分类号 TP242.6 学号 21053048

UDC 681.5 密级 公开

工学硕士学位论文

面向地铁排爆环境的救援机器人 RGB-D 语义分割方法研究

硕士生姓名 郝以慈

学科专业 控制科学与工程

研究方向 智能机器人技术

指导教师 周宗潭 教授

协助指导教师 XXX 讲师

国防科技大学研究生院

二〇二四年十月

RGB-D Semantic Segmentation for Rescue Robots in Explosive Ordnance Disposal Environments

Candidate: Yici Hao

Supervisor: Prof. Zongtan Zhou

Associate Supervisor: Lec. Jingsheng Tang

A dissertation

Submitted in partial fulfillment of the requirements

for the degree of Master of Engineering

in Control Science and Engineering

Graduate School of National University of Defense Technology

Changsha, Hunan, P. R. China

October, 2024

独 创 性 声 明

本人声明所呈交的学位论文是我本人在导师指导下进行的研究工作及取得的研究成果。尽我所知，除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表和撰写过的研究成果，也不包含为获得国防科技大学或其它教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示谢意。

学位论文题目：面向地铁排爆环境的救援机器人 RGB-D 语义分割方法研究

学位论文作者签名：_____ 日期： 年 月 日

学位论文版权使用授权书

本人完全了解国防科技大学有关保留、使用学位论文的规定。本人授权国防科技大学可以保留并向国家有关部门或机构送交论文的复印件和电子文档，允许论文被查阅和借阅; 可以将学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存、汇编学位论文。

（保密学位论文在解密后适用本授权书。）

学位论文题目：面向地铁排爆环境的救援机器人 RGB-D 语义分割方法研究

学位论文作者签名：_____ 日期： 年 月 日

作者指导教师签名：_____ 日期： 年 月 日

目 录

表 目 录

图 目 录

摘　要

国防科技大学是一所直属中央军委的综合性大学。1984 年, 学校经国务院、中央军委和教育部批准首批成立研究生院，肩负着为全军培养高级科学和工程技术人才与指挥人才，培训高级领导干部，从事先进武器装备和国防关键技术研究的重要任务。国防科技大学是全国重点大学，也是全国首批进入国家“211 工程”建设并获中央专项经费支持的全国重点院校之一。学校前身是 1953 年创建于哈尔滨的中国人民解放军军事工程学院，简称“哈军工”。

关键词: **国防科技大学**; 211; **哈军工**

Abstract

National University of Defense Technology is a comprehensive national key university based in Changsha, Hunan Province, China. It is under the dual supervision of the Ministry of National Defense and the Ministry of Education, designated for Project 211 and Project 985, the two national plans for facilitating the development of Chinese higher education.

NUDT was originally founded in 1953 as the Military Academy of Engineering in Harbin of Heilongjiang Province. In 1970 the Academy of Engineering moved southwards to Changsha and was renamed Changsha Institute of Technology.The Institute changed its name to National University of Defense Technology in 1978.

Key Words: NUDT; MND; ME

符号使用说明

HPC	高性能计算 (High Performance Computing)
cluster	集群
Itanium	安腾
SMP	对称多处理
API	应用程序编程接口
PI	聚酰亚胺
MPI	聚酰亚胺模型化合物，N- 苯基邻苯酰亚胺
PBI	聚苯并咪唑
MPBI	聚苯并咪唑模型化合物，N- 苯基苯并咪唑
PY	聚吡咙
PMDA-BDA	均苯四酸二酐与联苯四胺合成的聚吡咙薄膜
ΔG	活化自由能 (Activation Free Energy)
χ	传输系数 (Transmission Coefficient)
E	能量
m	质量
c	光速
P	概率
T	时间
v	速度

第一章　绪论

1.1　研究背景与意义

1.1.1　研究背景

地铁能够在短时间内运送大量乘客，从而减少城市拥堵、缓解地面交通压力，是现代城市交通的重要组成部分。但是，受到公司运营成本、人员流动量大和人员流动速度较快等因素的限制，我国很多城市的地铁安检工作却较为宽松，并不能有效地防止爆炸物进入地铁环境。爆炸事件一旦发生，将对地下从横交错的城市水网、城市电网等公共设施造成破坏，不仅会对疏散乘客和修复基础设施等工作带来诸多不便，危害公众的生命财产安全，还会对城市的高效运转和社会经济的快速发展产生巨大的不利影响。

传统的地铁排爆工作主要依赖排爆专家等技术人员深入爆炸环境排除爆炸物。然而，传统排爆方法却有很多弊端。首先，爆炸物危害极大，一旦发生爆炸，将对排爆专家生命安全造成巨大危害。其次，地铁内部的排椅和扶杆等结构化物体挤占了地铁内部空间，导致地铁内部空间较为狭窄。如果排爆专家身着排爆服进行排爆作业，空间移动将会收到很大限制。

随着智能机器人技术的快速发展，救援机器人在排爆领域逐渐开展应用。相比于传统排爆方法，救援机器人排爆展现出很多优势。首先，救援机器人具有一定的自主能力，可以代替人类进入危险区域进行排爆作业，极大地降低了人员伤亡的风险。其次，救援机器人体型较小便于在地铁排爆环境中移动，视角较低便于发现座椅下等隐匿的爆炸物，方便快速推进排爆工作。再次，救援机器人可以携带多种传感器和设备，提高排爆的效率和准确性。

推动救援机器人在排爆环境中的自主作业，需要救援机器人感知周边环境。常见的传感器有激光雷达、RGB 相机和 RGB-D 相机。其中，激光雷达通过发射激光并捕捉物体表面反射的激光来生成高精度的三维点云数据。但是，激光雷达一方面存在近距离盲区导致无法感知距离较近的物体，在狭窄的地铁环境中不利于感知环境。另一方面，激光雷达获取环境数据稀疏，并且没有颜色信息和纹理信息。此外，激光雷达的成本也比较高。RGB 相机可以获取场景中物体丰富的颜色信息、形状信息和纹理信息，但是无法获取深度信息。RGB-D 相机集成了 RGB 相机和深度相机的功能。其中，RGB 相机捕捉彩色图像，深度相机负责测量每个像素点到相机的距离，生成深度图像。因此，RGB-D 相机能够提供丰富的视觉信息，助力在狭窄的地铁环境中感知环境。

语义分割技术是地铁排爆场景语义理解的核心技术之一。相比于基于边框的目标检测而言，语义分割能够对场景进行更加精细和准确的解析。语义分割通过通过给图像中的每个像素分配一个特定的类别，实现像素级图像理解。通过像素级的图像理解，语义分割可以理解场景结构和布局，精确地识别和分类图像中的不同对象，为救援机器人在地铁排爆场景中的定位、建图、规划和决策等任务提供丰富的语义信息。

RGB-D 语义分割的数据由彩色图像和对应的深度图像组合而成，两种图像虽然具有相同的数据结构，但是模态不同、数据内容不同，属于异质数据。如何获取不同模态数据的有效信息、利用不同模态数据的互补性来提高语义分割的准确度是 RGB-D 语义分割的核心问题。为了提升救援机器人感知系统的能力，本文的研究聚焦于 RGB-D 语义分割的异质数据特征提取与融合这个关键内容。

1.1.2 研究意义

本文的研究意义在于以下 XX 个方面：（1）本文构建了一个地铁排爆场景的语义分割数据集，包括地铁闸机、楼梯、地铁内部常见物体以及模拟的管状爆炸物，覆盖了地铁场景从进站到乘车的几种常见场景。（2）本文提出了一种 XX 方法。（3）本文提出了一种 XX 方法。（4）本文将上述两种算法应用到实际的救援机器人上，能够使救援机器人更好地理解地铁排爆场景，为救援工作的定位、建图、规划和决策提供丰富语义信息。由此可见，本文的研究能为救援机器人提供可靠的语义信息，理解地铁排爆场景，进一步推动救援机器人全自主地铁排爆，具有较强的实际应用价值。

1.2 国内外研究现状

1.2.1 三维传感器的研究现状

相比于二维传感器而言，三维传感器可以捕获深度，获得场景的空间信息。常见的三维传感器主要包括激光雷达和 RGB-D 相机。如1.1 所示。

(a) 激光雷达　(b) 双目深度相机　(c) TOF 深度相机　(d) 结构光深度相机

图 1.1　常见的三维传感器

在激光雷达中，由激光器发射出的脉冲激光在打到周围物体的表面时会发生反射，一部分反射激光会被激光雷达的接收器捕获，通过分析激光遇到目标对象

后的在空中的折返时间，可以计算出激光雷达到目标对象的距离。激光雷达通过从上到下逐层发射脉冲激光，得到目标对象上全部目标点的数据，比如空间坐标、反射率和表面纹理等，通过对这些数据进行成像处理，得到精准的三维点云图像。根据成像原理，激光雷达有效探测距离可达上百米。但是激光雷达存在近距离盲区，无法捕捉自身周围 1 米内的物体信息。因此，激光雷达更适用于室外场景。

RGB-D 相机整合了 RGB 相机和深度相机的优势，将两者集于一身。在获得彩色图像的同时，也获得了对应的深度信息，继而实现对周围环境的三维信息捕获。深度相机的探测距离较短，但是近距离视野盲区半径可低至 30 厘米左右，相比于激光雷达，更适合在室内场景作业的机器人进行环境感知。

根据获取三维数据时是否主动发射出光波，RGB-D 相机主要分为三种：被动式、主动式和多模态融合式。依据这三种分类，发展出了不同的商用方案，主要有：双目视觉方案、飞行时间 (Time of Flight, TOF) 方案、结构光方案。

在双目视觉方案中，相机有两个类似人眼位置排列的 RGB 相机，利用这两个 RGB 相机获得不同位置得到的图像数据，基于图像特征点匹配原理对有视角差异的图像数据进行映射，通过计算图像特征点间的位置偏差，可以获取物体三维几何信息。双目相机主要优点在于其硬件要求低，因此成本也低。但是缺点也很明显。一方面，由于纯视觉方案对图像进行特征点的提取和匹配计算量大，因此对算法要求高，实时性效果较差。另一方面，因为只有 RGB 一种传感器，对环境特征点的提取和匹配误差会较大，因此算法产生的深度精度不高。如果场景特征不明显还会导致匹配失败，因此对纹理单调的场景不适用。目前，双目深度相机在国外比较知名的生产商有：大疆，Intel, Stereolabs，Leap Motion 等。

在飞行时间方案中，相机通过连续主动发射激光脉冲，再用传感器接收返回的激光脉冲，利用激光脉冲在空中的飞行时间来计算相机到目标物体的深度信息。由于使用激光脉冲进行深度特测，因此 TOF 深度相机检测距离远，在激光能量充足的条件下可达几十米。此外，激光脉冲在受环境光干扰的情况下，对成像效果影响较小。但是，TOF 深度相机的成本相对较高、体积较大。目前，TOF 深度相机在国外比较知名的生产商有：海康威视、联想、MicroSoft 等。

在结构光方案中，相机通过近红外发射器，将具有一定结构特征的红外光线连续投影到目标三维空间的表面，然后基于接收模组收集并解析被物体表面反射回来的红外光线，得到物体的位置和深度信息。因为，这种具备一定结构特征的红外光线会随着被摄物体深度的不同而产生不同的相位信息，运算单元可以将这种结构特征的变化换算成深度信息，并以此来获得三维结构。结构光深度相机在近距离范围内精度和分辨率较高高，帧率可达 60FPS。但是，缺点也比较明显。一方面，结构光容易受环境光干扰，室外成像效果比室内差。另一方面，随检测距

离增加精度会变差，因此，在远距离视野的精度较差。目前，结构光深度相机在国外比较知名的生产商有：奥比中光，Apple，MicroSoft，Intel。

三种商用方案的粗略对比如1.1 所示。

表 1.1　深度相机性能对比

相机类型	双目深度相机	TOF 深度相机	结构光深度相机
成像原理	双目特征匹配	飞行时间	激光散斑编码
分辨率	高	低	中
成像精度	中	中	中
制造成本	低	高	中

通过对比以上三种方案可以看出，相比于双目视觉方案，结构光方案在环境适应性、近距离场景数据精度、实时性等方面表现较好；相比于飞行时间方案，结构光方案的功耗更小，技术更成熟，制造成本更低。因此，本文选择的传感器是 Intel 公司的 Realsense 系列的 D435i，该款 RGB-D 相机采用结构光原理。

1.2.2　传统的 RGB-D 语义分割

传统的语义分割方法指不使用深度学习的方法。这种方法主要分为三个阶段。在选择特征阶段，这类方法依赖手工设计的特征，比如颜色、纹理、形状等局部的外观属性，和一些局部特征描述子。在分类输出阶段，将这些特征送入浅层机器学习分类模型，比如支持向量机（Support Vector Machine，SVM）和随机森林等，从而预测类别。在结果优化阶段，用图模型，比如马尔可夫随机场 (Markov Random Fields，MRF) 和条件随机场 (Conditional Random Fields，CRF)，从而优化预测结果。

有一部分研究聚焦于手工特征的设计。Shotton 等人 [**?**] 设计了一种新型低级特征语义纹理森林进行语义分割。Scharwächter 等人 [**?**] 联合处理了颜色、纹理和深度信息，通过随机决策森林快速推断出街景的粗略布局。

由于基于像素点的语义分割方法没有考虑像素与临近像素之间的关系，因此容易产生不一致的结果。基于超像素的方法在一定程度上缓解了这个问题，该方法将在同一个局部区域中的像素强制预测成相同类别。Gupta 等人 [**?**] 通过通用性和特异性特征来编码物体的外观和几何结构，将数据集中的超像素分类做为主要的物体类别，用随机森林和 SVM 分类，进一步提高了语义分割的准确性。

虽然基于超像素的方法对像素特征的提取更加鲁棒，但是超像素中的像素依然不能保持完全一致。概率图模型在一定程度上增强了空间一致性，从而缓解了

这个问题。条件随机场提供一个概率框架，将输出间的关系描述为观测特征的函数。Ladicky 等人 [**?**] 针对 CRF 模型定义了一个全局能量函数，它结合了滑动窗口检测器的结果，以及基于像素的低水平一元和成对关系。Cadena 等人 [**?**] 提出了一种有效的策略来诱导用于推理的 CRF 的图结构，增强了空间一致性。

传统的语义分割方法虽然取得了一些成果，但是人工设计的特征很难准确地表述复杂的场景环境，并且浅层机器学习分类模型对非线性函数的拟合能力有限。因此，针对较为复杂的分类问题，传统的语义分割方法效果很难提升。

1.2.3 基于深度学习的 RGB-D 语义分割

Hinton 等人 [**?**] 在 2006 年首次提出深度学习 (Deep Learning，DL) 之后，深度学习快速发展，语义分割研究也取得了突破性进展。与传统的语义分割方法相比，基于深度学习的语义分割方法能利用深度神经网络超强的非线性拟合能力，获取更多、更高级的语义信息来表达图像中的信息。

1.2.3.1 基于卷积神经网络的 RGB-D 语义分割

卷积神经网络 (Convolutional Neural Network，CNN) 的出现，极大地提升了语义分割地性能。Long 等人 [**?**] 提出的全卷积神经网络 (Fully Convolutional Network，FCN)，第一次将深度神经网络引入语义分割领域。FCN 推广了原有的 CNN 结构，利用特定的卷积层替换了常规卷积网络的全连接层，使得用于分类的 CNN 网络被转化为分割网络。同时，利用转置卷积层实现上采样，使得网络可以执行密集推理并学习到图像中每个像素的语义标签。此外，FCN 可以处理任意大小的图像，语义分割速度也得到显著提升。由于第一次实现了对图片进行端到端的训练，所以后续关于语义分割的研究几乎都借鉴了全卷积神经网络结构。

