arXiv:2512.15331v1 [cs.MM] 17 Dec 2025

压缩条件下视频机器视觉的预处理框架

Fei Zhao*[1]、Mengxi Guo†、Shijie Zhao†[2]、Junlin Li†、Li Zhang† 和Xiaodong Xie*

*计算机科学学院，
*北京大学
北京市颐和园路5号
中国北京市，北京，100871
feizhao@stu.pku.edu.cn
donxie@pku.edu.cn

†字节跳动
知春路48号
中国北京市，北京，100098
guomengxi.qoelab@字节跳动.com
zhaoshijie.0526@bytedance.com
lijunlin.li@bytedance.com
lizhang.idm@bytedance.com

摘要

近年来，针对机器视觉任务对终端视频进行压缩和传输的趋势日益增长。然而，大多数视频编码优化方法专注于根据人类感知指标最小化失真，忽视了机器视觉系统提出的更高要求。本文提出了一种面向机器视觉任务的视频预处理框架，以应对这一挑战。所提方法引入了一个能够保留后续任务关键信息的神经网络预处理器，从而提升码率-准确率性能。我们进一步引入可微虚拟编解码器，在训练阶段对码率和失真进行约束。测试时，我们直接采用广泛使用的标准编解码器。因此，我们的方案能够轻松应用于现实场景。我们在采用各种骨干网络的两个典型下游任务上，对所提出的压缩方法进行了广泛实验评估。实验结果表明，与仅使用标准编解码器锚定版本相比，我们的方法可节省超过15%的比特率。

引言

随着深度学习的成功，近年来机器视觉方法取得了显著进展。与此同时，边缘设备的视频采集量激增，采集的视频随后会被传输以用于机器视觉任务。视频压缩优化面临的挑战在于，在减少传输带宽的同时，保持下游机器视觉任务的准确性。

尽管存在 H.264/AVC[1] 和 H.265/HEVC[2], 等多种传统视频压缩标准，但这些标准优先考虑减少人眼可见的压缩失真，而非满足机器视觉任务的需求。此外，大多数标准编解码器都不可微，因此难以与基于学习的模块进行端到端优化。除此之外，压缩操作可能会导致机器视觉任务性能显著下降。通过实验测试，我们观察到机器视觉性能显著下滑，尤其是在低比特率压缩条件下。

针对这些局限性，我们提出了一种面向压缩场景中机器视觉的预处理框架，如图1所示。

[1]字节跳动实习生。 [2]通讯作者。

我们提出了一种基于学习的预处理器，旨在提升压缩条件下下游视觉任务的性能，同时不损害人眼观看时的感知质量。为实现端到端优化，我们进一步设计了一种可微虚拟视频编解码器，用于模拟失真和码率，以提供训练监督，并使梯度能够传播至基于学习的预处理器。在训练过程中，我们利用虚拟编解码器生成率失真损失，而视觉任务分析器则提供准确率损失。这三项损失共同构成预处理器训练期间的优化函数。在测试过程中，我们使用真实编解码器评估经过预处理和压缩的视频的机器视觉性能，并根据比特率和准确率进行评估。我们针对两项重要的机器视觉任务——视频动作识别和视频目标跟踪——开展了大量实验，并采用了多种下游骨干网络。实验结果表明，与 H.264/AVC 和 H.265/HEVC 等传统编解码器相比，我们提出的方法可节省超过 15% 的码率。

我们的工作作出了以下主要贡献：

(1)我们提出了一种深度神经预处理器，用于处理输入视频，并在施加高强度压缩时仍维持较高的下游机器视觉准确率。

(2)我们提出了一种可微虚拟视频编解码器，使预处理器能够在训练过程中感知压缩的影响。

(3)实验结果表明，我们的方法在码率-准确率性能方面优于锚点方法。此外，我们的方法还具有出色的鲁棒性和普适性。

相关工作

视频压缩

视频序列通常包含大量可视为冗余的信息。视频序列中通常可识别出三种主要的冗余：空间冗余、时间冗余和统计冗余。利用这些冗余可以减少表示视频序列所需的数据量，从而提高压缩效率并降低存储需求。现代编码标准采用混合编码方案，基于这些冗余消除技术构建而成，这也是 H.264/AVC 和 H.265/HEVC 等标准编解码器的基本框架。

