用使用注意力机制的时空金字塔特征进行视频实例分割

学生：张瑞阳，复旦大学计算机科学技术学院
指导老师：张文强教授，复旦大学计算机科学技术学院

摘要

在图像和视频中作分割是计算机视觉中的基础性问题之一。图像实例分割任务因为它的重要性在计算机视觉领域引起了非常多的关注。将实例分割从图像领域推广到视频领域，就产生了视频实例分割任务。视频实例分割任务包括对视频中的实例的同时的检测、分类、分割和跟踪。视频实例分割任务比图像实例分割任务更有挑战性，它不仅需要在视频的每一帧中进行实例分割，还要在帧与帧之间对实例进行跟踪。另一方面，视频和图像相比包含了更多的信息，比如不同实例的运动模式和时序上的连续性，者可以给实例的识别和分割提供更多线索。视频实例分割任务中现有的工作的模型都是基于提议的模型，它们通常包含多个模型，并且计算量很大。与这些工作相反，这个工作提出了一个单阶段的模型解决视频实例分割任务。它会将视频编码成时空特征金字塔，时空特征金字塔能够充分使用视频中的时空信息。在时空特征金字塔的基础上，进一步引入了通道注意力机制和时空注意力机制。通过实验，这个工作在 Youtbu-VIS[31] 上的结果超过了之前的工作的结果。

1. 介绍

在图像和视频中做分割是计算机视觉中的基础性问题之一。在图像领域，实例分割任务（同时对目标进行检测、分类和分割）由 Hariharanet al. [8] 首次提出，从此以后因为它的重要性在计算机视觉领域中引起了非常多的关注。将实例分割任务从图像领域推广到视频领域，就产生了视频实例分割任务。与图像实例分割任务不同，这个任务包括对视频中的目标的同时的检测、分类、分割和跟踪。这是因为视频比图像多一个维度，即时序，所以自然的要求对视频中的目标的跟踪。视频实例分割任务在很多需要视频级目标遮罩的地方都能得到应用，比如视频编辑、自动驾驶和增强现实等。

视频实例分割任务比图像实例分割任务更有挑战性，它不仅需要在每一帧中进行实例分割，还需要在帧与帧之间对实例进行跟踪。另一方面，视频和图像相比包含了更多的信息，比如不同目标的运动模式和时序上的连续性，这可以给目标的识别和分割提供更多线索。视频实例分割任务和很多其他的任务都有联系。比如，视频目标分割任务 [3][19][20] 要对视频中的目标进行分割和跟踪，但不需要识别出目标的类别。视频目标检测任务要对视频中的目标进行检测和跟踪，但不需要处理目标分割。

视频实例分割任务中现有的工作的模型都是多阶段的模型，这些模型通常会先有一个模型产生提议，然后之后的模型会进行检测、分割和跟踪，它们通常包含多个模型，并且计算量很大。

与这些工作相反，这个工作提出了一个单阶段的模型进行视频实例分割。这个模型的输入是视频片段，它会先使用视频片段生成空间特征金字塔，然后使用空间特征金字塔进一步生成时空特征金字塔，此时的特征在空间和时序上都有从大到小的不同的尺度，在得到的时空特征金字塔上还会使用通道注意力机制和时空注意力机制。提出的模型的结构见图6。通过实验，这个工作在 Youtube-VIS[31] 上的结果超过了之前的工作的结果。

总结来说，这个工作的贡献如下：

- 提出了一种充分使用视频中的时空信息的方法，即时空特征金字塔。
- 在时空特征金字塔的基础上，进一步引入了通道注意力机制和时空注意力机制。
- 提出了一个单阶段的模型解决视频实例分割任务。通过实验，这个工作在 Youtube-VIS[31] 上的结果超过了之前的工作的结果。

2. 相关工作

2.1. 图片实例分割

2.1.1 综述

基于提议的图片实例分割 [8][4][13][9] 将目标检测和实例分割结合了起来，这些方法在图片实例分割任务上取得了最好的结果。还有一类方法是基于 RNN 的图片实例分割 [33][21]，这些方法在小数据集上结果很好，但当图片中的实例数很多时，梯度消失问题就会变得很严重。还有一类方法是无提议的图片实例分割 [5][23]，这些方法使用神经网络把图片的所有像素嵌入到一个多维隐空间中，其中同一个实例的像素的嵌入应该互相靠近，不同实例的像素的嵌入应该相互远离，最后使用聚类算法就可以分割出实例。

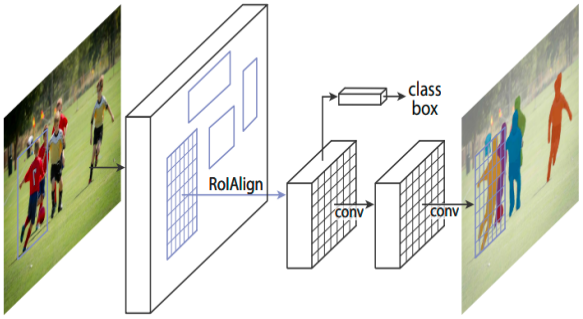

Figure 1. Mask R-CNN[9] 的结构。

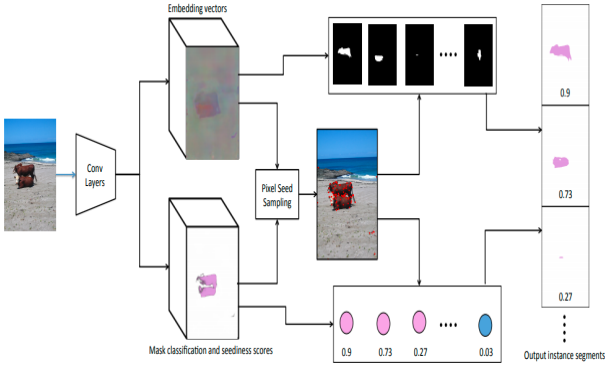

Figure 2. Fathi et al. [5] 的结构。

2.1.2 经典模型

Mask R-CNN[9] Mask R-CNN[9] 提出于 2017 年，是在 Faster R-CNN[22] 的基础上改进后被用于解决图像实例分割的问题。相对于原来的 Faster R-CNN 主干框架，它在网络的头上引入了另外一条 FCN 并行分支用来检测 ROI 的遮罩信息。这样最终它的头部共有三条并行的分支分别用来处理 ROI 区域的类别识别、目标框位置回归及相应的遮罩回归。