针对 RGB-D 双模态的语义分割，Couprie 等人 [**?**] 提出了一种前融合方案。具体做法是将视频流逐帧分解，将彩色图片和深度图片拼连起来构成一个四通道的输入，放入卷积神经网络的四个输入，该方法实现了视频流的 RGB-D 实时语义分割，但是并没有区分不同模态的输入。Gupta 等人 [**?**] 对深度数据进行编码为三个通道，包含水平视差 (Horizontal disparity)，地上高度 (Height above groud) 和重力夹角 (the Angle the pixel’s local surface normal makes with the inferred gravity direction) 等信息, 该编码结构加强了对深度信息的利用，但是不足在于需要相机的位姿等额外数据，并且编码计算代价较高。

1.2.3.2 基于 transformer 的 RGB-D 语义分割

1.2.3.3 基于 mamba 的 RGB-D 语义分割

1.3 主要研究内容与论文组织结构

1.3.1 主要研究内容

1.3.2 论文组织结构

第二章　论文正文

本章将进入论文排版的正文,按元素分主要包括：字体段落，图片表格，公式定理，参考文献这几部分。这个样例文件将包括模板中使用到的所有格式、模板中自定义命令到或者特有的东西，都将被一一介绍，希望大家在排版自己的学位论文前能细致的看一遍，记住样例的格式和方法，方便上手。

2.1　字体段落

陈赓（1903 年 2 月 27 日－ 1961 年 3 月 16 日），原名陈庶康，中国湖南湘乡人，军事家。出生将门，其祖父为湘军将领陈翼怀。

Adobe 中文字体有四种：

楷体\kai：陈赓，中国湖南湘乡人，军事家。出生将门，其祖父为湘军将领陈翼怀。1952 年筹办并任人民解放军军事工程学院第一任院长兼政委，培养国防科技人才。1955 年被授予大将军衔。

仿宋\fs：陈赓，中国湖南湘乡人，军事家。出生将门，其祖父为湘军将领陈翼怀。1952 年筹办并任人民解放军军事工程学院第一任院长兼政委，培养国防科技人才。1955 年被授予大将军衔。

黑体\hei：陈赓，中国湖南湘乡人，军事家。出生将门，其祖父为湘军将领陈翼怀。1952 年筹办并任人民解放军军事工程学院第一任院长兼政委，培养国防科技人才。1955 年被授予大将军衔。

宋体就是正文字体了。下面测试字体大小，LaTeX 默认的列表环境会在条目之间插入过多的行距，在下面这种情况可能正好，若用户需要正文行距的列表环境，可以使用 compactitem 环境，记住这点很重要，不要再用那种自己修改itemsep的傻傻的办法了。

初号 陈赓大将

小初 陈赓大将

一号 陈赓大将

小一 陈赓大将

二号 陈赓大将

小二 陈赓大将

三号 陈赓大将

小三 陈赓大将

四号 陈赓大将

小四 陈赓大将

五号 陈赓大将

小五 陈赓大将

2.2 表格明细

表格是论文的重要组成部分，我们从简单的表格讲起，到复杂的表格为止。

模板中关于表格的宏包有三个：booktabs、array 和 longtabular。三线表建议使用 booktabs 中提供的，包含 toprule、midrule 和 bottomrule 三条命令，简单干脆！它们与 longtable 能很好的配合使用。下面来看一个表格实例：

表 2.1 模板文件。如果表格的标题很长，那么在表格索引中就会很不美观，所以要像 chapter 那样在前面用中括号写一个简短的标题。这个标题会出现在索引中。

文件名	描述
nudtpaper.ins	LaTeX 安装文件，docstrip[a]
nudtpaper.dtx	所有的一切都在这里面[b]。
nudtpaper.cls	模板类文件。
nudtpaper.cfg	模板配置文。cls 和 cfg 由前两个文件生成。
bstutf8.bst	参考文献 Bibtex 样式文件。
mynudt.sty	常用的包和命令写在这里，减轻主文件的负担。

[a] 表格中的脚注

[b] 再来一个

表 2.1 列举了本模板主要文件及其功能，基本上来说论文中最可能用到的就是这种表格形式了。请大家注意三线表中各条线对应的命令。这个例子还展示了如何在表格中正确使用脚注。如果你不需要在表格中插入脚注，可以将 minipage 环境去掉。由于 LaTeX 本身不支持在表格中使用`\footnote`，所以我们不得不将

表格放在小页中，而且最好将表格的宽度设置为小页的宽度，这样脚注看起来才更美观。

另外六院的同学在使用模板时需要使用一种固定宽度（往往是页宽，下面的例子由 rongdonghu 提供）的表格，内容需要居中且可以自动调整。解决办法是自定义了一种`tabularx`中的 **Z** 环境，在论文模板中，该命令已添加到`mynudt.sty`中。下面是这种情况的实例：

表 2.2　Reed Solomon 码的典型应用

应用领域	编码方案
磁盘驱动器	RS(32,28,5) 码 [a]
CD	交叉交织 RS 码 (CIRC)
DVD	RS(208,192,17) 码、RS(182,172,11) 码
光纤通信	RS(255,229,17) 码

我们经常会在表格下方标注数据来源，或者对表格里面的条目进行解释。前面的脚注是一种不错的方法，如果你不喜欢 minipage 方法的脚注。那么完全可以在表格后面自己写注释，比如表 2.3。

表 2.3　复杂表格示例 1

y / x	First Half		Second Half	
	1st Qtr	2nd Qtr	3rd Qtr	4th Qtr
East*	20.4	27.4	90	20.4
	30.6	38.6	34.6	31.6
West**	30.6	38.6	34.6	31.6

*：东部

**：西部

此外，表 2.3 同时还演示了另外三个功能：1）通过 tabularx 的 `|X|` 扩展实现表格内容自动调整；2）通过命令 `\backslashbox` 在表头部分插入反斜线（WORD 中很简单，但 LaTeX 做表格需要一定的（极大的）想象力）；3）就是使用`multirow`和`multicolumn`命令。

不可否认 LaTeX 的表格功能没有想象中的那么强大，不过只要你足够认真，足够细致，那么同样可以排出来非常复杂非常漂亮的表格。可是科技论文中那么复杂表格有什么用呢？上面那个表格就够用啦。

浮动体的并排放置一般有两种情况：1）二者没有关系，为两个独立的浮动体；2）二者隶属于同一个浮动体。对表格来说并排表格既可以像表 2.4、表 2.5 使用小

页环境，也可以如表 2.6使用子表格来做。图与表同出一源，后面我们将讲解子图 (subfloat) 的例子。

表 2.4　第一个并排子表格

111	222
222	333

表 2.5　第二个并排子表格

111	222
222	333

表 2.6　并排子表格

(a) 第一个子表格

111	222
222	333

(b) 第二个子表格

111	222
222	333

如果您要排版的表格长度超过一页，那么推荐使用 longtable 命令。这里随便敲入一些无关的文字，使得正文看上去不是那么的少。表 2.7 就是 longtable 的简单示例。

表 2.7　实验数据

测试程序	正常运行时间 (s)	同步时间 (s)	检查点时间 (s)	卷回恢复时间 (s)	进程迁移时间 (s)	检查点文件（KB）
CG.A.2	23.05	0.002	0.116	0.035	0.589	32491
CG.A.4	15.06	0.003	0.067	0.021	0.351	18211
CG.A.8	13.38	0.004	0.072	0.023	0.210	9890
CG.B.2	867.45	0.002	0.864	0.232	3.256	228562
CG.B.4	501.61	0.003	0.438	0.136	2.075	123862
CG.B.8	384.65	0.004	0.457	0.108	1.235	63777
MG.A.2	112.27	0.002	0.846	0.237	3.930	236473
MG.A.4	59.84	0.003	0.442	0.128	2.070	123875
MG.A.8	31.38	0.003	0.476	0.114	1.041	60627
MG.B.2	526.28	0.002	0.821	0.238	4.176	236635
MG.B.4	280.11	0.003	0.432	0.130	1.706	123793
MG.B.8	148.29	0.003	0.442	0.116	0.893	60600
LU.A.2	2116.54	0.002	0.110	0.030	0.532	28754

续下页

续表 2.7　实验数据

测试程序	正常运行时间 (s)	同步时间 (s)	检查点时间 (s)	卷回恢复时间 (s)	进程迁移时间 (s)	检查点文件（KB）
LU.A.4	1102.50	0.002	0.069	0.017	0.255	14915
LU.A.8	574.47	0.003	0.067	0.016	0.192	8655
LU.B.2	9712.87	0.002	0.357	0.104	1.734	101975
LU.B.4	4757.80	0.003	0.190	0.056	0.808	53522
LU.B.8	2444.05	0.004	0.222	0.057	0.548	30134
EP.A.2	123.81	0.002	0.010	0.003	0.074	1834
EP.A.4	61.92	0.003	0.011	0.004	0.073	1743
EP.A.8	31.06	0.004	0.017	0.005	0.073	1661
EP.B.2	495.49	0.001	0.009	0.003	0.196	2011
EP.B.4	247.69	0.002	0.012	0.004	0.122	1663
EP.B.8	126.74	0.003	0.017	0.005	0.083	1656

另外，有的同学不想让某个表格或者图片出现在索引里面，那么请使用命令`\caption*{}`，这个命令不会给表格编号，也就是出来的只有标题文字而没有“表 XX”，“图 XX”，否则索引里面序号不连续就显得不伦不类，这也是 LaTeX 里星号命令默认的规则。

2.3　绘图插图

本模板不再预先装载任何绘图包（如 pstricks，pgf 等），完全由你自己来决定。个人觉得 pgf 不错，不依赖于 Postscript。此外还有很多针对 LaTeX 的 GUI 作图工具，如 XFig(jFig), WinFig, Tpx, Ipe, Dia, Inkscape, LaTeXPiX, jPicEdt 等等。本人强烈推荐 Ipe。

一般图形都是处在浮动环境中。之所以称为浮动是指最终排版效果图形的位置不一定与源文件中的位置对应，这也是刚使用 LaTeX 同学可能遇到的问题。如果要强制固定浮动图形的位置，请使用 float 宏包，它提供了`[H]`（意思是图片就给我放在这里Here）参数，但是除非特别需要，不建议使用`[H]`，而是推荐使用`[htbp]`，给 LaTeX 更多选择。比如图 2.1。

若子图共用一个计数器，那么请看图 2.2，它包含两个小图，分别是图 2.2(a)和图 2.2(b)。这里推荐使用`\subfloat`，**不要再用**`\subfigure`和`\subtable`。

而下面这个例子显示并排 3×2 的图片，见图2.3:

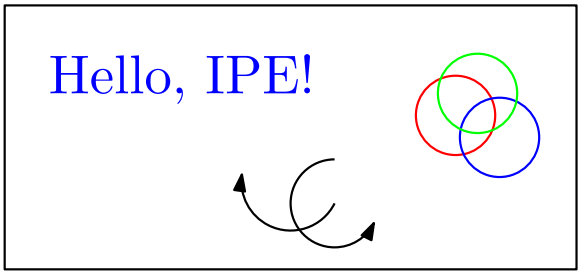


图 2.1　利用 IPE 制图

(a) 第一个小图形

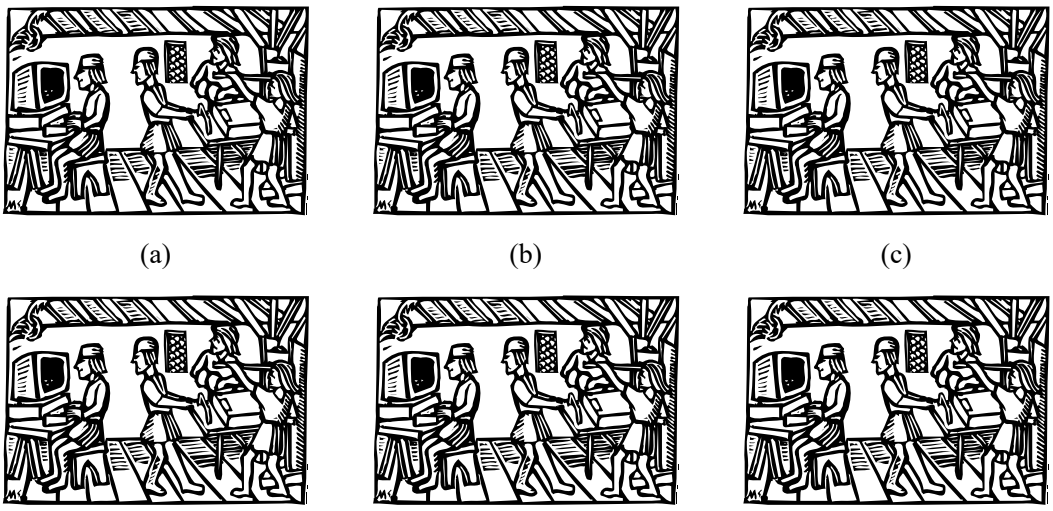
(b) 第二个小图形。如果标题很长的话，它会自动换行，这个 caption 就是这样的例子

图 2.2　包含子图形的大图形

要注意，图2.3例中 `qquad` 相当于`\hspace{2em}`，也就是 2 个字符的宽度，约 0.08 倍页宽，图片宽度设定为 0.27 倍页宽是合适的；在该环境中，尽量不要手动换行，所以，不妨自己计算一下！

如果要把编号的两个图形并排，那么小页 (minipage) 就非常有用了，可以分别参考图2.4和图2.5。其实这个例子和表格一节中并排放置的表格一摸一样。

图形就说这么多，因为大家在写论文是遇到的最大问题不是怎么把图插进去，而是怎样做出专业的、诡异的、震撼的图片来，记得在这时参考前面推荐的那些



(a)　(b)　(c)

(d)　(e)　(f)

图 2.3　并排图片

图 2.4　并排第一个图

图 2.5　并排第二个图

工具吧，当然必不可少的是 Matlab 了，至于如何加入中文标注、支持中文等等可以上网去查，但这里推荐一点，用好 export 命令，使得插入图片时尽可能的不要缩放，保证图文的一致性。

2.4　公式定理

贝叶斯公式如式 (2.1)，其中 $p(y|\mathbf{x})$ 为后验；$p(\mathbf{x})$ 为先验；分母 $p(\mathbf{x})$ 为归一化因子，这是实际应用中十分恐怖的一个积分式。

$$p(y|\mathbf{x}) = \frac{p(\mathbf{x},y)}{p(\mathbf{x})} = \frac{p(\mathbf{x}|y)p(y)}{p(\mathbf{x})} \tag{2.1}$$

论文里面公式越多，TEX 就越 happy。再看一个 amsmath 的例子：

$$\det \mathbf{K}(t=1,t_1,\ldots,t_n) = \sum_{I\in\mathbf{n}} (-1)^{|I|} \prod_{i\in I} t_i \prod_{j\in I} (D_j + \lambda_j t_j) \det \mathbf{A}^{(\lambda)}(\overline{I}|\overline{I}) = 0. \tag{2.2}$$

大家在写公式的时候一定要好好看 amsmath 的文档，并参考模板中的用法：

$$\begin{aligned}\int_a^b \left\{ \int_a^b [f(x)^2 g(y)^2 + f(y)^2 g(x)^2] - 2f(x)g(x)f(y)g(y)\,dx \right\} dy \\ = \int_a^b \left\{ g(y)^2 \int_a^b f^2 + f(y)^2 \int_a^b g^2 - 2f(y)g(y) \int_a^b fg \right\} dy\end{aligned}$$

再看2.3:

$$\begin{aligned} C(z) &= [z^n]\left[\frac{e^{3/4}}{\sqrt{1-z}} + e^{-3/4}(1-z)^{1/2} + \frac{e^{-3/4}}{4}(1-z)^{3/2} + O\Big((1-z)^{5/2}\Big) \right] \\ &= \frac{e^{-3/4}}{\sqrt{\pi n}} - \frac{5e^{-3/4}}{8\sqrt{\pi n^3}} + \frac{e^{-3/4}}{128\sqrt{\pi n^5}} + O\left(\frac{1}{\sqrt{\pi n^7}}\right) \end{aligned} \tag{2.3}$$

当然了，数学中必不可少的是定理和证明：

定理 2.1: *假定 X 的二阶矩存在:*

$$O_R(\mathbf{x},F) = \sqrt{\frac{\mathbf{u}_1^T \mathbf{A} \mathbf{u}_1}{\mathbf{u}_1^T \mathbf{B} \mathbf{u}_1}} = \sqrt{\lambda_1}, \tag{2.4}$$

