**自然语言处理实践报告**

|  |  |
| --- | --- |
| **题目：** | 目标检测Swin Transformer |

|  |  |
| --- | --- |
| 院系： | 计算机科学与工程学院 |
| 专业： | 计算机科学与技术 |
| 姓名： | 宋松波 |
| 学号： | 2201857 |
| 授课教师： | 肖桐、朱靖波 |

2022年 12 月 15 日

目 录

[**1 文章来源 1**](#_Toc104711321)

[**2 研究内容的应用背景与意义 1**](#_Toc104711322)

[**3 对涉及的研究内容进行综述 2**](#_Toc104711323)

[**3.1 网络结构图 3**](#_Toc104711324)

[**3.2 代码实现 4**](#_Toc104711325)

[**3.2.1 环境介绍 4**](#_Toc104711326)

[**3.2.2 数据集介绍 5**](#_Toc104711327)

[**3.2.3 训练模型 5**](#_Toc104711328)

[**3.2.4 测试训练结果 7**](#_Toc104711329)

[**4 进行算法描述与优劣分析 7**](#_Toc104711330)

[**4.1 Patch Partition和Linear Embedding 7**](#_Toc104711331)

[**4.2 Patch Merging 8**](#_Toc104711332)

[**4.3 Swin Transformer Block 9**](#_Toc104711333)

[**4.3.1 W-MSA模块 9**](#_Toc104711334)

[**4.3.2 SW-MSA模块 10**](#_Toc104711335)

[**4.4 Relative Position Bias 11**](#_Toc104711336)

[**4.5 优劣性 11**](#_Toc104711337)

[**5 心得体会 12**](#_Toc104711338)

[**6 参考文献 12**](#_Toc104711339)

**目标检测Swin Transformer**

# 1 文章来源

Swin Transformer是由Microsoft Research Asia中的研究者提出，同时它还获得了ICCV2021年的最佳论文。文章标题是Swin Transformer: Hierarchical Vision Transformer using Shifted Windows[1]，即使用了移动窗口的层级式的Vision Transformer。视觉Transformer具有指导意义的工作是谷歌在ICLR2021上发表的ViT[2]，ViT的提出使得越来越多的研究者、学者相继投入到该方向的研究中，ViT可以说是继2012年AlexNet提出之后的一种全新的研究思路和研究方向。在过去的十几年里，CNN（卷积神经网络）主导了计算机视觉领域，可以说CNN在CV领域具有绝对的统治地位，而ViT的提出使得计算机视觉领域迎来了一波新的浪潮。

Swin Transformer的横空出世使得研究者们更加的相信了transformer可以应用于视觉领域。transformer不仅可以在NLP上大放异彩，也可以在CV上面取得不错的甚至更优的结果。

# 2 研究内容的应用背景与意义

基于自注意力的体系结构，尤其是Transformer，已成为自然语言处理方面的首选模型。Transformer的计算效率和可扩展性使得它能够在NLP上崭露锋芒，同时它随着模型和数据集的增长，性能上并未出现饱和迹象，而且隐隐有种趋向于更好的迹象。受NLP中Transformer扩展成功的启发，我们尝试将标准Transformer[5]直接应用于图像。ViT是把图像分成多个图像块（例如16x16像素大小），并把这些图像块比作 NLP 中的 token。然后直接将 NLP 中的标准 Transformer 编码器应用于这些 “token”，并据此进行图像分类。该工作结合了海量的预训练数据（如谷歌内部3亿图片分类训练库 JFT-300M），在 ImageNet-1K 的 validation 评测集上取得了88.55%的准确率，刷新了该榜单上的纪录。

ViT开创了里程碑式的工作，但ViT 应用 Transformer 比较简单直接，因为其没有仔细考虑视觉信号本身的特点，所以它主要适应于图像分类任务，对于区域级别和像素级别的任务并不是很友好，例如物体检测和语义分割[4]等。进一步说，ViT打破了CV和NLP之间的壁垒，但是由于Transformer与CNN相比，少了一些归纳偏置，使得其在数据集较小的时候性能较差，另外由于其使用低分辨率特征映射且计算复杂度是图像大小的二次方，其体系结构不适合用于密集视觉任务的通用主干网络或输入图像分辨率较高时，Swin Transformer就是在ViT的基础上将层次性、局部性和平移不变性等先验引入Transformer网络结构设计从而能在视觉任务中取得更好的性能，能适用于多种CV任务，且其复杂度相对图片大小为线性相关，计算效率也十分不错。