Mask-RCNN 的主干网络同 Faster R-CNN 一样可以选用任何 CNN 如 VGG[24]、resnet[10] 等。论文中作者使用了 resnet[10] 与 FPN 来分别作为模型的主干网络。在主干网络最终产生的特征图集合之上，我们使用 RPN 生成多个区域提议出来。然后再将这些区域提议分别生成对应的 ROI 窗口，进而用于后续的分类、目标框定位和遮罩预测。在这里作者对之前的粗粒度 ROI pool 进行了改进，提出了更加细粒度的，可在子像素级别上提取 ROI 窗口区域特征的 ROI align 层。它可以有效地避免像素错位，能够取得更加准确的区域位置信息。遮罩分支通过在 ROI Align 层后输出的 ROI 特征上接上一个 FCN，最终得到一个与 ROI 区域相对应的遮罩图出来。这个遮罩图有 K 个通道，K 在这里表示目标可能的类别数目。另外每个通道的遮罩图上面都是些二元信息，分别表示 ROI 区域上面的某位置点是前景还是背景。模型的结构见图1。

Fathi et al. [5] 度量学习（Metric Learning）也就是常说的相似度学习。如果需要计算两张图片之间的相似度，如何度量图片之间的相似度使得不同类别的图片相似度小而相同类别的图片相似度大就是度量学习的目标。

在这篇论文中，度量学习用于计算像素之间的相似度，让属于同一目标的像素之间相似度大而不同目标的像素之间相似度小。整个模型有两个输出端，其中，第一个输出端为每个像素产生嵌入向量，在第二个输出端，模型预测以每个像素为中心生成的遮罩的类别标签（C+1 种），以及这个像素将成为创建遮罩的良好“种子”的置信度。最后，模型会选出一定数量的种子，这些种子在峰值处，而且有一定的位置多样性，然后由阀值确定最后的实例和它的类别。模型的结构见图2。

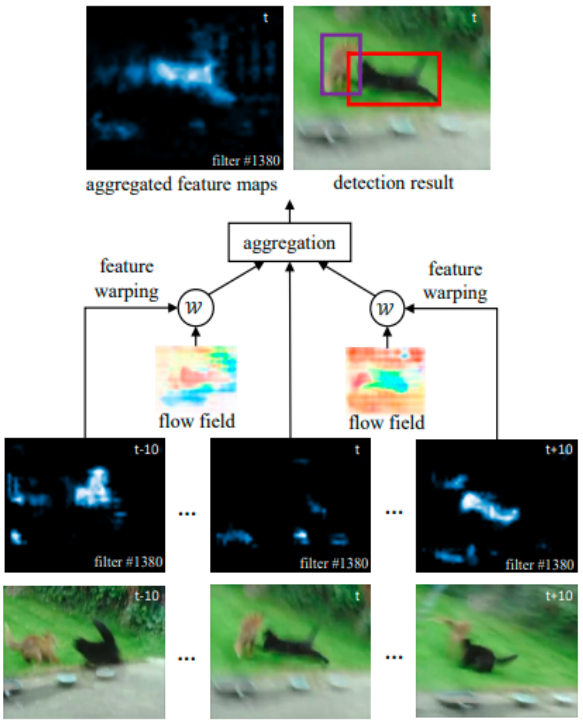

Figure 3. FGFA[34] 的结构。

2.2. 视频目标检测

2.2.1 综述

视频目标检测 [34][6][2][30] 可以分类和定位给定视频的每一帧中的目标。多数的视频目标检测方法实现了一些形式的时空特征对齐来提高在单个视频帧中的目标检测准确率。但是，这些方法通常不会跟踪目标。

2.2.2 经典模型

FGFA[34] 视频目标检测中，存在图像质量退化的问题，有运动模糊、视频失焦、部分遮挡、奇特姿势等。现有的目标检测算法不能很好地应对这些问题。作者

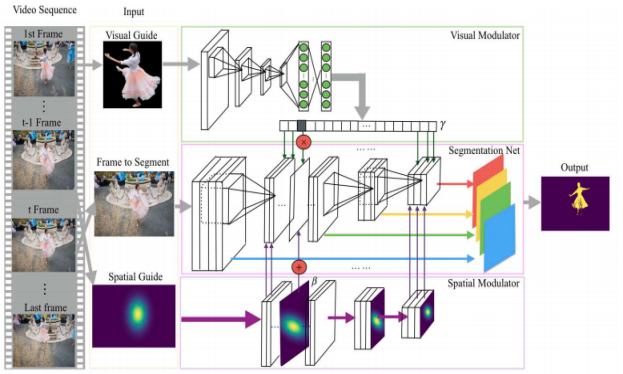

Figure 4. OSNM[32] 的结构。

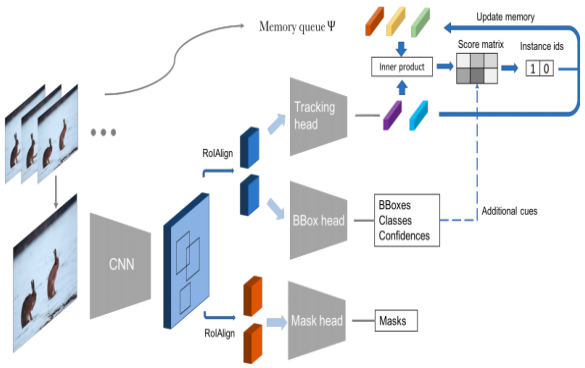

Figure 5. MaskTrack R-CNN[16] 的结构。

打算利用视频的连续性，视频中短时间内同一目标会被多次观察到，用质量好的图像特征来增强质量差的图像特征，来达到提高检测精度的目的，这里也用到了光流对特征进行 warp。

FGFA[34] 追求精度而不考虑速度，对于视频中的所有帧都做同样的处理，就是说把每一帧都当作关键帧。同时提取关键帧和其前后相邻 K 帧的特征，一共提取 2K+1 帧的图片特征，并计算每一相邻帧和关键帧的特征光流图，然后将相邻帧的特征 warp 到关键帧，最后通过加权求和得到关键帧的增强特征，这里对特征图上每一个点都求一个权值，这里每一个点的权值代表了这个点对关键帧上对应点的重要性，如果 warp 后的特征与关键帧的特征越接近，就给予更大的权值，否则分配较小的权值。这里用余弦相似度来衡量两个特征的相似度，最后通过 softmax 对权值进行归一化，将增强后的特征送入检测网络，得到检测结果。模型的结构见图3。

