver for instance-level low-shot learning. In *ICCV*, 2019.

[131] Xiaosheng Yan, Feigege Wang, Wenxi Liu, Yuanlong Yu, Shengfeng He, and Jia Pan. Visualizing the invisible: Occluded vehicle segmentation and recovery. In *ICCV*, 2019.

[132] Boyu Yang, Chang Liu, Bohao Li, Jianbin Jiao, and Qixiang Ye. Prototype mixture models for few-shot semantic segmentation. In *ECCV*, 2020.

[133] Linjie Yang, Yuchen Fan, and Ning Xu. Video instance segmentation. In *ICCV*, 2019.

[134] Linjie Yang, Yanran Wang, Xuehan Xiong, Jianchao Yang, and Aggelos K Katsaggelos. Efficient video object segmentation via network modulation. In *CVPR*, 2018.

[135] Zongxin Yang, Yunchao Wei, and Yi Yang. Collaborative video object segmentation by foreground-background integration. In *ECCV*, 2020.

[136] Cunjun Yu, Xiao Ma, Jiawei Ren, Haiyu Zhao, and Shuai Yi. Spatio-temporal graph transformer networks for pedestrian trajectory prediction. In *ECCV*, 2020.

[137] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In *CVPR*, 2020.

[138] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Generative image inpainting with contextual attention. In *CVPR*, 2018.

[139] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Free-form image inpainting with gated convolution. In *CVPR*, 2019.

[140] Yanhong Zeng, Jianlong Fu, and Hongyang Chao. Learning joint spatial-temporal transformations for video inpainting. In *ECCV*, 2020.

[141] Xiaohang Zhan, Xingang Pan, Bo Dai, Ziwei Liu, Dahua Lin, and Chen Change Loy. Self-supervised scene de-occlusion. In *CVPR*, 2020.

[142] Chi Zhang, Guosheng Lin, Fayao Liu, Rui Yao, and Chunhua Shen. Canet: Class-agnostic segmentation networks with iterative refinement and attentive few-shot learning. In *CVPR*, 2019.

[143] Haotian Zhang, Long Mai, Ning Xu, Zhaowen Wang, John Collomosse, and Hailin Jin. An internal learning approach to video inpainting. In *ICCV*, 2019.

[144] Li Zhang, Xiangtai Li, Anurag Arnab, Kuiyuan Yang, Yunhai Tong, and Philip HS Torr. Dual graph convolutional network for semantic segmentation. In *BMVC*, 2019.

[145] Shifeng Zhang, Longyin Wen, Xiao Bian, Zhen Lei, and Stan Z Li. Occlusion-aware r-cnn: detecting pedestrians in a crowd. In *ECCV*, 2018.

[146] Songyang Zhang, Shipeng Yan, and Xuming He. LatentGNN: Learning efficient non-local relations for visual recognition. In *ICML*, 2019.

[147] Chunluan Zhou and Junsong Yuan. Bi-box regression for pedestrian detection and occlusion estimation. In *ECCV*, 2018.

[148] Yanzhao Zhou, Yi Zhu, Qixiang Ye, Qiang Qiu, and Jianbin Jiao. Weakly supervised instance segmentation using class peak response. In *CVPR*, 2018.

[149] Yan Zhu, Yuandong Tian, Dimitris Metaxas, and Piotr Dollár. Semantic amodal segmentation. In *CVPR*, 2017.