综述｜视觉Transformer在CV中的现状、趋势和未来方向

极市平台 2022-11-08 22:00:53 发表于广东 手机阅读

以下文章来源于自动驾驶之心，作者汽车人

作者丨汽车人

来源丨自动驾驶之心

编辑丨极市平台

极市导读

本综述根据三个基本的CV任务和不同的数据流类型，全面调查了100多种不同的视觉Transformer，并提出了一种分类法，根据其动机、结构和应用场景来组织代表性方法。由于它们在训练设置和专用视觉任务上的差异，论文还评估并比较了不同配置下的所有现有视觉Transformer。 >>加入极市CV技术交流群，走在计算机视觉的最前沿

摘要

Transformer，一种基于注意力的编码器-解码器模型，已经彻底改变了自然语言处理（NLP）领域。受这些重大成就的启发，最近在计算机视觉（CV）领域采用类似Transformer的架构进行了一些开创性的工作，这些工作证明了它们在三个基本CV任务（分类、检测和分割）以及多传感器数据（图像、点云和视觉-语言数据）上的有效性。由于其具有竞争力的建模能力，与现代卷积神经网络（CNN）相比，视觉Transformer在多个基准测试中取得了令人印象深刻的性能改进。

本综述根据三个基本的CV任务和不同的数据流类型，全面调查了100多种不同的视觉Transformer，并提出了一种分类法，根据其动机、结构和应用场景来组织代表性方法。由于它们在训练设置和专用视觉任务上的差异，论文还评估并比较了不同配置下的所有现有视觉Transformer。

此外，论文还揭示了一系列重要但尚未开发的方面，这些方面可能使此类视觉Transformer能够从众多架构中脱颖而出，例如，松散的高级语义嵌入，以弥合视觉Transformer与序列式之间的差距。最后，提出了未来有前景的研究方向。仓库地址：https://github.com/liuyang-ict/awesome-visual-transformers

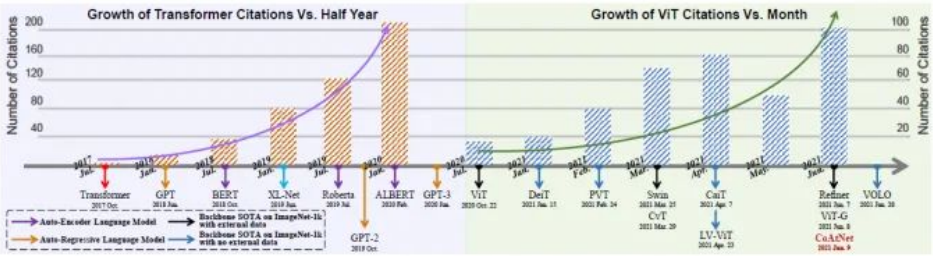
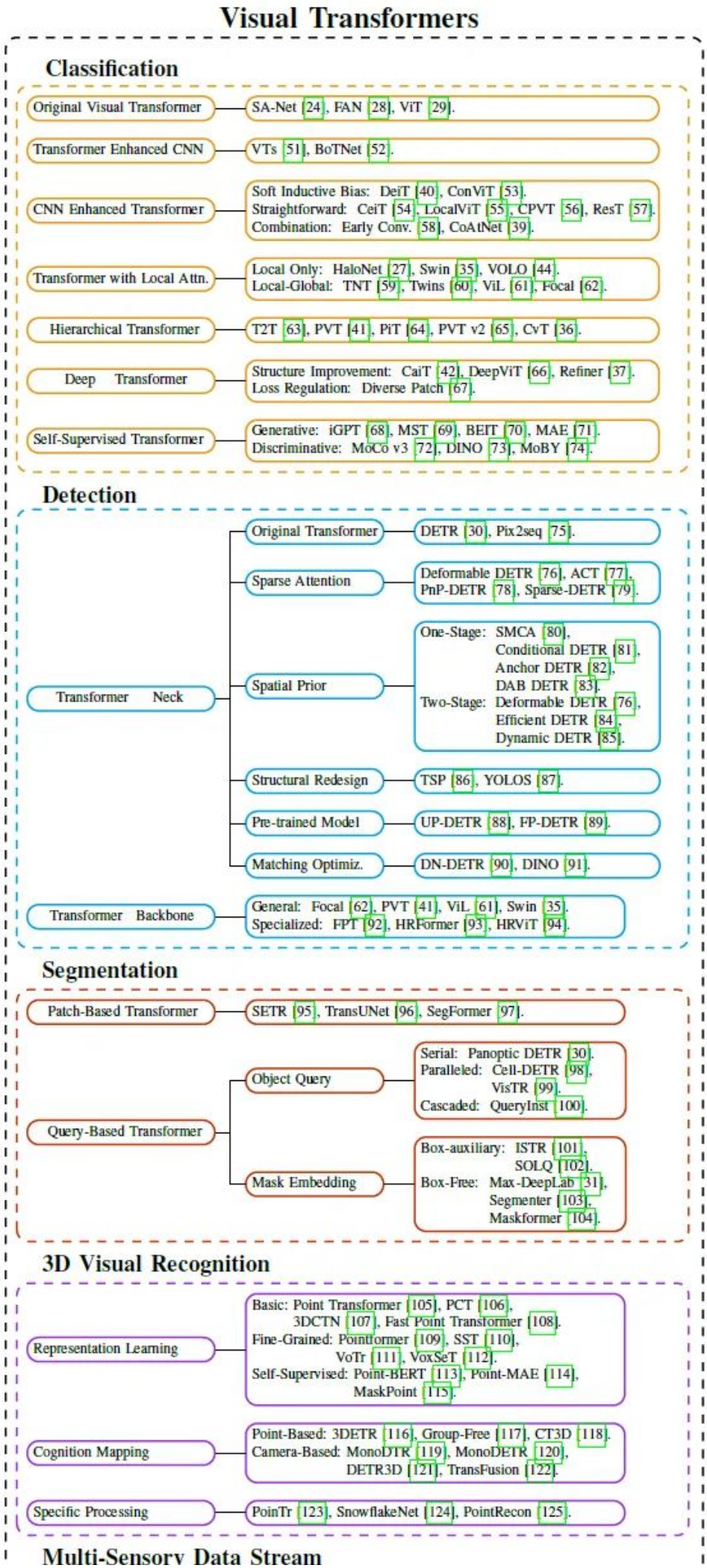


Fig. 1. Odyssey of Transformer application & Growth of both Transformer [1] and ViT [29] citations according to Google Scholar. (Upper Left) Growth of Transformer citations in multiple conference publication including: NIPS, ACL, ICML, IJCAI, ICLR, and ICASSP. (Upper Right) Growth of ViT citations in Arxiv publications. (Bottom Left) Odyssey of language model [1]–[8]. (Bottom Right) Odyssey of visual Transformer backbone where the black [29], [35]–[39] is the SOTA with external data and the blue [40]–[44] refers to the SOTA without external data (best viewed in color).

本文旨在对最新的视觉Transformer进行更全面的回顾，并对其进行系统分类：

- **全面性和可读性：** 本文根据它们在三个基本CV任务（即分类、检测和分割）和数据流类型（即图像、点云、多流数据）上的应用，全面回顾了100多个视觉Transformer。论文选择了更具代表性的方法，并进行了详细的描述和分析，但简要介绍了其他相关工作。本文不仅从一个角度对每个模型进行了详尽的分析，而且还从某种意义上建立了它们的内部联系，如渐进、对比和多视角分析。
- **直观的比较：** 由于现有的视觉Transformer针对各种视觉任务遵循不同的训练方案和超参数设置，本文对不同的数据集和限制进行了多次横向比较。更重要的是，总结了为每个任务设计的一系列有效组件，包括：（a）具有层次结构的浅局部卷积；（b） neck detector的稀疏注意力空间先验加速；（c）以及用于分割的通用掩模预测方案；
- **深入分析：** 论文进一步深入分析了以下几个方面：（a）从传统序列任务到视觉任务的转换过程；（b） 视觉Transformer和其他神经网络之间的对应关系；（c）以及不同任务和数据流类型中使用的可学习嵌入（即class token、object query、mask embedding）的相关性。最后，论文概述了一些未来的研究方向。例如，编码器-解码器Transformer主干可以通过query embedding来统一多个视觉任务和数据流类型。


Visual Transformers
Classification
Original Visual Transformer
SA-Net [24], FAN [28], ViT [29].
Transformer Enhanced CNN
VTs [51], BoTNet [52].
CNN Enhanced Transformer
Soft Inductive Bias: DeiT [40], ConViT [53].
Straightforward: CeiT [54], LocalViT [55], CPVT [56], ResT [57].
Combination: Early Conv. [58], CoAtNet [39].
Transformer with Local Attn.
Local Only: HaloNet [27], Swin [35], VOLO [44].
Local-Global: TNT [59], Twins [60], ViL [61], Focal [62].
Hierarchical Transformer
T2T [63], PVT [41], PiT [64], PVT v2 [65], CvT [36].
Deep Transformer
Structure Improvement: CaiT [42], DeepViT [66], Refiner [37].
Loss Regulation: Diverse Patch [67].
Self-Supervised Transformer
Generative: iGPT [68], MST [69], BEIT [70], MAE [71].
Discriminative: MoCo v3 [72], DINO [73], MoBY [74].
Detection
Transformer Neck
Original Transformer
DETR [30], Pix2seq [75].
Sparse Attention
Deformable DETR [76], ACT [77],
PnP-DETR [78], Sparse-DETR [79].
Spatial Prior
One-Stage: SMCA [80],
Conditional DETR [81],
Anchor DETR [82],
DAB DETR [83].
Two-Stage: Deformable DETR [76],
Efficient DETR [84],
Dynamic DETR [85].
Structural Redesign
TSP [86], YOLOS [87].
Pre-trained Model
UP-DETR [88], FP-DETR [89].
Matching Optimiz.
DN-DETR [90], DINO [91].
Transformer Backbone
General: Focal [62], PVT [41], ViL [61], Swin [35].
Specialized: FPT [92], HRFormer [93], HRViT [94].
Segmentation
Patch-Based Transformer
SETR [95], TransUNet [96], SegFormer [97].
Query-Based Transformer
Object Query
Serial: Panoptic DETR [30].
Paralleled: Cell-DETR [98],
VisTR [99].
Cascaded: QueryInst [100].
Mask Embedding
Box-auxiliary: ISTR [101],
SOLQ [102].
Box-Free: Max-DeepLab [31],
Segmenter [103],
Maskformer [104].
3D Visual Recognition
Representation Learning
Basic: Point Transformer [105], PCT [106],
3DCTN [107], Fast Point Transformer [108].
Fine-Grained: Pointformer [109], SST [110],
VoTr [111], VoxSeT [112].
Self-Supervised: Point-BERT [113], Point-MAE [114],
MaskPoint [115].
Cognition Mapping
Point-Based: 3DETR [116], Group-Free [117], CT3D [118].
Camera-Based: MonoDTR [119], MonoDETR [120],
DETR3D [121], TransFusion [122].
Specific Processing
PoinTr [123], SnowflakeNet [124], PointRecon [125].
Multi-Sensory Data Stream