压缩条件下的机器视觉

机器视觉任务与人类视觉任务在目标和评估指标方面有所不同。为机器处理而编码视频内容，是一项引人入胜且具有挑战性的问题。人工智能驱动的视频智能解决方案应运而生，在应对最紧迫的挑战方面发挥着关键作用。例如，

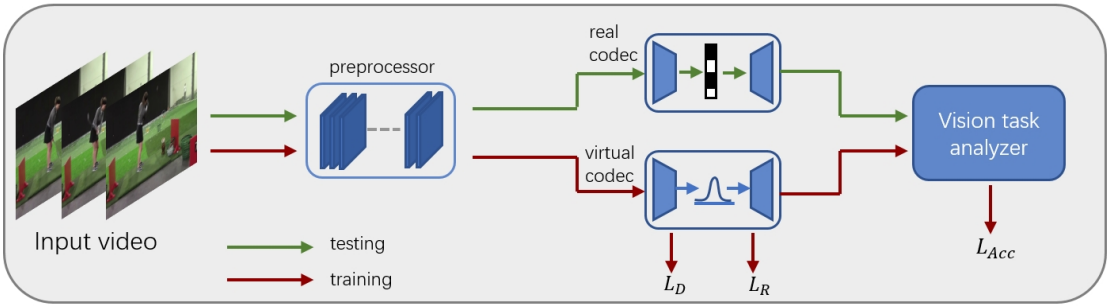


图1：所提出框架的整体处理流程。在训练阶段，我们使用虚拟编解码器生成监督所需的失真损失和码率损失，而准确率损失则由视觉任务生成。在测试阶段，我们使用真实的标准编解码器对经预处理器处理的视频进行编码，以获得比特率。我们还收集下游视觉任务的相应准确率指标，以评估整体性能。

Torfason 等人[3] 提出了一种深度神经压缩框架，可扩展用于联合训练分类和分割任务的网络，从而促进图像理解。Zhu 等人[4] 提出了一种用于视频编码的语义引导纹理合成框架，旨在提高视频中纹理区域的压缩效率。现有的大多数方法依赖基于学习的压缩方法来实现端到端优化，而这在现实应用中可能并不可行。

视频编码预处理

以往研究表明，从视频数据中去除噪声或不重要的细节，可以提升视频编码的感知质量。例如， [5]提出了一种预处理方法，可根据输入视频的内容自适应地调整滤波强度。Talebi 等人[6] 在使用 JPEG 对图像进行编码之前，先通过卷积神经网络对图像进行预处理，以减轻压缩伪影。最近，Chadha 等人[7] 提出了一种基于卷积网络的感知预处理模型，以提升视频编码性能。然而，现有研究主要面向人类视觉系统，对下游机器视觉任务的关注有限。

所提方法

概述

我们提出的方法的处理流程如图1所示。在训练阶段，我们使用预处理器处理输入视频。处理后的视频随后经过我们的虚拟编解码器，生成相应的码率和失真，作为训练监督的一部分。失真由虚拟编解码器生成的重建视频与源输入视频之间的差异决定，而码率则根据重建