2.3. 视频目标分割

2.3.1 综述

视频目标分割 [12][25][32][27] 可以分割用于类别无关的方式分割前景物体，通常是在推理时使用可用的第一帧的真实遮罩。

2.3.2 经典模型

OSNM[32] 由于 fine-tuning 过程太过耗时，作者添加另外的网络，即 modulator，通过学习被标注目标的 appearance 信息和前一帧的 spatial 信息，来调整 Segmentation Network 的相关参数，以达到分割我们想要的目标的结果。此方法是受 CBN 启发。CBN，即 Conditional Batch Normalization，有别于传统的 BN layer，它的 scale 和 bias 参数是通过另外的 Controller Network 学习而来的，受此些参数的影响，主网络会受到不同程度的控制。

整个模型共有三个分支。一个分支是 Segmentation Net，它是模型的主体，使用的主干网络是 VGG16。还有一个分支是 Visual Modulator，它负责从被标注的目标中学习相关的语义信息。主要作用是让 Segmentation Net 的注意力集中在我们想要的目标上。此网络的输入是从第一帧中裁剪而来的 224*224 的 annotated object image，即 visual guide。输出是 channel-wise 的 scale 参数。其维度是 segmentation Net 中对应的特征图层数。还有一个分支是 Spatial Modulator，它负责提取目标的 spatial 信息。此网络的输入是前一帧所预测到的遮罩的位置信息，即 spatial guide。本文用二维高斯分布来编码其位置信息。Spatial Modulator 利用 pooling 层和 1*1 conv 层来将二维高斯分布的 scale 与 Segmentation Network 中特征图的 size 相匹配。模型的结构见图4。

2.4. 视频实例分割

2.4.1 综述

MaskTrack R-CNN[31] 是一个用于视频实例分割的统一的模型。它使用了一个跟踪分支增强了 Mask R-CNN，这个跟踪分支可以在不同帧中分割出的实例间确立联系。ICCV 2019 视频实例分割挑战冠军 [16] 的方法是把视频实例分割分解为四个子问题：分类、检测、分割和跟踪，一个单独的模型或者几个模型组成的整体会用来解决每个子问题，然后子问题的结果会组合起来产生视频实例分割的结果。

2.4.2 经典模型

MaskTrack R-CNN[16] MaskTrack R-CNN[16] 就是 Mask R-CNN[9] 再加一个 Tracking head 分支。整个网络也是 two-stage 的，first stage 就是对每一帧都产生各自的一系列 bounding box。在 BBox Head 和 Mask Head 计算的同时，加一个 Tracking Head（2 个全连接层），用于对每一个候选框分配一个实例标签。假设已经计算出有 N 个实例，那么当前帧的候选框所属的标签要么属于这 N 个，要么属于一个新的标签。所以把这个当作 N+1 类的分类问题。这个分类问题使用一个 Memory queue 来解决，如果当前帧的候选框对应的标签属于已有的 N 个实例中的一个，那么 Memory queue 里的特征会更新，如果是一个新的标签，那么 Memory

queue 里也会添加一个新的特征。训练的时候，随机挑选一对 frames，一个做 reference，一个做 query。对 reference frame 只提取 Ground truth 里 instance region 里的特征存到 Memory queue 里，query frame 会先在 first stage 里选出 positive candidate bounding box，再对它分配标签。作者选择 IoU 和 Ground truth bounding box overlap 超过 0.7 的作为 positive。模型的结构见图5。

2.5. 计算机视觉中的注意力机制

2.5.1 **综述**

SENet[11] 中，对提取出来的特征做全局平均池化，然后使用一个线性层将通道数降下来，再使用一个线性层将通道数还原，通过一个 Sigmoid 层，然后将输出和原特征相乘，得到最终结果，从而形成了通道注意力机制。CBAM[29] 在通道注意力机制的基础上，对提取出来的特征做全局平均池化和全局最大池化，并将两个输出并起来，然后通过一个卷积层，通过一个 Sigmoid 层，然后将输出和原特征相乘，得到最终结果，从而形成了空间注意力机制。Non-local Neural Network[28] 提出了非局部注意力机制，这种注意力机制能考虑到全局中任意一个部分对另外一个部分的影响。DANet[7] 在 CBAM[29] 的基础上，引入了 Non-local Neural Network[28]，它将 CBAM[29] 中的注意力机制，替换为 Non-local Neural Network[28] 中的非局部注意力机制。以上工作没能在视频实例分割任务中得到应用。

3. **任务定义**

视频实例分割将图像实例分割任务从图像域扩展到视频域。该任务的目标是同时检测、分割和跟踪视频中的目标实例。给定一个测试视频，该任务不仅需要对预定义类别集的所有实例的遮罩进行标记，还需要把每个实例的所有遮罩关联到一起。具体来说，设 $\mathbb{V} \in \mathbb{R}^{T\times 3\times H\times W}$ 表示一个有 T 帧，每帧大小为 $H\times W$ 的视频，$C=\{1,...,K\}$ 是已知的类别，需要输出遮罩 $M^i \in \mathbb{R}^{T\times H\times W}$，类别 $c^i \in \{1,...,K\}$ 和置信度 $s^i \in [0,1]$，其中 i 是每个在视频中出现的实例。在对视频实例分割的结果进行评价时，不仅要看对每帧中的实例的分类和分割是否准确，还要看对每个实例在整个视频中的跟踪是否准确。