目前Transformer应用到图像领域主要有两大挑战：（1）视觉实体变化大，在不同场景下视觉Transformer性能未必很好；（2）图像分辨率高，像素点多，Transformer基于全局自注意力的计算导致计算量较大。为此，学术界展开了大量的改进工作。其中，Swin Transformer 骨干网络在物体检测和语义分割任务中大幅刷新了此前的纪录，让学术界进一步确信 Transformer 结构将会成为视觉建模的新主流。

# 3 对涉及的研究内容进行综述

Swin Transformer采用了分层结构，通过从小尺寸的patchs（以灰色勾勒）开始，并逐渐将相邻patchs合并到更深的Transformer层中来构建层次表示。有了这些分层特征映射，Swin Transformer模型可以方便地利用先进技术进行密集预测，如特征金字塔网络（FPN）[3]或U-Net。

图1所提出的Swin Transformer通过在更深的层中合并图像面片（以灰色显示）来构建层次特征图，并且由于只在每个局部窗口（以红色显示）内计算自我注意，因此对于输入图像大小具有线性计算复杂性。同时，这种层次结构具有在不同尺度下建模的灵活性，因此，它能够兼容广泛的视觉任务，即可以作为图像分类和密集识别任务的通用主干。

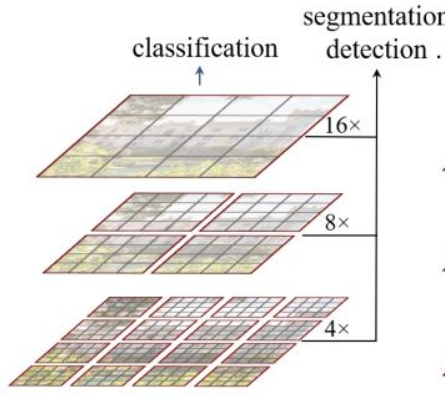


图1 Swin Transformer层级结构

## 3.1 网络结构图

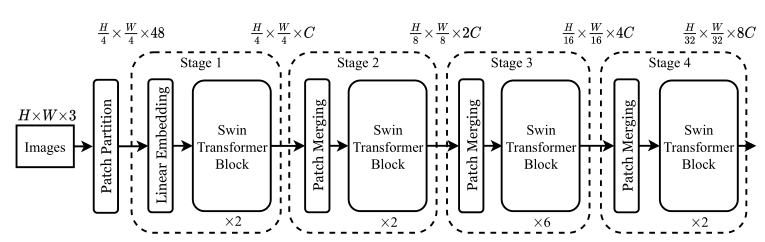


图2 网络骨干结构（Swin-T）

整个模型采取层次化的设计，一共包含4个Stage，每个stage都会缩小输入特征图的分辨率，像CNN一样逐层扩大感受野。下面将简单介绍一下网络的前向传播过程，针对网络结构图中的每一层将在后面部分详细介绍。

网络的前向传播过程：输入图片的尺寸为，通过Patch Partition将输入图片划分为不重合的patch集合，其中每个patch尺寸为，那么每个patch的特征维度为，patch块的数量为；此时，进入Stage1部分，先通过一个Linear Embedding层将输入划分后的patch特征维度变成C，这里的C为超参数，然后输入Swin Transformer Block，在这里先把Swin Transformer Block看成黑盒，若没有多加约束，它的输入输出维度不变；之后进入Stage2部分，先通过一个Patch Merging，Patch Merging会让输入进行降维，同时通道变为原来的二倍，此时维度为，往后的Stage3、Stage4输入输出维度同理，因此最终的输出维度为。

通过Swin Transformer的骨干网络，完成了图片的特征提取，之后根据任务的不同设置不同的头结构，比如进行ImageNet图片分类，会经过Global average pooling层，最终得到输出的类别数。