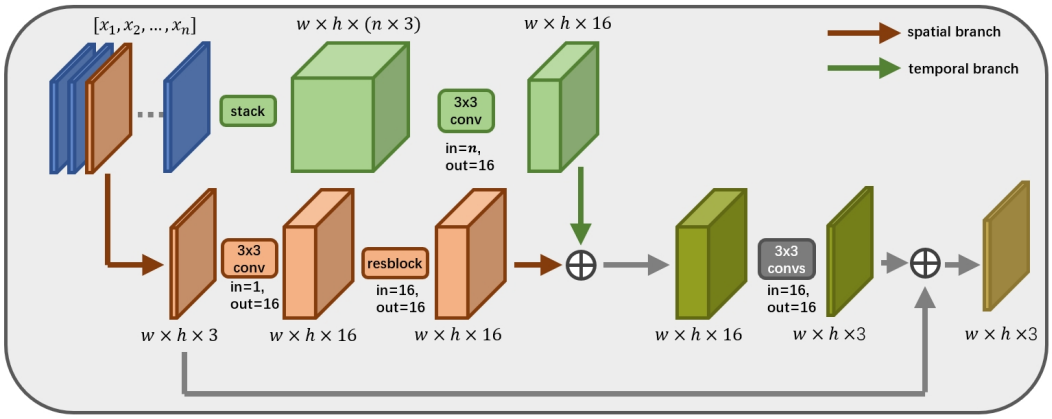


图2：所提出的预处理器结构可分为两个主要分支，分别负责从输入视频中提取时间特征和空间特征，并随后将二者合并。

残差的分布进行估算。虚拟编解码器设计的详细信息将在下一节介绍。虚拟编解码器输出的重建视频作为机器视觉分析器的输入，由此获得准确率指标并将其纳入训练监督。训练阶段的优化函数如下所示，

$$L = \alpha(L_D + \lambda \times L_R) + L_{Acc}, \tag{1}$$

其中， L_D 表示失真损失， L_R 表示码率损失， L_{Acc} 表示与相应下游机器视觉任务相关的准确率损失。请注意，训练期间视觉任务分析器保持固定，因此在每次训练迭代中，我们只更新预处理器的参数。 λ 是用于平衡码率与失真之间权衡的权重系数，而 α 表示压缩相关的率失真损失与机器视觉任务准确率之间的权重系数。对于每个机器视觉网络，我们都单独训练了一个相应的预处理器，用于测试。

测试期间，我们应用训练好的预处理器处理源视频。随后，使用实际的标准编解码器对处理后的视频进行压缩，从而得到相应的失真和码率值。然后，我们使用压缩视频执行下游机器视觉任务，并评估其性能。

预处理网络

预处理模型的目标有两个：首先，感知编码失真对整个处理流程的影响；其次，辨别在这种条件下下游机器视觉任务所需特征的特性。面对需要多帧输入的下游视频机器视觉任务，我们的预处理器必须有效利用帧间时序信息进行视频处理。此外，由于编码器会消除视频中的时序

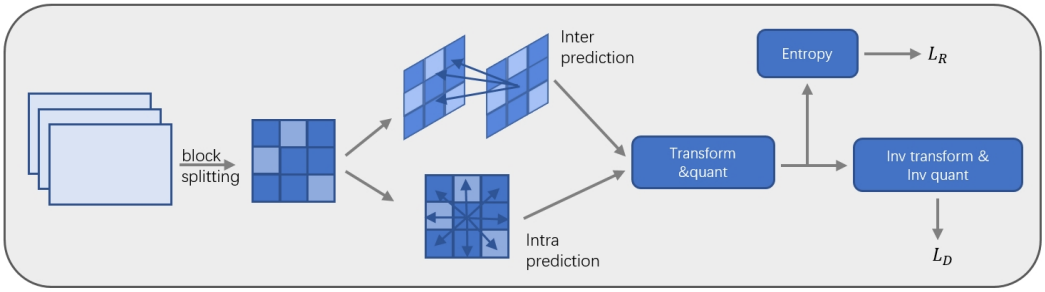


图3：所提出的虚拟编解码器模拟了视频编码所采用的基本逻辑，并通过张量运算实现。在训练阶段，虚拟编解码器负责提供码率损失和失真损失。

和空间冗余，预处理器必须对视频的帧间时序特征和帧内空间特征足够敏感。

如图2所示，我们采用具有多分支残差连接的卷积神经网络来构建预处理模型。具体而言，在时序分支中，我们沿帧间维度执行卷积，生成包含时序信息的特征图。在空间分支中，我们仅对各帧内部进行卷积，生成主要包含空间信息的特征图。随后，我们通过条件注意力融合这两组特征图，并利用残差连接将其融入原始图像，从而得到最终输出。