4. **方法**

在这部分中，会依次介绍时空特征金字塔、CBAM[29] 这种注意力机制、最终模型的结构、模型的损失函数和模型的推理过程。

4.1. 时空特征金字塔

时空特征金字塔的产生分为两步。首先，使用常用的 Feature Pyramid Network[14] 就可以完成空间特征金字塔的生成，Feature Pyramid Network[14] 会输出 $\frac{1}{32}$、$\frac{1}{16}$、$\frac{1}{8}$、$\frac{1}{4}$ 四种不同尺度的空间特征，从而形成了空间特征金字塔。空间特征金字塔的意义是能让模型同时关注到在空间维度上从大到小的目标。然后，使用 Squeeze 模块改变空间特征金字塔的时间尺度，出于 GPU 显存友好的角度出发，对空间尺度大的特征，通过较少的 Squeeze 模块，对空间尺度小的特征，通过较多的 Squeeze 模块。其中 Squeeze 模块由四个网络层组成，依次是核大小为 $3\times3\times3$ 的 Conv 层、BN 层、ReLU 层和核大小为 $3\times1\times1$ 的 AvgPool 层，Squeeze 模块可以使输入特征的时间维度减半。最终会生成 $\frac{1}{4}$、$\frac{1}{2}$、1 三种不同时间尺度的特征。具有三种不同时间尺度的意义是能让模型同时关注到在时间维度上从大到小的信息。这样最终就形成了时空特征金字塔，时空特征金字塔中的特征在空间维度上和时间维度上都有不同的尺度，会使模型关注到时空维度上不同大小的信息，让模型能够充分使用输入的视频中的时空信息。最终大小为 $T, \frac{H}{4}, \frac{W}{4}$ 的特征会经过插值变成大小为 T, H, W 的特征。

4.2. CBAM

CBAM[29] 即卷积块注意力模块，是一种计算机视觉中的注意力机制。它既使用了通道注意力机制，也使用了空间注意力机制。对于卷积层提取出的特征，它会先使用通道注意力机制，再使用空间注意力机制。CBAM[29] 原本是使用在图像任务中的，在这个工作中对它进行了拓展，把它使用在了视频任务中，将其中的空间注意力机制拓展成了时空注意力机制。

具体来说，记 F 是卷积层提取出的特征，$\mathrm{M_c}(\mathrm{F}) \in \mathbb{R}^{1\times C\times1\times1}$ 是求出的通道注意力图，则：

$$\begin{aligned}\mathrm{M_c}(\mathrm{F}) &= \sigma\left(MLP\left(AvgPool3D\left(\mathrm{F}\right)\right) + MLP\left(MaxPool3D\left(\mathrm{F}\right)\right)\right) \\ &= \sigma\left(\mathrm{W_1}\left(\mathrm{W_0}\left(\mathrm{F^c_{avg}}\right)\right) + \mathrm{W_1}\left(\mathrm{W_0}\left(\mathrm{F^c_{max}}\right)\right)\right)\end{aligned} \tag{1}$$

其中 $\mathrm{F^c_{avg}}$ 为 F 在时序高宽维做三维平均池化的结果，$\mathrm{F^c_{max}}$ 为 F 在时序高宽维做三维最大池化的结果，MLP 有三层，包括线性层、修正线性单元、线性层，其中第一个线性层的参数为 $\mathrm{W_0} \in \mathbb{R}^{\mathrm{C/r}\times\mathrm{C}}$，第二个线性层的参数为 $\mathrm{W_1} \in \mathbb{R}^{\mathrm{C}\times\mathrm{C/r}}$，其中 r 为减少率，σ 为 sigmoid 函数。

记 $\mathrm{M_s}(\mathrm{F}) \in \mathbb{R}^{\mathrm{T}\times1\times\mathrm{H}\times\mathrm{W}}$ 为求出的空间注意力图，则：

$$\begin{aligned}\mathrm{M_s}(\mathrm{F}) &= \sigma\left(f^{7\times7\times7}\left(\left[AvgPool\left(\mathrm{F}\right); MaxPool\left(\mathrm{F}\right)\right]\right)\right) \\ &= \sigma\left(f^{7\times7\times7}\left(\left[\mathrm{F^s_{avg}}; \mathrm{F^s_{max}}\right]\right)\right)\end{aligned} \tag{2}$$

其中 $\mathrm{F^s_{avg}}$ 为 F 在通道维做平均池化的结果，$\mathrm{F^s_{max}}$ 为 F 在通道维做最大池化的结果，$f^{7\times7\times7}$ 为一个卷积核大小为 $7\times7\times7$ 的卷积运算，σ 为 sigmoid 函数。

