分类号：TP242.6　　单位代码：10363
密　级：公开　　学　　号：2210920112

题　　目　基于注意力机制的移动机器人动态环境定位方法研究

题　　目　RESEARCH ON DYNAMIC ENVIRONMENT LOCALIZATION METHOD OF MOBILE ROBOT BASED ON ATTENTION MECHANISM

学　生　姓　名：杨海波

校　内　教　师：曹雏清

校　外　教　师：陈智君

专业 学位 类别：计算机技术

研究方向（领域）：人工智能及应用

论文答辩日期：2024 年 5 月 24 日

安徽工程大学学位论文原创性声明

本人郑重声明：我恪守学术道德，崇尚严谨学风。所呈交的学位论文，是本人在导师的指导下，独立进行研究工作所取得的成果。除文中已明确注明和引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的作品及成果的内容。论文为本人亲自撰写，我对所写的内容负责，并完全意识到本声明的法律后果由本人承担。

学位论文作者签名：杨海波

日期：2024年5月24日

安徽工程大学学位论文版权使用授权书

学位论文作者完全了解学校有关保留、使用学位论文的规定，同意学校保留并向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅或借阅。本人授权安徽工程大学可以将本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和汇编本学位论文。

保密 □，在 ____ 年解密后适用本版权书。

本学位论文属于

不保密 ☑。

学位论文作者签名：杨海涛　　指导教师签名：李永涛

日期：2024年5月24日　　日期：2024年5月24日

基于注意力机制的移动机器人动态环境定位方法研究

摘 要

近几年来，随着科技不断地发展，智能移动机器人在商业、工业、军事等诸多领域中泛用性越来越广泛。由于定位在智能移动机器人对环境的感知环节中起到关键作用，这使得视觉定位成为学者关注的热点，并且基于视觉的定位方法不仅成本低廉，而且可以获得更多环境信息。作为智能移动机器人的核心技术，视觉定位在没有动态物体干扰时，定位精度与鲁棒性较高。但当环境中有移动的物体例如行人、车等干扰时，会形成不正常的特征匹配，导致定位精度急剧下降。

针对以上问题，本文使用微软的 RGB-D 相机进行室内动态环境下视觉定位研究，提出了一个新的基于 EMA 注意力机制的 YOLOv8n 视觉定位算法，来实现动态环境下的位姿估计，既保证了系统的实时性，也解决了动态物体对系统的干扰。本文主要有以下研究工作：

1. 针对视觉定位将面对现实环境中存在的动态物体，本文提出了实时性较好的 YOLOv8n 算法，通过将 EMA 注意力机制模块添加至 YOLOv8n 模型的 C2F 模块中来加强模型的检测能力，并在 POSCAL VOC 数据集上进行训练评估，为后续的视觉定位系统提供准确度高的先验目标框信息。

2. 提出了一种基于 ORB-SLAM3 且适用于高动态环境的视觉定位方法，该系统了融合 EMA 注意力的 YOLOv8n 目标检测算法，将

准确的先验目标框用于定位系统的视觉模块中，实现了对动态特征点的快速检测以及剔除，最后将剩下的静态特征点用于后续的特征匹配以及位姿估计。最后，在 TUM 数据集上进行定位实验，实验结果表明，本文提出的方法在动态环境下拥有更好的的稳定性与精度。

关键词：视觉定位；室内动态环境；YOLOv8；目标检测

RESEARCH ON DYNAMIC ENVIRONMENT LOCALIZATION METHOD OF MOBILE ROBOT BASED ON ATTENTION MECHANISM

ABSTRACT

In recent years, with the continuous development of science and technology, intelligent mobile robots have become more and more widely used in commercial, industrial, military and other fields. Since localization plays a key role in the perception of mobile robot'senvironment, visual localization has attracted a lot of attention fromscholars, and vision-based localization methods are not only inexpensive but also rich in semantic information, which has gradually become a hot research direction. As a key technology in mobile robots, visual localization has achieved good results in static environments, with high localization accuracy and robustness. However, whenthere are dynamic objects such as pedestrians, cars and other interferences in the environment, wrong matching will be generated, resulting in a sharp degradation of positioning accuracy and a decrease in the robustness of the positioning system.

Aiming at the above problems, this paper conducts a visual localization study in indoor dynamic environments using an RGB-D ca

mera, and proposes a new YOLOv8 visual localization algorithm ba sed on the EMA attention mechanism to realize the position estimat ion in dynamic environments, which not only ensures the real-time performance of the system, but also solves the interference of dyna mic objects to the system. This paper has the following main resea rch work:

1. For visual localization will face dynamic objects existing in the real environment, this paper introduces the YOLOv8 target dete ction algorithm with better real-time performance, strengthens the de tection ability of the model by adding the EMA attention mechanis m module to the C2F module of the Yolov8n model, and conducts the training and evaluation on the POSCAL VOC dataset, which will provide the subsequent visual localization system with highly a ccurate a priori target frame information.

2. A visual localization method based on ORB-SLAM3 for hig hly dynamic environments is proposed, which incorporates the YOL Ov8n target detection algorithm with EMA attention, and uses the a ccurate a priori target frames in the vision module of the localizati on system to achieve fast detection and elimination of dynamic feat ure points, and then the remaining static feature points are used for the subsequent feature matching and position estimation. Finally, the localization experiments are conducted on the TUM dataset, and the

experimental results show that the method proposed in this paper h as better stability and accuracy in dynamic environments.