虚拟视频编解码器

由于传统视频编解码器具有不可微分性，因此无法将其集成到基于神经网络训练的联合优化方案中。为解决这一限制，我们在训练过程中采用虚拟编解码器结构代替实际编解码器。此结构旨在对失真和码率施加约束，从而构成整个处理流程优化函数的一部分。

虚拟视频编解码器主要通过生成类似于视频编解码器中残差图的残差图来产生失真和码率。我们按照图3所示对虚拟编解码器进行建模。通过帧内搜索和帧间搜索（以前一帧作为参考帧），我们得到当前帧的预测残差。随后，依次进行变换、量化、反量化和逆变换，以获得重建残差。这里，我们采用量化因子 f_q，该因子是代表量化参数（QP）的等效参数，用于调节量化强度。利用重建残差即可还原重建帧，从而获得相应的失真。在码率计算方面，我们采用 Balle 等人提出的因子化先验熵方法。[8]

表 1：动作识别的 BD-Rate(%) 性能

		骨干网络					
编解码器	指标	Slow-only [10]	SlowFast [10]	C2D [11]	I3D [12]	Swin [13]	TPN [14]
H.264	Top-1 准确率	-17.6	-12.3	-18.1	-19.0	-16.4	-18.8
	Top-5 准确率	-17.1	-12.6	-18.5	-18.9	-16.8	-19.5
	平均准确率	-17.2	-13.0	-18.8	-19.2	-16.6	-19.6
	VMAF	-5.6					
H.265	Top-1 准确率	-16.2	-11.7	-16.9	-16.6	-15.4	-18.1
	Top-5 准确率	-16.2	-11.5	-16.5	-16.9	-15.8	-18.3
	平均准确率	-16.0	-11.8	-16.8	-17.1	-15.6	-18.4
	VMAF	-5.1					

实验

实现

对于动作识别任务，我们的框架在 Kinetics400[9] 数据集上进行训练和测试。我们在各种压缩设置下评估所提出框架的性能，并报告 Top-1 至 Top-5 准确率。在实验中，我们采用六种广受认可的动作识别网络进行评估：Slow- only[10], SlowFast[10], C2D[11], I3D[12], Swin[13]和 TPN[14] 。对于视频目标跟踪任务，我们采用 GOT-10K[15] 数据集。实验结果以 ROC曲线下面积（AUC）表示。为展示我们方法的有效性，我们采用骨干网络 KYS[16],DiMP[17],ATOM[18] 和 PrDiMP[19] 进行性能评估。

在我们的实验中，我们使用每像素比特数（bpp）来衡量压缩过程中产生的编码开销。测试时，我们采用了广泛使用的 H.264 和 H.265 开源实现，即 x264[20] 和 x265[21], ，作为标准编解码器。测试的编码预设设为“medium”，量化参数为 QP=[30,35,40,45,50]。损失函数 (1) 中的失真损失通过均方误差（MSE）计算，而码率损失则通过计算估计的 bpp 得出。损失函数 α 和 λ中的权重参数分别设为 10 和 0.001。

我们的整个框架使用 PyTorch[22] 开发，支持 CUDA，并在 A100 GPU 上训练。虚拟编解码器的量化因子 f_q 设为在 30 到 50 之间随机选取。下游网络的权重在整个训练过程中保持不变。由于我们只需训练完整框架中的预处理网络，因此训练流程只需要进行一次端到端训练迭代。我们采用 Adam优化方法，初始学习率设为 1e-4。

实验结果

我们对增强预处理的视频压缩方法与传统编解码器 H.264 和 H.265 进行了比较分析。此外，我们

表 2：目标跟踪的 BD-Rate(%) 性能

		骨干网络			
编解码器	指标	KYS[16]	DiMP[17]	ATOM[18]	PrDiMP[19]
H.264	AUC	-15.5	-15.2	-17.2	-12.4
	VMAF	-6.3			
H.265	AUC	-13.6	-12.4	-16.6	-12.0
	VMAF	-5.7			