图 7是通道注意力机制和时空注意力机制的结构。

4.3. 最终模型

最终模型的结构见图6。最终的模型从整体上看，由一个 Encoder 和一个 Decoder 组成。Encoder 包括上

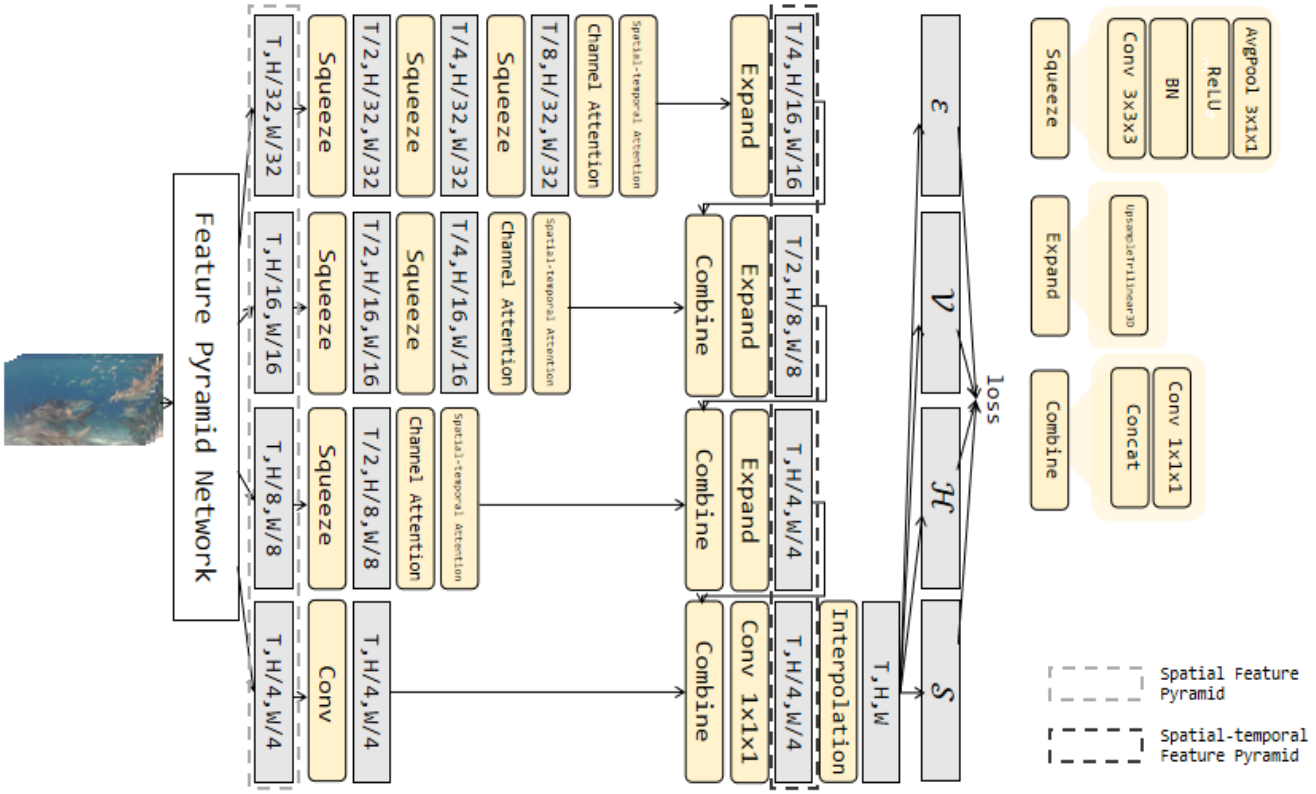

Figure 6. 提出的模型的结构。模型的输入为视频片段。模型会首先使用 Feature Pyramid Network[14] 生成空间特征金字塔，然后会使用空间特征金字塔生成时空特征金字塔，不同尺度的特征都会通过通道注意力机制和时空注意力机制。

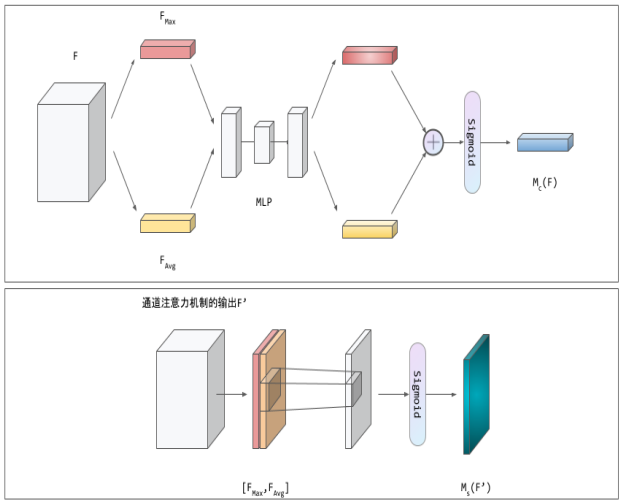

Figure 7. 通道注意力机制和时空注意力机制的结构。

面时空特征金字塔的生成中涉及的结构，而且在每个尺度的时空特征之后，都有通道注意力机制和时空注意力机制。在特征的时空尺度具有多样性之后，再加入通道注意力机制和时空注意力机制，能让模型不仅能够关注到时空维度上从大到小的信息，而且对每个尺度的时空特征能在语义、空间和时序上都有所侧重。

Decoder 把 Encoder 编码出的特征进行解码，最终得到输出。Decoder 中包含 Expand 模块和 Combine 模块，Expand 模块由一个网络层组成，为 UpsampleTrilinear3D 层，Expand 模块可以使输入特征的时序高宽维翻倍，Combine 模块由两个网络层组成，依次是 Concat 层和核大小为 $1 \times 1 \times 1$ 的 Conv 层，Combine 模块可以把两个时序高宽维相同的特征在通道维并起来，并且会通过一个 Conv 层增加模型的深度。尺度最小的特征会通过 Expand 模块，然后和尺度更大的特征合并，然后再通过 Expand 模块，最终所有尺度的特征会生成一个大小为 $\mathbb{R}^{N \times C \times T \times \frac{H}{4} \times \frac{W}{4}}$ 的特征，这个特征会再通过一个核大小为 $1 \times 1 \times 1$ 的 Conv 层，得到的特征会用来生成输出。Decoder 会使用 Combine 模块把不同尺度的特征逐步融合在一起，从而使最终得到的特征能够包含不同尺度的特征中包含的信息。

4.4. 损失函数

整个模型的输入是一个视频片段，它包含 T 帧，每帧的分辨率是 $H \times W$。这个视频片段也可以表示为一个像素集 $\mathcal{X} \in \mathbb{R}^{N \times 3}$，其中 $N = T \times H \times W$，N 表示像素的个数。整个模型可以看作一个将像素集 $\mathcal{X}$ 映射到四个输出的函数：(1) $\mathcal{E} \in \mathbb{R}^{N \times E}$：每个像素的 E 维嵌入，(2) $\mathcal{V} \in \mathbb{R}_{+}^{N \times E}$：每个像素的嵌入的每个维度的方差，(3) $\mathcal{H} \in [0,1]^{N}$：每个像素作为实例中心的分数的热力图，(4) $\mathcal{S} \in \mathbb{R}^{N \times C}$，其中 C 是数据集的目标

的种类数：每个像素属于每个类的分数。设 K 为输入的视频片段中的实例数量，j 表示第 j 个实例，$\mathcal{M}_j$ 表示第 j 个实例的所有遮罩的所有像素组成的集合，N_j 表示第 j 个实例的所有遮罩的所有像素的数量，$\mathcal{E}_j$ 表示第 j 个实例的所有遮罩的所有像素的嵌入组成的集合，$\mathcal{E}_j \subset \mathcal{E}$，$\mathcal{E}_j \in \mathbb{R}^{N_j \times E}$，$\mathcal{V}_j$ 表示第 j 个实例的所有遮罩的所有像素的嵌入的每个维度的方差组成的集合，$\mathcal{V}_j \subset \mathcal{V},$，$\mathcal{V}_j \in \mathbb{R}_+^{N_j \times E}$。然后进行如下计算：

$$\vec{\mu}_j = \frac{1}{N_j} \sum_{\vec{e} \in \mathcal{E}_j} \vec{e} \in \mathbb{R}^E, \qquad \Sigma_j = \frac{1}{N_j} \operatorname{diag}\left(\sum_{\vec{v} \in \mathcal{V}_j} \vec{v} \right) \in \mathbb{R}^{E \times E} \tag{3}$$