这里假设输入图片大小为，基本的维度变化图如下所示。

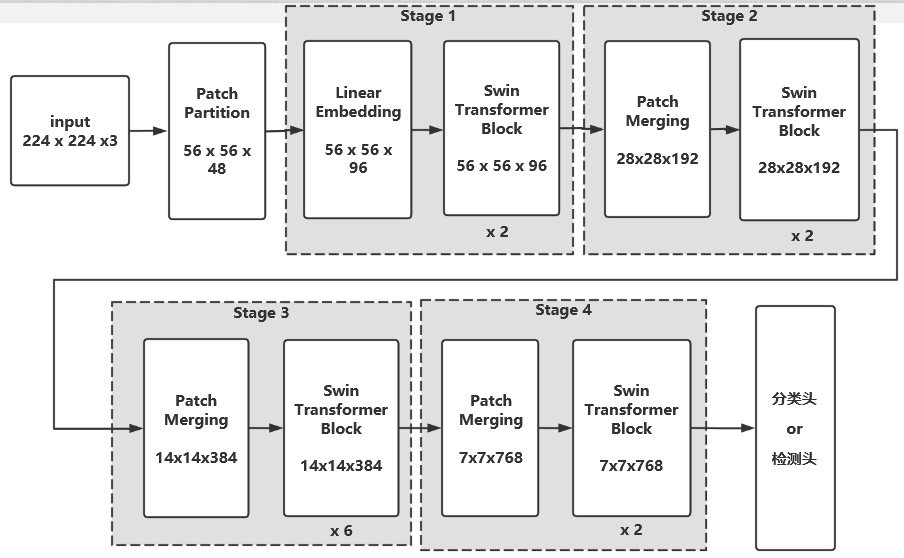


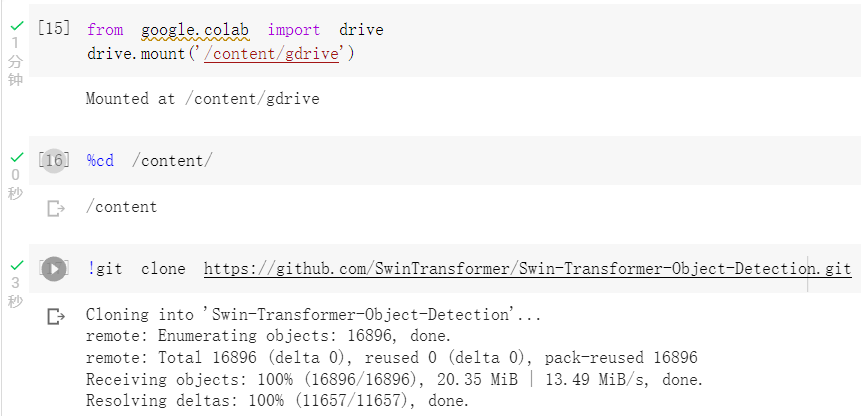
图3 维度变化流程图

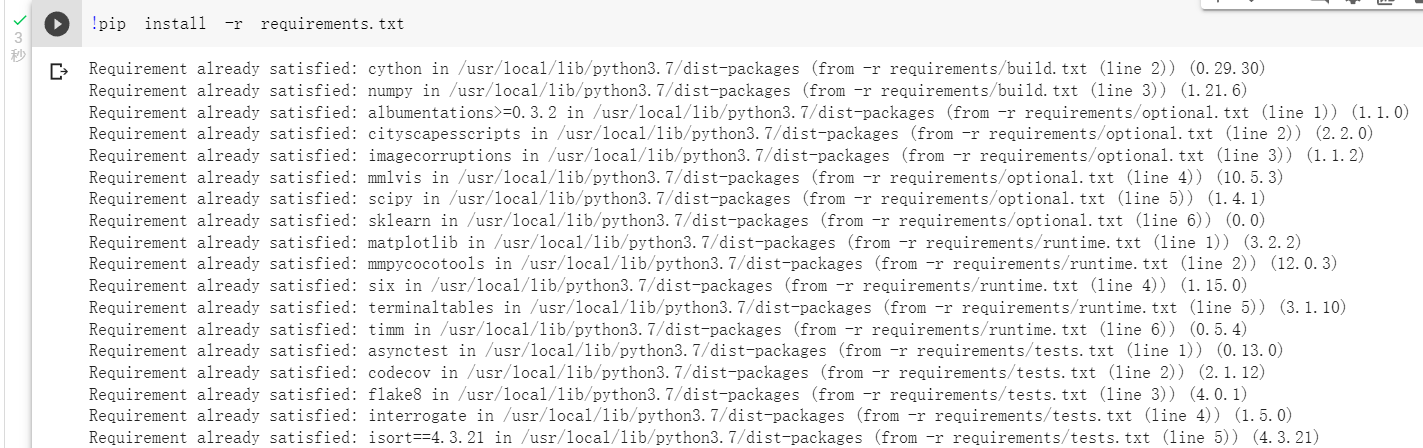
## 3.2 代码实现

### 3.2.1 环境介绍

实现环境选用Google Colab，并在其环境上安装相应的开源包，这里主要介绍开源包，第一个MMDetection：是一个基于 PyTorch 的目标检测开源工具箱，但是依赖 mmcv，支持 pytorch 1.5+。第二个mmcv：类似OpenCV，是一个开源的面向计算机视觉的基础库，高质量实现CUDA算子，使用它的话需要 python 3.6+。