采用 BD-Rate 指标量化在保持相同准确率水平的同时所节省的比特率百分比。视频动作识别和视频目标跟踪的默认骨干网络分别列于表 1 和表 2 中，我们也据此训练了相应的预处理模块。

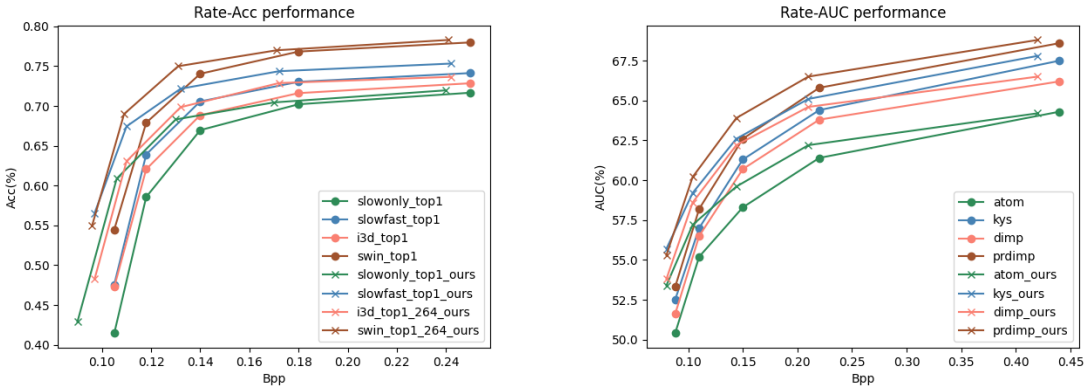
显然，与基线方法相比，我们的预处理增强型视频压缩方法在下游动作识别任务中显著改善了压缩率与准确率之间的权衡。如表1所示，在视频动作识别任务中，我们在所有四种骨干网络上都显著提升了 BD-Rate 性能。值得注意的是，在 H.264 上，我们取得了最高达 -17.6% 的提升；在 H.265 上，最高提升达 -16.2%。我们还针对视频目标跟踪任务，将我们的方法与传统编解码器进行了比较分析。表2展示了使用 GOT-10k 数据集，通过不同压缩方法得到的码率-准确率曲线。值得强调的是，与传统编解码器基线相比，我们的方法具有更优的码率-准确率性能，码率最高可节省 17.5%。相应的码率-准确率曲线如图4所示。

还值得注意的是，应用我们提出的预处理器后，输出能够与原始输入保持高度相似。此外，即使经过压缩，我们在视频多方法评估融合[23], 等感知导向指标上也能取得提升，而视频多方法评估融合以人类视觉系统为导向。这一点在表1和表2中均有体现。

进一步分析

为验证所提出的预处理模块的有效性，我们进行了额外测试。在这些测试中，我们直接使用视频编解码器生成的压缩数据，对下游任务中的分析器进行微调。我们采用了与所提出方法相同的训练策略和锚点。随后，将训练好的分析器与我们提出的方案进行比较。实验结果如表3所示，表明使用压缩数据进行微调确实能在一定程度上提升分析器在压缩数据上的性能。然而，与我们提出的方案相比，其表现仍明显落后。

如图5所示，我们提供了一些可视化结果。如(a)中的目标跟踪结果所示，即使比特率大幅降低，我们提出的方法仍能成功跟踪目标，而作为对比的锚点



(a) (b)

图4：视频动作识别(a)和视频目标跟踪(b)测试结果的率-准确率示意图。

表3：微调版本的 BD-Rate(%) 性能比较

	骨干网络			
	Slow-only[10]		SlowFast[10]	
指标	微调	ours	微调	ours
Top-1 准确率	-5.1	-17.6	-5.5	-12.3
Top-5 准确率	-5.6	-17.1	-5.8	-12.6