这样就得到了一个多维高斯分布 $\mathcal{N}(\vec{\mu}_j, \Sigma_j)$，使用 $\mathcal{N}(\vec{\mu}_j, \Sigma_j)$ 就可以计算输入的视频片段中每个像素 i 属于第 j 个实例的概率 p_{ij}，这种建模方式受 [18] 启发：

$$p_{ij} = \frac{1}{(2\pi)^{\frac{E}{2}} |\boldsymbol{\Sigma_j}|^{\frac{1}{2}}} \exp\left(-\frac{1}{2} (\vec{e}_i - \vec{\mu}_j)^T \Sigma_j^{-1} (\vec{e}_i - \vec{\mu}_j) \right) . \tag{4}$$

通过把概率的阈值设置为 0.5，就可以得到第 j 个实例的所有预测遮罩 $\mathcal{M}_j$：

$$\widehat{\mathcal{M}}_j = \{ (x_i, y_i, t_i) \,|\, i \in \{1, .., N\},\ p_{ij} > 0.5 \} . \tag{5}$$

整个模型的损失函数是四个项的和：

$$L_{\text{total}} = L_{\text{embedding}} + L_{\text{smooth}} + L_{\text{center}} + L_{\text{classification}} \tag{6}$$

嵌入损失 $L_{\text{embedding}}$：使用 Lovàsz 铰链损失 [18] 优化 $\mathcal{M}_j$ 和 $\widehat{\mathcal{M}}_j$ 之间的 IoU，Lovàsz 铰链损失是一种可以直接优化 IoU 的损失函数。输入的视频片段中的所有实例的 Lovàsz 铰链损失的平均作为 $L_{\text{embedding}}$。方差平滑损失 L_{smooth}：为保证每个实例的所有遮罩的所有像素的方差保持一致，使用了一个和 [18] 相似的方差平滑损失 L_{smooth}。这会使 $\mathcal{V}_j$ 中的所有方差趋向于 $\mathcal{V}_j$ 的平均值。实例中心热力图损失 L_{center}：使用了一个 L_2 损失，对于第 j 个实例的所有遮罩的所有像素，实例中心热力图 $\mathcal{H}_j$ 中的值趋向于公式4的输出，对于背景中的像素实例中心热力图 $\mathcal{H}_j$ 中的值趋向于 0。分类损失 $L_{\text{classification}}$：把 $\widehat{\mathcal{M}}_j$ 中的所有像素属于每个类的分数求平均，平均分数最高的那个类就是第 j 个实例的类别，使用一个交叉熵损失作为分类损失 $L_{\text{classification}}$。

4.5. 推理

对于每个像素 $\vec{c}_i = (x_i, y_i, t_i)$，$\mathcal{H}(\vec{c}_i) \in [0, 1]$ 表示像素 $\vec{c}_i$ 是一个实例的中心的分数。推理出输入的视频片段中的所有实例的过程如下：1. 找出分数最高的实例中心 $\vec{c}_j = \operatorname{argmax}_i \mathcal{H}(\vec{c}_i)$。2. 找到对应的 $\mathcal{E}(\vec{c}_j)$ 和 $\mathcal{V}(\vec{c}_j)$。3.$\vec{\mu}_j \leftarrow \mathcal{E}(\vec{c}_j)$，$\Sigma_j \leftarrow \operatorname{diag}(\mathcal{V}(\vec{c}_j))$，然后使用公式4和公式5计算出 $\widehat{\mathcal{M}}_j$。4. 因为 $\widehat{\mathcal{M}}_j$ 中的所有像素都已经被赋予了一个实例，这些像素的嵌入、方差和实例中心热力图都要被移除：

$$\mathcal{E} \leftarrow \mathcal{E} \setminus \widehat{\mathcal{E}}_j , \qquad \mathcal{V} \leftarrow \mathcal{V} \setminus \widehat{\mathcal{V}}_j , \qquad \mathcal{H} \leftarrow \mathcal{H} \setminus \widehat{\mathcal{H}}_j . \tag{7}$$

5. 重复步骤 1-4 直到或者 $\mathcal{E} = \mathcal{V} = \mathcal{H} = \emptyset$，或者实例中心热力图中下一个最高的分数低于某个阈值。

5. 实验

5.1. 数据集

使用的数据集是 Youtube-VIS[31]，这个数据集有 2019 年版和 2021 年版两个版本。2019 年版有 2,883 个高分辨率 YouTube 视频，其中训练集有 2,238 个视频，验证集有 302 个视频，测试集有 343 个视频。这个数据集的类别标签集包括 40 种常见对象，如人、动物和车辆。一共有 4,883 个不同的视频实例，131,000 个高质量的人工标注。2021 年版有 3,859 个高分辨率 YouTube 视频，其中训练集有 2,985 个视频，验证集有 421 个视频，测试集有 453 个视频。对类别标签集进行了改进，还是包括 40 种常见的对象，将鹰类和猫头鹰类合并成了鸟类，将猿类并入了猴类，删除了手类，并增加了飞碟类、松鼠类和鲸鱼类。因为 2019 年版的数据集上之前工作的实验结果较多，方便进行比较，所以实验中采用的是 2019 年版的数据集。因为数据集的测试集评测服务器现在是关闭的，所以下面的实验结果都是在验证集上的结果。

5.2. 评价指标

使用了图像实例分割中的标准的评价指标，并根据视频实例分割的特点对它们进行了适当的修改。具体来说，使用的两个评价指标是平均准确率（AP）和平均召回率（AR）。平均准确率（AP）定义为准确率-召回率（PR）曲线下的面积。绘制准确率-召回率（PR）曲线需要一个实例是一个预测类别的置信度（介于 0 到 1 之间）。平均准确率（AP）是多个使用不同 IoU 阈值的准确率的平均。这里遵循 COCO 的评价指标，使用 10 个 IoU 阈值，从 50% 到 95%，步长为%5。平均召回率（AR）定义为给定每个视频的实例的数量的最大召回率。这两个指标都是先按类别进行计算，然后对所有类别的结果进行平均。根据新任务，对两个评价指标做的唯一的修改是 IoU 计算。因为与图像实例分割不同，视频中的每个实例都有一个掩码序列。因此，IoU 计算不仅要在空间域中进行，还要在时间域中进行，即每一帧的交的和除以每一帧的并的和。

5.3. 实现细节

使用的 Feature Pyramid Network[14] 的骨架网络是一个 ResNet-101[1]，它的初始化参数来自于在 COCO[15] 上做图像实例分割任务的 Mask R-CNN[9]。模型中其他部分的参数是随机初始化的。模型使用 SGD 进行优化，动量设置为 0.9，指数衰减率设置为