Yang Haibo(Computer Technology)

Supervised by Cao Chuqing

KEY WORDS: Visual Localization;Indoor Dynamic environment; YOLOv8; Target Detection

目 录

第 1 章 绪论

1.1 研究背景和意义

移动机器人的研究始于上个世纪末，并随着计算机相关领域的蓬勃发展,它集合了很多学科的技术，如人工智能、自动化、计算机科学，是高度智能化的结果，在国家的科研技术方向发挥着重要功能，例如军事、工业、农业、医疗、民用，甚至在海洋探索[1]与太空探索[2]中也发挥着重要作用。

目前，我国年轻人正处于结婚率与生育意愿低下的境况，在未来，人口老龄化问题会愈加严重，以应对老龄化而产生的劳动力不足等问题，发展智能移动机器人来替代人去工作就显得十分重要。

(a)送餐机器人

(b)扫地机器人

(c)仓储机器人

(d)巡逻机器人

图 1-1 智能移动机器人

事实上，智能化移动服务机器人已经成为大众关注的热点，社会的需求量也逐年增加，智能化移动服务机器人已经进入应用发展时期。例如送餐机器人、扫地机器人、仓储搬运机器人、巡逻机器人，它们的作用是代替人类完成日常工作，

同时具备高度的智能化与自主学习能力。

目前主流的定位主要分为激光定位[3]和视觉定位[4]。激光定位尽管激光定位有较高精度，但其获取到的环境信息较为有限。另外，激光定位的传感器价格较高，会增加机器人的制造成本。而视觉定位以例如单目相机,双目相机,RGB-D 相机等传感器成本低廉和可以用于感知的语义信息十分丰富等优势。

目前，视觉定位系统 ORB-SLAM3[5]、LSD-SLAM[6]等在常规场景下有着很好的精度，但当场景中含有人、车等动态物体时，会产生错误的匹配，导致视觉定位精度下降，鲁棒性降低。因此滤除动态场景内的移动的物体来提高定位系统的性能成为了一个热门研究问题。

1.2 国内外研究现状和问题

1.2.1 视觉定位研究进展

定位问题由 R.C.Smith[7]首次提出，后续定位问题在视觉领域主要发展成两种主流方法，一是特征点法，二是直接法。

2007 年 Davison 等[8]人提出了 MonoSLAM 算法，该算法是第一个基于单目相机的视觉定位算法，其视觉里程计中通过提取的 Shi-Tomasi[9]特征点进行追踪,构建稀疏特征地图。但是该方法在后端使用 EKF 扩展卡尔曼滤波[10], 导致无法处理大场景和动态场景,且所有计算都包含在一个线程内，计算夯余量大。2007 年，Klein[11]等人优化 MonoSLAM 的单线程计算的问题，提出一个双线程视觉定位算法(Parallel Tracking and Mapping,PTAM)，这项算法使用 FAST 角点[12]作为特征点提取的方法，并使用基于光束法平差(Bundle Adjustment,BA)进行非线性优化，该算法首次提出双线程的前后端架构，是第一个使用光束平差法的算法，摒弃了 EFK 滤波器在后端的局限性。

2010 年，微软发布了 Kinect 相机，能够实时获取 RGB 图像与不通过计算直接获取三维空间的深度图像，RGB-D 相机也因此迅速进入视觉定位领域。Endres F 等[13]在 2014 年提出 RGB-D SLAMv2，该系统在前端提供了两种可以选择的特征点提取算法 SIFT 与 ORB[14]，使用 RANSAC 与 ICP 算法来估计两帧图像之间的平移与旋转。

2015 年，在 PTAM 的基础上，Mur-Artal[15]等人提出了基于 ORB 特征点的 ORB-SLAM 算法，该算法使用非线性优化方法来进行局部和全局优化，具有较高

的定位精度和建图效果。随后，在 2021 年该团队在原算法的基础上提出了 ORB-SLAM3[16]系统，该系统支持单目、双目和 RGB-D 相机，并加入了 IMU 惯性测量单元。

目前 ORB-SLAM3 算法在许多数据集中得到了验证，其效果十分理想，系统的鲁棒性也优于其它视觉定位系统，后续有很多学者在此系统上进行补充。2018 年，Wang 等人在文献[17]中对 ORB-SLAM 进行了改进，提出了 CubemapSLAM 算法，该算法能够支持单目鱼眼相机，并且在一定程度上减小了鱼眼相机畸变的影响,同时充分应用到了鱼眼相机大视野的好处，增加了相邻图像帧的重叠度,进而增强稳定性。2016 年，S.Urban[18]等人基于 ORB-SLAM2 算法提出了 MultiCol-SLAM 算法,该算法支持鱼眼和多鱼眼相机。

上述系统基本都是基于特征点法的视觉定位系统，随着一些使用直接法的开源项目出现，直接法逐渐成为视觉里程计中重要的部分之一，2013 年，Kerl[19]基于直接法提出了使用 RGB-D 相机进行视觉定位的 DVO-SLAM(Dense Visual Odometry)算法，该算法通过最小化光度误差与深度误差,最大限度的利用图像的彩色与深度数据，从而提高位姿估计的精度。2014 年，Forster 等人在文献[20]提出了基于稀疏直接法的 SVO 系统(Semi-Direct Monocular Visual Odometry)，在视觉里程计中结合特征点法和直接法进行位姿估计，这种稀疏直接法不用计算描述子,因此速度相对较快，但是无法进行稠密的地图重建。同年,Engel[21]等人提出了 LSD-SLAM(Large-Scale Direct)，该算法可以在 CPU 上实时构建半稠密场景,通过回环检测能够构建大规模场景的地图,同时提出了关键帧之间的漂移尺度公式,在局部和全局的优化中将融合不同尺度的场景，减小尺度漂移的现象。随后在 2016 年，又提出了 DSO(Direct Sparse Odometry)[22]，该算法使用滑动窗口的优化方式，同时结合了相机的光度标定，进一步提升了该算法的精确度和鲁棒性。直接法由于基于灰度不变的强假设，导致该方法对光线强度变化较为敏感，在室外效果较差。