其中 $\mathbf{A}$ 等于 $(\mathbf{x}-EX)(\mathbf{x}-EX)^T$，$\mathbf{B}$ 表示协方差阵 $E(X-EX)(X-EX)^T$，λ_1 $\mathbf{u}_1$ 是 λ_1 对应的特征向量，

对于希腊符号使用mathbf命令可能有些问题，所以建议对符号用bm加粗，记得用\up<greek>切换正体符号，下面看几个例子：\gamma斜体代表变量 γ，

\bm{\upgamma}正体代表向量$\boldsymbol{\upgamma}$,。\Gamma正体代表操作符号Γ,\bm{\Gamma}正体粗体代表矩阵形式$\boldsymbol{\Gamma}$，\varGamma斜体代表变量$\varGamma$。另外对于大小写斜体的加粗可以见$\boldsymbol{\gamma}$和$\boldsymbol{\varGamma}$，但是这两种科技论文中很少出现，这里只做测试。非符号普通向量就用\mathbf吧：$\mathbf{x}_k, \mathbf{X}_k$。完整测试如下$\omega, \boldsymbol{\omega}, \mathbf{\omega}, \boldsymbol{\mathbf{\omega}}, \Omega, \mathbf{\Omega}, \varOmega, \boldsymbol{\varOmega}$。

证明：上述优化问题显然是一个 Rayleigh 商问题。我们有

$$O_R(\mathbf{x}, F) = \sqrt{\frac{\mathbf{u}_1^T \mathbf{A}\mathbf{u}_1}{\mathbf{u}_1^T \mathbf{B}\mathbf{u}_1}} = \sqrt{\lambda_1}, \tag{2.5}$$

其中 λ_1 下列广义特征值问题的最大特征值：

$$\mathbf{A}\mathbf{z} = \lambda \mathbf{B}\mathbf{z}, \mathbf{z} \neq 0.$$

$\mathbf{u}_1$ 是 λ_1 对应的特征向量。结论成立。

下面来看看算法环境的定义和使用。我们知道，故障诊断的最终目的，是将故障定位到部件，而由于信号 – 部件依赖矩阵的存在，因此，实质性的工作是找出由故障部件发出异常信号，不妨称为源异常信号，而如前所述，源异常信号与异常信号依赖矩阵 $\mathbf{S_a}$ 的全零列是存在一一对应的关系的。因此，我们只要获得了 $\mathbf{S_a}$ 的全零列的相关信息，也就获得了源异常信号的信息，从而能进一步找到故障源。通过以上分析，我们构造算法2.1，用于实现非回路故障诊断。

算法 2.1 非回路故障诊断算法

已知: 信号 – 部件依赖矩阵 $\mathbf{A}$，信号依赖矩阵 $\mathbf{S}$，信号状态向量 α
求: 部件状态向量 γ

1: $\mathbf{P} \leftarrow (< \alpha >)$
2: $\mathbf{S_a} \leftarrow \mathbf{P^T S P}$
3: **for** $i = 1$ to S_a 的阶数 m **do**
4: $\quad s_i \leftarrow s_i$ 的第 i 个行向量
5: **end for**
6: $\beta_a \leftarrow \neg (s_1 \vee s_2 \vee \cdots \vee s_m)^T$
7: $\beta \leftarrow \mathbf{P}\beta_a$
8: $\gamma \leftarrow \mathbf{A}\beta$

第一类故障回路推理与非回路故障推理是算法基本相同，稍微不同的是 β_a 的计算。因为第一类故障回路中的信号全部可能是源异常信号，因此我们不必计算 $\beta_a = \neg \left([s_1 \vee s_2 \vee \cdots \vee s_m]^T \right)$，而直接取 $\beta_a = \underbrace{\left[\begin{array}{cccc} 1 & 1 & \cdots & 1 \end{array} \right]^T}_{m}$，将 β_a 代入算

法2.1，有

$$\beta = \mathbf{P}\beta_a = \mathbf{P}\underbrace{\left[\begin{array}{cccc} 1 & 1 & \cdots & 1 \end{array}\right]^T}_{m} = \alpha$$

因此一类故障回路的推理算法变得相当简单，例如算法2.2

算法 2.2 第一类故障回路诊断算法

已知: 信号 – 部件依赖矩阵 $\mathbf{A}$，信号状态向量 α
求: 部件状态向量 γ
1: $\gamma \leftarrow \mathbf{A}\alpha$

2.5 参考文献

当然参考文献可以直接写 bibitem，虽然费点功夫，但是好控制，各种格式可以自己随意改写，在 nudtpaper 里面，建议使用 JabRef 编辑和管理文献，再结合`bstutf8.bst`，对中文的支持非常不错，格式也很规范。

本模板推荐使用 BIBTEX，样式文件为 bstutf8.bst，符合学校的参考文献格式（如专利等引用未加详细测试）。看看这个例子，关于书的[? ?]，还有这些[? ? ?]，关于杂志的[? ? ?]，硕士论文[? ?]，博士论文 [? ?]，标准文件[?]，会议论文[? ?]，技术报告[?]。中文参考文献[?] **特别注意**，需要在`bibitem`中增加`language`域并设为`zh`，英文此项可不填，之后由`bstutf8`统一处理 (具体就是决定一些文献在中英文不同环境下的显示格式，如等、etc)。若使用`JabRef`，则你可按下面步骤来设置：选择 Options→Set Up General Fields，在`General:`后加入`language`就可以了。

有时候不想要上标，那么可以这样 [**?**]，这个非常重要。

2.6 代码高亮

有些时候我们需要在论文中引入一段代码，用来衬托正文的内容，或者体现关键思路的实现。在模板中，统一使用 `listings` 宏包，并且设置了基本的内容格式，并建议用户只使用三个接口，分别控制：编程语言，行号以及边框。简洁达意即可，下面分别举例说明。

首先是设定语言，来一个 C 的，使用的是默认设置：

```
1 void sort(int arr[], int beg, int end)
2 {
3   if (end > beg + 1)
4   {
5     int piv = arr[beg], l = beg + 1, r = end;
6     while (l < r)
```

```
7       {
8         if (arr[l] <= piv)
9           l++;
10        else
11          swap(&arr[l], &arr[--r]);
12      }
13      swap(&arr[--l], &arr[beg]);
14      sort(arr, beg, l);
15      sort(arr, r, end);
16    }
17  }
```

当我们需要高亮 Java 代码，不需要行号，不需要边框时，可以：

```
// A program to display the message
// "Hello World!" on standard output

public class HelloWorld {

  public static void main(String[] args) {
    System.out.println("Hello World!");
  }

}  // end of class HelloWorld
```

细心的用户可能发现，行号被放在了正文框之外，事实上这样是比较美观的，如果有些用户希望在正文框架之内布置所有内容，可以：

```
1 #!/usr/bin/perl
2 print "Hello, world!\n";
```

好了，就这么多，listings 宏包的功能很强大也很复杂，如果需要自己定制，可以查看其手册，耐心阅读总会找到答案。**注意:** 当前代码环境中文注释的处理还不是很完善，对于注释请妥善处理。在本模板中，推荐算法环境或者去掉中文的 listings 代码环境。如果需要包含中文注释，不要求代码高亮，就用 code 环境，这个环境是 Verbatim 的定制版，简单有效，调用的是 fancyvbr 宏包，用户可在 mynudt.sty 中修改它的外观等等。这里我们还可以给代码加上标签。

hello.c

```
1 public class HelloWorld {
2     public static void main(String[] args) {
3         System.out.println("Hello World!");
4     }
5 }     // 世界，你好!
```

2.7 符号列表

前面的话：2.2 版本后默认使用 nomencl 环境，如果你还是希望使用传统的definition.tex，那么只需注释掉顶层文件中的 nomenclature 即可。

符号列表使用的是nomencl包，自己简单定制了下，使用方法分为四步：

1. 将\makenomenclature语句放在正文前，即\begin{document}前面；
2. 将\printnomenclature放在论文中，我在例子中将符号列表放在了英文摘要的后面，正文第一章的前面，当然，你可以根据自己的需要或者教研室的规范放置在合理的位置上，为了页面引用的正确，在这句话前面放上\cleardoublepage；
3. 使用\nomenclature命令在论文的各个位置上添加符号定义，语法后面会讲到；
4. 编译。编译需要首先运行一遍 xelatex，之后运行

```
makeindex -s nomencl.ist -o thesis.nls thesis.nlo
```

你可以把这句编译命令放在makepdf.bat中第一个xelatex thesis下面。然后双击makepdf.bat就可以了，论文模板中已经为你添加上了，如果你强烈不想使用 nomencl 环境，只要把它注释掉（前面加rem）就可以。另外，由于我使用的是 VIM 来编辑 TEX 代码，具体到每个编辑器（诸如 WinEDT，TeXWorks 等）如何设定该命令的快捷按钮，诸位可以搜索网上的教程。

下面简单说明下\nomenclature命令，语法为。这里插入一些随机的文字，希望对你在阅读帮助中的思维没有什么不良的影响。

```
\nomenclature[<prefix>]{<symbol>}{<desc>}{<null>}
```

nomencl模板的默认排序方法可能（大多都）不满足要求，论文模板里，我们通过设定<prefix>来实现符号列表的排序。它分为两部分，比如如[Aa]，第一个字母的含义是：

‘A’ 符号归为拉丁字母

‘G’ 希腊字母

‘X’ 上标

‘Z’ 下标

每个标识后边的字幕a-z作为当前符号组内的排列顺序，比如 β 就可以写成 [Gb]，诸如此类。当然你一定注意到了，这个排序分组的设定只是为了记忆方便，并不是强制的，因此你可以有自己的方案，比如 Z 是 Greek，R 是 Roman 什么的，只要统一就好，只需记住，组间排列是按字母顺序排的。

注意符号表分四列，前三列的含义与命令中相同，最后一列是符号定义时所在的页码。效果看例子，对于下式:

$$\dot{Q} = k \cdot A \cdot \Delta T \tag{2.6}$$

或者:

$$\frac{1}{k}=\left[\frac{1}{\alpha_{\mathrm{i}}\,r_{\mathrm{i}}}+\sum_{j=1}^{n}\frac{1}{\lambda_j}\,\ln\frac{r_{\mathrm{a},j}}{r_{\mathrm{i},j}}+\frac{1}{\alpha_{\mathrm{a}}\,r_{\mathrm{a}}}\right]\cdot r_{\mathrm{reference}} \tag{2.7}$$

注意事项：模板中定制的 nomencl 格式在 mynudt.sty 中，默认是三栏的，分别是："符号"，"定义"，"首次出现页码"，注意这里的符号列表都没有单位，如果你需要额外的栏输入单位（呵呵，聪明的读者可能看出来了，`nomenclature`命令最后一个是空的，就是用来让你赋予她各种意义的）。此时就需要你有一点点动手能力了（其实只要会修改表格就行），方法很简单，比如需要添加"国际单位制"这一栏，则

1. 论文中`\nomenclature`命令的第三个参数就让他代表单位，也可留空；
2. 将`mynudt.sty`中 longtable 的表头添加"国际单位制"几个字，你也可以取其他的名字，放在那个应该出现的位置上；
3. 由于增加了 5 个字，就把前面栏的宽度数字减 5，同时设定第三栏宽度为 5，注意这一步需要你自己调整，记得不要让表格超出边界就行。

2.8 中文习惯

对于 itermize 过大的行间距，用户可以使用 compactitem 环境来替代，但是模板中不进行默认替代，因为只有用户真正发现列表不好看才会找到这里，而且在示例文件中，陈赓大将那个列表环境如果压缩了行距会很不好看。谢谢 ZhangLei 的建议！

一个重要的提示：作者自己的定义命令、包等，不要放在模板里面，请放到`mynudt.sty`中，这样模板时，只要覆盖`nudtpaper.cls`即可。

中文破折号为一个两个字宽垂直居中的直线，输入法直接得到的破折号是两个断开的小短线（——），这看起来不舒服。所以模板中定义了一个破折号的命令`\pozhehao`，请看：

厚德博学，强军兴国 ——国防科大校训

第三章　基于通道交换机制的跨模态 RGB-D 语义分割网络

语义分割是计算机视觉领域的一个重要研究任务，该任务旨在对给定图像的每一个像素进行分类。在地铁排爆场景中，语义分割可以帮助机器人在复杂的场景下理解周边场景，进行定位、导航、排爆等任务。

近年来，卷积神经网络（Convolutional Neural Network，CNN）在高分辨率图像处理方面展示了强大的能力。其中，全卷积网络（Fully Convolutional Network，FCN）是卷积神经网络在语义分割任务的重量级工作。语义分割在彩色数据集上的迅速发展，随着 RGB-D 相机的发展，使用 RGB-D 数据的语义分割任务发展也如火如荼。

彩色数据和深度数据是同质化的图像信息，但是仍然是分属不同模态的数据，其中蕴含着的信息既有本模态的独特性，又有不同模态的互补性。

针对提取模态信息独特性而言，在传统的卷积操作中，卷积操作是为彩色信息设计的，并且在网络的训练过程中会和深度信息共享卷积参数，这样并不利于提取深度信息。

针对提取模态信息互补性而言，如何融合这种互补性信息，继而增强语义分割在 RGB-D 数据上的性能，是 RGB-D 语义分割领域的重中之重。融合策略分为以下三种：前融合、中间融合、决策融合。前融合就是指在输入进网络时，将彩色信息和深度信息做一个拼接操作，一起输入语义分割的网络之中。中间融合就是在进行网络生成特征图之后，对生成的特征图使用融合策略，从而加强语义分割网络对特征的提取能力。决策融合就是在网络进行最终分割结果运算时，将彩色信息的预测结果和深度信息的预测结果进行加强，得到最终的语义分割结构。这三种融合策略并不是互相排斥的，他们可以共存在同一个分割网络之中。由于前融合和决策融合较为简单，因此目前的研究重点主要是在中间层融合。

针对以上两个问题，本章首先提出了一种针对 RGB-D 双模态信息的卷积模块，该模块可以替换传统的卷积操作，应用在 RGB-D 语义分割网络中。其次，采用三种融合策略，构建一种基于特征图交换机制的中间融合机制，以加强 RGB-D 语义分割的性能。具体而言，本章的主要工作包括几个部分：

(1) 针对使用 RGB-D 双模态信息的语义分割网络，设计了一个跨模态卷积（Cross-modal convolution，CMConv），该部分可以更好地提取 RGB-D 信息，促进分割性能的提升。

(2) 针对使用视觉 transformer（vision　transformer，VIT）的结构，提出一种特征交换机制（Feature exchange），该结构可以有效地检测并剔除训练过程中的冗余特征信息，从而增强 RGB-D 语义分割网络的性能。

3.1 模型方法

3.1.1 框架结构

3.1.2 跨模态卷积

彩色信息和深度信息属于不同模态的信息。彩色图片通过对使用 RGB 色彩空间对颜色进行三通道赋值来表示捕获到的信息，而深度图片通过对捕获到的深度信息进行赋值产生的单通道灰白图片来表示捕获到的信息。因此，从本质上说，这两种数据是不同的。

从其表示信息的原理上来说，RGB 值捕获投影图像空间中的光度外观属性，而深度特征编码局部形状信息及其在上下文中的位置信息。对于同样形状的物体，我们希望网络可以提取出相同的特征。然而使用普通的卷积运算时，由于其位置的不同，提取出的特征是不同的，这阻碍了形状不变性的学习。但是，也不能因为追求当前层的形状不变性而简单地将位置信息直接舍弃，因为位置信息在具有更大上下文的后续处理过程中会形成形状信息。与位置相比，形状是物体更固有的属性，与语义联系更为紧密，因而对分割精度更关键。因此，广泛用于使用彩色数据的卷积运算在处理深度数据时，并不高效。

基于深度特征可以表征形状信息和位置信息的模态特性，本章引入跨模态卷积来处理深度特征，以学习形状信息和位置信息重要性之间的自适应平衡，使网络有机会在必要时更多地关注形状信息，从而有利于 RGB-D 语义分割任务。