Method	mAP	AP@50	AP@75	AR@1	AR@10
OSMN MaskProp[32]	23.4	36.5	25.7	28.9	31.1
IoUTracker+[31]	23.6	39.2	25.5	26.2	30.9
DeepSORT[17]	26.1	42.9	26.1	27.8	31.3
FEELVOS[26]	26.9	42.0	29.7	29.9	33.4
OSMN[32]	27.5	45.1	29.1	28.6	33.1
SeqTracker[31]	27.5	45.7	28.7	29.7	32.5
MaskTrack R-CNN[31]	30.3	51.1	32.6	31.0	35.5
Ours	31.6	49.9	34.6	31.3	37.6

Table 1. 在 Youtube-VIS[31] 验证集上的结果。使用的评价指标有五个，分别是 mAP、AP@50、AP@75、AR@1、AR@10。这个工作的结果超过了之前的工作的结果。

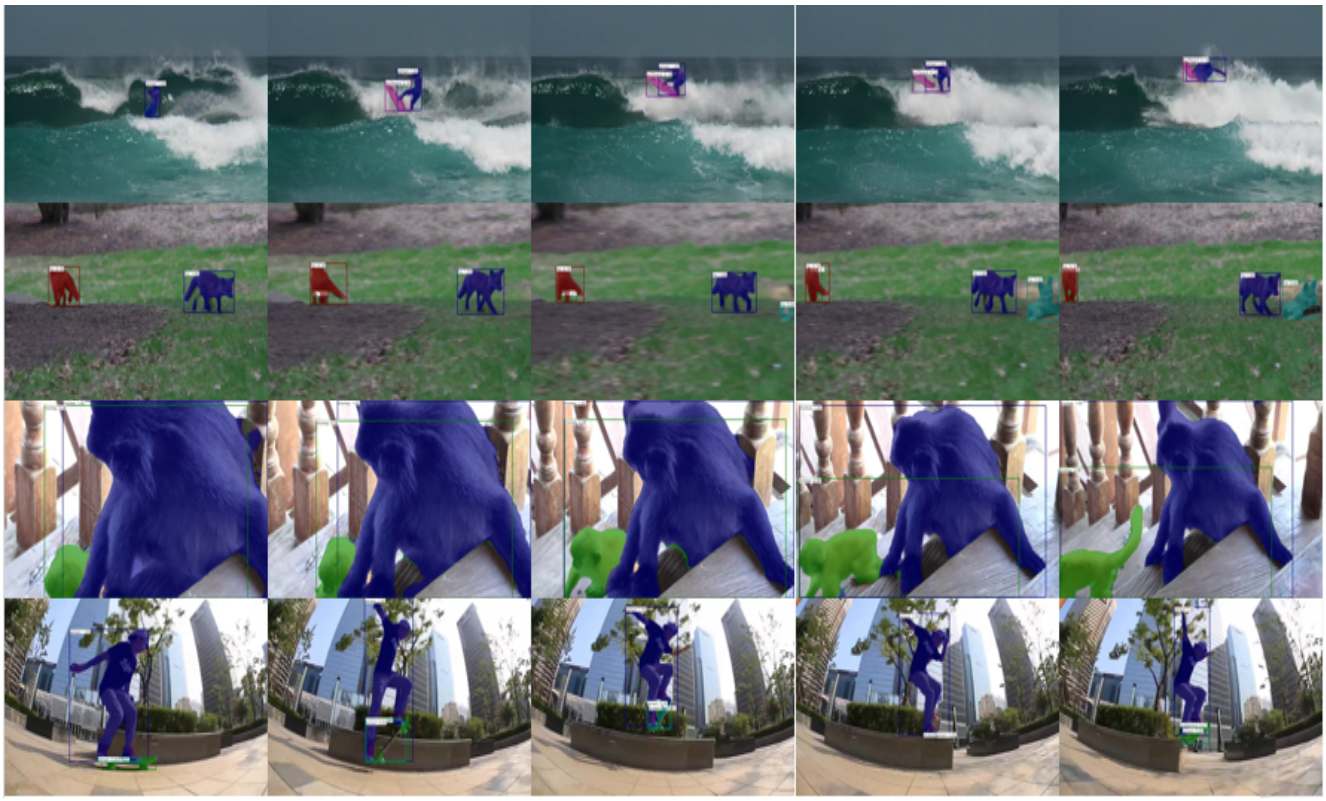

Figure 8. 在 Youtube-VIS[31] 验证集上的定性结果。每行包含的图像来自于同一个视频。对于每个视频，相同的颜色代表同一实例的遮罩序列（电脑屏幕上查看最佳）。

10^{-4}。初始学习率设置为 10^{-3}，而且呈指数衰减，指数衰减从 6×10^4 次迭代时开始，持续 10^5 次，衰减率为 10^{-2}。

因为目前能够用于视频实例分割任务的视频数据集的数据数量有限，这个工作使用了图像实例分割任务的数据集中的图片来合成视频片段，从而增广了可以用来训练的数据的数量。在生成视频片段的过程中使用了随即仿射变换和运动模糊的方法。使用的图像实例分割任务的数据集是 COCO2017[15] 和 VOC2012 的训练集。

5.4. 定量结果

视频实例分割任务是一个最近才出现的计算机视觉任务，所以，之后很少的基准可以用来进行比较。这个工作在比较中包含了 MaskTrack R-CNN[31]，同时也包含了一些为了解决其他任务的工作，不过对这些工作进行了调整，使它们能够应用在视频实例分割任务中。

定量结果见表1。用到的评价指标有：1）视频平均准确率均值（mAP），2）视频平均准确率，IoU 阈值为 50%（AP@50），3）视频平均准确率，IoU 阈值为 75%（AP@75），4）视频平均召回率，给定每个视频第一高分的实例（AR@1），5）视频平均召回率，给定每个视频前十高分的实例（AR@10）。从结果中可以看到，这个工作的结果在 mAP 这个评价指标上超过了所有基准的结果，取得了在 Youtube-VIS[31] 上的视频实例分割任务的最好结果。这些用来比较的基准都是多阶段的模型，IoUTracker+、OSMN 和 DeepSORT 都要先生成提议，然后根据提议生成检测、分割和跟踪的结果，OSMN MaskProp 和 FEELVOS 需要视频的第一帧的