1.2.2 动态环境视觉定位研究进展

大多数视觉定位系统都是以静态场景为前提，才能拥有良好的定位精度与鲁棒性，例如前面所提到的 ORB-SLAM3 系统。但是在实际场景中，通常会存在

移动的物体例如行走的人或者车辆会对视觉定位系统产生干扰，影响了定位效果。对于动态场景，目前主要分为依靠语义和不依靠语义。

不依靠语义实则是依赖于外部传感器，外部传感器可以提供有效的环境数据以及自身的状态信息。因此，2015 年，Kim 等[23]针对高动态环境提出一种结合 RGB-D 相机和 IMU 的视觉里程计算法，通过整合深度信息和 RGB 图像生成三维点，利用 IMU 旋转三维点，在相邻两帧图像间计算旋转分量，并根据此过程过滤动态特征点。2017 年，Bloesch 等人[24]提出了一种基于卡尔曼滤波器的算法，将单目视觉和 IMU 相融合，将视觉信息用于滤波器的更新，IMU 的测量值用于状态传递。

RANSAC 是从含异常数据的样本集中拟合模型的迭代方法，用于计算 F 矩阵或 H 矩阵。张慧娟等人[25]提出了一种针对动态环境的方法，该方法是一种基于线特征的动态环境视觉里程计，通过特征点匹配和变换矩阵求解实现。利用外观和几何约束，检测静态环境中的直线特征，并根据结果选择动态环境中的特征，实行位姿估计。在 2017 年，Sun 等[26]人的研究中，先计算得到单应矩阵，对前一帧的像素点进行处理，利用两张图像不同来筛选对象，若图像中动态物体较多时，比如人流密集区时，其表现就会很差。

在 2016 年，章国峰等学者发明 RK-SLAM 算法[27]利用多视图几何约束进行动态特征点的判断，并借助 RANSAC 算法来去特征点，从而在实现靠谱的位姿估计。尽管该方法能够判断特征点是否为动态，但无法准确获得动态目标的轮廓信息，存在运动目标边界模糊、对动态目标描述不完整的问题，Wang 等[28]发明了新方法，借助数学模型和几何约束，通过极线约束排除外部点，分割运动对象，但对于高动态识别上会出现较大误差。

深度学习在视觉领域的快速发展推动了目标检测和语义分割技术的进步，依靠语义先验成为可能，为含有动态物体场景下的视觉定位提供了新思路。

2018 年，Bescos[29]等人提出了 DynaSLAM 算法，结合了 Mask-RCNN[30]网络来对动态特征点进行筛选，并通过区域生长算法分割出动态目标区域，从而消除了动态物体上的特征点在位姿估计阶段的影响，实现了动态物体的检测及背景修复功能，但该系统实时性较差。2017 年，YU[31]等人在同一年提出了 DS-SLAM 算法，该方法应用 SegNet[32]进行语义分割以获取语义掩模，同时运用极线几

何检测来获得动态特征点。在此过程里，若特征点高于阀值，则进行清零，从而减少动态目标对视觉里程计的影响。2022 年，Fan 等[33]借助 BlitzNet 卷积神经取得掩膜，并滤除图像中的动态物体。

2019 年，王金戈等人[34]提出了 Dynamic-SLAM，该方法利用卷积神经网络来构建 SSD 检测器，以先验语义信息为基础检测动态物体，从而减小动态物体对算法的影响，提高了位姿估计和地图构建的精度。在 2020 年，VDO-SLAM 由 Jun Zhang[35]提出，该算法摆脱了对运动先验信息的依赖，鲁棒性和准确性较为理想。2022 年，Xing Z 提出了 DE-SLAM 算法[36]，算法结合了语义信息和深度信息，提出了一种能够同时处理短期和长期动态物体的视觉定位系统。极大地提高了定位精度。在 2021 年，Chang 等[37]提出借助 YOLACT 进行等分的 SLAM 方法，同时融合了自适应阈值的光流检测，来识别被人为挪动的物体。而 2022 年，Wu[38]则在文章中举出 YOLO-SLAM，借助 YOLOv3 的特性，对正在行进中的目标进行检测，并输出检测框，从而剔除动态物体对定位系统的影响。

1.2.3 注意力机制与目标检测的研究进展

注意力机制在计算机视觉领域是一种非常重要的技术，在计算机视觉的不同的任务例如识别、分割、关键点检测、目标跟踪等技术领域中，都担当着重要的角色[39]。2017 年，文献[40]首次提出了 SENet 模型，并阐述了通道注意力的定义。该方法主要是使用 Squeeze 与 Excitation 操作来所有通道数的权重，但 Squeeze 操作只能获取低阶的汇总信息，Excitation 操作又增加了模型的复杂性。2015 年，Jaderberg 等人[41]提出了关于空间注意力的经典论文，该论文提出了空间变换网络，该网络可以使模型在不同位置、不同空间以及不同尺度下对感兴趣的目标进行权重提升并将该变换网络应用于目标检测的神经网络，并且在后续的目标检测工作[42]中取得很好的进展。2020 年，基于自注意力机制，Donsovitskiy[43]等学者提出了一个首次将 Transformer[44]应用于计算机视觉的 Vision Transformer 模型，该模型将图片转化成序列信息与编码融合，送入 Transformer 网络中，从而对图像中的目标进行预测。2020 年，由 Facebook 团队提出的 DETR 模型[45]完成了全局注意力的目标检测功能，并且该模型不需要预定的锚框来生成标签，提升了该模型在各个场景的泛用性。2023 年，由 Daliang[46]提出的多尺度 EMA 注意力机制，保留每个通道上的信息和降低计算开销为目标，将部分通道重塑为批量维度，