实验环境使用python 3.7，pytorch1.11，GPU型号为Tesla K80，安装开源包结果如下：

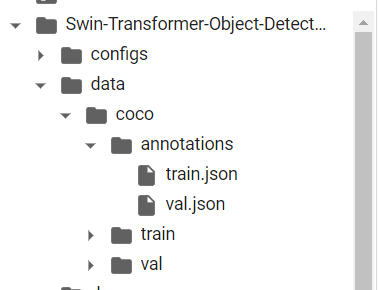
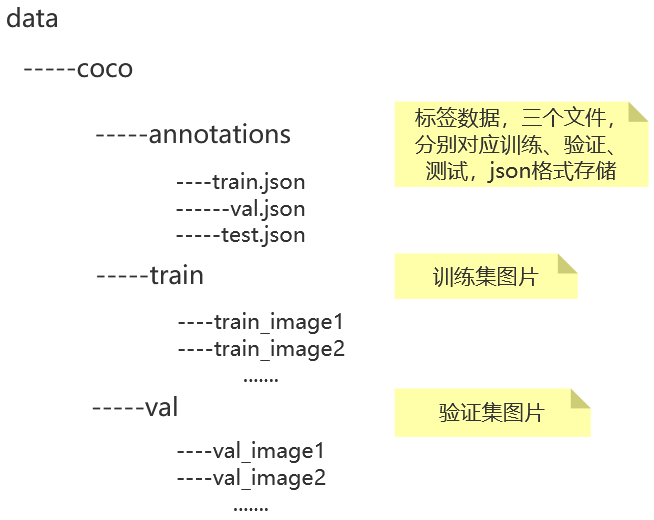




### 3.2.2 数据集介绍

数据集为口罩数据集，有大致8000张图片，训练测试比8:2，类别为0：no-mask，1：mask。

数据集格式以及在代码中存放目录：



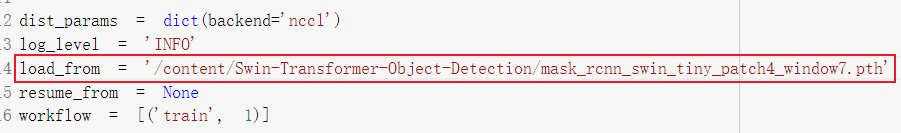
### 3.2.3 训练模型

1、修改类别数目

修改 configs/base/models/mask\_rcnn\_swin\_fpn.py 中 num\_classes 为自己数据集的类别（有两处地方需要修改），这里修改为2。

2、配置权重信息

修改 configs/base/default\_runtime.py 中的 interval，load\_from。interval：dict(interval=1)，表示多少个 epoch 验证一次，然后保存一次权重信息。Load\_from：表示加载哪一个训练好的权重。权重如下：



3、配置数据集路径

修改configs/base/datasets/coco\_instance.py 文件的最上面指定了数据集的路径，因此在项目下新建 data/coco目录，下面四个子目录 annotations和train，val，test，这里test使用了和val相同的数据集。