首先，将深度图片产生的补丁（patch）蕴含的形状信息和位置信息分解为两个独立的部分，得到形状分量（shape component）和位置分量（local component）。深度补丁的平均值描述了该补丁在更大范围内的位置，从而构成了位置分量，表示距离观测点的距离。而剩余的部分描述补丁的相对变化，描述了底层的几何形状，从而构成了形状分量，表示物体的语义。

然后，引入两个可学习的权值分别处理形状分量和位置分量得到形状核（shape kernal）和位置核（local kernal），最后对形状核和位置核加权组合，形成一个形状感知的补丁，并进一步与一个正常的卷积核进行卷积。与原始补丁相比，形状感知补丁能够利用形状核自适应学习形状特征，利用位置核平衡形状和位置对最终预测的贡献。此外，由于形状核和位置核在推理阶段成为常量，将它们融合到下面的卷积核中可以得到一个与普通卷积层相同的网络。

对于一个输入的补丁 $\mathbb{P} \in R^{K_h \times K_w \times C_{in}}$，$K_h$ 和 K_w 是核的空间维度，C_{in} 表示输入特征映射中的通道数，传统卷积层得到的输出特征

$$\mathbb{F} = Conv(\mathbb{K}, \mathbb{P}), \tag{3.1}$$

其中，$\mathbb{K} \in R^{K_h \times K_w \times C_{in} \times C_{out}}$ 表示卷积层中核的可学习权值，C_{out} 表示输出特征映射中的通道数。$\mathbb{F} \in R^{C_{out}}$ 的每个元素计算为:

$$\mathbb{F}_{c_{out}} = \sum_{i}^{K_h \times K_w \times C_{in}} (\mathbb{K}_{i,c_{out}} \times \mathbb{P}_i).$$

可以很容易地看出，$\mathbb{F}$ 通常会随着 $\mathbb{P}$ 的不同值而变化。假设有两个一样的物体在不同的位置，分别用补丁 $\mathbb{P}_1$ 和补丁 $\mathbb{P}_2$ 表示。对应的输出特征：$\mathbb{F}_1$ and $\mathbb{F}_2$ 从卷积层学习：$\mathbb{F}_1 = Conv(\mathbb{K}, \mathbb{P}_1)$，$\mathbb{F}_2 = Conv(\mathbb{K}, \mathbb{P}_2)$。由于 $\mathbb{P}_1$ 和 $\mathbb{P}_2$ 与观测点的距离并不相同，因此它们的特征通常不同，这可能导致不同的预测结果。然而实际上，$\mathbb{P}_1$ 和 $\mathbb{P}_2$ 属于同一个类别，传统的卷积层不能很好地处理这种情况。

但是，这两个补丁的形状是不变量。形状特征用来表征局部特征下的相对深度差异，而这一点却被现有的方法所忽略。鉴于此，我们建议通过对 RGB-D 语义分割的形状进行有效建模来填补这一空白。

基于上述分析，本文提出将输入补丁分解为两个分量：描述补丁位置的位置分量（local component）和表示补丁是什么的形状分量（shape component）。

我们将补丁的平均值称为 $\mathbb{P}_B$，其相对值称为 $\mathbb{P}_S$。

$$\mathbb{P}_B = m(\mathbb{P}),$$
$$\mathbb{P}_S = \mathbb{P} - m(\mathbb{P}),$$

其中 $m(\mathbb{P})$ 是 $\mathbb{P}$ 上的平均函数（在 $K_h \times K_w$ 维上），$\mathbb{P}_B \in R^{1 \times 1 \times C_{in}}$，$\mathbb{P}_S \in R^{K_h \times K_w \times C_{in}}$。

注意，在等式3.1中直接卷积的 $\mathbb{P}_S$ with $\mathbb{K}$ 是次优的，因为来自 $\mathbb{P}_B$ 的值有助于跨块的类别区分。因此，我们的 ShapeConv 利用两个可学习的权重 $\mathbb{W}_B \in R^1$ and $\mathbb{W}_S \in R^{K_h \times K_w \times K_h \times K_w \times C_{in}}$ 来分别消耗上述两个分量。然后以逐元素添加的方式组合输出的特征，这形成具有与原始 $\mathbb{P}$ 相同大小的新的形状感知贴片。

$$\begin{aligned} \mathbb{F} &= ShapeConv(\mathbb{K}, \mathbb{W}_B, \mathbb{W}_S, \mathbb{P}) \\ &= Conv(\mathbb{K}, \mathbb{W}_B \diamond \mathbb{P}_B + \mathbb{W}_S * \mathbb{P}_S) \\ &= Conv(\mathbb{K}, \mathbf{P_B} + \mathbf{P_S}) \\ &= Conv((\mathbb{K}, \mathbf{P_{BS}}), \end{aligned} \tag{3.2}$$

其中，$\diamond$ and $*$ 分别表示基本乘积和形状乘积算子，其被定义为

$$\begin{cases} \mathbf{P_B} = \mathbb{W}_B \diamond \mathbb{P}_B \\ \mathbf{P_{B}}_{1,1,c_{in}} = \mathbb{W}_B \times \mathbb{P}_{B_{1,1,c_{in}}}, \end{cases} \tag{3.3}$$

$$\begin{cases} \mathbf{P_S} = \mathbb{W}_S * \mathbb{P}_S \\ \mathbf{P_S}_{k_h,k_w,c_{in}} = \sum_i^{K_h \times K_w} (\mathbb{W}_{S_{i,k_h,k_w,c_{in}}} \times \mathbb{P}_{S_{i,c_{in}}}), \end{cases} \tag{3.4}$$

其中 c_{in}, k_h, k_w 分别是 C_{in}, K_h, K_w 维度中的元素的索引。

我们通过 $\mathbf{P_B}$ and $\mathbf{P_S}$ 的相加来重建形状感知补丁 $\mathbf{P_{BS}}$，$\mathbf{P_B}$ and $\mathbf{P_S}$，这使得它能够被香草卷积层的内核 $\mathbb{K}$ 平滑卷积。然而，$\mathbf{P_{BS}}$ 配备了通过两个额外权重学习的重要形状信息，使得卷积层专注于仅使用深度值失败的情况。

第 3.1 节中提出的形状转换可以有效地利用补丁。然而，在 CNNs 中用 ShapeConv 代替香草卷积层引入了更多的计算成本，这是由于等式 3 和 4 中的两个乘积运算。为了解决这个问题，我们提出将这两个操作从补丁转移到内核

$$\begin{cases} \mathbf{K_B} = \mathbb{W}_B \diamond \mathbb{K}_B \\ \mathbf{K_B}_{1,1,c_{in},c_{out}} = \mathbb{W}_B \times \mathbb{K}_{B_{1,1,c_{in},c_{out}}}, \end{cases}$$

$$\begin{cases} \mathbf{K_S} = \mathbb{W}_S * \mathbb{K}_S \\ \mathbf{K_S}_{k_h,k_w,c_{in},c_{out}} = \sum_i^{K_h \times K_w} (\mathbb{W}_{S_{i,k_h,k_w,c_{in}}} \times \mathbb{K}_{S_{i,c_{in},c_{out}}}), \end{cases}$$

其中 $\mathbb{K}_B \in R^{1\times1\times C_{in}\times C_{out}}$ and $\mathbb{K}_S \in R^{K_h\times K_w\times C_{in}\times C_{out}}$ 分别表示核的基分量和形状分量，$\mathbb{K} = \mathbb{K}_B + \mathbb{K}_S$。因此，我们将方程 2 的形状转换重新形式化为以下：

$$\begin{aligned} \mathbb{F} &= ShapeConv(\mathbb{K}, \mathbb{W}_B, \mathbb{W}_S, \mathbb{P}) \\ &= Conv(\mathbb{W}_B \diamond m(\mathbb{K}) + \mathbb{W}_S * (\mathbb{K} - m(\mathbb{K})), \mathbb{P}) \\ &= Conv(\mathbb{W}_B \diamond \mathbb{K}_B + \mathbb{W}_S * \mathbb{K}_S, \mathbb{P}) \\ &= Conv(\mathbf{K_B} + \mathbf{K_S}, \mathbb{P}) \\ &= Conv(\mathbf{K_{BS}}, \mathbb{P}), \end{aligned} \tag{3.5}$$

其中 $m(\mathbb{K})$ 是 $\mathbb{K}$ 上的平均函数（在 $K_h \times K_w$ 维度上）。我们要求 $\mathbf{K_{BS}} = \mathbf{K_B} + \mathbf{K_S}$, $\mathbf{K_{BS}} \in R^{K_h\times K_w\times C_{in}\times C_{out}}$。

事实上，ShpeConv 的两个公式，即，等式3.2和等式3.5在数学上是等价的，即，

$$\begin{aligned} \mathbb{F} &= ShapeConv(\mathbb{K}, \mathbb{W}_B, \mathbb{W}_S, \mathbb{P}) \\ &= Conv(\mathbb{K}, \mathbf{P_{BS}}) \\ &= Conv(\mathbf{K_{BS}}, \mathbb{P}), \end{aligned} \tag{3.6}$$

$$\begin{aligned}\mathbb{F}_{c_{out}} &= \sum_{i}^{K_h \times K_w \times C_{in}} (\mathbb{K}_{i,c_{out}} \times \mathbf{P}_{\mathbf{BS}_i}) \\ &= \sum_{i}^{K_h \times K_w \times C_{in}} (\mathbf{K}_{\mathbf{BS}_{i,c_{out}}} \times \mathbb{P}_i),\end{aligned} \tag{3.7}$$

推论阶段。在推理过程中，由于 $\mathbb{W}_B$ and $\mathbb{W}_S$ 这两个附加权重变为常数，因此我们可以将它们融合成 $\mathbf{K_{BS}}$，如**??**(c) 所示 $\mathbf{K_{BS}} = \mathbb{W}_B \diamond \mathbb{K}_B + \mathbb{W}_S * \mathbb{K}_S$。$\mathbf{K_{BS}}$ 与等式3.1中的 $\mathbb{K}$ 共享相同的张量大小，因此，我们的 ShapeConv 实际上与**??**(a) 中的香草卷积层相同。换句话说，当用 ShapeConv 代替香草卷积时，不会引入额外的推理时间。

如引言中所讨论的，深度多模态融合方法主要可以分为基于聚合的融合和基于融合的融合 [4]。由于模态内处理的弱点，最近的基于聚合的工作执行特征融合，同时仍然保持所有模态的子网络 [12，30]。此外，[19] 指出，熔合的性能受到选择熔合哪一层的高度影响。基于对齐的融合方法通过应用相似性规则来对齐多模态特征，其中最大均值差异 MMD)[16] 通常用于测量。然而，仅仅关注统一整个分布可能会忽略每个领域 / 模态中的特定模式 [6，44]。因此，[47] 提供了一种可以缓解这一问题的方法，该方法将模态共同特征相关联，同时保持模态特定信息。还有一部分基于调制的多模态学习文献 [11，13，46]。不同于这些类型的融合方法，我们提出了一种新的融合方法，通过通道交换，这可能享有充分的模型间的相互作用和模态内学习的保证。使用 BN 缩放因子来评估 CNN 通道重要性的想法已经在网络修剪 [33，49] 和表示学习 [40] 中进行了研究。[33] 对缩放因子实施了 101 范数惩罚，并显式地修剪掉满足稀疏性标准的过滤器。在这里，我们将这个想法作为一种自适应工具来确定在哪里交换和融合。CBN [46] 通过以另一种模态为条件调制一种模态的 BN 来执行跨模态消息传递，这显然不同于我们在不同模态之间直接交换信道以进行融合的方法。ShuffleNet [53] 提出在多个组之间移动一部分信道，以在轻量级网络中进行有效传播，这类似于我们交换信道进行消息融合的想法。然而，虽然我们的论文的动机是非常不同的，但交换过程是由 BN 缩放因子自决定的，而不是 ShuffleNet 中的随机交换。

Transformer 最初在自然语言社区中被广泛研究为非递归序列模型 [40]，并且很快被扩展以使视觉语言任务受益。最近，许多研究进一步采用变压器进行计算机视觉任务，具有良好的适应性架构和优化时间表。因此，视觉跨以前的变体已经在许多单模视觉任务中显示出巨大的潜力，例如分类 [6，21]、分割 [44，47]、检测 [3，8，22，48]、图像生成 [16]。然而，直到这项工作的日期，尝试扩展视觉转换器，以处理多模态数据仍然很少。当引入具有复杂对齐关系的多模态数据时，

对模型体系结构的融合方案设计提出了很大的挑战。要回答的关键问题是，不同模态的特征之间的交互应如何以及在何处发生。已有几种基于变换器的视觉语言融合方法，VL-BERT [37] 和 ViLT [17] 中所述的方法。在这些方法中，视觉和语言标记在每个 Transformer 层之前直接级联，使得整体架构与原始 Transformer 非常相似。这种融合通常是比对不可知的，这表明没有明确地利用模态间比对。我们还尝试将类似的融合方法应用于多模态视觉任务（第 4）、第四章。不幸的是，这种直观的 Transformer 融合不能带来有希望的增益，或者甚至可能导致比单模态对应物更差的性能，这主要是由于没有充分利用模态间的相互作用。也有几种尝试用于融合多种视觉模态。例如，TransFuser [26] 利用 Transformer 模块来连接图像的 CNN 主干和 LiDAR 点。与已有的试验不同，本文旨在寻求一种有效且通用的方法，将多个单模态变压器组合起来，并在模型中插入模态间的对齐。这项工作有利于多模态数据的学习过程，同时利用模态间对齐。这种对准在许多视觉任务中自然可用，例如，利用摄像机内函数 / 外函数，世界空间点可以被投影并且对应于摄像机平面上的像素。与不可知论融合（Sec.3.1），该 martaware 融合明确涉及的对齐关系，不同的模式。然而，由于在 Transformer 中引入了模态间投影，因此对准感知融合可能会极大地改变原始模型结构和数据流，这可能会破坏单模态架构设计的成功或预训练期间习得的注意力。因此，可能必须为多模投影和融合确定“正确的”层 / 记号 / 通道，并且还必须为新模型重新设计架构或重新调整优化设置。为了避免处理这些具有挑战性的问题并继承原始单模设计的大部分，我们提出了多模式令牌融合，称为 TokenFusion，它自适应地并有效地融合多个单模变换器。我们的 TokenFusion 的基本思想是修剪多个单模态变换器，然后重新利用修剪后的单元进行多模态融合。我们对每个单模态 Transformer 应用单独的修剪，并且每个修剪的单元由来自其他模态的投影对准特征代替。假设该融合方案对原始的单模态变压器具有有限的影响，因为它保持了重要单元的相对注意关系。TokenFusion 在允许多模态转换器继承来自单模态预训练的参数方面也被证明是上级的，在 ImageNet 上。展示优势

融合视觉变形。与多模态数据的深度融合一直是一个重要的主题，它可能通过利用多个输入源来提高性能，并且还可能进一步释放变压器的力量。然而，将多个现成的单变压器联合收割机组合在一起，同时保证这种组合不会影响其精心设计的单模态设计，这是具有挑战性的。信号装置.[2] 以及 [20] 利用变换器处理连续的视频帧，用于空间 - 时间对准，并通过使多个帧相关来捕获细粒度模式。关于多模态数据，[26，41] 利用 Transformer 模块的动态特性来联合收割机 CNN 主干，以融合红外 / 可见光图像或 LiDAR 点。[9] 将从粗到精的经验从 CNN 融合方法扩展到用于图像处理任务的变换器。[14] 采用变换器联合收割机高光谱图像进

行简单的特征拼接。[24] 在图像补片和音频频谱图补片之间插入中间标记作为瓶颈以隐式地学习模态间对准。然而，这些工作与我们的工作不同，因为我们希望构建一个通用的融合管道，用于组合现成的视觉转换器，而无需重新设计其结构或重新调整其优化设置，同时明确利用模态间的对齐关系。