并将通道维度分组为多个子特征，使空间语义特征在每个特征组中均匀分布，该网络可用于目标检测骨干网络中，以提升对物体的检测准确度。

1.3 本文的主要研究内容

本文为了解决视觉定位系统在动态环境下鲁棒性差，从深度学习的角度分析研究了该问题，本文选择将注意力机制融入动态目标检测，再将动态目标检测结果融入视觉定位系统进行位姿估计，本文的主要研究内容概括如下：

（1）将 EMA 注意力机制与 C2f 模块融合成新的 C2f-EMA 模块，再将该模块把目标检测网络 YOLOv8n 中的 C2f 模块全部替换，构建改进后的 YOLOv8n-EMA 模型，从而提高 YOLOv8n 对物体的检测精度，并在 PASCAL VOC 数据集下进行训练验证。

（2）根据室内动态场景的定位需求，将 YOLOv8n-EMA 目标检测模型加入到 ORB-SLAM3 系统中，并使用该模型检测后提供的先验目标框来检测场景中的动态物体，并在高动态的场景数据集 TUM 下进行定位精度的测试验证，并将本文改进的算法与其他同类算法进行精度对比以及性能能对比。

1.4 本文的章节安排

本文将 EMA 注意力机制融入 YOLOv8 动态目标检测，再将动态目标检测结果融入定位系统进行位姿估计。章节安排如下：

第 1 章绪论，介绍本文的研究的背景和意义，简单说明了本文的主要工作内容，再介绍近年来，学者们对面向动态环境定位的研究现状，最后总结本文主要贡献。

第 2 章相关理论介绍，主要介绍神经网络的组成原理、注意力机制的种类以及与本文相关的视觉定位前后端框架流程。

第 3 章 YOLOv8 融合注意力机制，主要是将 EMA 注意力机制与 YOLOv8n 模型进行融合，其原理是将 EMA 注意力机制与 C2f 模块融合成新的 C2f-EMA 模块，再将该模块把目标检测网络 YOLOv8n 中的 C2f 模块全部替换，最后用改进后的 YOLOv8-EMA 模型在数据集上进行训练、验证、测试。

第 4 章动态环境下融合注意力机制的 YOLOv8 视觉定位，该章节主要是将 YOLOv8n-EMA 目标检测网络与 ORB-SLAM3 视觉定位系统进行融合，并在高动态的数据集上进行定位测试，同时与同类算法进行精度与性能对比。