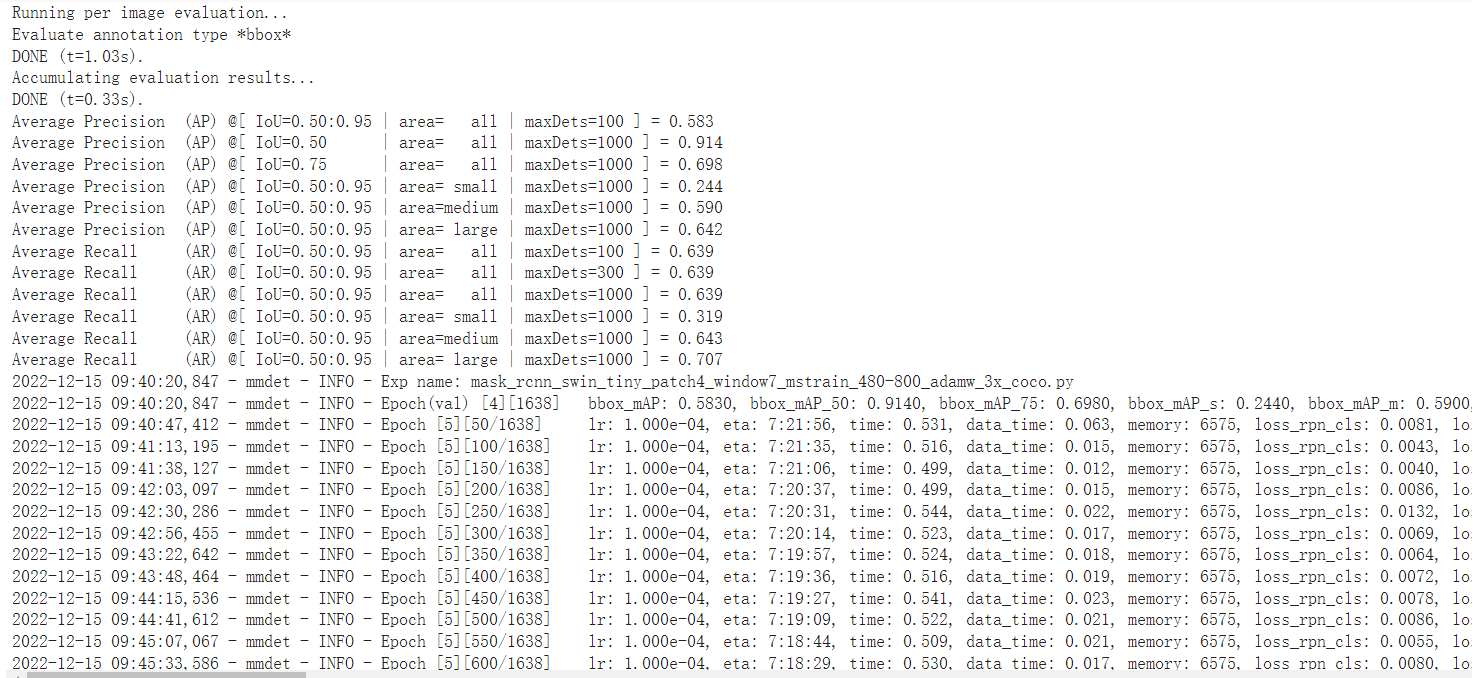
4、修改分类数组

修改mmdet/datasets/coco.py，将CLASSES中填写自己的分类：CLASSES = ('no-mask', 'mask')。

5、开始训练

执行命令：python tools/train.py configs\swin\mask\_rcnn\_swin\_tiny\_patch4\_window7\_mstrain\_480-800\_adamw\_3x\_coco.py。

6、训练过程



### 3.2.4 测试训练结果

选取一张或者几张图片，然后使用latest.pth ，也就是自己训练好的最新的权重文件，默认会放在workdir下，之后执行命令：python demo/image\_demo.py demo/000019.jpg configs\swin\mask\_rcnn\_swin\_tiny\_patch4\_window7\_mstrain\_480-800\_adamw\_3x\_coco.py work\_dirs/mask\_rcnn\_swin\_tiny\_patch4\_window7\_mstrain\_480-800\_adamw\_3x\_coco/latest.pth，

查看结果。





# 4 进行算法描述与优劣分析

## 4.1 Patch Partition和Linear Embedding

我们假设输入的图像为，那么在Patch Partition的过程中会将其分割成大小的块，然后将每一个块在维度方向进行展平，也就是说先前的的矩阵经过此步骤后，长宽会变成，至于最后一个维度，将会变为维，因为我们将其分割成大小的块后，是在维度方向对其进行展平的。展平后，再通过Linear Embedding层，该层就是对原来的第三维度为48维的三维矩阵进行一个映射，然后三维矩阵的最后一个维度的大小就会变为C。

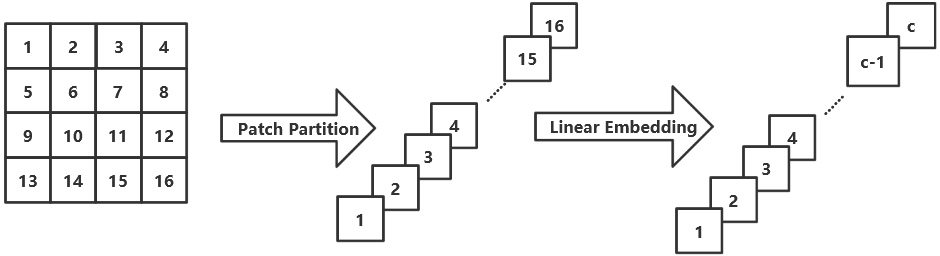


图4 Patch Partition

## 4.2 Patch Merging

Patch Merging层所起到的作用就是将图片下采样，然后在深度方向进行拼接。首先是以的格子为一组，将每组中相同位置的像素抽出，形成一个新的矩阵。以下图为例，原特征矩阵为，那么以格子为1组，会形成4组新的矩阵，每组的矩阵大小为，也就是原来的一半。然后将这4组新形成的矩阵，在深度方向做连接，然后完成LayerNorm，最后再在深度维度进行映射，将深度维度减半。此时就完成了最终的Patch Merging层的输出。