Vision Transformers 的基本融合

假设我们有第 i 个输入数据 $\boldsymbol{x}^{(i)}$，它包含 M 个模态：$\boldsymbol{x}^{(i)} = \{\boldsymbol{x}_m^{(i)} \in \mathbb{R}^{N\times C}\}_{m=1}^{M}$，其中 N and C 分别表示令牌和输入通道的数量。为了简单起见，我们将在接下来的部分中省略下标 $^{(i)}$。深度多模态融合的目标是确定一个多层模型 $f(\boldsymbol{x})$，期望其输出尽可能接近目标 $\boldsymbol{y}$。具体来说，在这项工作中，$f(\boldsymbol{x})$ 是近似基于 transformerbased 网络架构。假设模型总共包含 L 层，我们表示第 l 层（$l = 1, \ldots, L$）为 $\boldsymbol{e}^l = \{\boldsymbol{e}_m^l \in \mathbb{R}^{N\times C'}\}_{m=1}^{M}$，其中 C' 表示范围内的层的特征通道的数量。最初，使用 $\boldsymbol{e}_m^1$ 的线性投影来获得 $\boldsymbol{x}_m$，这是一种广泛采用的对输入标记（例如，图像块）进行矢量化的方法，使得第一 Transformer 层可以接受标记作为输入。

我们对输入模态使用不同的变换器，并将 $f_m(\boldsymbol{x}) = \boldsymbol{e}_m^{L+1}$ 表示为第 m 个 Transformer 的最终预测。给定第 m 个模态的令牌特征 $\boldsymbol{e}_m^l$，第 l 层计算

翻译到这里懂？1 这是第三章节懂？1 还有什么问题？我试试不动了喊你 1

3.1.3 Vision Transformer 的基本融合

假设我们有第 i 个输入数据 $\boldsymbol{x}^{(i)}$ 它包含 M 个模态：$\boldsymbol{x}^{(i)} = \{\boldsymbol{x}_m^{(i)} \in \mathbb{R}^{N\times C}\}_{m=1}^{M}$，其中，$N$ 和 C 分别表示令牌和输入通道的数量。为了简单起见，我们将在接下来的部分中省略下标 $^{(i)}$。深度多模态融合的目标是确定一个多层模型 $f(\boldsymbol{x})$，期望其输出尽可能接近目标 $\boldsymbol{y}$。具体来说，在这项工作中，$f(\boldsymbol{x})$ 是近似基于 transformerbased 的网络架构。假设模型总共包含 L 层，我们表示第 l 层 $(l = 1, \ldots, L)$ 为 $\boldsymbol{e}^l = \{\boldsymbol{e}_m^l \in \mathbb{R}^{N\times C'}\}_{m=1}^{M}$, 其中 C' 表示范围内的层的特征通道的数量。最初，使用 $\boldsymbol{x}_m$ 的线性投影来获得 $\boldsymbol{e}_m^1$，这是一种广泛采用的对输入标记 (例如，图像块) 进行矢量化的方法，使得第一 Transformer 层可以接受标记作为输入。

我们对输入模态使用不同的变换器，并将 $f_m(\boldsymbol{x}) = \boldsymbol{e}_m^{L+1}$ 表示为第 m 个 Transformer 的最终预测。给定第 m 个模态的令牌特征 $\boldsymbol{e}_m^l$，第 l 层计算

$$\hat{\boldsymbol{e}}_m^l = \mathrm{MSA}\big(\mathrm{LN}(\boldsymbol{e}_m^l)\big),\ \boldsymbol{e}_m^{l+1} = \mathrm{MLP}\big(\mathrm{LN}(\hat{\boldsymbol{e}}_m^l)\big), \tag{3.8}$$

其中 MSA，MLP，和 LN 表示多头自注意、多层感知和层归一化，$\hat{\boldsymbol{e}}_m^l$ 代表 MSA 的输出。

在视觉任务的多模态融合过程中，不同模态的对齐关系可以显式地可用。例如，像素位置通常用于确定图像深度相关性; 并且相机内函数 / 外函数在将 3D 点投影到图像中时很重要。基于对齐信息的参与，我们考虑了以下两种 Transformer

融合方法。

对齐不可知融合不明确使用模态之间的对齐关系。该算法期望从大量的数据中隐式地学习到对齐。对齐不可知融合的一种常用方法是直接拼接多模态输入标记，广泛应用于视觉语言模型。类似地，用于第 l 层的输入特征 $\boldsymbol{e}_l$ 也是不同模态的令牌式级联。尽管对准不可知的融合是简单的并且可以对原始 Transformer 模型具有最小的修改，但是很难直接受益于已知的多模态对准关系。

明确地利用模态间对齐。例如，这可以通过选择对应于相同像素或 3D 坐标的标记来实现。假设 $\boldsymbol{x}_m[n]$ 是第 m 个模态输入 $\boldsymbol{x}_m$ 的第 n 个令牌，其中 $n = 1, \cdots, N_m$。我们将从第 m 个模态到第 m' 个模态的“标记投影”定义为：

$$\mathrm{Proj}^{\mathrm{T}}_{m'}(\boldsymbol{x}_m[n_m]) = h(\boldsymbol{x}_{m'}[n_{m'}]), \tag{3.9}$$

其中 h 可以简单地是身份函数（对于同质模态）或浅多层感知（对于异质模态）。当考虑整个 N 个 token 时，我们可以方便地将“模态投影”定义为 token 投影的串联：projections:

$$\mathrm{Proj}^{\mathrm{M}}_{m'}(\boldsymbol{x}_m) = \big[\mathrm{Proj}^{\mathrm{T}}_{m'}(\boldsymbol{x}_m[1]); \cdots; \mathrm{Proj}^{\mathrm{T}}_{m'}(\boldsymbol{x}_m[N])\big]. \tag{3.10}$$

3.10 仅示出了输入侧的融合策略。我们还可以通过投影和聚合特征嵌入 $\boldsymbol{e}_m$，在不同模态特定模型之间执行中间层或多层融合，这可能实现更多样化和更精确的特征交互。然而，随着基于变换器的模型的复杂性的增长，搜索仅用于两种模态 (例如 2D 和 3D 检测变换器）的最佳融合策略 (例如应用投影和聚集的层和标记）可能会增长为极难解决的问题。为了解决这一问题，我们在第 2.3 节中提出了多模式令牌融合 3.1.4.

3.1.4 多模式令牌融合

正如 **??**描述的, 多模态令牌融合 (TokenFusion) 首先修剪单模态变换器，并进一步重新利用修剪后的单元进行融合。通过这种方式，原始的单模态变压器的信息单元被假定为在很大程度上被保留，而多模态的相互作用可以被涉及以提高性能。

如之前在 [**?**] 中所示，视觉变换器的标记可以在保持性能的同时以分层方式被修剪。类似地，我们可以通过采用评分函数 $s^l(\boldsymbol{e}^l) = \mathrm{MLP}(\boldsymbol{e}^l) \in [0, 1]^N$ 来选择较少信息的标记，该评分函数动态地预测第 l 层和第 m 模态的标记的重要性。为了在 $s^l(\boldsymbol{e}^l)$ 上实现反向传播，我们重新公式化方程中的 MSA 输出 $\hat{\boldsymbol{e}}^l_m$ in 3.8 as

$$\hat{\boldsymbol{e}}^l_m = \mathrm{MSA}\big(\mathrm{LN}(\boldsymbol{e}^l_m) \cdot s^l(\boldsymbol{e}^l_m)\big). \tag{3.11}$$

我们使用 $\mathcal{L}_m$ 来表示第 m 个模态的任务特定损失。为了修剪无信息的标记，我们进一步在 $s^l(\boldsymbol{e}_m^l)$ 上添加标记式修剪损失（l_1-norm）。因此，用于优化的总损失函数被导出为：

$$\mathcal{L} = \sum_{m=1}^{M} \Big(\mathcal{L}_m + \lambda \sum_{l=1}^{L} \big| s^l(\boldsymbol{e}_m^l) \big| \Big), \tag{3.12}$$

其中 λ 是用于平衡不同损耗的超参数。

对于特征 $\boldsymbol{e}_m^l \in \mathbb{R}^{N \times C'}$，令牌式剪枝从所有 N 个令牌中动态检测不重要的令牌。改变不重要的标记或用其他嵌入替换它们，预计对其他信息标记的影响有限。因此，我们提出了一种用于多模态变换器的标记融合过程，因此，我们为多模态变换器提出了一种标记融合程序，用其他模态的标记投影（定义见第 **??**节）替代不重要的标记。由于修剪过程是动态的，*i.e.*，即，在输入特征的条件下，融合过程也是动态的。该过程在每个 Transformer 层之前执行令牌替换，因此第 l 层的输入特征，*i.e.*，$\boldsymbol{e}_m^l$，被重新表述为

$$\boldsymbol{e}_m^l = \boldsymbol{e}_m^l \odot \mathbb{I}_{s^l(\boldsymbol{e}_m^l) \geq \theta} + \mathrm{Proj}_{m'}^{\mathrm{M}}(\boldsymbol{e}_m^l) \odot \mathbb{I}_{s^l(\boldsymbol{e}_m^l) < \theta}, \tag{3.13}$$

其中 $\mathbb{I}$ 是一个断言下标条件的指示符，因此它输出一个掩码张量 $\in \{0,1\}^N$；参数 θ 是一个小阈值（我们在实验中采用 10^{-2}）; 运算符 $\odot$ 表示逐元素乘法。

In 3.13, 如果只有两个模态作为输入，则 m' 将仅仅是除 m 之外的另一模态。对于两个以上的模态，我们将标记预先分配为 $M-1$ 个部分，每个部分都与其他模态中的一个绑定，而不是与它绑定。此预分配的更多细节将在**??**.

3.1.5 剩余位置对准

直接替换代币将冒着完全破坏其原始位置信息的风险。因此，模型仍然可以忽略来自另一模态的投影特征的对齐。为了缓解这个问题，我们采用了残余位置对齐（RPA），利用位置嵌入（PE）的多模式对齐。图3.1 and 图3.2 所示。稍后将详细介绍，RPA 的关键理念在于向后续层注入等效 PE。此外，PE 的反向传播在第一层之后停止，这意味着在整个训练过程中，仅保留第一层处的 PE 的梯度，而对于其余的层则冻结。以这种方式，PE 服务于对齐多模态令牌的目的，而不管原始令牌的替换状态。总之，即使替换了标记，我们仍然保留从另一个模态添加到投影特征的原始 PE。

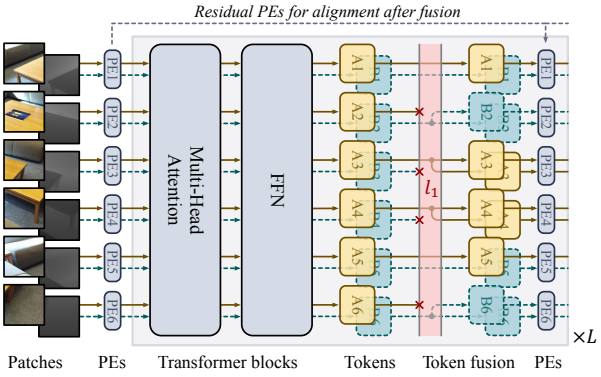


图 3.1　以 RGB 和深度为例，介绍了一种用于同质模态的 TokenFusion 框架。这两种模态都被发送到共享的 Transformer，其中还具有共享的位置嵌入。

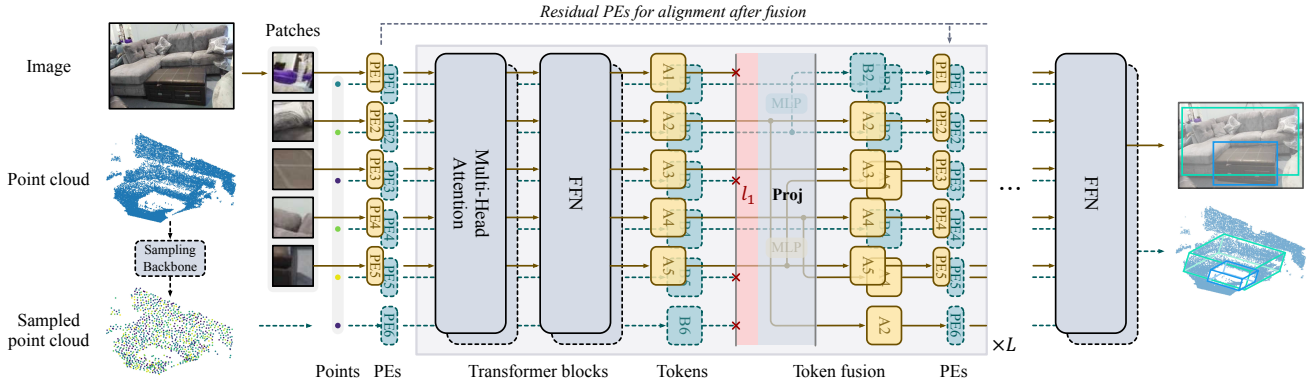


图 3.2　用于点云和图像的异构模态的 TokenFusion 框架。两种模态都被发送到具有单独位置嵌入的单独 Transformer 模块。需要额外的模态间投影（Proj），这与同质模态的融合不同

第四章　非对称结构的实时跨模态 RGB-D 语义分割网络

此处主要是与上一章节的逻辑联系。要写一页。

本章基于非对称的 Transformer+Transformer 结构，提出了一种轻量化的实时跨模态 RGB-D 语义分割网络 CMFormer。具体而言，本章的主要工作包括几个部分：

(1) 提出了一种基于非对称结构的跨模态 RGB-D 语义分割框架，分别处理 RGB 信息和深度信息的语义分割。在 RGB 分支的 VIT（Vision Transformer）中使用 top-k 稀疏注意力（top-k Sparse Attention, top-k SA）, 用以减少注意力机制计算时的信息冗余，降低模型大小，提高模型计算速度。在 RGB 分支的 VIT（Vision Transformer）使用轻量级的 mix-transformer 处理深度特征，该结构在处理深度信息的同时极大的压缩了模型大小。

(2) 跨模态融合模块中使用特征选择模块，用以提取 RGB 模态和深度模态的有效信息。使用基于跨模态注意力引导的特征融合模块，用以融合 RGB 模态和深度模态，最后将融合的模态替换深度模态的原有信息。

(3) 使用轻量级 MLP 解码器来解码浅层特征的语义信息，实现语义分割。

4.1　模型方法

4.1.1　框架结构

本章设计的 CMFormer 算法采用双分支结构处理 RGB 信息和深度信息，通过四个阶段的降采样对不同尺度的信息进行特征编码，采取中间层融合策略对不同尺寸的不同模态的信息进行融合，最后对融合的特征图进行解码实现 RGB-D 语义分割。

本章的算法主要有以下四个部分：基于 top-k transformer 的 RGB 信息处理分支、基于 mix-transformer 的深度信息处理分支、基于注意力引导的跨模态信息融合结构和轻量级的 MLP 解码器。如所示。网络以三通道的 RGB 信息和三通道的深度信息作为输入，top-k transformer 负责处理 RGB 信息，mix-transformer 负责处理深度信息。由于第一阶段浅层特征比较明显，因此不进行特征融合，但是在之后的阶段都进行特征融合，基于注意力引导的跨模态信息融合结构负责融合不同阶段不同尺度的彩色信息和深度信息，将得到的融合信息放入深度信息处理通道。轻量级的 MLP 解码器负责第四阶段后的解码。