第 5 章总结与展望，主要是对整篇论文进行概括总结，并提出目前使用的算法不足之处。

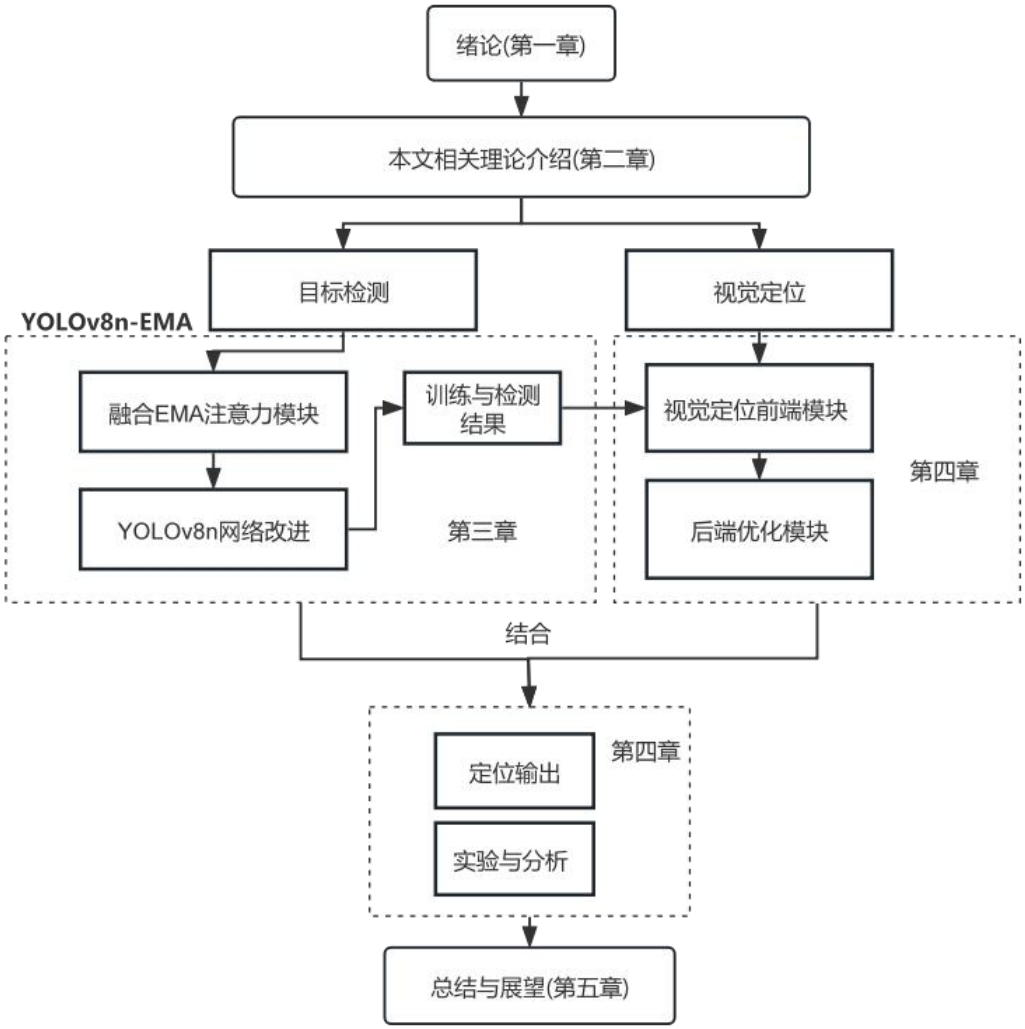


图 1-2 论文主要内容与章节安排

第 2 章 相关理论介绍

本章主要围绕基本原理进行说明，详细解释包括卷积神经网络的基本概念与组成、注意力机制的原理以及视觉定位原理描述。

2.1 卷积神经网络基本概念

2.1.1 卷积神经网络介绍

最初，Hubel 等人提出感受野(receptive field)[47]，随后，Lecun[48]提出了包含五个部分（输入层、卷积层、BN 层、激活函数、池化层、全连接层）的卷积神经网络（Convolutional Neural Network，CNN），卷积神经网络针对图像数据的特点，将数据传输和处理的维度定义为三维，即宽度、高度和深度。这使得网络能保留图像特征空间信息上的局部相关性，还能在深度维度上对像素点进行编码，从而分析出更丰富的高维度语义信息，减少了图像信息的丢失。

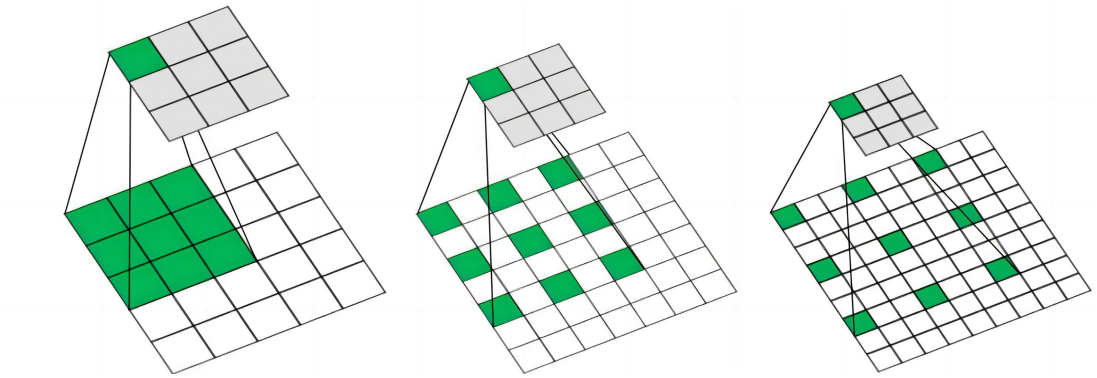
2.1.2 卷积层

卷积神经网络的关键为卷积层，其主要目标为借助卷积核计算图形特征，即将所有数据进行等分，并进行卷积运算，生成对应值，以此生成特征图。通常，卷积层有很多类型的卷积核，且对于不同的特征由不同卷积核进行计算。所使用的滤波器数量与所得特征图的深度一致，特征图尺寸输出大小的计算公式如式 2.1：

$$N=\frac{(W-F+2P)}{S}+1 \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots (2.1)$$

上述公式中，S(stride)即调整卷积步伐大小，从而调整特征图大小以及对输入数据的提取采样率进行控制，从而影响 CNN 的性能和特征提取能力。而 P(padding)则可以防止信息过分丢失。

另外，相关学者还提出扩张卷积[49]，也称为空洞卷积。如图 2-1，其可在同等条件下提升感受野。与传统卷积不同，它借助扩张率来解决窗口过小的问题，以达到快速加大感受野。以 3×3 为例，图(a)是传统卷积核，扩张率是 1，因此其有 9 个像素的感受野，图(b)的扩张率是 2，感受野扩大到 25 个像素，图(c)扩张率为 3，由图 2-1 所见，感受野有了明显提升。



(a) (b) (c)

图 2-1 扩张卷积过程

此外，对于一个好的网络模型而言,应同时注重准确率以及复杂度，因此有人提出带状卷积[50]，可以提升正确率，同时极大优化复杂度，如图 2-3,这里采取了一个较为极端的策略,直接将 $k*k$ 的基本卷积分解为 $k*1$ 和 $1*k$ 的两个卷积。在具体使用上，借助串联的方式，输出即为输入，可极大降低复杂度。