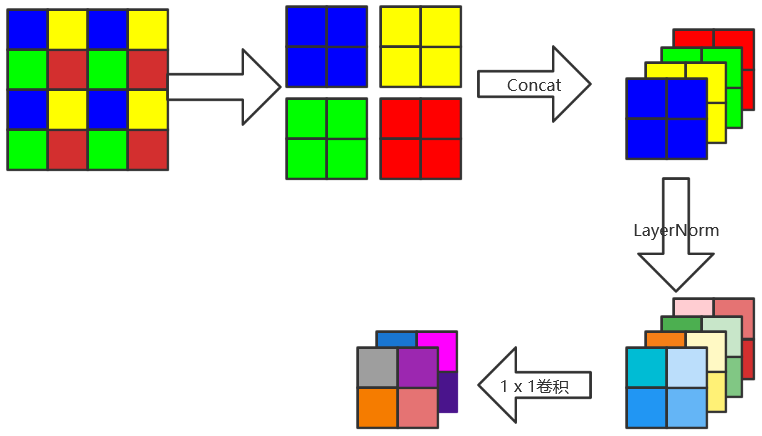


图5 Patch Merging层

## 4.3 Swin Transformer Block

Swin Transformer Block的输入输出不变，每两个连续Block为一组，即一个Window Multi-head Self-Attention和一个Shifted Window Multi-head Self-Attention。使用公式表述Swin Transformer Block中的前向传播过程：

，

，

，

，

这里，和分别表示块的模块和模块的输出特性。

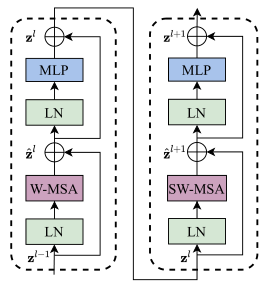


图6 Swin Transformer Block

### 4.3.1 W-MSA模块

W-MSA为Windows Multi-head Self-Attention也就是窗口化的Self-Attention机制。在W-MSA中，首先会将图片特征图划分成若干个 Window窗口，然后再在每个窗口内部进行单独的Self-Attention的计算。也就是说，每个像素只需要和自己所属Window窗口内部的像素进行自注意力即可。这样可以大大减少计算量，计算复杂度的公式如下，但是窗口之间的像素也因此而无法进行通信，从而导致我们的感受野变小，并未起到全局建模的能力。

(1)

(2)

公式（1）是多头注意力机制的计算复杂度，公式（2）是以窗口进行多头注意机制的计算复杂度，可以看出与的数量级差别还是较大的，故W-MSA计算复杂度较小。

### 4.3.2 SW-MSA模块

论文的核心就是SW-MSA，它与W-MSA不同的地方在于这个模块存在滑动，所以叫做Shifted Window，它弥补了W-MSA窗口与窗口间无法进行信息交互的缺陷。

运用移动窗口将之前的4个Windows划分成新的窗口，此时窗口数量为9个，如图7所示。划分出的新的窗口，如果将小块窗口进行填充后，然后合并成一个batch进行计算，这样无形之中增加了计算复杂度。

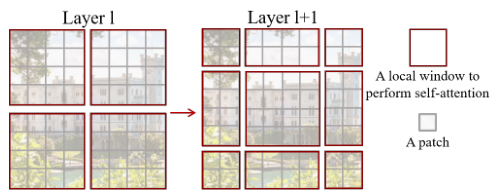


图7 窗口结构

图8为整个模块的计算流程图，在这里，我们将window窗口进行一个划分，如下图9（a）所示，0所在的部分标志为A，1，2所在的标志为C，3，6所在的标志为B，然后进行一个平移，也就是变成图9（b）的形式，这样子的话我们发现又可以将其组成4个新的window进行self-Attention的计算，计算量和以前一致。

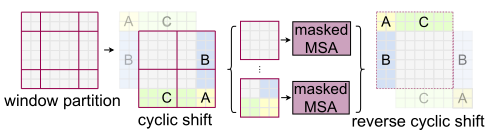


图8 SW-MSA计算流程

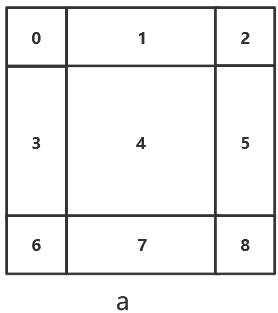
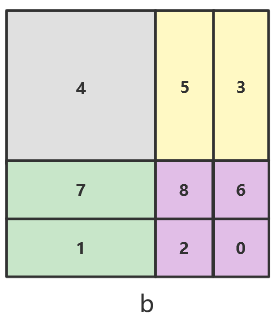
 