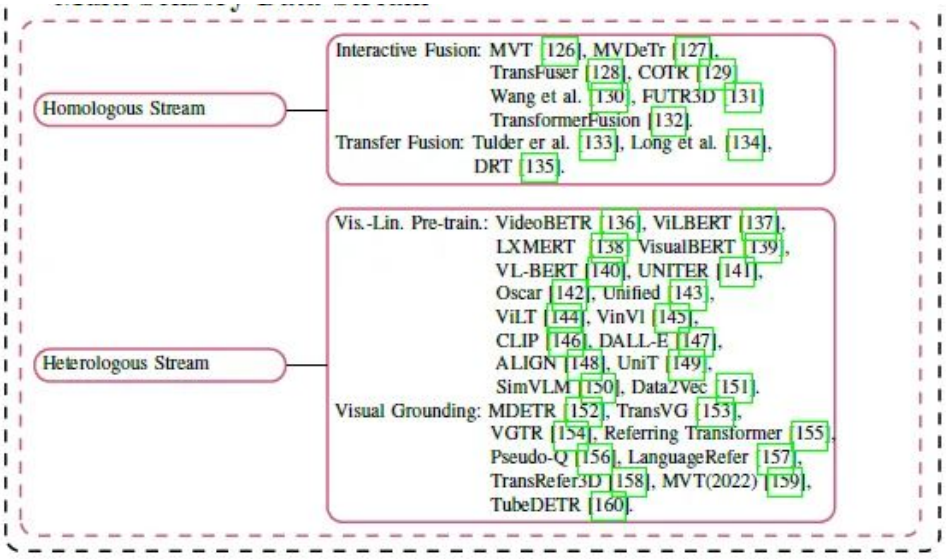


Fig. 3. Taxonomy of Visual Transformers

原始Transformer

最初的Transformer[1]首先应用于序列到序列自动回归的任务。与先前的序列转导模型[49]、[50]相比，这种原始的Transformer继承了编码器-解码器结构，但通过使用multi-head attention机制和point-wise feed-forward网络，完全放弃了递归和卷积。图4展示了带有编码器-解码器架构的整体Transformer模型。具体而言，它由N个连续的编码器模块组成，每个编码器由两个子层组成。1） MHSA层聚合编码器嵌入内的关系；2） 逐位置FFN层提取特征表示。

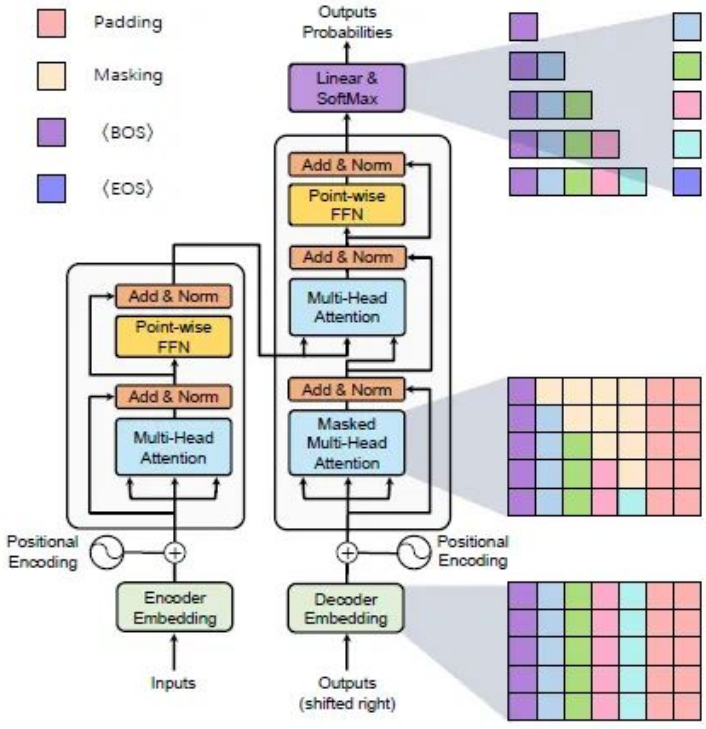


Fig. 4. The overall architecture of Transformer [1] that follows an encoder-decoder structure. The 2D lattice visualizes the states of each part of the decoder during training (best viewed in color).

在自然语言回归模型中，Transformer源于机器翻译任务。给定一个单词序列，Transformer将输入序列矢量化为单词嵌入，添加位置编码，并将生成的向量序列输入编码器。在训练期间，如图4所示，Vaswani等人根据自回归任务的规则设计了masking操作，其中当前位置仅取决于先前位置的输出。基于这种masking，Transformer解码器能够并行处理输入标签的序列。在推理期间，通过相同的操作处理先前预测的单词序列以预测下一个单词。

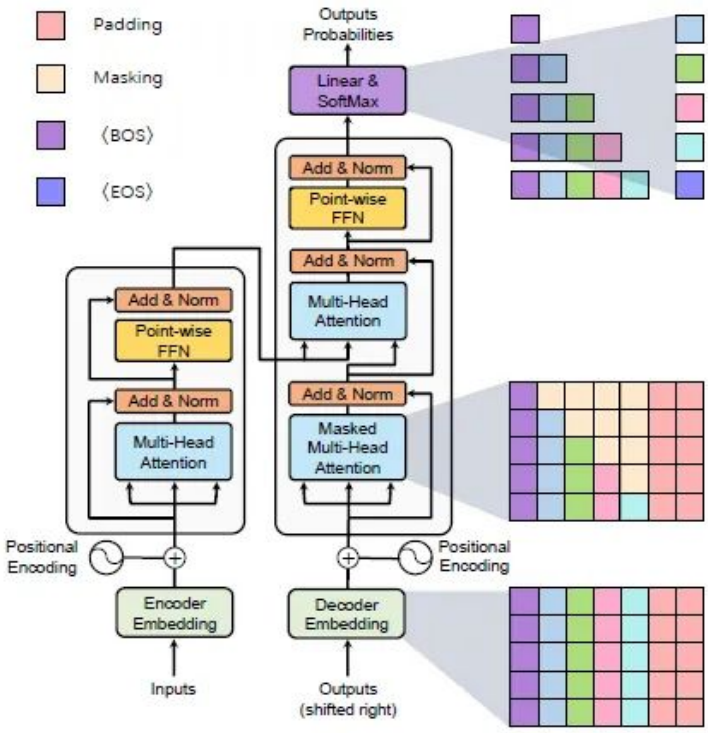


Fig. 4. The overall architecture of Transformer [1] that follows an encoder-decoder structure. The 2D lattice visualizes the states of each part of the decoder during training (best viewed in color).

分类Transformer

随着NLP中Transformer的显著发展[2]–[5]，最近的工作试图引入视觉Transformer来进行图像分类。本节全面回顾了40多个视觉Transformer，并将其分为六类，如图5所示。

首先介绍了Fully-Attentional网络[24]、[28]和Vision Transformer（ViT）[29]，这种原始ViT首先证明了其在多个分类基准上的功效。然后讨论了Transformer增强的CNN方法，该方法利用Transformer来增强CNN的表示学习。由于忽略了原始ViT中的局部信息，CNN增强型Transformer采用了适当的卷积inductive bias来增强ViT，而局部注意力增强型Transformer重新设计了patch分区和注意力块，以提高其局部性。

继CNN[162]中的分层和深层结构之后，分层Transformer用金字塔代替了固定分辨率的柱状结构，而Deep Transformer防止了注意力图过于平滑，并增加了其在深层中的多样性。此外，论文还回顾了现有的基于自监督学习的ViT。

最后，本文根据直观的比较进行了简短的讨论，组织了一个ViT的里程碑，并讨论了一个共同的问题以供进一步研究。

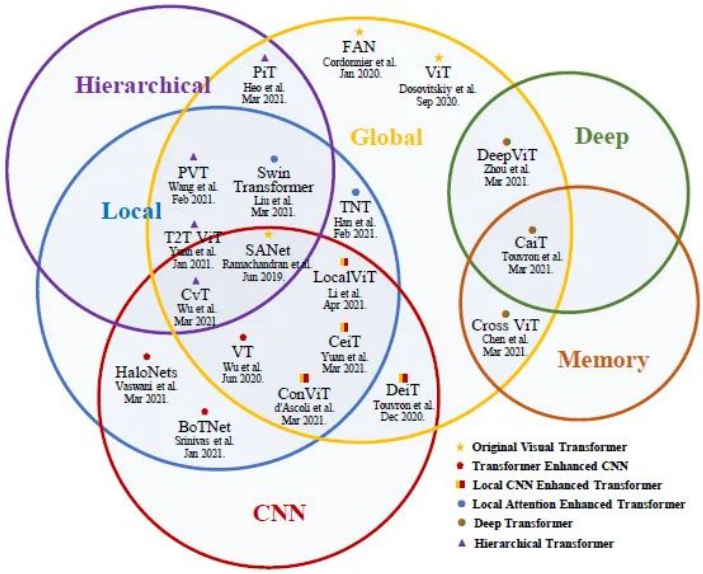


Fig. 5. Taxonomy of Visual Transformer Backbone (best viewed in color).

Original Visual Transformer

受Transformer在NLP领域取得的巨大成就的启发[2]-[5]，先前视觉任务的技术趋势[14]-[17]，[163]将注意力机制与卷积模型相结合，以增强模型的感受野和全局依赖性。除了这种混合模型，Ramachandran等人考虑了注意力是否可以完全取代卷积，然后提出了一个Stand-Alone自注意力网络（SANet）[24]，与原始基线相比，该网络在视觉任务上取得了优异的性能。给定ResNet[11]架构，作者直接将每个bottleneck中的空间卷积层（3*3）替换为局部空间自注意力层，并保持其他结构与ResNet中的原始设置相同。此外，大量消融已经表明，位置编码和卷积可以进一步提高网络效率。

继[24]之后，Cordonnier等人设计了一个原型（称为“Fully-Attentional Network”）[28]，包括一个fully vanilla Transformer和一个二次位置编码。作者还从理论上证明了卷积层可以用具有相对位置编码和足够head的单个MHSA层来近似。通过在CIFAR-10上的消融实验[164]，他们进一步验证了这样的原型设计确实能够学习到每个query像素周围的网格状图案，这是他们的理论结论。与[28]只关注小尺度模型不同，ViT[29]通过大规模预训练学习进一步探索了vanilla Transformer的有效性，这样的先锋工作对社区产生了重大影响。因为vanilla Transformer只接受序列输入，ViT中的输入图像首先被拆分成一系列不重叠的patch，然后被投影到patch嵌入中。将一维可学习位置编码添加到patch embeddings上以保留空间信息，然后将joint embeddings馈送到编码器中，如图6所示。

与BERT[5]类似，将学习的 [class] token与patch embeddings附加在一起，以聚合全局表示，并将其用作分类的输入。此外，2D插值补充了预训练的位置编码，以在馈送图像是任意分辨率时保持patche的一致顺序。通过使用大规模私有数据集（JFT-300M[165]）进行预训练，与最流行的CNN方法相比，ViT在多个图像识别基准（ImageNet[166]和CIFAR-100[164]）上取得了相似甚至更好的结果。然而，它的泛化能力往往会受到有限训练数据的侵蚀。

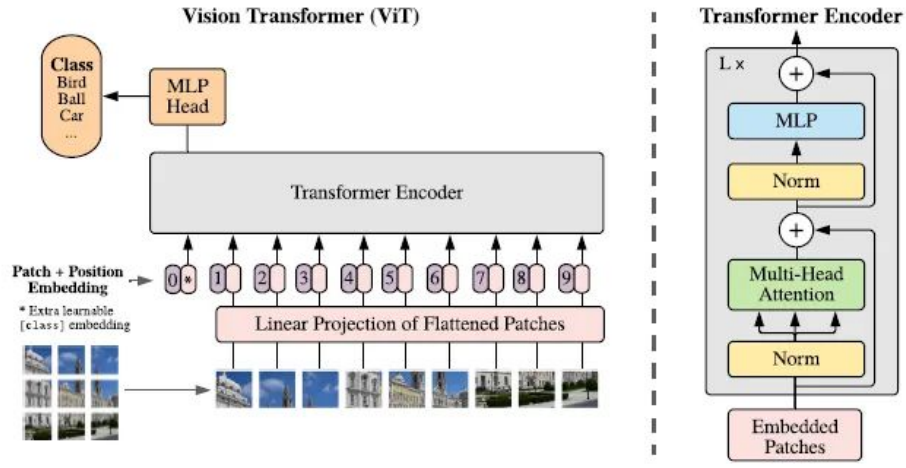


Fig. 6. Illustration of ViT. The flatten image patches with an additional class token are fed into the vanilla Transformer encoder after positional encoding. Only the class token can be predicted for classification. (from [29].)