方法则丢失了目标。在(b)中的动作识别结果中，即使比特率大幅降低，我们提出的方法仍能正确识别。相比之下，锚点方法在比特率降低后出现了严重的识别错误。

结论

在视频机器视觉领域，本文提出了一种视频预处理框架，旨在提升压缩条件下下游视觉任务的性能。我们提出一种基于神经网络的预处理器，用于处理输入视频，使处理后的视频能够更好地兼顾视频压缩与机器视觉任务。此外，我们引入一种虚拟视频编解码器，以便进行端到端训练优化。实验结果表明，与仅使用标准编解码器相比，我们提出的方法可显著节省码率。

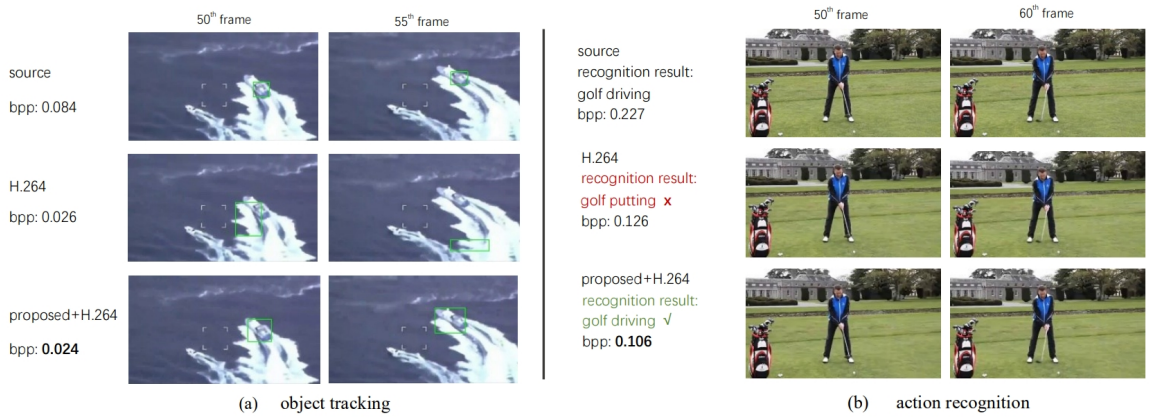


图5：所提方法输出结果的可视化示例。(a) 展示视频目标跟踪的样例结果，图中帧取自 GOT-10k 数据集中的 Test000004。(b) 展示视频动作识别的样例结果，图像取自 Kinetics400 数据集中的测试序列 0u4c8Cel91U。

参考文献

[1] Thomas Wiegand、Gary J Sullivan、Gisle Bjontegaard 和 Ajay Luthra，《H.264/AVC视频编码标准概述》，*IEEE*视频技术电路与系统汇刊，第13卷，第7期，第560–576页，2003年。[2] Gary J Sullivan、Jens-Rainer Ohm、Woo-Jin Han 和 Thomas Wiegand，《高效视频编码（HEVC）标准概述》，*IEEE*视频技术电路与系统汇刊，第22卷，第12期，第1649–1668页，2012年。[3] Robert Torfason、Fabian Mentzer、Eirikur Agustsson、Michael Tschannen、Radu Timo- fte 和 Luc Van Gool，《无需解码、通过深度压缩实现图像理解》，*arXiv*预印本 *arXiv:1803.06131*，2018年。[4] Chen Zhu、Guo Lu、Rong Xie 和 Li Song，《基于语义引导纹理检测与合成的感知视频编码》，载于*2022*年图像编码研讨会（*PCS*）。IEEE，2022年，第217–221页。[5] Guoqing Xiang、Huizhu Jia、Jie Liu、Binbin Cai、Yuan Li 和 Xiaodong Xie，《用于视频编码的自适应感知预处理》，载于*2016*年*IEEE*国际电路与系统研讨会（*ISCAS*）。IEEE，2016年，第2535–2538页。[6] Hossein Talebi、Damien Kelly、Xiyang Luo、Ignacio Garcia Dorado、Feng Yang、Pey- man Milanfar 和 Michael Elad，《通过深度预编辑实现更好的压缩》，*IEEE*图像处理汇刊，第30卷，第6673–6685页，2021年。[7] Aaron Chadha 和 Yiannis Andreopoulos，《用于视频编码的深度感知预处理》，载于*IEEE/CVF*计算机视觉与模式识别会议论文集，2021年，第14852–14861页。[8] Johannes Ball´e、Valero Laparra 和 Eero P Simoncelli，《端到端优化图像压缩》，*arXiv*预印本 *arXiv:1611.01704*，2016年。[9] Will Kay、Joao Carreira、Karen Simonyan、Brian Zhang、Chloe Hillier、Sudheendra Vijayanarasimhan、Fabio Viola、Tim Green、Trevor Back、Paul Natsev 等，《Kinetics人体动作视频数据集》，*arXiv*预印本 *arXiv:1705.06950*，2017年。