图9 窗口移动

这时，我们发现，这些大块中出现了不同的小块，但我们不希望内部的两个原来的window之间信息有交互和混合。当平移后，4号块无交集，内部像素本身就是连续的，因此无关紧要，但5、3号块合起来算，1、7号块合起来算，0、2、6、8号块合起来算时，它们之间存在交集，我们希望在计算区域5的时候不要引入区域3的信息，因此为了解决这一问题，引入mask（掩码）计算，过程如图10所示。通过mask这一巧妙的设计，就有力地解决了这一问题。

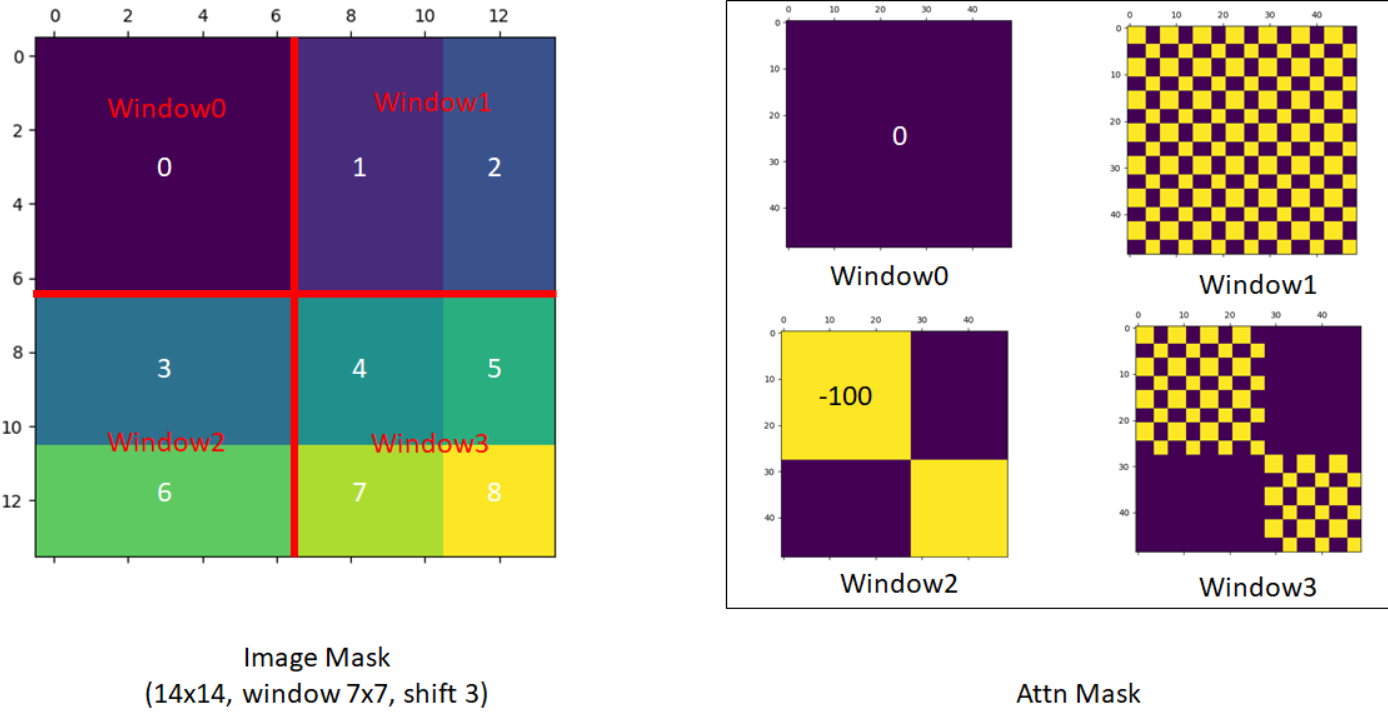


图10 mask计算

## 4.4 Relative Position Bias

Relative Position Bias是相对位置偏移的计算机制。这种机制应用于计算Self-Attention的时候。基本公式如下：

其中是token vector的数目，在计算Attention时加入了相对位置编码，有助于提升模型性能。

## 4.5 优劣性

Swin transformer很好地解决了ViT中对于区域级别和像素级别的任务并不是很友好这一问题，同时它层级化的设计结构能够促进不同的视觉下游任务，在提高模型全局注意力的同时，对于一些局部信息的提取也能够做到不忽视。图片的先验知识的引入，可能会使得模型具有更强的鲁棒性和泛化能力，Swin Transformer在ViT的基础上将层次性、局部性和平移不变性等归纳偏置引入Transformer网络结构设计从而能在视觉任务中取得更好的性能，能适用于多种CV任务。