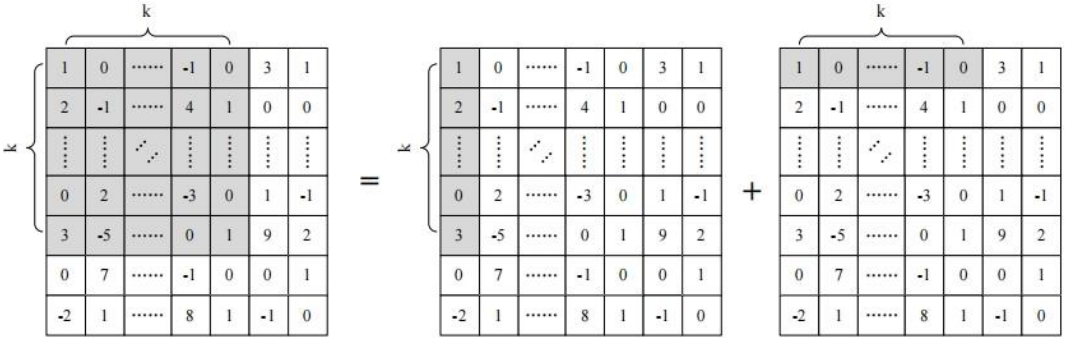


图 2-2 带状卷积过程

2.1.3 池化层

池化层是深度卷积神经网络中的一种层级。它的主要功能是通过降采样来获得数据量更小的特征图，以此来减少网络模型的计算量。

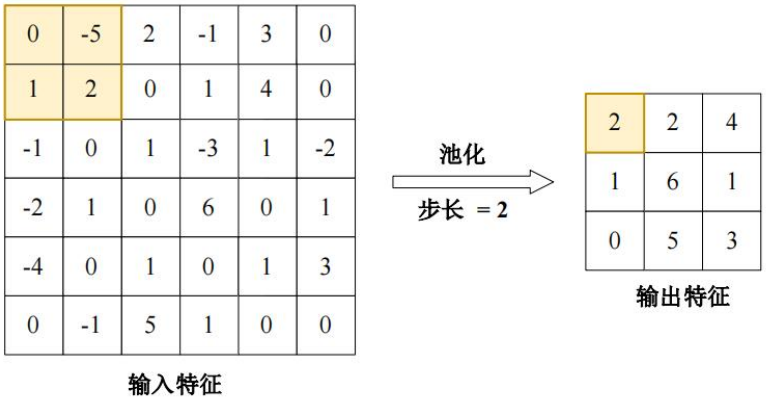


图 2-3 池化操作过程

如图 2-3 所示，池化操作是深度卷积神经网络中的一种降维技术，以 2×2 窗口和以 2 的步长为例，本次操作使用的是最大池化，选择在 2×2 的矩阵中选出最大值进行输出，然后再跨 2 个步长，最后获得一个只含有最大值的 3×3 的矩阵。除了最大池化，同样也有最小池化等操作。

2.1.4 激活层

非线性映射主要依赖于激活层，神经网络会借助激活层来习得复杂的数据。若不使用非线性映射，则每层之间进行简单的矩阵相乘，因此会无法拟合实际的数据，且不同激活函数有不同的功效。期中常见的主要有以下：

(1) Sigmoid 激活函数

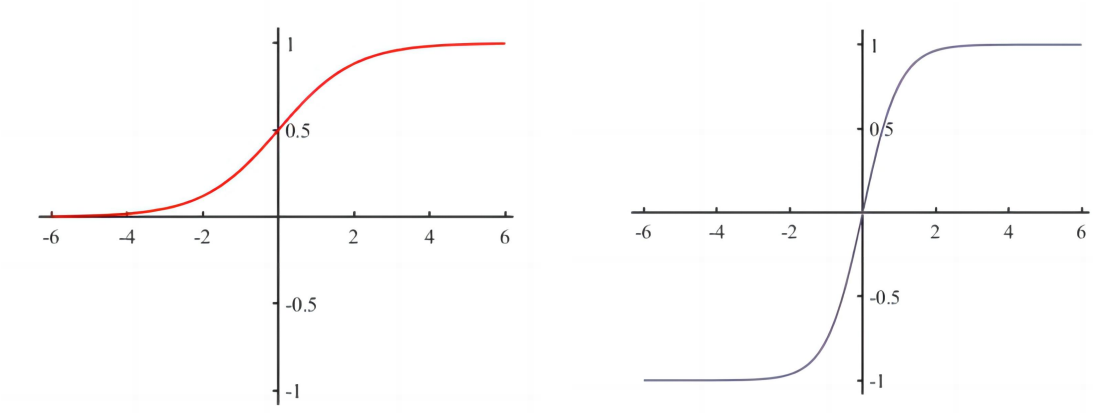
Logistic 函数及 Tanh 函数是 Sigmoid 型函数最为典型的代表，图 2-4 所示。Logistic 函数可将数据进行压缩。其数学表达式如下：

$$f_L(x)=\frac{1}{1+e^{-x}} \quad \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots (2.2)$$

神经网络具备了非线性建模功能，可以达成概率分布输出。但实际使用过程中，Logistic 函数有一定概率损坏数据，进一步加大训练难度，因此有学者提出 Tanh 函数：

$$f_T(x)=\frac{e^{x}-e^{-x}}{e^{x}+e^{-x}} \quad \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots (2.3)$$

如图 2-4，Sigmoid 型激活函数有可能导致梯度消失，因此不能大量使用。



(a)Logistic 函数　　(b)Tanh 函数

图 2-4 Sigmoid 型函数示意图

(2) Relu 函数

ReLU 函数包括其相关变体，为目前使用最多的激活函数。具有简洁高效的

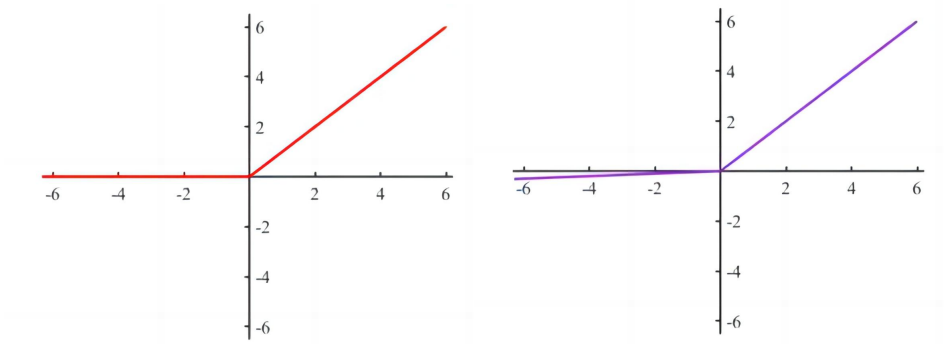
性能，对于梯度消失现象有极大的克制作用。表达式如 2.4：

$$f_R(x)=\begin{cases}x, & x\geq 0\\ 0, & x<0\end{cases}=\max(x,0) \quad \cdots\cdots\cdots\cdots\cdots\cdots (2.4)$$