标签信息，然后把视频的第一帧的标签信息作为提议，生成最终的结果，MaskTrack R-CNN 需要事先生成光流，然后再进行训练。这个工作的模型是一个单阶段的模型，但这个工作的结果仍然超过了这些用来比较的基准。说明这个工作提出的时空特征金字塔能够充分的使用视频片段中包含的时空信息，使用的通道注意力机制和时空注意力机制能让模型使用视频片段中包含的时空信息时在语义、空间和时序上有所侧重。

5.5. 定性结果

在 Youtube-VIS[31] 验证集上的定性结果见图8，每一行包含的图像来自于同一个视频。从第一行可以看出，冲浪的人和冲浪板很好的被区分了开来，说明这个工作提出的模型能够很好地解决实例重叠的情况。从第二行可以看出，一只新的狐狸进入视频时，它被和其他的狐狸正确的区分了开来，说明这个工作提出的模型能够跟踪实例的移入和移出。从第三行可以看出，两只猴子被很好的区分了开来，说明这个工作提出的模型能够解决实例重叠的情况，即使重叠的实例的类别相同。从第四行可以看出，虽然滑滑板的人的动作幅度很大，他仍然被准确的分割了出来，说明这个工作提出的模型能够很好地解决实例快速运动的情况。

6. 结论

这个工作中，提出了一个单阶段的模型解决视频实例分割任务。它会将视频编码成时空特征金字塔，时空特征金字塔能够充分使用视频中的时空信息。在时空特征金字塔的基础上，进一步引入了通道注意力机制和时空注意力机制。通过实验，这个工作在 Youtube-VIS[31] 上的结果超过了之前的工作的结果。

References

[1] Deep residual learning for image recognition. In IEEE Conference on Computer Vision Pattern Recognition, 2016.

[2] G. Bertasius, L. Torresani, and J. Shi. Object Detection in Video with Spatiotemporal Sampling Networks. Springer, Cham, 2018.

[3] S. Caelles, K. K. Maninis, J. Pont-Tuset, L Leal-Taixé, and L. V. Gool. One-shot video object segmentation. 2016.

[4] J. Dai, K. He, Y. Li, S. Ren, and J. Sun. Instance-sensitive fully convolutional networks. European Conference on Computer Vision, 2016.

[5] Alireza Fathi, Zbigniew Wojna, Vivek Rathod, Peng Wang, Hyun Oh Song, Sergio Guadarrama, and Kevin P. Murphy. Semantic instance segmentation via deep metric learning. 2017.

[6] C. Feichtenhofer, A. Pinz, and A. Zisserman. Detect to track and track to detect. IEEE, 2017.

[7] Jun Fu, Jing Liu, Haijie Tian, Zhiwei Fang, and Hanqing Lu. Dual attention network for scene segmentation. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 3141–3149, 2019.

[8] B. Hariharan, P Arbeláez, R. Girshick, and J. Malik. Simultaneous detection and segmentation. In European Conference on Computer Vision, 2014.

[9] K. He, G. Gkioxari, P Dollár, and R. Girshick. Mask r-cnn. IEEE Transactions on Pattern Analysis Machine Intelligence, 2017.

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

[11] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. IEEE transactions on pattern analysis and machine intelligence, 2019.

[12] S. D. Jain, B. Xiong, and K. Grauman. Fusionseg: Learning to combine motion and appearance for fully automatic segmention of generic objects in videos. IEEE Computer Society, 2017.

[13] Y. Li, H. Qi, J. Dai, X. Ji, and Y. Wei. Fully convolutional instance-aware semantic segmentation. IEEE, 2017.

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

[15] T. Y. Lin, M. Maire, S. Belongie, J. Hays, and C. L. Zitnick. Microsoft coco: Common objects in context. Springer International Publishing, 2014.

[16] J. Luiten, P. Torr, and B. Leibe. Video instance segmentation 2019: A winning approach for combined detection, segmentation, classification and tracking. In 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW), 2019.

[17] D. Munoz, N. Vandapel, and M. Hebert. Onboard contextual classification of 3-d point clouds with learned high-order markov random fields. In Robotics and Automation, 2009. ICRA '09. IEEE International Conference on, 2009.

[18] D. Neven, B De Brabandere, M. Proesmans, and L Van Gool. Instance segmentation by jointly optimizing spatial embeddings and clustering bandwidth. In 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019.

[19] F. Perazzi, J. Pont-Tuset, B. Mcwilliams, L. V. Gool, and A. Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

[20] J Pont-Tuset, F. Perazzi, S. Caelles, P Arbeláez, A. Sorkine-Hornung, and L Van Gool. The 2017 davis challenge on video object segmentation. 2017.

[21] M. Ren and R. S. Zemel. End-to-end instance segmentation with recurrent attention. In Computer Vision Pattern Recognition, 2017.

[22] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. IEEE Transactions on Pattern Analysis Machine Intelligence, 39(6):1137–1149, 2017.

[23] K. Shu and C. Fowlkes. Recurrent pixel embedding for instance grouping. 2017.

[24] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. Computer Science, 2014.

[25] P. Tokmakov, K. Alahari, and C. Schmid. Learning video object segmentation with visual memory. In IEEE Computer Society, 2017.

[26] P. Voigtlaender, Y. Chai, F. Schroff, H. Adam, B. Leibe, and L. C. Chen. Feelvos: Fast end-to-end embedding learning for video object segmentation. IEEE, 2019.

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

[28] Xiaolong Wang, Ross B. Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 7794–7803, 2018.

[29] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In-So Kweon. Cbam: Convolutional block attention module. ArXiv, abs/1807.06521, 2018.

[30] F Xiao and Y. J. Lee. Video object detection with an aligned spatial-temporal memory. arXiv, 2017.

[31] Linjie Yang, Yuchen Fan, and Ning Xu. Video instance segmentation, 2019.

[32] L. Yang, Y. Wang, X. Xiong, J. Yang, and A. K. Katsaggelos. Efficient video object segmentation via network modulation. IEEE, 2018.

[33] Z. Zhang, S. Fidler, and R. Urtasun. Recurrent instance segmentation.

[34] Xizhou Zhu, Yujie Wang, Jifeng Dai, Lu Yuan, and Yichen Wei. Flow-guided feature aggregation for video object detection, 2017.