虽然Swin transformer在视觉的各个任务中表现突出，但它模型的参数或者是计算量对于“平民玩家”来说还是不太友好的，对于计算资源的需求还是比较大。

# 5 心得体会

科技的快速发展，国家的繁荣稳定，为整个计算机行业的发展提供了最有力的保证。伴随着大量各式各样的数据，研究者们通过建模分析，并融入不同且极具巧妙的设计，能够将如科幻电影般的世界变成现实——人工智能战胜日本将棋、国际象棋冠军，阿尔法狗战胜围棋冠军；智能手机不仅可以理解人们说的话，还能在视频通话中进行实时的“机器翻译”；配备了摄像头的“自动防撞的车”保护着人们的生命安全。环顾我们的四周，原来被认为只有人类才能做到的事情，现在人工智能都能毫无差错地完成，甚至试图超越人类。因为人工智能的发展，我们所处的世界正在逐渐变成一个崭新的世界。

基于 Transformer 的预训练模型在 NLP 领域不断展现出惊人的能力，越来越多的工作将 Transformer 引入到了图像以及相关的跨模态领域，Transformer 的自注意力机制以其领域无关性和高效的计算，极大地推动了图像相关任务的发展。视觉—语言预训练任务属于图像领域，其目标是利用大规模图片和语言对应的数据集，通过设计预训练任务学习更加鲁棒且具有代表性的跨模态特征，从而提高下游视觉—语言任务的性能。

无论是transformer还是CNN，它们都在特定的时期贡献着自己独有的特色，transformer和CNN的不断借鉴，将两者的特点不断融合，我相信在现阶段不能解决的问题，终将在某年某月中的某一天找到合适的解决方法。

# 6 参考文献

[1] Liu Z , Lin Y , Cao Y , et al. Swin Transformer: Hierarchical Vision Transformer using Shifted Windows[J]. 2021.

[2] Dosovitskiy A , Beyer L , Kolesnikov A , et al. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale[J]. 2020.

[3] Tsung-Yi Lin, Piotr Dollar, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), July 2017.

[4]Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In European Conference on Computer Vision, pages 213–229. Springer, 2020.

[5] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszko-reit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In Advances in Neural In-formation Processing Systems, pages 5998–6008, 2017.

[6] Josh Beal, Eric Kim, Eric Tzeng, Dong Huk Park, Andrew Zhai, and Dmitry Kislyuk. Toward transformer-based object detection. arXiv preprint arXiv:2012.09958, 2020.

[7] Irwan Bello, Barret Zoph, Ashish Vaswani, Jonathon Shlens, and Quoc V. Le. Attention augmented convolutional net-works, 2020.

[8] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934, 2020.

[9] Navaneeth Bodla, Bharat Singh, Rama Chellappa, and Larry S. Davis. Soft-nms– improving object detection with one line of code. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Oct 2017.

[10] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delv-ing into high quality object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recogni-tion, pages 6154–6162, 2018.

[11] Yue Cao, Jiarui Xu, Stephen Lin, Fangyun Wei, and Han Hu. Gcnet: Non-local networks meet squeeze-excitation net-works and beyond. In Proceedings of the IEEE/CVF Inter-national Conference on Computer Vision (ICCV) Workshops,Oct 2019.

[12]Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural net-works. In Advances in neural information processing sys-tems, pages 1097–1105, 2012.

[13] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. arXiv preprint arXiv:2102.12122, 2021.

[14] Minghao Yin, Zhuliang Yao, Yue Cao, Xiu Li, Zheng Zhang, Stephen Lin, and Han Hu. Disentangled non-local neural networks. In Proceedings of the European conference on computer vision (ECCV), 2020.

[15] Li Yuan, Yunpeng Chen, Tao Wang, Weihao Yu, Yujun Shi, Francis EH Tay, Jiashi Feng, and Shuicheng Yan. Tokens-to-token vit: Training vision transformers from scratch on imagenet. arXiv preprint arXiv:2101.11986, 2021.