Transformer Enhanced CNNs

如上所述Transformer有两个关键：MHSA和FFN。卷积层和MHSA之间存在近似值[28]，Dong等人认为，Transformer可以借助跳跃连接和FFN[167]进一步减轻MHSA的强偏置。最近，一些方法试图将Transformer集成到CNN中以增强表示学习。VTs[51]将输入图像的语义概念解耦到不同的通道中，并通过编码器块（即VT块）将它们紧密关联。这种VT块替代了最后的卷积阶段，以增强CNN模型的语义建模能力。与先前直接用注意力结构代替卷积的方法不同，Vaswani等人提出了一种概念上的重新定义，即具有MHSA的连续bottleneck block可以被表述为Bottleneck Transformer（BoTNet）[52]块。采用相对位置编码[168]进一步模拟原始Transformer.。基于ResNet[11]，BoTNet在ImageNet基准上的参数设置类似，优于大多数CNN模型，并进一步证明了混合模型的有效性。

CNN Enhanced Transformer

Inductive bias被定义为关于数据分布和解空间的一组假设，其在卷积中的表现为局部性和平移不变性[169]。由于局部邻域内的协方差很大，并且在图像中逐渐趋于平稳，CNN可以在偏差的帮助下有效地处理图像。然而，当有足够的数据可用时，强偏差也限制了CNN的上限。最近的努力试图利用适当的CNN bias来增强Transformer。相关算法有DeiT[40]、ConViT[53]、CeiT[54]、LocalViT[55]、ResT[57]、CPVT[56]、CvT[36]、CoAtNet)[39]等。

Local Attention Enhanced Transformer

ViT[29]中的coarse patchify过程忽略了局部图像信息。除了卷积，研究人员提出了一种局部注意力机制，以动态关注相邻元素并增强局部提取能力。代表性方法之一是Swin Transformer[35]。类似于TSM[173]（图7（a）），Swin利用沿空间维度的移位窗口来建模全局和边界特征。具体而言，两个连续的window-wise attention可以促进cross-window相互作用（图7（b）-（c）），类似于CNN中的感受野扩展。这种操作将计算量由降低至。其他相关算法TNT[59]、Twins[60]、ViL[61]、VOLO[44]可以参考具体论文。

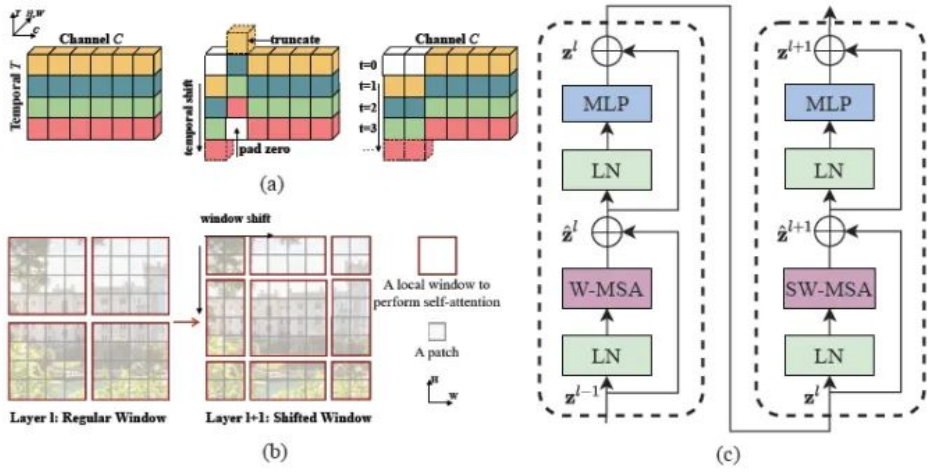


Fig. 7. An overview of Swin Transformer and TSM. (a) TSM with bi-direction and uni-direction operation. (b) The shifted window method. (c) Two successive Transformer blocks of Swin Transformer. The regular and shifted window correspond to W-MSA and SW-MSA, respectively. (from [35], [173]).

Hierarchical Transformer

由于ViT[29]在整个Transformer层中采用具有固定分辨率的柱状结构，忽略了细粒度特征，并带来了沉重的计算成本。继分层模型之后，Tokens to Token ViT（T2T-ViT）首先引入了分层Transformer的范例，并采用overlapping unfold操作进行下采样。然而，这种操作带来了沉重的内存和计算成本。因此，Pyramid Vision Transformer（PVT）[41]利用非重叠的patch分区来减少特征大小。此外，PVT中的spatial-reduction attention（SRA）层被应用于通过学习低分辨率key-value pairs来进一步降低计算成本。在经验上，PVT使Transformer适应许多基准上的密集预测任务，这些基准需要大量输入和细粒度特征，并且具有计算效率。此外，PiT[64]和CvT[36]都分别利用池化和卷积来进行token下采样。具体而言，CvT[36]通过用卷积投影替换线性层来改进PVT[41]的SRA。基于convolutional bias，CvT[36]可以适应任意大小的输入，而无需位置编码。

Deep Transformer

经验上，增加模型的深度是可以增强其学习能力[11]的。最近的工作将深度结构应用于Transformer，并进行了大量实验，通过分析cross-patch[67]和cross-layer[37]、[66]的相似性以及残差的贡献[42]来研究其可扩展性。在Deep Transformer中，来自较深层的特征往往不太具有代表性（attention collapse[66]），并且patch被映射到不可区分的潜在表示中（patch over-smoothing[67]）。为了解决上述限制，这些方法从两个方面提出了相应的解决方案。从模型结构的角度来看，Touvron等人在图像Transformers中提出了有效的Class-attention（CaiT[42]），包括两个阶段：1）没有class token的多个self-attention阶段。在每一层中，利用由small values初始化的可学习对角矩阵来动态更新channel权重，从而为channel调整提供一定的自由度；2） 最后class-attention阶段是冻结patch embeddings。之后的class token被插入到模型全局表示中，类似于具有编码器-解码器结构的DETR。这种显式分离基于这样一个假设，即class token对于前向传递中的patch embeddings梯度无效。通过蒸馏训练策略[40]，CaiT在没有外部数据的情况下在imagenet-1k上实现了新的SOTA（86.5%的TOP1精度）。Deep Transformer遭受attention collapse和过度平滑问题的困扰，但仍在很大程度上保留了不同head之间注意力图的多样性。基于这一观察，Zhou等人提出了Deep Vision Transformer（DeepViT）[66]，该Transformer聚合cross-head attention maps，并通过使用线性层重新生成新的注意力图，以增加跨层特征多样性。此外，Refiner[37]应用线性层来扩展注意力图的维度（间接增加head数量），以促进多样性。然后，采用分布式局部注意力（DLA）来实现对局部特征和全局特征的更好建模，这是通过影响注意力图的head-wise卷积来实现的。从训练策略的角度来看，Gong等人提出了deep Transformer的三个Patch Diversity损失，这可以显著鼓励patch的多样性并抵消过度平滑问题[67]。类似于[175]，patch-wise余弦损失最小化

了patch之间的成对余弦相似性。patch-wise对比度损失通过其在早期层中的对应patch使较深的patch正则化。受Cutmix[176]的启发，patch-wise混合损失混合了两个不同的图像，并迫使每个patch只关注来自同一图像的patch，而忽略不相关的patch。与LV-ViT[43]相比，它们具有相似的损失功能，但动机不同。前者侧重于patch多样性，而后者侧重于关于token标记的数据增强。

Transformers with Self-Supervised Learning

自监督Transformer在NLP领域取得了巨大的成功[5]，但视觉Transformer仍停留在监督的预训练阶段[35]，[40]。最近的工作还试图以生成和判别的方式为ViT设计各种自监督学习方案。生成式的相关工作有iGPT[68]、BEiT[70]、dVAE[147]。判别式的相关工作有[72]、DINO[73]。

讨论

算法评估和比较分析：在论文的分类法中，所有现有的监督模型被分为六类。表一总结了这些现有ViT在ImageNet-1k基准上的性能。为了客观直观地评估它们，论文使用以下三张图来说明它们在不同配置下在ImageNet-1k上的性能。图8（a）总结了2242个输入大小下每个模型的精度。图8（b）以FLOP为水平轴，重点关注其在更高分辨率下的性能。图8（c）侧重于具有外部数据集的预训练模型。根据这些比较结果，论文简要总结了在效率和可伸缩性方面的几项性能改进，如下所示：

- 与大多数结构改进方法相比，DeiT[40]和LV-ViT[43]等基本训练策略更适用于各种模型、任务和输入；
- 局部性对于Transformer是必不可少的，这反映在VOLO[44]和Swin[35]分别在分类和密集预测任务上的优势；
- 卷积patchify stem（ViTc[58]）和早期卷积阶段（CoAtNet[39]）可以显著提高Transformer器的精度，尤其是对于大模型。论文推测原因是因为这些设计引入了比ViT中的non-overlapping patch projection更严格的高级特征[29]；
- deep Transformer，如Refined-ViT[37]和CaiT[42]，具有巨大的潜力。随着模型尺寸与channel尺寸成二次增长，未来可以进一步研究deep Transformer中的相关权衡策略；
- CeiT[54]和CvT[36]在训练中小型模型（0到40M）有显著优势，这表明轻量模型的这种混合注意力block值得进一步探索。

TABLE I

TOP-1 ACCURACY COMPARISON OF VISUAL TRANSFORMERS ON IMAGENET-1K. "1K ONLY" DENOTES TRAINING ON IMAGENET-1K ONLY; "21K PRE." DENOTES PRE-TRAINING ON IMAGENET-21K AND FINE-TUNING ON IMAGENET-1K; "DISTILL." DENOTES APPLYING DISTILLATION TRAINING SCHEME OF DEIT [40]; THE COLOR OF "LEGEND" CORRESPONDING TO EACH MODEL ALSO DENOTES SAME MODEL IN FIG. 8.

Method	#Params. (M)	FLOPs (G)	ImageNet-1k Top-1 Acc. 1K	21K/Distill.(†/ϒ)	Legend
ViT-B/16↑384 [29]	86	743	77.9	83.97†	
ViT-L/16↑384 [29]	307	5172	76.5	85.15†	
VT-Resnet18 [51]	11.7	1.57	76.8	-	
VT-Resnet34 [51]	19.2	3.24	79.9	-	
VT-Resnet50 [51]	21.4	3.41	80.6	-	
VT-Resnet101 [51]	41.5	7.13	82.3	-	
BoTNet-T2 [52]	33.5	7.3	81.7	-	
BoTNet-T4 [52]	54.7	10.9	82.8	-	
BoTNet-T5↑256 [52]	75.1	19.3	83.5	-	
DeiT-Ti [40]	5.7	1.3	72.2	74.5ϒ	
DeiT-S [40]	22.1	4.6	79.8	81.2ϒ	
DeiT-B [40]	86.6	17.6	81.8	83.4ϒ	
DeiT-B↑384 [40]	86.6	52.8	83.1	84.5ϒ	
ConViT-Ti [53]	6	1	73.1	-	
ConViT-S [53]	27	5.4	81.3	-	
ConViT-B [53]	86	17	82.4	-	
LocalViT-T [55]	5.9	1.3	74.8	-	
LocalViT-S [55]	22.4	4.6	80.8	-	
CeiT-T [54]	6.4	1.2	76.4	-	
CeiT-S [54]	24.2	4.5	82.0	-	
CeiT-T↑384 [54]	6.4	3.6	78.8	-	
CeiT-S↑384 [54]	24.2	12.9	83.3	-	
ResT-Small [57]	13.7	1.9	79.6	-	
ResT-Base [57]	30.3	4.3	81.6	-	
ResT-Large [57]	51.6	7.9	83.6	-	
ViT$_C$-1GF [61]	4.6	1.1	75.3	-	
ViT$_C$-4GF [61]	17.8	4.0	81.4	81.2†	
ViT$_C$-18GF [61]	81.6	17.7	83.0	84.9†	
ViT$_C$-36GF [61]	167.8	35	84.2	85.8†	
CoAtNet-0 [39]	25	4.2	81.6	-	
CoAtNet-1 [39]	42	8.4	83.3	-	
CoAtNet-2 [39]	75	15.7	84.1	87.1†	
CoAtNet-3 [39]	168	34.7	84.5	87.6†	
CoAtNet-4↑384 [39]	275	189.5	-	88.4†	
TNT-S [59]	23.8	5.2	81.3	-	
TNT-B [59]	65.6	14.1	82.8	-	
TNT-S↑384 [59]	23.8	-	83.1	-	
TNT-B↑384 [59]	65.6	-	83.9	-	
Swin-T [35]	29	4.5	81.3	-	
Swin-S [35]	50	8.7	83.0	-	
Swin-B [35]	88	15.4	83.3	85.2†	
Swin-L↑384 [35]	197	104	-	86.4†	
LV-ViT-S [43]	26	6.6	83.3	-	
LV-ViT-M [43]	56	16.0	84.0	-	
LV-ViT-L↑288 [43]	150	59.0	85.3	-	
LV-ViT-M↑384 [43]	56	42.2	85.4	-	
LV-ViT-L↑448 [43]	150	157.2	85.9	-	