总体结构如4.7 所示。

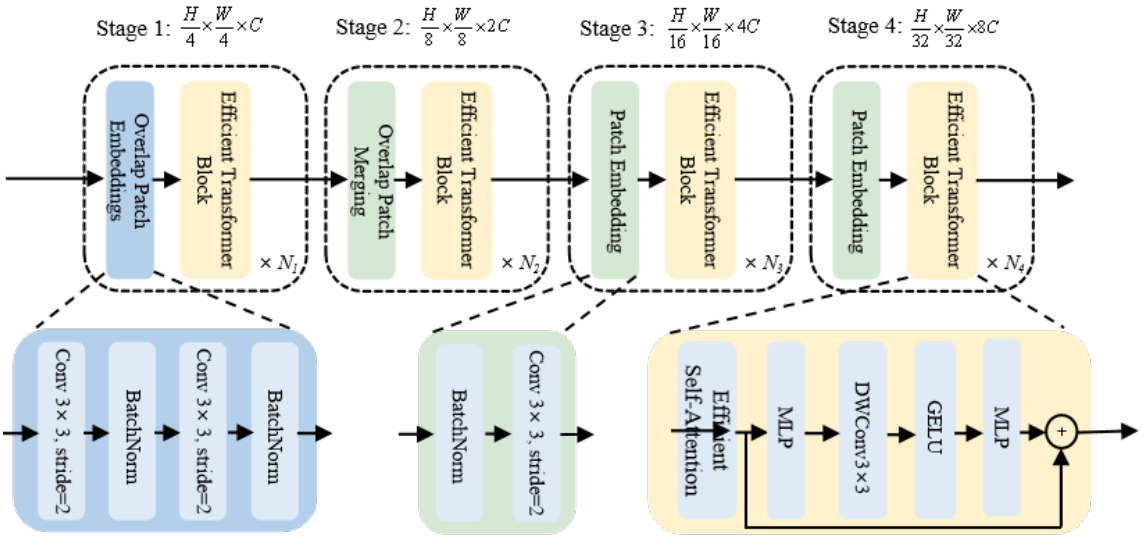


图 4.1　efficient

4.1.2　基于 top-k 区域注意力机制的彩色信息处理分支

常见的视觉 transformer（vision transformer, VIT）在进行图像处理时，会先把图片切割成小块，然后将这些小块展平为序列作为输入。VIT 使用的是自注意力机制（Multi-Head Self-Attention, MHSA）。MHSA 在进行计算的时候，需要计算输入序列中每个序列与其他所有序列之间的相似度，此时产生的计算复杂度为 $O(N^2)$，其中，N 是序列的长度，由小块的尺寸决定。因此，在图片分辨率较大的时候，序列长度也较长，但是序列长度二次方增长的复杂度会导致计算量的急剧增长。此外，在语义分割任务中，序列的分类更多的跟其周围的序列相关，不是所有的序列都有必要和其他的序列进行注意力的计算。

针对上述问题，本章算法提出基于 top-k 区域注意力机制（top-k regions attention，TRA）设计了处理彩色信息的视觉 transformer。top-k 区域注意力原理如下：首先，将图片切割成包含若干个小块的区域。然后，计算区域之间的相似度，保留相似度最高的 k 个区域。最后，对区域内的小块使用稀疏注意力机制。top-k 区域注意力机制的引入有效地减少了序列长度，降低了计算量。

划分区域。给定一个二维的特征图 $\mathbf{X}\in\mathbb{R}^{H\times W\times C}$，将其划分为 $S\times S$ 个非不重合的区域，使每个区域包含 $\frac{HW}{S^2}$ 特征向量，这时，$\mathbf{X}$ 重塑为 $\mathbf{X}^r\in\mathbb{R}^{S^2\times\frac{HW}{S^2}\times C}$，继而通过线性投影可以得到 $\mathbf{Q},\mathbf{K},\mathbf{V}\in\mathbb{R}^{S^2\times\frac{HW}{S^2}\times C}$：

$$\mathbf{Q}=\mathbf{X}^r\mathbf{W}^q,\quad \mathbf{K}=\mathbf{X}^r\mathbf{W}^k,\quad \mathbf{V}=\mathbf{X}^r\mathbf{W}^v, \tag{4.1}$$

其中，$\mathbf{W}^q,\mathbf{W}^k,\mathbf{W}^v\in\mathbb{R}^{C\times C}$ 分别是 $\mathbf{Q},\mathbf{K},\mathbf{V}$ 的投影权重。

选择 top-k 区域。通过构造有向图来选择与给定区域相关度排列最高的 k 个区

域。对之前得到的 $\mathbf{Q},\mathbf{K}$ 分别求取区域内的均值得到区域层面的 $\mathbf{Q}^r,\mathbf{K}^r \in \mathbb{R}^{S^2\times C}$，将 $\mathbf{Q}^r$ 和 $\mathbf{K}^r$ 的转置进行矩阵乘法运算可以得到邻接矩阵 $\mathbf{A}^r \in \mathbb{R}^{S^2\times S^2}$。

$$\mathbf{A}^r = \mathbf{Q}^r(\mathbf{K}^r)^T. \tag{4.2}$$

邻接矩阵 $\mathbf{A}^r$ 是两个区域语义相关的度量。然后，通过为每个区域保留关联度最高的 k 个区域，得到稀疏邻接矩阵 $\mathbf{I}_r \in \mathbb{N}^{S^2\times k}$。

$$\mathbf{I}^r = \text{topkIndex}(\mathbf{A}^r). \tag{4.3}$$

其中，$\mathbf{I}^r$ 的第 i 行包含了 k 个最相关区域的索引。

计算稀疏注意力。得到了稀疏邻接矩阵 $\mathbf{I}_r$，就可以使用稀疏注意力。首先，合并 k 个相关区域的 $\mathbf{K},\mathbf{V}$。

$$\mathbf{K}^g = \text{gather}(\mathbf{K},\mathbf{I}^r),\ \ \mathbf{V}^g = \text{gather}(\mathbf{V},\mathbf{I}^r), \tag{4.4}$$

其中，$\mathbf{K}^g,\mathbf{V}^g \in \mathbb{R}^{S^2\times \frac{kHW}{S^2}\times C}$ 由 k 个相关区域对应参数合并产生。

然后，对收集的键值对使用稀疏注意力。

$$\mathbf{O} = \text{Attention}(\mathbf{Q},\mathbf{K}^g,\mathbf{V}^g). \tag{4.5}$$

复杂度的理论计算。相比与普通注意力机制的复杂度 $O((HW)^2)$，TRA 的复杂度降低到了 $O((HW)^{\frac{4}{3}})$。TRA 的复杂度计算包括三个部分：划分区域、选择 top-k 区域、计算稀疏注意力。因此，总体的复杂度

$$\begin{aligned}
\text{FLOPs} &= \text{FLOPs}_{region} + \text{FLOPs}_{top-k} + \text{FLOPs}_{attention} \\
&= 3HWC^2 + 2(S^2)^2C + 2HWk\frac{HW}{S^2}C \\
&= 3HWC^2 + C(2S^4 + \frac{k(HW)^2}{S^2} + \frac{k(HW)^2}{S^2}) \\
&\geq 3HWC^2 + 3C(2S^4 \cdot \frac{k(HW)^2}{S^2} \cdot \frac{k(HW)^2}{S^2})^{\frac{1}{3}} \\
&= 3HWC^2 + 3Ck^{\frac{2}{3}}(2HW)^{\frac{4}{3}}
\end{aligned} \tag{4.6}$$

其中，C 是特征映射的通道数，k 是参与的区域数。公式中的放缩使用了均值不等式，当且仅当 $2S^4 = \frac{k(HW)^2}{S^2}$ 时等式成立。因此:

$$S = (\frac{k}{2}(HW)^2)^{\frac{1}{6}}. \tag{4.7}$$

当根据公式4.7划分区域大小时，TRA 的复杂度可以降低到 $O((HW)^{\frac{4}{3}})$。

TRA 的伪代码如 XX 所示。

TRA 过程如图如4.2 所示。

算法 4.1 TRA

已知: 特征图尺寸是 (H, W, C)。k 是区域数量。S 区域数量的算术平方根。

求: TRA 处理后的特征图 (H, W, C)。

$x = patchify(input, patch\_size = H//S)$

$query, key, value = linear\_qkv(x).chunk(3, dim = -1)$

$query\_r, key\_r = query.mean(dim = 1), key.mean(dim = 1)$

$A_r = mm(query\_r, key\_r.transpose(-1, -2))$

$I_r = topk(A_r, k).index$

$key\_g = gather(key, I_r)$

$value\_g = gather(value, I_r)$

$A = bmm(query, key\_g.transpose(-2, -1))$

$A = softmax(A, dim = -1)$

$output = bmm(A, value\_g) + dwconv(value)$

$output = unpatchify(output, patch\_size = H//S)$

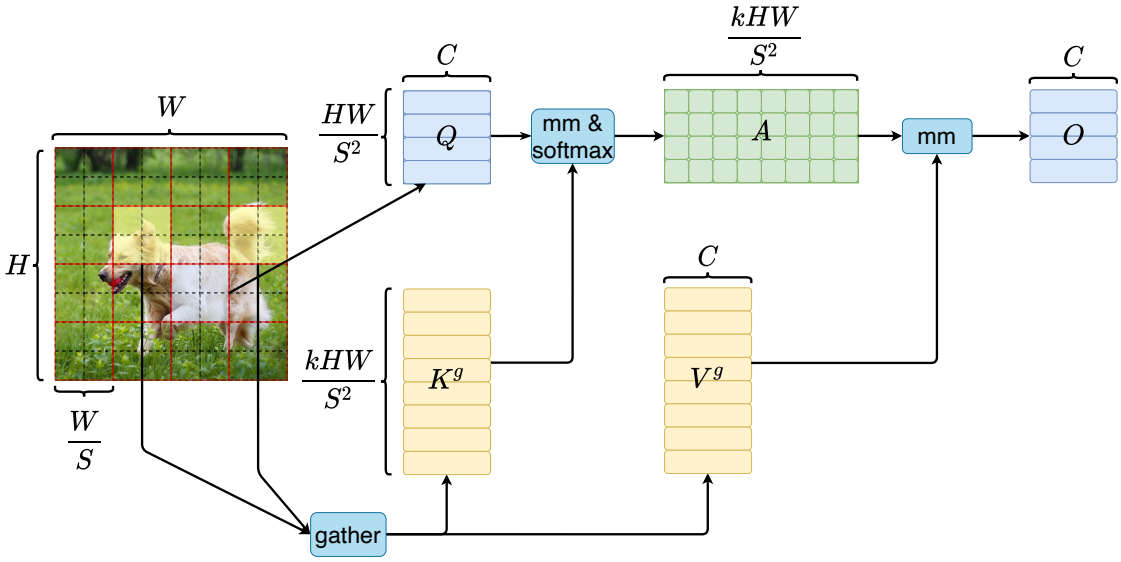


图 4.2　top-k 区域注意力机制

基于 TRA 的视觉 transofrmer 如图如4.3所示。

4.1.3　基于稀疏自注意力的深度信息处理分支

VIT 依靠多头自注意力机制强大的特征提取能力，对输入信息进行高效特征提取，在语义分割领域取得了很好的进展。但是，不同于三通道的彩色信息，深度信息是单通道的，所以其包含的有效信息比彩色信息更少。

针对这个问题，本文在深度信息处理分支，改进 VIT 使用的自注意力机制，使用稀疏自注意力构建更高效的 transformer，获取更加轻量化的语义分割模型。

(1) 重叠补丁嵌入。对于一个输入的图片来说，ViT 使用的不重叠的补丁嵌入

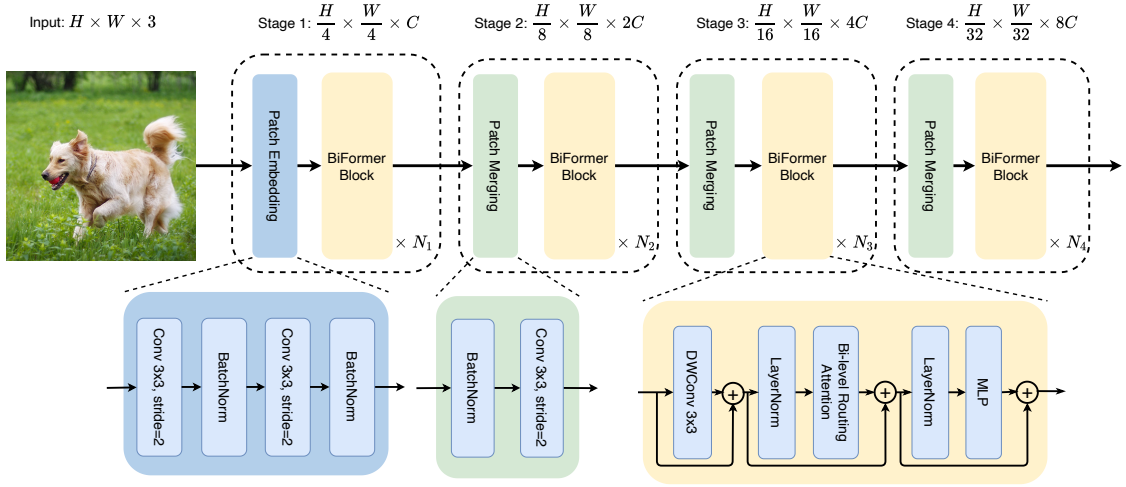
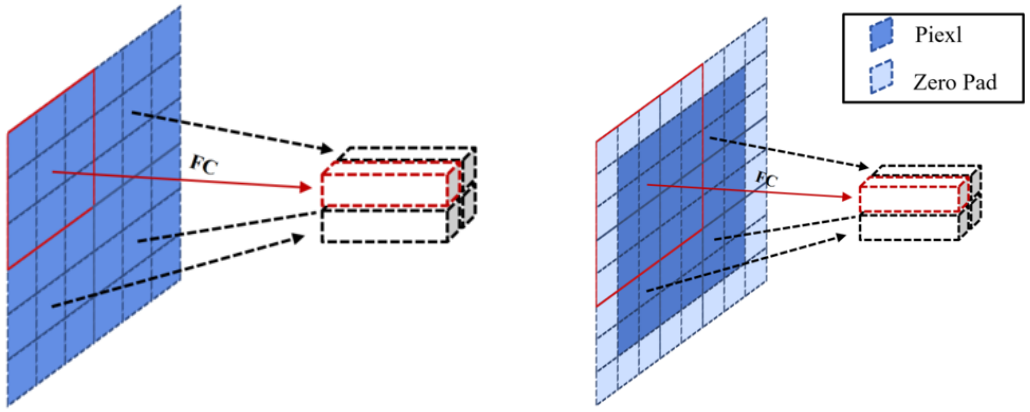


图 4.3　基于 top-k 区域注意力机制的 RGB 处理分支

操作，将一个 $N \times N \times 3$ 的补丁统一为一个 $1 \times 1 \times C$ 的向量。这种操作可以很容易地将一个 $2 \times 2 \times C_i$ 的特征统一为 $1 \times 1 \times C_{i+1}$ 的向量，从而获得分层特征映射。因此，我们可以将不同的层次特征不断缩小，从而获得预期大小尺寸的特征映射。但是，不重叠的补丁嵌入不能保证补丁的局部连续性。因此，我们使用重叠补丁嵌入。通过定义补丁大小、两个相邻补丁之间的步幅和填充大小，可以产生与不重叠补丁嵌入一样大小的特征。重叠补丁嵌入如4.4 所示。



(a) 非重叠补丁嵌入　　(b) 重叠补丁嵌入

图 4.4　补丁嵌入

(2) 稀疏自注意力。传统的自注意力机制如下所示：

$$\text{Attention}(Q, K, V) = \text{Softmax}(\frac{QK^{\mathsf{T}}}{\sqrt{d_{head}}})V. \tag{4.8}$$

本文采用 XX 介绍的序列简约算法，该算法使用稀疏因子 R 来缩减序列的长度。如下所示：

$$\begin{aligned}\hat{K} &= \text{Reshape}(\frac{N}{R}, C \cdot R)(K) \\ K &= \text{Linear}(C \cdot R, C)(\hat{K}),\end{aligned} \tag{4.9}$$

其中，K 是待稀疏的序列，R 是一个固定的稀疏因子。在实验中的阶段 1 到阶段 4，R 被设定为 [64, 16, 4, 1]。在上述公式中，第一个公式将 K 的形状由 $N \times C$ 重塑为 $\frac{N}{R} \times (C \cdot R)$，第二个公式将重塑的 K 线性操作，将其形状展开为 $\frac{N}{R} \times C$。因此，该操作可以使自注意机制的复杂性从 $O(N^2)$ 降低到 $O(\frac{N^2}{R})$。如4.5 所示。