[10] Christoph Feichtenhofer、Haoqi Fan、Jitendra Malik 和 Kaiming He，“用于视频识别的 SlowFast 网络”，载于*IEEE/CVF*国际计算机视觉会议论文集，2019年，第6202–6211页。[11] Xiaolong Wang、Ross Girshick、Abhinav Gupta 和 Kaiming He，“非局部神经网络”，载于*IEEE*计算机视觉与模式识别会议论文集，2018年，第7794–7803页。[12] Joao Carreira 和 Andrew Zisserman，“Quo vadis，动作识别？一种新模型与 Kinetics 数据集”，载于*IEEE*计算机视觉与模式识别会议论文集，2017年，第6299–6308页。[13] Ze Liu、Jia Ning、Yue Cao、Yixuan Wei、Zheng Zhang、Stephen Lin 和 Han Hu，“视频Swin Transformer”，载于*IEEE/CVF*计算机视觉与模式识别会议论文集，2022年，第3202–3211页。[14] Ceyuan Yang、Yinghao Xu、Jianping Shi、Bo Dai 和 Bolei Zhou，“用于动作识别的时序金字塔网络”，载于*IEEE/CVF*计算机视觉与模式识别会议论文集，2020年，第591–600页。[15] Lianghua Huang、Xin Zhao 和 Kaiqi Huang，“GOT-10k：用于野外通用目标跟踪的大规模高多样性基准”，*IEEE*模式分析与机器智能汇刊，第43卷，第5期，第1562–1577页，2019年。[16] Goutam Bhat、Martin Danelljan、Luc Van Gool 和 Radu Timofte，“了解周围环境：利用场景信息进行目标跟踪”，载于《计算机视觉——*ECCV 2020*：第*16*届欧洲会议》，英国格拉斯哥，*2020*年*8*月*23*日至*28*日，论文集，第*XXIII*卷第*16*部分。Springer，2020年，第205–221页。[17] Goutam Bhat、Martin Danelljan、Luc Van Gool 和 Radu Timofte，“学习用于跟踪的判别式模型预测”，载于*IEEE/CVF*国际计算机视觉会议论文集，2019年，第6182–6191页。[18] Martin Danelljan、Goutam Bhat、Fahad Shahbaz Khan 和 Michael Felsberg，“ATOM：通过重叠最大化实现精准跟踪”，载于*IEEE/CVF*计算机视觉与模式识别会议论文集，2019年，第4660–4669页。[19] Martin Danelljan、Luc Van Gool 和 Radu Timofte，“用于视觉跟踪的概率回归”，载于*IEEE/CVF*计算机视觉与模式识别会议论文集，2020年，第7183–7192页。[20] VideoLan，“x264”，https://www.videolan.org/developers/x264.html。[21] VideoLan，“x265”，https://www.videolan.org/developers/x265.html。[22] Sagar Imambi、Kolla Bhanu Prakash 和 GR Kanagachidambaresan，“PyTorch”，*TensorFlow*编程：边缘计算应用解决方案，第87–104页，2021年。[23] Zhi Li、Christos Bampis、Julie Novak、Anne Aaron、Kyle Swanson、Anush Moorthy 和 JD Cock，“VMAF：旅程仍在继续”，*Netflix*技术博客，第25卷，2018年。