Method	#Params. (M)	FLOPs (G)	ImageNet-1k Top-1 Acc. 1K	21K/Distill.(†/ϒ)	Legend
VOLO-D1 [44]	27	6.8	84.2	-	
VOLO-D2 [44]	59	14.1	85.2	-	
VOLO-D3 [44]	86	20.6	85.4	-	
VOLO-D4 [44]	193	43.8	85.7	-	
VOLO-D5 [44]	296	69.0	86.1	-	
VOLO-D3↑448 [44]	86	67.9	86.3	-	
VOLO-D4↑448 [44]	193	197	86.8	-	
VOLO-D5↑448 [44]	296	304	87.0	-	
T2T-ViT-14 [63]	21.5	5.2	81.5	-	
T2T-ViT-19 [63]	39.2	8.9	81.9	-	
PVT-Ti [41]	13.2	1.9	75.1	-	
PVT-S [41]	24.5	3.8	79.8	-	
PVT-M [41]	44.1	6.7	81.2	-	
PVT-L [41]	61.4	9.8	81.7	-	
PVTv2-B2 [65]	25.4	4.0	82.0	-	
PVTv2-B4 [65]	62.6	10.1	83.6	-	
PiT-Ti [64]	4.9	0.7	73.0	74.6ϒ	
PiT-XS [64]	10.6	1.4	78.1	79.1ϒ	
PiT-S [64]	23.5	2.9	80.9	81.9ϒ	
PiT-B [64]	73.8	12.5	82.0	84.0ϒ	
CvT-13 [36]	20	4.5	81.6	-	
CvT-21 [36]	32	7.1	82.5	-	
CvT-13↑384 [36]	20	16.3	83.0	83.3†	
CvT-21↑384 [36]	32	24.9	83.3	84.9†	
CvT-W24↑384 [36]	277	193.2	-	87.7†	
DeepViT-S [66]	27	6.2	82.3	-	
DeepViT-L [66]	55	12.5	83.1	-	
CaiT-XS-24 [42]	26.6	5.4	81.8	82.0ϒ	
CaiT-S-24 [42]	46.9	9.4	82.7	83.5ϒ	
CaiT-S-36 [42]	68.2	13.9	83.3	84.0ϒ	
CaiT-M-24 [42]	185.9	36.0	83.4	84.7ϒ	
CaiT-M-36 [42]	270.9	53.7	83.8	85.1ϒ	
DiversePatch-S12 [67]	22	-	81.2	-	
DiversePatch-S24 [67]	44	-	82.2	-	
DiversePatch-B12 [67]	86	-	82.9	-	
DiversePatch-B24 [67]	172	-	83.3	-	
DiversePath-B12↑384 [67]	86	-	84.2	-	
Refined-ViT-S [37]	25	7.2	83.6	-	
Refined-ViT-M [37]	55	13.5	84.6	-	
Refined-ViT-L [37]	81	19.1	84.9	-	
Refined-ViT-M↑384 [37]	55	49.2	85.6	-	
Refined-ViT-L↑384 [37]	81	69.1	85.7	-	
CrossViT-9 [178]	8.6	1.8	73.9	-	
CrossViT-15 [178]	27.4	5.8	81.5	-	
CrossViT-18 [178]	43.3	9.0	82.5	-	
CrossViT-15*↑384 [178]	28.5	21.4	83.5	-	
CrossViT-18*↑384 [178]	44.6	32.4	83.9	-	

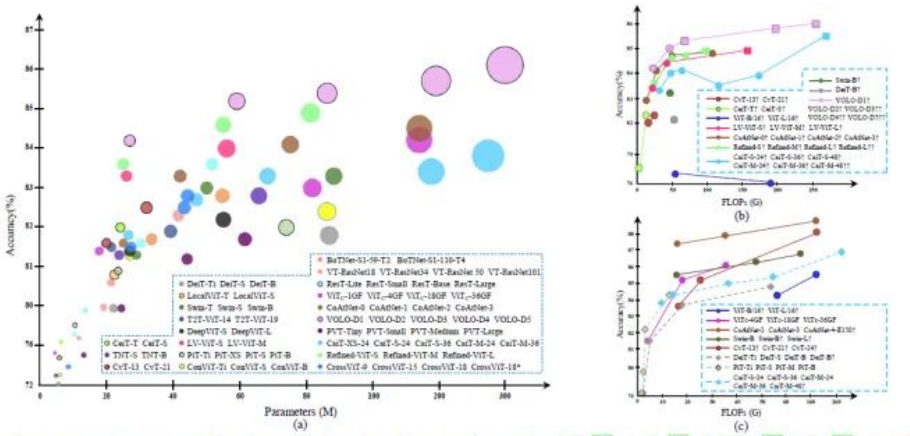


Fig. 8. Comparisons of recent visual Transformers on ImageNet-1k benchmark, including ViT [29], DeiT [40], BoTNet [52], VTs [51], ConViT [53], CeiT [54], LocalViT [55], TNT [59], Swin [35], PiT [64], T2T-ViT [63], PVT [41], CvT [36], DeepViT [66], CaiT [42], Cross ViT [178] (best viewed in color). (a) The bubble plot of the mentioned models with 224^2 resolution input, the size of cycle denotes GFLOPs. (b) Comparison of visual Transformers with high-resolution inputs, the square indicates 448^2 input resolution. (c) The accuracy plot of some pre-trained models on ImageNet-21k.

ViT发展趋势综述

当论文的系统学与这些模型的时间线匹配时，我们可以清楚地跟踪Transformer用于图像分类的发展趋势（图1）。作为一种自注意机制，视觉Transformer主要根据NLP中的朴素结构（ViT[29]和iGPT[68]）或CV中基于注意力的模型（VTs[51]和BoTNet[52]）进行重新设计。然后，许多方法开始将CNN的层次结构或深层结构扩展到ViT。T2T-ViT[63]、PVT[41]、CvT[36]和PiT[64]都有一个动机，即将分层结构迁移到Transformer中，但它们实现的下采样方式不同。CaiT[42]、Diverse Patch[67]、DeepViT[66]和Refiner[37]关注deep Transformer中的问题。此外，一些方法转向内部组件以进一步增强先前Transformer的图像处理能力，即位置编码[56]、[179]、[180]、MHSA[28]和MLP[167]。下一波Transformer是局部范式。其中大多数通过引入局部注意机制[35]、[44]、[59]、[60]或卷积[53]–[55]将局部性引入Transformer。如今，最新监督Transformer正在探索结构组合[39]、[58]和scaling laws[38]、[181]。除了有监督的Transformer，自监督学习在ViT[68]–[70]、[72]–[74]中占据了很大一部分。然而，目前尚不清楚哪些任务和结构对CV中的自监督Transformer更有利。关于备选方案的简要讨论：在ViT的开发过程中，最常见的问题是ViT能否完全取代传统的卷积。通过回顾过去一年的性能改进历史，这里没有任何相对劣势的迹象。ViT已经从一个纯粹的结构回归到一个混合的形式，而全局信息已经逐渐回归到带有局部bias的混合阶段。尽管ViT可以等同于CNN，

甚至具有更好的建模能力，但这种简单有效的卷积运算足以处理浅层中的局部性和语义特征。未来，两者结合的精神将推动图像分类取得更多突破。

检测Transformer

在本节中，论文将回顾用于目标检测的ViT，它可以分为两个部分：Transformer Neck和Transformer Backbone。对于neck，论文主要关注为Transformer结构指定的一种新表示，称为object query，即一组可学习的参数等价地聚集了全局特征。最近的变体试图在收敛加速或性能改进方面解决最优融合范式。除了专门为检测任务设计的neck外，一部分主干检测器还考虑了特定的策略。最后，论文对它们进行了评估，并分析了这些检测器的一些潜在方法。

Transformer Neck

首先回顾DETR[30]和Pix2seq[75]，它们是最初的Transformer检测器，重新定义了两种不同的目标检测范式。随后，论文主要关注基于DETR的变体，从五个方面改进了Transformer检测器的准确性和收敛性：稀疏注意力、空间先验、结构重新设计、分配优化和预训练模型。原始检测器：DETR[30]是第一个端到端Transformer检测器，它消除了手工设计的表示[182]-[185]和非最大抑制（NMS）后处理，这将目标检测重新定义为集合预测问题。详细地说，一小组可学习的位置编码，称为object query，被并行馈送到Transformer解码器中，以从图像特征中聚合实例信息。然后，预测头直接从解码器的输出query产生检测结果。在训练过程中，在预测目标和GT之间使用二分匹配策略，以识别一对一的标签分配，从而在没有NMS的情况下消除推理时的冗余预测。在反向传播中，匈牙利损失包括所有分类结果的对数似然损失和所有匹配对的box损失。总之，DETR为端到端目标检测提供了一种新的范例。object query在与图像特征交互期间逐渐学习实例表示。二分匹配允许直接的集合预测很容易适应一对一的标签分配，从而消除了传统的后处理。DETR在COCO基准上实现了具有竞争力的性能，但在小目标上存在收敛速度慢和性能差的问题。另一项开创性工作是Pix2seq[75]，将通用目标检测视为一项语言建模任务。给定一个图像输入，执行一个vanilla sequential Transformer来提取特征并自动回归生成一系列目标描述（即类标签和边界框）。这种简化但更复杂的图像caption方法是在这样的假设下得出的，即如果模型同时了解目标的位置和标签，则可以教导其生成具有指定序列的描述[75]。与DETR相比，Pix2seq在小目标上获得了更好的结果。如何将这两种概念结合起来值得进一步考虑。稀疏注意力：在DETR中，query和特征图之间的密集交互耗费了难以承受的资源，并减缓了DETR的收敛速度。因此，最近的努力旨在设计依赖于数据的稀疏注意力来解决这些问题。继[186]之后，Zhu等人开发了Deformable DETR，以通过多尺度deformable attention显著改善训练收敛性和检测性能[76]。与原始DETR相比，deformable attention模块仅对一小部分关键点进行采样，以进行全特征聚合。这种稀疏注意力可以很容易地扩展到多尺度特征融合，而无需FPN[187]的帮助，因此称为多尺度可定义注意力（MSDA），如图10所示。其他相关算法ACT[77]、PnP[78]、Sparse DETR[79]可以参考具体论文。

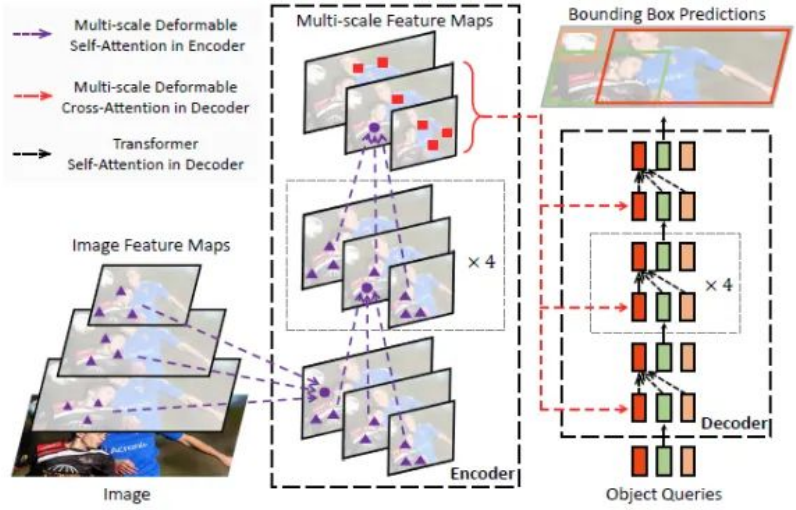


Fig. 10. Illustration of Deformable DETR. A fixed number of key samples in each scale feature interacting with all queries. (from [76].)