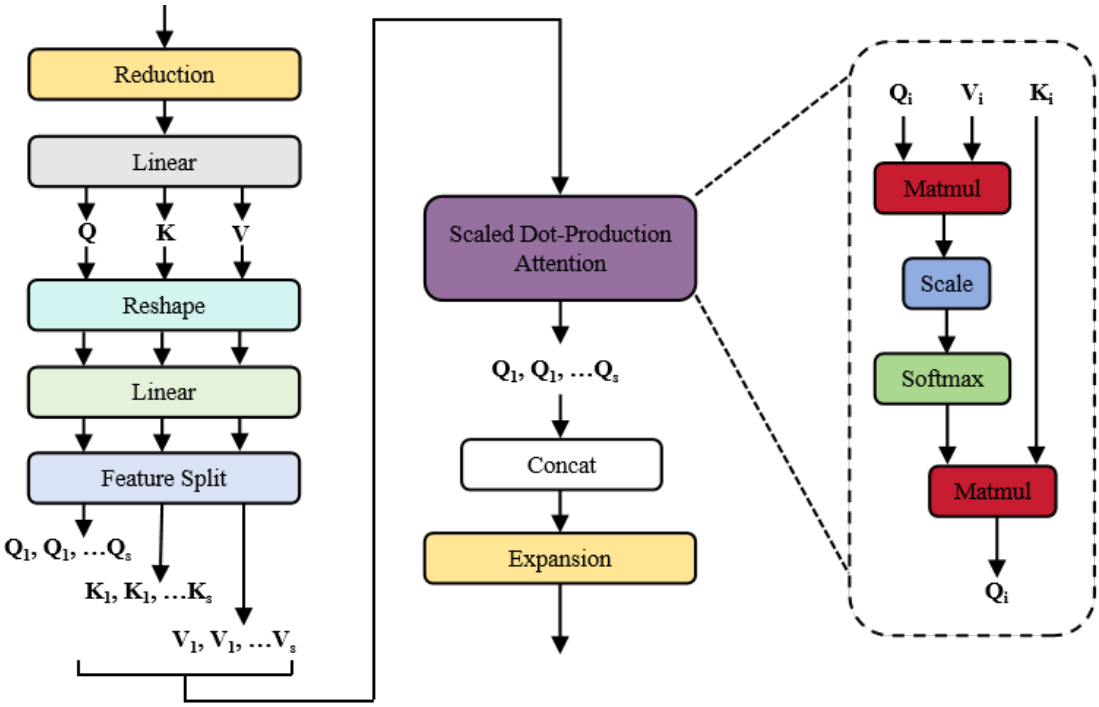


图 4.5　稀疏自注意力

(3) 高效的前馈网络。在 VIT 中使用的传统 Transformer 模型中，由于注意力机制并没有考虑图片小块的先后顺序信息，因此需要通过位置编码（Positional Encoding，PE）这种方式把前后的位置信息加在输入的图片小块上，这样让 Transformer 保留图片小块的位置信息，可以提高模型对序列的理解能力。然而，位置编码的分辨率是固定的。因此，当测试分辨率与训练分辨率不同时，需要对位置编码进行插值，这往往会导致预测准确性下降。

为了缓解这个问题，本文引入 Mix-FFN(Mixed Feed-Forward Network)，该结构考虑了零填充对泄漏位置信息的影响，通过在前馈网络中直接使用 3 × 3 卷积来实现位置编码。

传统的 FFN 在每个位置上都采用相同的非线性变换，而 Mix-FFN 则允许对不同位置应用不同的非线性变换，从而增强模型的表达能力。具体来说，Mix-FFN 使用了全局前馈神经网络（Global FFN）和局部前馈神经网络（Local FFN）两种不同的前馈神经网络结构。全局 FFN 是一个具有较大感受野的前馈神经网络，能够更好地捕捉全局上下文信息。而局部 FFN 是一个具有较小感受野的前馈神经网络，能够更好地捕捉局部细节信息。通过同时使用全局 FFN 和局部 FFN，Mix-FFN 能够在处理不同位置的特征时更加灵活和准确。全局 FFN 可以帮助模型捕捉到更长范围的依赖关系和语义信息，而局部 FFN 则可以更好地处理局部细节和细微变化。此外，Mix-FFN 在每个 FFN 中将一个 3 × 3 卷积和一个 MLP 混合在一起，该结构可以为 Transformer 提供位置信息。并且，我们使用深度可分离卷积来减少参数数量并提高效率。Mix-FFN 可以表示为：

$$\mathbf{x}_{out} = \mathrm{MLP}(\mathrm{GELU}(\mathrm{Conv}_{3\times3}(\mathrm{MLP}(\mathbf{x}_{in})))) + \mathbf{x}_{in}, \tag{4.10}$$

Mix-FFN 如4.6 所示。

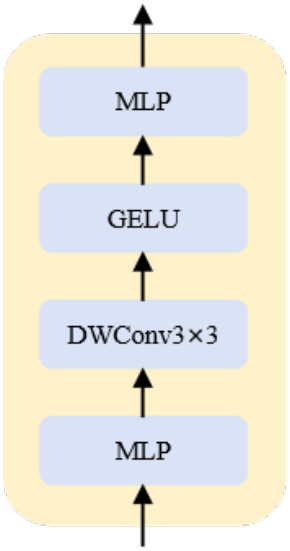


图 4.6 mixffn

(4) 网络结构网络结构如4.7 所示。

4.1.4 基于交叉注意力的跨模态特征融合

(1) 基于空间和通道注意力的跨模态特征选择

空间注意力和通道注意力可以从特定模态中压缩特征并选择特征，提高语义分割的准确性。但是，现有的空间注意力和通道注意力机制采取不可学习的方法来压缩特征，这种方法对单模态的特征选择较为充分，但是对多模态输入，就无法兼顾不同模态特征之间的差异性，进而不利于不同模态信息的利用。

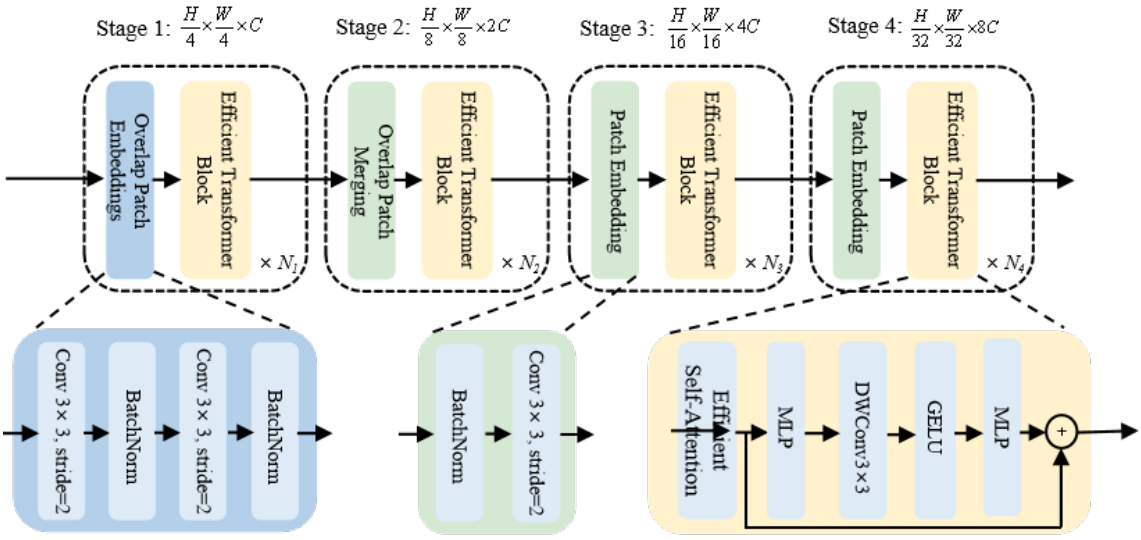


图 4.7　efficient

针对上述问题，本文提出基于空间和通道注意力的跨模态特征选择方法。该方法通过可学习的策略对彩色信息和深度信息进行特征压缩和特征选择。

首先，对彩色信息和深度信息拼接的特征图进行均值池化和最大池化，并将两种池化信息加权求和得到特征图的全局信息向量。其次，在通道注意力部分，全局信息向量被输入到一个多层感知机用来产生表示不同通道的权重分配向量，然后将权重分配向量通过 Sigmoid 函数得到归一化的通道注意力权重。在空间注意力部分，全局信息向量被输入到另一个多层感知机产生表示不同通道的权重分配向量，通过与原始特征图相乘后在通道维度的相加，可以得到不同空间的空间权重分配向量，然后将空间权重分配向量通过 Sigmoid 函数得到归一化的空间注意力权重。最后，将原始特征图和通道注意力权重和空间注意力权重相乘得到跨模态的融合特征。

如图如4.8所示。

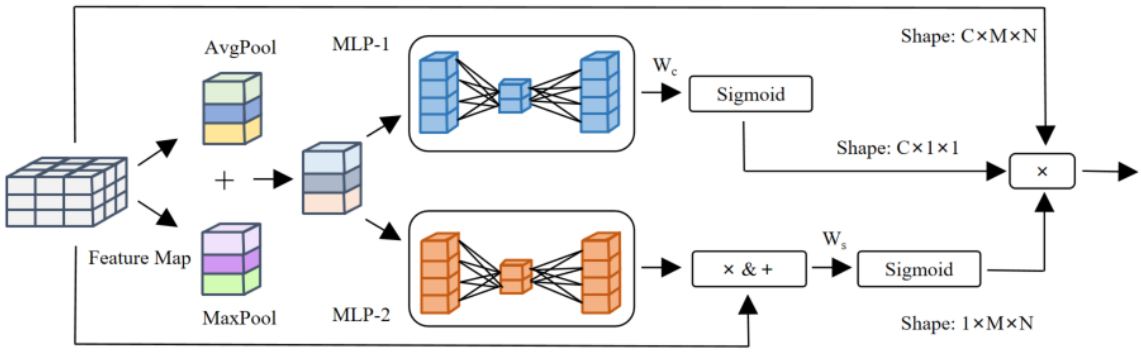


图 4.8　基于空间和通道注意力引导的跨模态特征选择

(2) 基于交叉注意力的跨模态特征嵌入

VIT 利用 transformer 中的多头注意力 (MultiHead Self-Attention, MHSA) 对输入的模态信息进行特征提取。但是 MHSA 只接受单一模态的信息，因而只能对同一模态的信息进行自相似性计算。在跨模态的语义分割任务中，单一模态的自相似性计算无法对来自不同模态的信息进行特征提取，因此无法充分利用来自不同模态信息的互补性来提高语义分割的性能。

针对这一问题，本文提出基于交叉注意力的跨模态特征嵌入方法。该方法借鉴自注意力机制中的自相似性计算，通过定义跨模态自相似性构造交叉注意力机制，提出一个跨模态特征嵌入模块，从而对彩色信息和深度信息进行特征融合。

特征混合重组。跨模态特征嵌入模块有三个输入，分别是彩色特征，深度特征和融合特征。首先，彩色特征和深度特征会被投影到向量子空间中，用来产生对应模态的 Key 和 $Query$。融合特征会被投影到第三个向量子空间中，用来产生融合模态对应的 $Value$。如 XX 所示。

然后，如果在不同的子空间中计算自相似性，那么就无法在不同的子空间同时包含彩色信息和深度信息包含的特征。因此，为了可以从不同的特征子空间学习特征，利用混合重组方法将彩色信息和深度信息产生的 Key 拼接后打乱重组。这样，新产生的 Key 就同时包含了来自彩色模态和深度模态的信息。将彩色信息和深度信息产生的 $Query$ 拼接后打乱重组。这样，新产生的 $Query$ 也同时包含了来自彩色模态和深度模态的信息。如 XX 所示。

跨模态自相似性。假设彩色信息和深度信息的特征被编码为 Key 和 $Query$，那么对于任意一个像素 (i_0, j_0)，它与其他像素 (i, j) 的跨模态自相似性可以被定义为：

$$W(i,j) = \sum_{n=1}^{N}(Krgb_{n,i,j} \cdot Qrgb_{n,i_0,j_0}) + \sum_{n=1}^{N}(Kdepth_{n,i,j} \cdot Qdepth_{n,i_0,j_0}) \tag{4.11}$$

其中，$Krgb_{n,i,j}$ 表示像素 (i_0, j_0) 产生的 Key 在彩色模态的第 n 个特征值，$Qrgb_{n,i,j}$ 表示像素 (i_0, j_0) 产生的 $Query$ 在彩色模态的第 n 个特征值，$Kdepth_{n,i,j}$ 表示像素 (i_0, j_0) 产生的 Key 在深度模态的第 n 个特征值，$Qdepth_{n,i,j}$ 表示像素 (i_0, j_0) 产生的 $Query$ 在深度模态的第 n 个特征值。

跨模态交叉注意力。在计算完跨模态自相似性后，还需要将计算的结果嵌入到融合特征 $Value$ 中。首先，计算 K_1 和 Q_1 的点积、K_2 和 Q_2 的点积，并将点积

结果通过 Softmax 函数归一化，得到特征子空间 W_1 和 W_2。

$$W_1 = \text{Softmax}(\frac{Q_1 \cdot K_1^T}{\sqrt{C_1/4}})$$
$$W_2 = \text{Softmax}(\frac{Q_2 \cdot K_2^T}{\sqrt{C_1/4}}) \tag{4.12}$$

然后，通过点积运算将信息嵌入到 V_1 和 V_2 中后，将 V_1 和 V_2 在通道维度拼接可以得到最后的融合特征 $Fused$。

$$Fused = \text{Cat}[W_1 \cdot V_1, W_2 \cdot V_2] \tag{4.13}$$

最后，将融合特征 $Fused$ 与 $Fused_1$ 进行残差连接，完成跨模态交叉注意力的全过程，得到最终的融合模态 $Fused_2$。

$$Fused_2 = Fused + Fused_1 \tag{4.14}$$

(3) 融合结构如图如4.9所示。

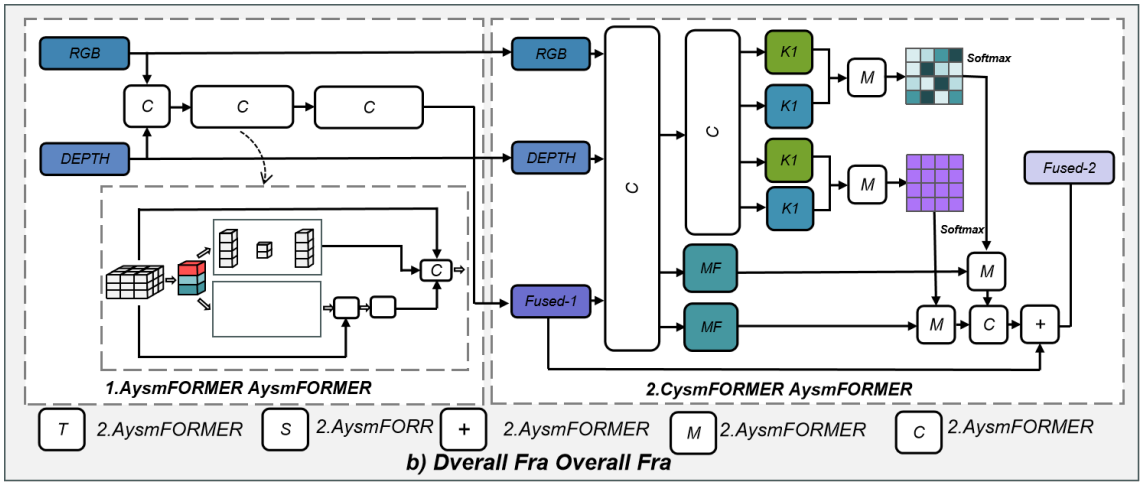


图 4.9　跨模态融合结构

4.1.5　轻量级 MLP 解码器

对于语义分割来说最重要的问题就是如何增大有效感受野。对于 Transformer encoder 来说，由于 self-attention 有效感受野变得非常大，因此 decoder 不需要更多操作来提高有效感受野，也因此可以设计更加简单的 decoder。