如图 2-5(a)的函数加速了模型收敛，防止了梯度消失，但他仍会破坏数据分布，且有可能出现梯度消失的情况。为了解决上述问题，研究人员提出了许多版本，如 Leaky ReLU 函数，公式如 2.5：

$$f_{LR}(x)=\begin{cases}x, & x>0\\ \gamma x, & x\leq 0\end{cases} \quad \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots (2.5)$$

图 2-5(b)借助 ϒ可以处理神经网络的收敛性差问题。



(a)ReLU　　(b)Leaky ReLU

图 2-5 ReLU 和 Leaky ReLU 的函数示意图

（3）Swish 函数

Swish 函数兼具 ReLU 和 Sigmoid 函数的优点，公式定义如式 2.6：

$$f_S(x)=xf_L(\beta x) \quad \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots (2.6)$$

当 $\beta=0$，Swish 为线性映射；当 β 增大到一定值时，$f_L(\beta x)$ 的输出近似为 0，可见 Swish 受 β 参数影响很大。

（4）GELU 函数

GELU 函数与 Swish 函数接近，如公式 2.7：

$$f_G(x)=xP(X\leq x) \quad \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots (2.7)$$

其中 $P(X\leq x)$ 为高斯分布的累积分布函数。

2.1.5 损失函数

损失函数可以借助预期结果与实际结果之间的区别来改善模型。该函数有两种模型——分类损失与回归损失。在实际的运用上，会根据具体的目标来选择损失函数，以便模型最佳化，本节将会进行详细阐。

(1)交叉熵损失

交叉嫡损失（CELoss）主要运用方向为分类模型。对于样本，假定 p 为分布概率，那可 CELoss 可以表示为式 2.8：

$$\text{Loss}_{BCE} = -\sum_{i=1}^{N} y_i log(p_i) \quad \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots (2.8)$$

其中 N 表示类别的数量，y_i 表示样本的真实标签，p_i 表示模型输出的概率分布。交叉嫡损失若匹配，损失为 0，若模型不准确，损失的值越大。

(2)均方误差损失

均方误差损失(MSE Loss)。主要用于回归问题,预测连续值。假设模型对于第 i 个样本的了预测值为 y_i，而它的真实标签为 t，则均可表示为公式 2.9：

$$\text{Loss}_{MSE} = \frac{1}{N}\sum_{i=1}^{N}(y_i - t_i)^2 \quad \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots (2.9)$$

其中 N 表示样本数量。均方误差损失计算了预测与实际数据之间的差距，差距越大值越大，模型越失真。为了使均方误差达到最优解，通常会借助梯度下降等算法来处理。

2.2 注意力机制

由人类视觉注意感官的启发，深度神经网络也可以借助注意力机制来处理相关机器视觉任务。该机制是以自动选择的方式，对于认为该重点关注的对象进行特殊的强化，且无视其余地方。其最为重要的是注意热点图，它体现了各个区域的重要程度，随着注意力机制的发展，视觉任务也有了极大的进步。本章节会阐述几种常用的该机制。

2.2.1 空间注意力

在各个任务里，最至关的机制就是空间注意力。通过借助空间位置的比重，来关注任务所感兴趣的区域，并忽略噪声的干扰。空间注意力对全通道进行池化

操作，借此来减少维度，随后为了维持与原图的维度相同，在进行卷积操作增加通道数，最后获得特征图不仅与原图尺度相同，也融合了全局的权重系数。

2.2.2 通道注意力

通道注意力顾名思义，图片的每个通道都会有一定的自适应权重，从而能够有效地概括整幅图像的主要特征。具体而言，通道注意力的原理是对全局通道进行池化操作获取特征，再把特征结果进行前反馈，以获取各个通道的重要性权重。最终，通过将通道注意力的权重与原权重进行融合，获得经过优化的图像特征。此外，引入通道注意力不仅能够使网络更专注于所需的信息，还能在一定程度上优化网络。

2.2.3 时间注意力

时间注意力常常因为其处理动态时序有着很大优势，所以在关于视频的处理上，学者往往会选择时间注意力。但与本文研究特性关联性较少，因此不过多赘述。

2.2.4 分支注意力

在使用多尺度的模型时，分支注意力是相当明智的选择，其主要有两个原因：一是深度模型的有效信息存在于各个层次，二是现今图像训练单个图像往往会包含多个任务对象，需要更多的分支进行特征筛选，而分支注意力可以很好的完成这项任务。

2.2.5 自注意力

自注意机制的输入为有序的上下文信息，其优势在于：可平行化运算以及每一个向量均看过完整的输入，不用进行堆叠。自注意机制的实现过程进行具体的数学表达式如式 2.10：

$$Attention(Q,K,V)=Softmax\left(\frac{QK^{\mathrm{T}}}{\sqrt{d_k}}\right)V \quad \cdots\cdots\cdots\cdots\cdots\cdots(2.10)$$

其中 Q,K,V 分别表示查询序列、键序列和值序列；d_k 为 Q 及 K 的维度。此外，部分学者还发明了多头自注意力机制，极大增加了信息的全面性进一步提升了序列上下文信息获取的全面性。