空间先验：与由内容和几何特征[182]、[188]直接生成的anchor或其他表示不同，object query通过随机初始化隐式建模空间信息，这与边界框弱相关。空间先验应用的主流是具有经验空间信息的一阶段检测器和具有几何坐标初始化或感兴趣区域（RoI）特征的两阶段检测器。一阶段相关算法有SMCA[80]、Conditional DETR[81]、Anchor DETR[82]、DAB-DETR[83]。二阶段相关算法有Efficient DETR[84]、Dynamic DETR[85]。结构重新设计：除了关注交叉注意力的修改之外，一些工作重新设计了仅编码器的结构，以直接避免解码器的问题。TSP[86]继承了集合预测[30]的思想，并去除了解码器和object query以加速收敛。这种仅编码器的DETR重用先前的表示[182]、[188]，并生成一组固定大小的感兴趣特征（FoI）[188]或proposal[182]，这些proposal随后被馈送到Transformer编码器。此外，匹配蒸馏被应用于解决二分匹配的不稳定性，特别是在早期训练阶段。Fang等人[87]将DETR的编码器-解码器 neck和ViT的仅编码器主干合并为仅编码器检测器，并开发了YOLOS，一种纯序列到序列转换器，以统一分类和检测任务。它继承了ViT的结构，并用固定大小的可学习检测token替换了单个类token。这些目标token首先在分类任务上进行预训练，然后在检测基准上进行微调。二分匹配优化：在DETR[30]中，二分匹配策略迫使预测结果在训练期间完成一对一的标签分配。这样的训练策略简化了检测管道，并在无需NMS帮助的情况下直接构建端到端系统。为了深入了解端到端检测器的功效，Sun等人致力于探索一对一预测的理论观点[192]。基于多次消融和理论分析，他们得出结论，一对一匹配策略的分类成本是显著避免重复预测的关键因素。即便如此，DETR仍面临着由二分匹配引起的多重问题。Li等人[90]利用denoisingDETR（DN-DETR）来减轻二部分匹配的不稳定性。具体地说，一系列有轻微扰动的目标应该重建它们的真实坐标和类。去噪（或重建）部分的主要成分是防止匹配部分和噪声部分之间信息泄漏的注意力掩码，以及指示扰动的指定标签嵌入。其他工作还有DINO[91]。预训练：灵感来自预训练的语言Transformer[3]，[5]，相关工作有UP-DETR[88]、FP-DETR[89]。

Transformer Backbone

前文已经回顾了许多基于Transformer的图像分类主干[29]、[40]。这些主干可以很容易地并入各种框架（例如，Mask R-CNN[189]、RetinaNet[184]、DETR[30]等），以执行密集预测任务。例如，像PVT[41]、[65]这样的分层结构将ViT构造为一个高分辨率到低分辨率的过程，以学习多尺度特征。局部增强结构将主干构建为局部到全局的组合，可以有效地提取短距离和长距离视觉相关性，并避免二次计算开销，如Swin Transformer[35]、ViL[61]和Focal Transformer[62]。表III包括密集预测任务的这些模型的更详细比较。除了通用Transformer主干，特征金字塔Transformer（FPT）[92]通过使用self-attention、自上而下的cross-attention和自底向上的cross channel attention，结合了空间和尺度的特性。继[193]之后，HRFormer[93]向Transformer介绍了多分辨率的优点以及非重叠的局部self-attention。HRViT[94]重新设计了异质分支和十字形注意力模块。

TABLE III
DENSE PREDICTION RESULTS OF COCO 2017 VAL. SET BASED ON RETINANET [184] AND MASK R-CNN [189], WHEN TRAINED WITH 3× SCHEDULE AND MULTI-SCALE INPUTS (MS). THE NUMBERS BEFORE AND AFTER "/" CORRESPOND TO THE PARAMETER OF RETINANET AND MASK R-CNN, RESPECTIVELY. (MOST OF DATA FROM [62].)

Backbone	#Params (M)	FLOPs (G)	RetinaNet 3× schedule + MS						Mask R-CNN 3× schedule + MS					
			AP^b	AP^b_{50}	AP^b_{75}	AP_S	AP_M	AP_L	AP^b	AP^b_{50}	AP^b_{75}	AP^m	AP^m_{50}	AP^m_{75}
ResNet50 [11]	38 / 44	239 / 260	39.0	58.4	41.8	22.4	42.8	51.6	41.0	61.7	44.9	37.1	58.4	40.1
PVTv1-Small [41]	34 / 44	226 / 245	42.2	62.7	45.0	26.2	45.2	57.2	43.0	65.3	46.9	39.9	62.5	42.8
ViL-Small [61]	36 / 45	252 / 174	42.9	63.8	45.6	27.8	46.4	56.3	43.4	64.9	47.0	39.6	62.1	42.4
Swin-Tiny [35]	39 / 48	245 / 264	45.0	65.9	48.4	29.7	48.9	58.1	46.0	68.1	50.3	41.6	65.1	44.9
PVTv2-B2-Li [65]	32 / 42	- / -	-	-	-	-	-	-	46.8	68.7	51.4	42.3	65.7	45.4
Focal-Tiny [62]	39 / 49	265 / 291	45.5	66.3	48.8	31.2	49.2	58.7	47.2	69.4	51.9	42.7	66.5	45.9
PVTv2-B2 [65]	35 / 45	- / -	-	-	-	-	-	-	47.8	69.7	52.6	43.1	66.8	46.7
ResNet101 [11]	57 / 63	315 / 336	40.9	60.1	44.0	23.7	45.0	53.8	42.8	63.2	47.1	38.5	60.1	41.3
ResNeXt101-32x4d [191]	56 / 63	319 / 340	41.4	61.0	44.3	23.9	45.5	53.7	44.0	64.4	48.0	39.2	61.4	41.9
PVTv1-Medium [41]	54 / 64	283 / 302	43.2	63.8	46.1	27.3	46.3	58.9	44.2	66.0	48.2	40.5	63.1	43.5
ViL-Medium [61]	51 / 60	339 / 261	43.7	64.6	46.4	27.9	47.1	56.9	44.6	66.3	48.5	40.7	63.8	43.7
Swin-Small [35]	60 / 69	335 / 354	46.4	67.0	50.1	31.0	50.1	60.3	48.5	70.2	53.5	43.3	67.3	46.6
Focal-Small [62]	62 / 71	367 / 401	47.3	67.8	51.0	31.6	50.9	61.1	48.8	70.5	53.6	43.8	67.7	47.2
ResNeXt101-64x4d [191]	96 / 102	473 / 493	41.8	61.5	44.4	25.2	45.4	54.6	44.4	64.9	48.8	39.7	61.9	42.6
PVTv1-Large [41]	71 / 81	345 / 364	43.4	63.6	46.1	26.1	46.0	59.5	44.5	66.0	48.3	40.7	63.4	43.7
ViL-Base [61]	67 / 76	443 / 365	44.7	65.5	47.6	29.9	48.0	58.1	45.7	67.2	49.9	41.3	64.4	44.5
Swin-Base [35]	98 / 107	477 / 496	45.8	66.4	49.1	29.9	49.4	60.3	48.5	69.8	53.2	43.4	66.8	46.9
Focal-Base [62]	101 / 110	514 / 533	46.9	67.8	50.3	31.9	50.3	61.5	49.0	70.1	53.6	43.7	67.6	47.0

讨论

论文在表II中总结了Transformer neck检测器的五个部分，密集预测任务的Transformer backbone的更多细节参见表III。大多数neck提升集中在以下五个方面：

- 1）提出了稀疏注意力模型和评分网络，以解决冗余特征交互问题。这些方法可以显著降低计算成本并加速模型收敛；
- 2）将显式空间先验分解为所选特征初始化和由可学习参数提取的位置信息，将使检测器能够精确预测结果；
- 3）在Transformer解码器中扩展了多尺度特征和逐层更新，用于小目标细化；
- 4）改进的二分匹配策略有利于避免冗余预测以及实现端到端目标检测；
- 5）仅编码器结构减少了整个Transformer堆栈层，但过度增加了FLOPs，而编码器-解码器结构是FLOPs和参数之间的良好权衡，但更深的解码器层可能会导致长时间训练过程和过度平滑的问题。

此外，有许多Transformer主干用于改进分类性能，但很少有针对密集预测任务的工作。未来，论文预计Transformer主干将与深度高分辨率网络合作，以解决密集预测任务。

TABLE II
COMPARISON BETWEEN TRANSFORMER NECKS AND REPRESENTATIVE CNNS ON COCO 2017 VAL SET. "GPUS TIME" DENOTES THE TRAINING TIME WITH 8×V100 GPUS; "MS" DENOTES TO MULTI-SCALE FEATURES.