本文设计一个轻量级 MLP 解码器。该解码器仅由 MLP 层组成，避免了其他方法中通常使用的手工设计和计算量较大的组件。提出的全 MLP 解码器包括四个主要步骤。首先，来自 MiT 编码器的多级特征 Fi 通过一个 MLP 层进行通道维度统一。然后，在第二步中，特征被上采样到 1/4 大小，并进行拼接。其次，采用

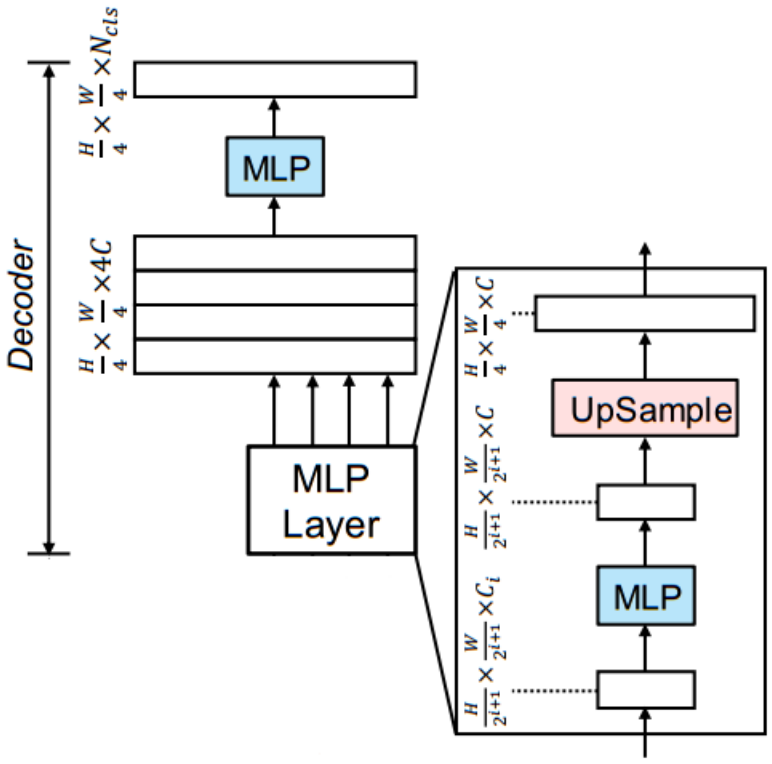
MLP 层来融合拼接后的特征 F。最后，另一个 MLP 层将融合后的特征输入，预测分割掩码 M，分辨率为 H/4 × W/4 × Ncls，其中 Ncls 是类别的数量。

可以将解码器表示为：

$$\begin{aligned}
\hat{F}_i &= \text{Linear}(C_i, C)(F_i), \forall i \\
\hat{F}_i &= \text{Upsample}(\frac{W}{4} \times \frac{W}{4})(\hat{F}_i), \forall i \\
F &= \text{Linear}(4C, C)(\text{Concat}(\hat{F}_i)), \forall i \\
M &= \text{Linear}(C, N_{cls})(F),
\end{aligned} \tag{4.15}$$

其中，M 是 XX，$\text{Linear}(C_{in}, C_{out})(\cdot)$ 是 XX，ZZ 是 XX。

如图如4.10所示。



CSDN @八十八岁扶墙敲码

图 4.10　decoder

4.2 实验结果分析

4.2.1 数据集与评价指标

(1) 数据集

本张使用公开数据集 NYUv2 和私有的地铁排爆数据集。NYUv2 是语义分割任务中使用最广泛的基准。NYUv2 原始数据集的数据来自 3 个城市的 464 个场景，并且绝大多数场景是室内场景，共 894 个类别标注。通过对原始的语义标签扩展，有 13 类和 40 类两种版本用于语义分割。根据做语义分割任务的大多数论文设定，本文采用 40 类的版本。该版本的语义标签主要包括墙壁、地板、窗户、桌子和椅子等室内物体。数据集主要包括彩色图片、深度图片以及标注图片。1449 张精细标注的图片被进一步分为 795 张和 654 张，分别用于训练和测试。图像尺寸 640×480。地铁排爆数据集是针对地铁排爆场景的自制数据集。基础场景是某城市的某个地铁站点，通过在地铁月台和地铁车厢内部布置管状模拟爆炸物、模拟爆炸物疑似藏匿箱体等物体，模拟真实的地铁排爆场景。该数据集一共有 XX 个语义分割类别，主要包括地铁闸机、地铁月台、地铁车厢内部座椅、模拟爆炸物、模拟爆炸物疑似藏匿箱体等物体。数据集格式依照 NYUv2 数据集设置，主要包括彩色图片、深度图片以及标注图片。XX 张精细标注的图片被进一步分为 XX 张和 XX 张，分别用于训练和测试。图像尺寸 640×480。两个数据集的具体对比如 XX 所示。

(2) 评价指标

实验中使用的 X 个指标来衡量语义分割算法的性能。第一个是参数量 ()，该指标反应，该指标越低越好。第二个是ＦＬＯＰｓ。该指标越低越好。第三个是 Miou。该指标越高越好。

4.2.2 参数设置

本章算法使用 pytorch 框架，使用一台服务器进行训练和测试。该服务器装配有 XX 型号的 CPU，四张 RTX4090GPU，XX 版本的 CUDA。表 XX 显示的是本章算法在 NYUv2 数据集和地铁排爆数据集上的详细参数设置。在 NYUv2 数据集上，GPU 数量设置为 XX，批大小设置为 XX，训练 XX 个 epoch。学习策略设置为 XX，初始学习率设置为 XX，学习率衰减参数设置为 XX，优化方法设置为 XX，损失函数设置为 XX。在地铁排爆数据集上，GPU 数量设置为 XX，批大小设置为 XX，训练 XX 个 epoch。学习策略设置为 XX，初始学习率设置为 XX，学习率衰减参数设置为 XX，优化方法设置为 XX，损失函数设置为 XX。此外，在两个数据集的训练初始阶段，都采用数据增加对原始的彩色图片和深度图片进行

处理，采用 XX 方法来提高模型的学习能力和泛化能力，但是在测试时，使用原始的图片，不涉及任何的数据增强，也不涉及任何对图片大小进行改变的操作。

4.2.3 消融实验

本小节通过在 XX 数据集上进行消融实验，验证 XX 算法中不同模块的有效性。因此，本章的算法 XX 模块表示 XX。XX 模块表示 XX。XX 模块表示 XX。%这里插入小消融实验的表格。表 XX 展示了 XX 算法消融实验的结果。

4.2.4 模型性能对比

4.2.5 对比实验

4.3 本章小结

本章通过提出 XX 算法，该算法通过 XX 解决了 XX 问题。实验结果表明，XX 算法的 XX 指标相较基准算法分别降低了 XX。

第五章　第一章题目

本章的主要内容与学校提供的 Word 模板中内容一致，图片与表格均采用原始设定大小，主要是为了说明格式的统一。但是，LaTeX 的一些禁则，专业排版的能力，对公式及文献的处理都是得天独厚的，我们不必刻意去追求与 Word 的完美匹配。而且你将会发现，用 LaTeX 书写论文的美！

5.1　（1.1 题目）

正文内容

5.1.1　（1.1.1 题目）

正文内容
正文内容

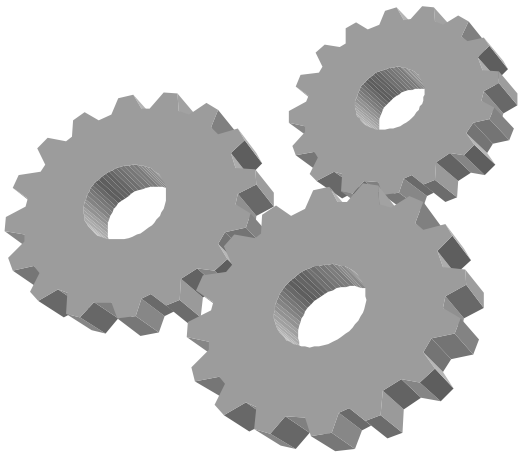


图 5.1　图 1.1 名称

5.1.1.1　（1.1.1.1 题目）

正文内容
正文内容
正文内容

5.1.1.2　（1.1.1.2 题目）

正文内容
正文内容
正文内容

5.1.2 （1.1.2 题目）

正文内容
正文内容

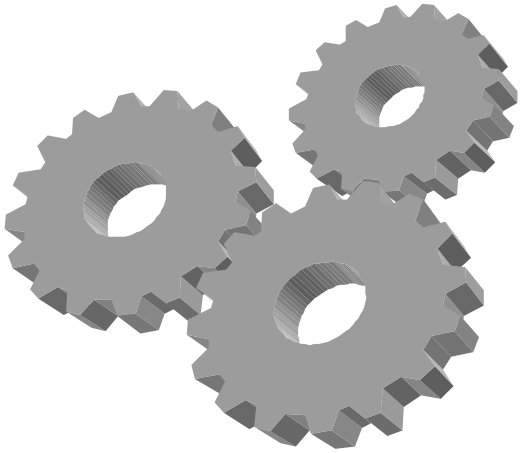


图 5.2 图 1.2 名称

5.2 （1.2 题目）

正文内容
正文内容

表 5.1

列 1	列 2

正文内容
正文内容
正文内容
正文内容

5.3 （1.3 题目）

正文内容
正文内容
正文内容
正文内容
正文内容
正文内容

5.3.1 （1.3.1 题目）

正文内容

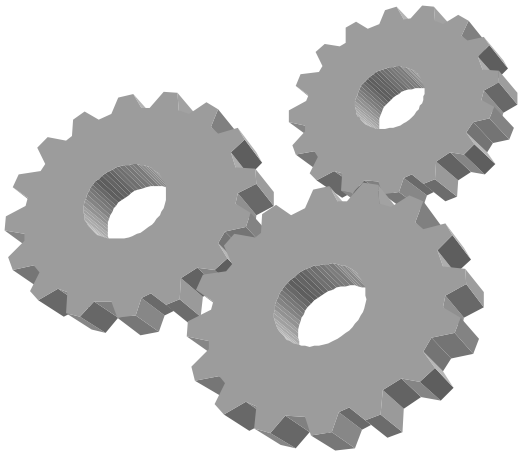


图 5.3 图 1.3 名称

5.3.2 （1.3.2 题目）

正文内容
正文内容

表 5.2

列 1	列 2

致　谢

衷心感谢导师 xxx 教授和 xxx 副教授对本人的精心指导。他们的言传身教将使我终生受益。

感谢 NUDTPAPER，它的存在让我的论文写作轻松自在了许多，让我的论文格式规整漂亮了许多。

作者在学期间取得的学术成果

发表的学术论文

[1] Yang Y, Ren T L, Zhang L T, et al. Miniature microphone with silicon- based ferroelectric thin films. Integrated Ferroelectrics, 2003, 52:229-235. (SCI 收录, 检索号:758FZ.)

[2] 杨轶, 张宁欣, 任天令, 等. 硅基铁电微声学器件中薄膜残余应力的研究. 中国机械工程, 2005, 16(14):1289-1291. (EI 收录, 检索号:0534931 2907.)

[3] 杨轶, 张宁欣, 任天令, 等. 集成铁电器件中的关键工艺研究. 仪器仪表学报, 2003, 24(S4):192-193. (EI 源刊.)

[4] Yang Y, Ren T L, Zhu Y P, et al. PMUTs for handwriting recognition. In press. (已被 Integrated Ferroelectrics 录用. SCI 源刊.)

[5] Wu X M, Yang Y, Cai J, et al. Measurements of ferroelectric MEMS microphones. Integrated Ferroelectrics, 2005, 69:417-429. (SCI 收录, 检索号:896KM.)

[6] 贾泽, 杨轶, 陈兢, 等. 用于压电和电容微麦克风的体硅腐蚀相关研究. 压电与声光, 2006, 28(1):117-119. (EI 收录, 检索号:06129773469.)

[7] 伍晓明, 杨轶, 张宁欣, 等. 基于 MEMS 技术的集成铁电硅微麦克风. 中国集成电路, 2003, 53:59-61.

研究成果

[1] 任天令, 杨轶, 朱一平, 等. 硅基铁电微声学传感器畴极化区域控制和电极连接的方法: 中国, CN1602118A. (中国专利公开号.)

[2] Ren T L, Yang Y, Zhu Y P, et al. Piezoelectric micro acoustic sensor based on ferroelectric materials: USA, No.11/215, 102. (美国发明专利申请号.)

公开评阅信息

序号	评阅人	职称	导师类型	工作单位	总分	结论	答辩建议	熟悉程度	备注
1	张三	教授	博导	XXX大学	95.8	达到	无需修改直接答辩	有深入了解	
2	李四	研究员	硕导	XXX大学	95	达到	修改后答辩	有深入了解	
3	王五	教授	博导						
4	赵六	教授	博导						
5	孙六	教授	博导	XXX大学	59	尚未达到	修改后复评	有深入了解	
	孙六	教授	博导	XXX大学	80	尚未达到	无需修改直接答辩	有深入了解	复评结果

说明：

1. 结论选项包括 2 个：“达到博士学位论文要求”、“尚未达到博士学位论文要求”。

2. 答辩建议选项包括 4 个：“无需修改直接答辩”、“修改后答辩”、“修改后复评”、“不予答辩”。

3. 熟悉程度选项包括 3 个：“有深入了解”、“比较熟悉”、“一般了解”。

提醒（正式成文后删除）：

1. 评阅版论文删除此页。
2. 采用双盲评阅方式的学位申请人撰写的学位论文删除此页。
3. 评阅总分无需取整。
4. 工作单位填至学校、科研院所即可。

附录 A　模板提供的希腊字母命令列表

大写希腊字母：

Γ \Gamma	Λ \Lambda	Σ \Sigma	Ψ \Psi
Δ \Delta	Ξ \Xi	Υ \Upsilon	Ω \Omega
Θ \Theta	Π \Pi	Φ \Phi	
$\varGamma$ \varGamma	$\varLambda$ \varLambda	$\varSigma$ \varSigma	$\varPsi$ \varPsi
$\varDelta$ \varDelta	$\varXi$ \varXi	$\varUpsilon$ \varUpsilon	$\varOmega$ \varOmega
$\varTheta$ \varTheta	$\varPi$ \varPi	$\varPhi$ \varPhi	

小写希腊字母：

α \alpha	θ \theta	o o	τ \tau
β \beta	ϑ \vartheta	π \pi	υ \upsilon
γ \gamma	ι \iota	ϖ \varpi	ϕ \phi
δ \delta	κ \kappa	ρ \rho	φ \varphi
ϵ \epsilon	λ \lambda	ϱ \varrho	χ \chi
ε \varepsilon	μ \mu	σ \sigma	ψ \psi
ζ \zeta	ν \nu	ς \varsigma	ω \omega
η \eta	ξ \xi	$\varkappa$ \varkappa	$\digamma$ \digamma
$\upalpha$ \upalpha	$\uptheta$ \uptheta	o \mathrm{o}	$\uptau$ \uptau
$\upbeta$ \upbeta	$\upvartheta$ \upvartheta	$\uppi$ \uppi	$\upupsilon$ \upupsilon
$\upgamma$ \upgamma	$\upiota$ \upiota	$\upvarpi$ \upvarpi	$\upphi$ \upphi
$\updelta$ \updelta	$\upkappa$ \upkappa	$\uprho$ \uprho	$\upvarphi$ \upvarphi
$\upepsilon$ \upepsilon	$\uplambda$ \uplambda	$\upvarrho$ \upvarrho	$\upchi$ \upchi
$\upvarepsilon$ \upvarepsilon	$\upmu$ \upmu	$\upsigma$ \upsigma	$\uppsi$ \uppsi
$\upzeta$ \upzeta	$\upnu$ \upnu	$\upvarsigma$ \upvarsigma	$\upomega$ \upomega
$\upeta$ \upeta	$\upxi$ \upxi		

希腊字母属于数学符号类别，请用\bm命令加粗，其余向量、矩阵可用\mathbf。