Method	GPUs Time	Epochs	FLOPs (G)	#Para. (M)	FPS	MS	AP/AP$_{50}$/AP$_{75}$	Ap$_S$/Ap$_M$/Ap$_L$
CNN Backbone with Other Representations								
FCOS [86], [188]	-	36	177	-	17	✓	41.0 /59.8/44.1	26.2/44.6/52.2
Faster R-CNN [182]	-	36	180	42	26	✓	40.2/61.0/43.8	24.2/43.5/52.0
Faster R-CNN+ [182]	-	108	180	42	26	✓	42.0 /62.1/45.5	26.6/45.4/53.4
Mask R-CNN [189]	-	36	260	44	-	✓	41.0/61.7/44.9	- / - / -
Cas. Mask R-CNN [190]	-	36	739	82	18	✓	46.3/64.3/50.5	- / - / -
Transformer Model as Neck								
DETR-R50 [30]	240h	500	86	41	28	✗	42.0/62.4/44.2	20.5/45.8/61.1
DETR-DC5 [30]	240h	500	187	41	12	✗	43.3/63.1/45.9	22.5/47.3/61.1
ACT-MTKD (L=16) [77]	W/o	-	156	-	14	✗	40.6/ - / -	18.5/44.3/59.7
ACT-MTKD (L=32) [77]	W/o	-	169	-	16	✗	43.1/ - / -	22.2/47.1/61.4
Defor. DETR [76]	20h	50	78	34	27	✗	39.7/60.1/42.4	21.2/44.3/56.0
Defor. DETR-DC5 [76]	27h	50	128	34	22	✗	41.5/61.8/44.9	24.1/45.3/56.0
Defor. DETR-Iter [76]	40h	50	173	40	19	✓	43.8/62.6/47.7	26.4/47.1/58.0
Defor. DETR-Two [76]	43h	50	173	40	19	✓	46.2/65.2/50.0	28.8/49.2/61.7
SMCA [35]	33h	50	152	40	22	✗	41.0/ - / -	21.9/44.3/59.1
SMCA+ [35]	70h	108	152	40	22	✗	42.7/ - / -	22.8/46.1/60.0
SMCA [35]	75h	50	152	40	10	✓	43.7/63.6/47.2	24.2/47.0 /60.4
SMCA+ [35]	160h	108	152	40	10	✓	45.6/65.5/49.1	25.9/49.3/62.6
Efficient DETR [84]	-	36	159	32	-	✓	44.2/62.2/48.0	28.4/47.5/56.6
Efficient DETR* [84]	-	36	210	35	-	✓	45.1/63.1/49.1	28.3/48.4/59.0
Condit. DETR [81]	30h	108	90	44	-	✗	43.0/64.0/45.7	22.7/46.7/61.5
Condit. DETR-DC5 [81]	30h	108	195	44	-	✗	45.1/65.4/48.5	25.3/49.0/62.2
UP-DETR [88]	72h	150	86	41	28	✗	40.5/60.8/42.6	19.0/44.4/60.0
UP-DETR+ [88]	144h	300	86	41	28	✗	42.8/63.0/45.3	20.8/47.1/61.7
TSP-FCOS [86]	15h	36	189	52	15	✓	43.1/62.3/47.0	26.6/46.8/55.9
TSP-RCNN [86]	15h	36	188	64	11	✓	43.8/63.3/48.3	28.6/46.9/55.7
TSP-RCNN+ [86]	40h	96	188	64	11	✓	45.0/64.5/49.6	29.7/47.7/58.0
YOLOS-S [87]	240h‡	150	200	31	7	-	36.1/56.4/37.1	15.3/38.5/56.1
YOLOS-S [87]	-	150	179	28	5	✓	37.6/57.6/39.2	15.9/40.2/57.3
YOLOS-B [87]	480h‡	150	537	127	-	-	42.0/62.2/44.5	19.5/45.3/62.1
Pix2seq [75]	384h+¶	300	-	37	-	✗	43.0/61.0/45.6	25.1/46.9/59.4
Pix2seq-DC5 [75]	384h+¶	300	-	38	-	✗	43.2/61.0/46.1	26.6/47/58.6
Sparse-DETR-0.1 [79]	23h	50	105	41	25	✓	45.3/65.8/49.3	28.4/48.3/60.1
Sparse-DETR-0.5 [79]	28h	50	136	41	21	✓	46.3/66.0/50.1	29.0/49.5/60.8
PnP-DETR-0.33 [78]	-	500	77.1	-	-	✗	41.1/61.5/43.7	20.8/44.6/60.0
PnP-DETR-0.5 [78]	-	500	78.9	-	-	✗	41.8/62.1/44.4	21.2/45.3/60.8
PnP-DETR-DC5-0.5 [78]	-	500	135.9	-	-	✗	43.1/63.4/45.3	22.7/46.5/61.1
Anchor-DETR [82]	22h	50	-	39	-	✗	42.1/63.1/44.9	22.3/46.2/60.0
Anchor-DETR-DC5 [82]	28h	50	172	39	19	✗	44.2/64.7/47.5	24.7/48.2/60.6
DAB-DETR [83]	-	50	100	44	22	✗	42.6/63.2/45.6	21.8/46.2/61.1
DAB-DETR-DC5 [83]	-	50	216	44	-	✗	45.7/66.2/49.0	26.1/29.4/63.1
Dynamic DETR [85]	-	50	-	58	-	✓	47.2/65.9/51.1	28.6/49.3/59.1
FP-DETR-Base [89]	-	50	-	36	-	✗	43.7/64.1/47.8	26.5/46.7/58.2
DN-DETR [90]	-	50	94	44	-	✗	44.1/64.4/46.7	22.9/48.0/63.4
DN-DETR-DC5 [90]	-	50	202	44	-	✗	46.3/66.4/49.7	26.7/50.0/64.3
DN-Defor.-DETR [90]	-	50	196	48	-	✓	46.3/66.4/49.7	26.7/50.0/64.3
DINO-4scale [91]	-	36	279	47	24	✓	50.5/68.3/55.1	32.7/53.9/64.9
DINO-5scale [91]	-	36	860	47	10	✓	51.0/69.0/55.6	34.1/53.6/65.6

‡ denotes 8 × 3090Ti GPUs, and ¶ denotes 8 × TPUs.

分割Transformer

Patch-Based 和 Query-Based Transformer是分割的两种主要应用方式。后者可以进一步细分为Object Query 和 Mask Embedding两类。

Patch-Based Transformer

由于感受野扩展策略[194]，CNN需要多个解码器堆栈来将高级特征映射到原始空间分辨率。相反，基于patch的Transformer由于其全局建模能力和分辨率不变性，可以很容易地与用于分割mask预测的简单解码器结合。Zheng等人扩展了用于语义分割任务的ViT[29]，并通过使用解码器的三种方式来实现逐像素分类，提出了SEgmentation TRansformer（SETR）[95]：naive上采样（naive）、渐进上采样（PUP）和多级特征聚合（MLA）。SETR展示了ViT用于分割任务的可行性，但它也带来了不可接受的额外GPU开销。TransUNet[96]是第一个用于医学图

像分割的方法。形式上，它可以被视为带有MLA解码器的SETR的变体[95]，或者是U-Net[195]和Transformer的混合模型。由于Transformer编码器强大的全局建模能力，Segformer[97]设计了一个只有四个MLP层的轻量级解码器。当使用多种损坏类型的图像进行测试时，Segformer显示出比CNN更好的性能和更强的鲁棒性。

Query-Based Transformer

Query embedding是一组从图像输入中逐渐学习的临时语义/实例表示。与patch嵌入不同，query可以更“公平”地集成来自特征的信息，并自然地与集合预测损失结合[30]，用于去除后处理。现有的基于query的模型可以分为两类。一种是由检测和分割任务同时驱动的（称为object queries）。另一个仅由分割任务（称为mask embeddings）监督。Object Queries：基于object queries的方法有三种训练方式（图11）。如图11(a)所示的Panoptic DETR[30]。图11(b)所示的Cell-DETR[98]和VisTR[99]，以及如图11(c)所示的QueryInst[100]

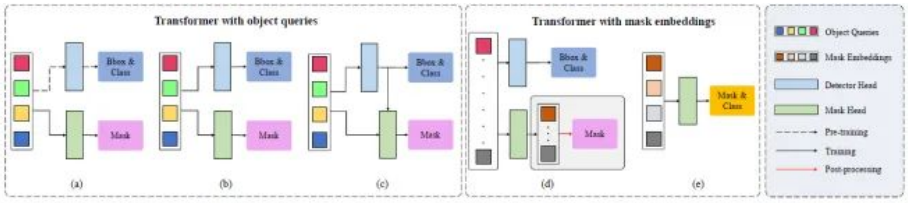


Fig. 11. Query-based frameworks for segmentation tasks. (a) The model is firstly trained on detection task and then fine-tuned on segmentation task following a serial training process. (b) The model is supervised on two dependent task simultaneously. (c) The cascade hybrid task model generates a fine-grained mask based on the coarse region predicted by detector head. (d) The query embeddings are dependently supervised by mask embeddings and boxes. (e) The box-free model directly predicts masks without box branch and views segmentation task as a mask prediction problem.

Mask Embeddings：另一个框架使用query直接预测掩码，论文将这种基于掩码的学习查询称为mask embedding。与object queries不同，mask embedding仅由分割任务监督。如图11（d）所示，两个不相交的query集被并行用于不同的任务，box学习被视为进一步增强的辅助损失，相关算法有ISTR[101]、SOLQ[102]。对于语义和box-free实例分割，一系列基于query的Transformer直接预测掩码，而不需要框分支的帮助（图11（e））， 相关算法如Max-DeepLab[31]、Segmenter[103]、Maskformer[104]等。

TABLE IV
COMPARISON BETWEEN CNN-BASED AND TRANSFORMER-BASED MODEL ON ADE20K AND COCO FOR DIFFERENT SEGMENTATION TASKS. "+MS" DENOTES THE PERFORMANCE TRAINED WITH MULTI-SCALE INPUTS.

(a) ADE20K Val. Set for Semantic Segmentation

Method	Backbone	image size	#Params. (M)	FLOPs (G)	FPS	mIoU	+MS
UperNet [104] [198] [199]	R-50 [11]	512	67	238	23.4	42.1	42.8
	R-101 [11]	512	86	257	20.3	43.8	44.9
	Swin-T [35]	512	60	236	18.5	44.5	46.1
	Swin-S [35]	512	81	259	15.2	47.6	49.3
	Swin-B† [35]	640	121	471	8.7	50.0	51.6
	Swin-L† [35]	640	234	647	6.2	52.0	53.5
Segformer [97]	MiT-B3	512	47.3	79	-	49.4	50.0
	MiT-B4	512	64.1	95.7	15.4	50.3	51.1
	MiT-B5	512	84.7	183.3	9.8	51.0	51.8
Segmenter [103]	ViT-S/16† [29]	512	27	-	34.8	45.3	46.9
	ViT-B/16† [29]	512	106	-	24.1	48.5	50.0
	ViT-L/16† [29]	640	334	-	-	51.8	53.6
MaskFormer [104]	R-50 [11]	512	41	53	24.5	44.5	46.7
	R-101 [11]	512	60	73	19.5	45.5	47.2
	Swin-T [35]	512	42	55	22.1	46.7	48.8
	Swin-S [35]	512	63	79	19.6	49.8	51.0
	Swin-B† [35]	640	102	195	12.6	52.7	53.9
	Swin-L† [35]	640	212	375	7.9	54.1	55.6

(b): COCO Test-Dev for Instance Segmentation

Method	Backbone	Epochs	Ap^b/Ap^m	AP^m_S	Ap^m_M	Ap^m_L	FPS
Mask R-CNN [189]	R-50-FPN [11]	36	41.3/37.5	21.1	39.6	48.3	15.3
	R-101-FPN [11]	36	43.1/38.8	21.8	41.4	50.5	11.8
Blend Mask [200]	R-50-FPN [11]	36	43.0/37.8	18.8	40.9	53.6	15.0
	R-101-FPN	36	44.7/39.6	22.4	42.2	51.4	11.5
SOLO v2 [201]	R-50-FPN [11]	36	40.7/38.2	16.0	41.2	55.4	10.5
	R-101-FPN [11]	36	42.6/39.7	17.3	42.9	57.4	9.0
ISTR [101]	R-50-FPN [11]	36	46.8/38.6	22.1	40.4	50.6	13.8
	R-101-FPN [11]	36	48.1/39.9	22.8	41.9	52.3	11.0
SOLQ [102]	R-50 [11]	50	47.8/39.7	21.5	42.5	53.1	-
	R-101 [11]	50	48.7/40.9	22.5	43.8	54.6	-
	Swin-L† [35]	50	55.4/45.9	27.8	49.3	60.5	-
QueryInst [100]	R-50-FPN [11]	36	44.8/40.1	23.3	42.1	52.0	10.5
	R-50-FPN [11]	36	45.6/40.6	23.4	42.5	52.8	7.0
	R-101-FPN [11]	36	47.0/41.7	24.2	43.9	53.9	6.1
	Swin-L† [35]	50	56.1/49.1	31.5	51.8	63.2	3.3

(c): COCO Panopticon Minival. for Panoptic Segmentation

Method	Backbone	Epochs	#Params. (M)	FLOPs (G)	PQ	PQ^{Th}	PQ^{St}
DETR [30]	R-50 [11]	150+25	42.8	137	43.4	48.2	36.3
	R-101 [11]		61.8	157	45.1	50.5	37.0
MaxDeepLab [31]	Max-S	54	61.9	162	48.4	53.0	41.5
	Max-L		451	1846	57.0	42.2	51.1
MaskFormer [202]	R-50 [11]	300	45	181	46.5	51.0	39.8
	R-101 [11]		64	248	47.6	52.5	40.3
	Swin-T [35]		42	179	47.7	51.7	41.7
	Swin-S [35]		63	259	49.7	54.4	42.6
	Swin-B [35]		102	411	51.1	56.3	43.2
	Swin-L† [35]		212	792	52.7	58.5	44

† denotes the model pre-trained on ImageNet-21k

讨论

论文根据三个不同的任务总结了上述Transformer。表IV（a）侧重于ADE20K（170类）。可以表明，当在具有大量类的数据集上进行训练时，ViT的分割性能显著提高。表IV（b）侧重于实例分割的COCO测试数据集。显然，在分割和检测任务中，带有掩模嵌入的ViT超过了大多数主流模型。然而，APbox和APseg之间存在巨大的性能差距。通过级联框架，QueryInst[100]在各种Transformer模型中实现了SOTA。将ViT与混合任务级联结构相结合，值得进一步研究。表IV（c）侧重于全景分割。Max-DeepLab[31]通常通过掩码预测形式解决全景分割任务中的前景和背景问题，而Maskformer[104]成功地将这种格式用于语义分割，并将语义和实例

分割任务统一为一个模型。基于它们在全景分割领域的表现，我们可以得出结论，ViT可以将多个分割任务统一到一个box-free框架中，并进行掩模预测。

3D视觉Transformer

随着3D采集技术的快速发展，双目/单目图像和LiDAR（Light Detection and Ranging）点云成为3D识别的流行传感数据。与RGB（D）数据不同，点云表示更关注距离、几何图形和形状信息。值得注意的是，由于其稀疏性、无序性和不规则性的特点，这种几何特征非常适合Transformer。随着2D ViT的成功，开发了大量的3D分析方法。本节展示了3D ViT在表示学习、认知映射和特定处理之后的简要回顾。

表示学习

与传统的手工设计的网络相比，ViT更适合于从点云学习语义表示，在点云中，这种不规则和排列不变的性质可以转化为一系列具有位置信息的并行嵌入。鉴于此，Point Transformer[105]和PCT[106]首先证明了ViT对3D表示学习的有效性。前者将hierarchical Transformer[105]与下采样策略[203]合并，并将其先前的vector attention block[25]扩展到3D点云。后者首先聚集相邻点云，然后在全局off-set Transformer上处理这些相邻嵌入，其中来自图卷积网络（GCN）的知识迁移被应用于噪声缓解。值得注意的是，由于点云的固有坐标信息，位置编码（ViT的重要操作）在两种方法中都有所减少。PCT直接处理坐标，无需位置编码，而Point Transformer添加了可学习的相对位置编码以进一步增强。继[105]、[106]之后，Lu等人利用local-global聚合模块3DCTN[107]来实现局部增强和成本效率。给定多步长下采样组，使用具有max-pooling操作的显式图卷积来聚合每个组内的局部信息。将得到的组嵌入级联并馈送到改进的Transformer[105]、[106]中，用于全局聚合。Park等人提出了Fast Point Transformer[108]，通过使用voxel-hashing邻域搜索、体素桥接相对位置编码和基于余弦相似性的局部关注来优化模型效率。为了进行密集预测，Pan等人提出了一种定制的基于点云的Transformer主干（Pointformer）[109]，用于在每个层中分别参与局部和全局交互。与以往的局部-全局形式不同，采用局部关注后的坐标细化操作来更新质心点而不是曲面点。局部-全局交叉注意力模型融合了高分辨率特征，然后是全局注意力。Fan等人返回到Single-stride Sparse Transformer（SST）[110]，而不是下采样操作，以解决小目标检测的问题。与Swin[35]类似，连续Transformer块中的移位组被用于分别处理每组token，这进一步缓解了计算问题。在基于体素的方法中，Voxel Transformer（VoTr）[111]采用两步voxel Transformer来有效地操作空和非空体素位置，包括通过local attention和dilated attention。VoxSeT[112]进一步将self-attention分解为两个交叉关注层，一组潜在编码将它们链接起来，以在隐藏空间中保存全局特征。一系列自监督Transformer也被扩展到3D空间，例如Point BERT[113]、Point MAE[114]和MaskPoint[115]。具体而言，Point BERT[113]和Point MAE[114]直接将先前的工作[70]、[71]转移到点云，而MaskPoint[115]通过使用与DINO（2022）[91]类似的对比解码器来改变生成训练方案，以进行自训练。基于大量实验，论文得出结论，这种生成/对比自训练方法使ViT能够在图像或点云中有效。

Cognition Mapping

鉴于丰富的表示特征，如何将实例/语义认知直接映射到目标输出也引起了相当大的兴趣。与2D图像不同，3D场景中的目标是独立的，可以由一系列离散的表面点直观地表示。为了弥补这一差距，一些现有的方法将领域知识转移到2D主流模型中。继[30]之后，3DETR[116]通过最远点采样和傅里叶位置嵌入将端到端模块扩展到3D目标检测，以用于object queries初始化。Group Free 3D DETR[117]应用了比[116]更具体和更强的结构。详细地说，当object queries时，它直接从提取的点云中选择一组候选采样点，并在解码器中逐层迭代地更新它们。Sheng等人提出了一种典型的两阶段方法，该方法利用Channel-wise Transformer3D检测器（CT3D）[118]同时聚合每个提案中的点云特征的proposal-aware嵌入和channel-wise上下文信息。对于单目传感器，MonoDTR[119]和MonoDETR[120]在训练过程中使用辅助深度监督来估计伪深度位置编码（DPE）。DETR3D[121]引入了一种多目3D目标检测范式，其中2D图像和3

D位置都通过摄像机变换矩阵和一组3D object queries相关联。TransFusion[122]通过连续通过两个Transformer解码器层与object queries交互，进一步利用了LiDAR点和RGB图像的优点。

Specific Processing

受传感器分辨率和视角的限制，点云在真实场景中存在不完整、噪声和稀疏性问题。为此，PoinTr[123]将原始点云表示为一组局部点云代理，并利用几何感知编码器-解码器Transformer将中心点云代理向不完整点云方向迁移。SnowflakeNet[124]将点云补全的过程公式化为类似雪花的生长，它通过point-wise splitting deconvolution策略从父点云逐步生成子点云。相邻层的skip-Transformer进一步细化父层和子层之间的空间上下文特征，以增强它们的连接区域。Choe等人将各种生成任务（例如降噪、补全和超分辨率）统一为点云重构问题，因此称为PointRecon[125]。基于体素散列，它覆盖了绝对尺度的局部几何结构，并利用PointTransformerlike[105]结构将每个体素（query）与其相邻体素（value-key）进行聚合，以便从离散体素到一组点云集进行细粒度转换。此外，增强的位置编码适用于体素局部attention方案，通过使用L1损失的负指数函数作为朴素位置编码的权重来实现。值得注意的是，与masked生成自训练相比，补全任务直接生成一组完整点云，而不需要不完整点云的显式空间先验。

多传感器数据流Transformer

在现实世界中，多个传感器总是互补使用，而不是单个传感器。为此，最近的工作开始探索不同的融合方法，以有效地协同多传感器数据流。与典型的CNN相比，Transformer自然适合于多流数据融合，因为它的非特定嵌入和动态交互注意机制。本节根据数据流源（同源流和异源流）详细介绍了这些方法。

Homologous Stream

同源流是一组具有相似内在特征的多传感器数据，如多视图、多维和多模态视觉流数据。根据融合机制，它们可以分为两类：交互融合和迁移融合。交互融合：CNN的经典融合模式采用channel级联操作。然而，来自不同模态的相同位置可能是各向异性的，这不适合CNN的平移不变偏差。相反，Transformer的空间级联操作使不同的模态能够超越局部限制进行交互。对于局部交互，MVT[126]在空间上连接来自不同视图的patch嵌入，并通过使用模式不可知的Transformer来加强它们的交互。为了减轻多模态特征的冗余信息，MVDeTr[127]将特征图的每个视图投影到地平面上，并将多尺度可变形注意力[76]扩展到多视图设计。其他相关算法TransFuser[128]、COTR[129]可参考论文。对于全局交互，Wang等人[130]利用共享主干提取不同视图的特征。代替COTR[129]中的逐像素/逐patch级联，提取的逐视图全局特征在空间上进行级联，以在Transformer中进行视图融合。考虑到不同相机视图之间的角度和位置差异，TransformerFusion[132]首先将每个视图特征转换为具有其相机视图的内部和外部的嵌入向量。这些嵌入然后被馈送到global Transformer中，该global Transformer的注意力权重用于帧选择，以便有效地计算。为了在3D检测中统一多传感器数据，FUTR3D[131]将类DETR解码器中的object queries投影到一组3D参考点中云。这些点云及其相关特征随后从不同的模态中采样并在空间上连接以更新object queries。迁移融合：与Transformer编码器通过self-attention实现的交互式融合不同，另一种融合形式更像是通过交叉关注机制从源数据到目标数据的迁移学习。例如，Tulder等人[133]在中间主干特征中插入了两个协作的交叉注意力Transformer，用于桥接未配准的多视图医学图像。代替pixel-wise 注意力形式，进一步开发了token-pixel交叉注意力，以减轻繁重的计算。Long等人[134]提出了一种用于多视图图像深度估计的对极时空Transformer。给定包含一系列静态多视点帧的单个视频，首先将相邻帧连接起来，然后将对极线扭曲到中心相机空间中。最终得到的帧volume作为源数据，通过交叉注意力与中心帧进行融合。对于空间对齐的数据流，DRT[135]首先通过使用卷积层显式地建模不同数据流之间的关系图。随后将生成的map输入到双路径交叉注意力中，以并行构建局部和全局关系，从而可以收集更多的区域信息用于青光眼诊断。

Heterologous Stream

ViT在异源数据融合方面也表现出色，尤其是在视觉语言表示学习方面。尽管不同的任务可能采用不同的训练方案，例如监督/自监督学习或紧凑/大规模数据集，但论文仅根据其认知形式将其分为两类：1）视觉语言-预训练，包括视觉-语言预训练（VLP）[204]和对比语言-图像预训练（CLIP）[146]；2）Visual Grounding如Phrase Grounding（PG）、参考表达理解（REC）。更多比较见表五。视觉-语言预训练：由于有限的标注数据，早期的VLP方法通常依赖于现成的目标检测器[204]和文本编码器[5]来提取数据特定的特征以进行联合分布学习。给定图像-文本对，在视觉基因组（VG）上预先训练的目标检测器[205]首先从图像中提取一组以目标为中心的RoI特征。然后将用作视觉标记的RoI特征与用于预定义任务预训练的文本嵌入合并。基本上，这些方法分为双流和单流融合。双流方法包括ViLBERT[137]、LXMERT[138]。单流方法包括VideoBERT[136]、VisualBERT[139]、VL-BERT[140]、UNITER[141]、Oscar[142]、Unified VLP[143]。然而，这些方法严重依赖于视觉提取器或其预定义的视觉词汇表，导致了降低VLP表达能力上限的瓶颈。一些算法如VinVL[145]、ViLT[144]、UniT[149]、SimVLM[150]尝试解决这个问题。除了传统的带有多任务监督的预训练方案外，另一条最新的对比学习路线已经开发出来。相关算法有CLIP[146]、ALIGN[148]、Data2Vec[151]。Visual Grounding：与VLP相比，Visual Grounding具有更具体的目标信号监督，其目标是根据目标对象的相应描述来定位目标对象。在图像空间中，Modulated DETR（MDETR）[152]将其先前的工作[30]扩展到phrase grounding预训练，该训练在一个描述中定位并将边界框分配给每个instance phrase。其他相关算法Referring Transformer[155]、VGTR[154]、TransVG[153]、LanguageRefer[157]、TransRefer3D[158]、MVT 2022[159]、TubeDETR[160]可以参考具体论文。

TABLE V
DETAILS OF VISUAL-LINGUISTIC PRE-TRAINING METHODS, WHERE ● AND ●● DENOTE SINGLE- AND DUAL-STREAM ARCHITECTURE, RESPECTIVELY, AND THE ZERO-SHOT DENOTES THE METHOD CAN BE ZERO-SHOT TRANSFERED INTO DOWN STREAM TASKS. IN THE PRE-TRAINING TASKS, MRM IS MASKED REGION MODELING, OD IS OBJECT DETECTION, SMLM AND BMLM DENOTE BOTH SEQUENTIALLY AND BIDIRECTIONALLY MASKED LANGUAGE MODELING, AND MVM IS MASED VISUAL-TOKEN MODELING.

Methods	Arch.	Visual Token	Pre-training			Zero Shot	Publiction
			Main Dataset(s)	Data Size	Tasks		
Region-Besed Methods							
VideoBERT [136]	●	S3D [209]/w k-means	YouTube Cooking [136]	312K	ITA, MLM, MVM	✓	ICCV 2019
ViLBERT [137]	●●	RoI [204]	CC3M [206]	3.1M	ITA, MLM, MRC-KL	-	NIPS 2019
LXMERT [138]	●●	RoI [204]	VG-QA VQAv2 [210], VG [205], COCO [211], GQA [212]	9.2M	ITA, MLM, MRM, MRC	-	IJCNLP 2019
VisualBERT [139]	●	Faster RCNN [182]	COCO [211]	0.9M	ITA, MLM	-	Arxiv 2019
VL-BERT [140]	●	RoI [204]	CC3M [206], BooksCorpus &English Wikipedia	11M	MLM, MRC	-	ICLR 2020
UNITER [141]	●	RoI [204]	CC3M [206], SBU [213], COCO [211], VG [205]	9.5M	ITA, WRA, MLM/MRM	-	ECCV 2020
Oscar [142]	●	RoI [204] +Tags	COCO [211], GQA [212], CC3M [206], SBU [213], VG [205], Fliker30K [214]	11.4M	ITA, MLM	-	ECCV 2020
Unified-VLP [143]	●	RoI [204]	CC3M [206]	3.1M	SMLM, BMLM	-	AAAI 2020
VinVL(Oscar+) [145]	●	RoI [204]/w NMS +Tags	SBU [213], VG-Qas [205], COCO [211], CC3M [206], GQA [212], Fliker30K [214], VQA [207], OpenImages [215]	8.9M	MLM, ITA	-	CVPR 2021
Feature-Based Methods							
ViLT [144]	●	Patches from ViT [29]	SBU [213], CC3M [206], COCO [211], VG [205]	10M	ITM, MLM	✓	ICML 2021
UniT [149]	●	DETR-ResNet50 [30]	COCO [211], VG [205], VQAv2 [210], SNLI-VE Four LM Datasets	-	OD, 4LM, 2ILM	✓	ICCV 2021
CLIP [146]	●●	ViT [29]	Internet Pairs [146]	400M	Contrasive	✓	ICML 2021
DALL-E [147]	●	dVAE	Extension [147] from COCO	250M	Contrasive	✓	ICML 2021
ALIGN [148]	●	EfficientNet [12]	Noise English al-text data [148]	1.8B	Contrasive	✓	ICML 2021
SimVLM [150]	●	CoAtNet [39]	Noise English al-text data [148]	1.8B	PLM	✓	ICLR 2022
Data2Vec [151]	●	ViT [29]	ImageNet-1k LS-960 Books Corpus & English Wikipedia data	1k 960h 1M	Self-Distillation	✓	Arxiv 2022

讨论和结论

近期改进总结

- 对于分类，深度分层Transformer主干对于降低计算复杂度[41]和避免深层中的过平滑特征[37]、[42]、[66]、[67]是有效的。同时，早期卷积[39]足以捕获低层特征，这可以显著增强鲁棒性并降低浅层的计算复杂性。此外，卷积投影[54]、[55]和局部注意

力机制[35]、[44]都可以改善ViT的局部性。前者[56]、[57]也可能是替代位置编码的新方法；

- 对于检测，Transformer neck从编码器-解码器结构中受益，其计算量比仅编码器Transformer 检测器少[87]。因此，解码器是必要的，但由于其收敛较慢，因此需要更多的空间先验[76]，[80]–[85]。此外，前景采样的稀疏注意力[76]和评分网络[78]、[79]有助于降低计算成本并加速ViT的收敛；
- 对于分割，编码器-解码器Transformer模型可以通过一组可学习的mask embedding[31]、[103]、[202]将三个分割子任务统一为mask预测问题。这种box-free方法在多个基准测试中实现了最新的SOTA性能[202]。此外，特定的混合任务与基于框的ViT[100]级联，该模型在实例分割方面表现出了更高的性能；
- 对于3D视觉，具有评分网络的局部分层Transformer可以有效地从点云数据中提取特征。全局建模能力使Transformer能够轻松聚合曲面点，而不是复杂的局部设计。此外，ViT可以处理3D视觉识别中的多传感器数据，如多视图和多维数据；
- 视觉-语言预训练的主流方法已经逐渐放弃了预训练的检测器[144]，并专注于基于大规模噪声数据集[148]的潜在空间中不同数据流之间的对齐[146]或相似性[151]。另一个问题是使下游视觉任务适应预训练方案，以进行zero-shot迁移[146]；
- 最近流行的多传感器数据融合架构是单流方法，它在空间上连接不同的数据流并同时执行交互。基于单流模型，最近的许多工作致力于寻找一个潜在空间，使不同的数据流语义一致。

ViT的讨论

尽管ViT模型有了很大的发展，但“基本”理解仍然不够。因此，论文将重点审查一些关键问题，以获得深入和全面的理解。

Transformer如何弥合语言和视觉之间的鸿沟

Transformer最初是为机器翻译任务设计的[1]，其中句子的每个单词都被视为表示高级语义信息的基本单元。这些词可以嵌入到低维向量空间中的表示中。对于视觉任务，图像的每个像素都不能携带语义信息，这与传统NLP任务中的特征嵌入不匹配。因此，将这种特征嵌入（即单词嵌入）转移到CV任务中的关键是构建图像到向量的转换并有效地保持图像的特征。例如，ViT[29]在强松弛条件下将图像转换为具有多个低层信息的patch嵌入。

Transformer、Self-Attention与CNN的关系

从CNN的角度来看，其inductive bias主要表现为局部性、平移不变性、权重共享和稀疏连接。这种简单的卷积内核可以在低级语义处理中高效地进行模板匹配，但由于过度的偏差，其上限低于Transformers。从self-attention机制的角度来看，当采用足够数量的head时，它们理论上可以表示任何卷积层[28]。这种完全注意力操作可以结合局部和全局注意力，并根据特征关系动态生成注意力权重。尽管如此，它的实用性仍然不如SOTA CNN，因为精度更低，计算成本更高。从Transformer的角度来看，Dong等人证明，当在没有short connection或FFN的深层上训练self-attention层时，self-attention表现出对“token uniformity”的强烈感应偏差[167]。可以得出结论，Transformer由两个关键组件组成：self-attention聚合token的关系，以及按位置的FFN从输入中提取特征。尽管ViT具有强大的全局建模能力，CNN可以有效地处理低级特征[39]、[58]，增强ViT的局部性[53]、[81]，并通过填充[56]、[57]、[172]附加位置特征。

不同视觉任务的可学习嵌入

各种可学习的嵌入被设计用于进行不同的视觉任务。从目标任务的角度来看，这些嵌入可以分为class token、object query和mask embedding。从结构上看，这些ViT主要采用两种不同的模式，编码器和编码器-解码器。如图15所示，每个结构由三个嵌入级别组成。在位置级别上，编码器Transformer中可学习嵌入的应用被分解为initial token[29]、[87]和later token[42]、[103]，而可学习位置编码[30]、[81]、，[202]和可学习的解码器输入嵌入[76]被应用于编码器-解码器结构。在数量层面上，编码器仅设计应用不同数量的token。例如，ViT[29]、[40]家族和YOLOS[87]将不同数量的token添加到初始层中，而CaiT[42]和Segmenter[103]利用这些token来表示不同任务中最后几层的特征。在编码器-解码器结构中，解码器的可学习位置编码（object query[30]、[81]或mask embedding[202]）被显式地加入到解码器输入[30]，[202]或隐式地加入到解码器输入[80]，[81]。与恒定输入不同，Deformable DETR[76]采用可学习嵌入作为输入，并关注编码器输出。

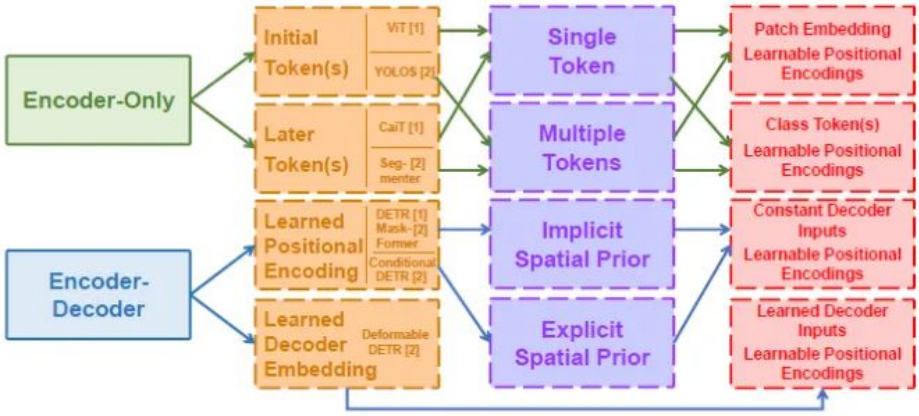


Fig. 15. Taxonomy of the learnable embedding.

在多头注意力机制的启发下，使用多个初始token的策略有望进一步提高分类性能。然而，DeiT[40]指出，这些额外的token将朝着相同的结果收敛，并且不会对ViT有利。从另一个角度来看，YOLOS[87]提供了一种范例，通过使用多个初始token来统一分类和检测任务，但这种编码器的设计只会导致更高的计算复杂度。根据CaiT[42]的观察，较后的class token可以略微降低Transformer的FLOPs并提高性能（从79.9%提高到80.5%）。Segmenter[103]也展示了分割任务的策略效率。与具有多个后期token的仅编码器Transformer不同，编码器-解码器结构减少了计算成本。它通过使用一小组object query（mask embedding）在检测[30]和分割[202]领域标准化了ViT。通过组合后期token和object query（mask embedding），像Deformable DETR[76]这样的结构，它将object query和可学习解码器嵌入（相当于后期token）作为输入，可以将不同任务的可学习嵌入统一到Transformer编码器解码器中。

未来研究方向

ViT已经取得了重大进展，并取得了令人鼓舞的结果，在多个基准上接近甚至超过了SOTA CNN方法。然而，ViT的一些关键技术仍然不足以应对CV领域的复杂挑战。基于上述分析，论文指出了未来研究的一些有前景的研究方向。集合预测：由于损失函数的相同梯度，多类token将一致收敛[40]。具有二分损失函数的集合预测策略已广泛应用于ViT，用于许多密集预测任务[30]，[202]。因此，自然要考虑分类任务的集合预测设计，例如，多类token Transformer通过集合预测预测混合patch中的图像，这与LV-ViT中的数据增强策略类似[43]。此外，集合预测策略中的一对一标签分配导致早期过程中的训练不稳定，这可能会降低最终结果的准确性。使用其他标签分配和损失改进集合预测可能有助于新的检测框架。自监督学习：Transformer的自监督预训练使NLP领域标准化，并在各种应用中取得巨大成功[2]，[5]。由于自监督范式在CV领域的流行，卷积孪生网络使用对比学习来实现自监督预训练，这与NLP领域中使用的masked自动编码器不同。最近，一些研究试图设计自监督的ViT，以弥合视觉和语言之间的预训练方法的差异。它们中的大多数继承了NLP领域中的masked自动编码器或CV领域中的对比学习方案。ViT没有特定的监督方法，但它彻底改变了GPT-3等NLP任务。如前文所述，编码器-解码

器结构可以通过联合学习解码器嵌入和位置编码来统一视觉任务。因此，值得进一步研究用于自监督学习的编码器-编码器Transformer。

结论

自从ViT证明了其在CV任务中的有效性之后，ViT受到了相当大的关注，并削弱了CNN在CV领域的主导地位。本文全面回顾了100多个ViT模型，这些模型相继应用于各种视觉任务（即分类、检测和分割）和数据流（如图像、点云、图像文本对和其他多个数据流）。对于每个视觉任务和数据流，提出了一种特定的分类法来组织最近开发的ViT，并在各种主流基准上进一步评估其性能。通过对所有这些现有方法的综合分析和系统比较，本文总结了显著的性能改进，还讨论了ViT的三个基本问题，并进一步提出了未来投资的几个潜在研究方向。我们希望这篇综述文章能帮助读者在决定进行深入探索之前更好地理解各种视觉Transformer。

参考

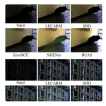
[1] A Survey of Visual Transformers

YOLOv5帮助母猪产仔？南京农业大学研发母猪产仔检测模型并部署到

极市平台

ICCV23｜将隐式神经表征用于低光增强，北大张健团队提出NeRCo

极市平台



ICCV 2023｜南开程明明团队提出适用于SR任务的新颖注意力机制（已开源）

极市平台