2.3 视觉定位原理

2.3.1 针孔相机模型

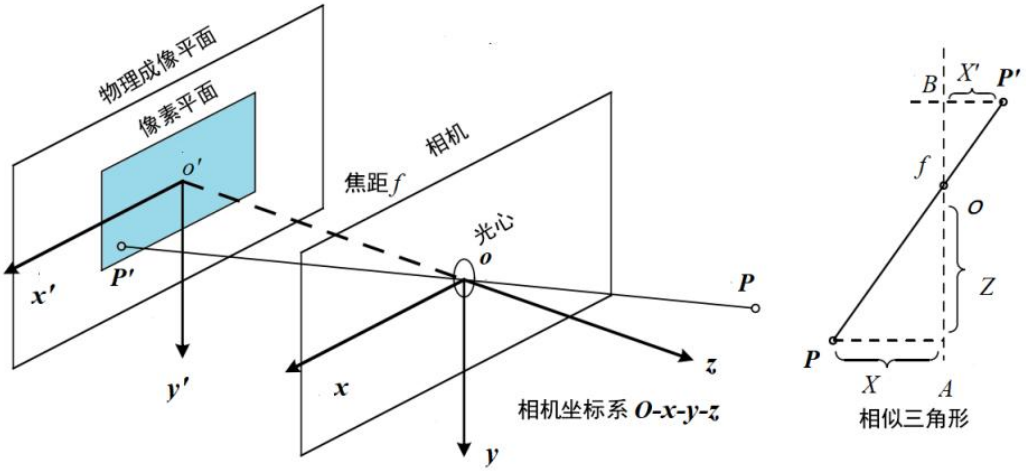


图 2-7 针孔相机模型

如图 2-7，O 为相机光心，P 为空间中一点，其坐标为。P 投影在$[X,Y,Z]^{\mathrm{T}}$。P *投影*成像平面上为点P'，设其坐标为$[X',Y',Z']^{\mathrm{T}}$，f 为焦距，根据相似三角形有如式 2.11：

$$\frac{Z}{f}=-\frac{X}{X'}=-\frac{Y}{Y'} \quad \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots (2.11)$$

式 2.11 中负号表示像是倒立的，整理得式 2.12：

$$\begin{cases} X'=f\dfrac{X}{Z} \\ Y'=f\dfrac{Y}{Z} \end{cases} \quad \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots (2.12)$$

图像上的每个位置都有固定的像素坐标，原点位于左上角，u 轴与 x 轴平行，v 轴与 y 轴平行，在像素坐标系中存在缩放和平移，同时原点平移了$[c_x,c_y]^{\mathrm{T}}$，则像素坐标$[u,v]^{\mathrm{T}}$，与成像平面上P'的关系为：

$$\begin{cases} u=\alpha X'+c_x \\ v=\beta Y'+c_y \end{cases} \quad \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots (2.13)$$

将式 2.12 代入式 2.13 并整理得:

$$\begin{cases} u = f_x \dfrac{X}{Z} + c_x \\ v = f_y \dfrac{Y}{Z} + c_y \end{cases} \quad \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots (2.14)$$

式 2.14 中，f_x 为 αf，f_y 为 βf，写成矩阵形式为式 2.15，K 为内参矩阵，

$$Z\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = KP \quad \cdots\cdots\cdots\cdots (2.15)$$

式 2.15 描述了空间中一点 P 在相机坐标系下与像素平面的关系。又由世界坐标系 P_w 根据相机位姿变化而来，可用式 2.16 表示：

$$ZP_{uv} = Z\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = K(RP_w + t) = KTP_w \quad \cdots\cdots\cdots\cdots\cdots (2.16)$$

式 2.16 式 2.16 描述了世界坐标系 P_w 与相素坐标系 P_{uv} 的关系。式中 R 和 t 称视觉定位要估计的目标。

2.3.2 经典视觉定位框架

视觉定位经过多年的发展，形成了如图 2-8 所示的经典框架，这一经典框架包括传感器数据获取与预处理、视觉里程计、回环检测与修正、后端优化与路径规划以及地图构建与更新，其中视觉里程计为系统框架中的前端，后端优化有回环检测等模块，首先使用视觉里程计获得初步的位姿估计，而后再使用后端优化对位姿进行修正，以此达到前后端共同实现机器人精准定位与环境建模的目的。

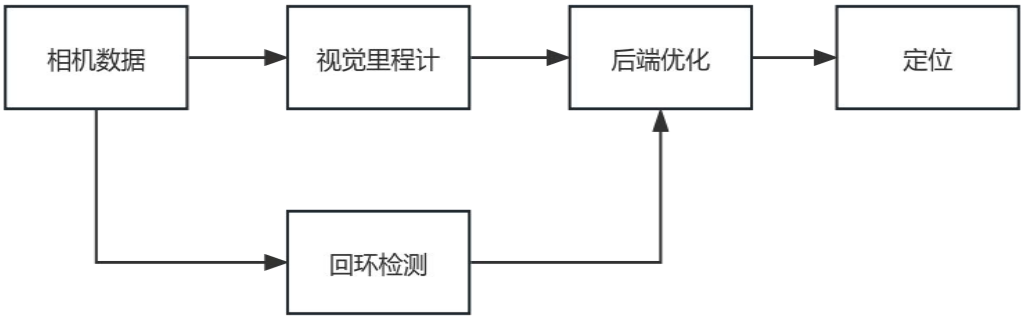


图 2-8 视觉定位框架

2.3.3 特征点法

特征点是图像中突出且具有特征的点，如角点、边缘、区块，用于定位与匹配，如图 2-9 所示。早期研究者在视觉上主要用角点，因为角点比区块更加具有代表性，如 Harris 和 FAST，但角点也有限制，比如在相机发生大幅度旋转或者遭遇到快速移动，角点很大概率会失效。所以 SIFT、SURF、ORB 等特征点被提出来去代表局部图像特征，比起角点更加具有可重复性、区别性、高效率和本地性，使其在计算机视觉中广泛应用。

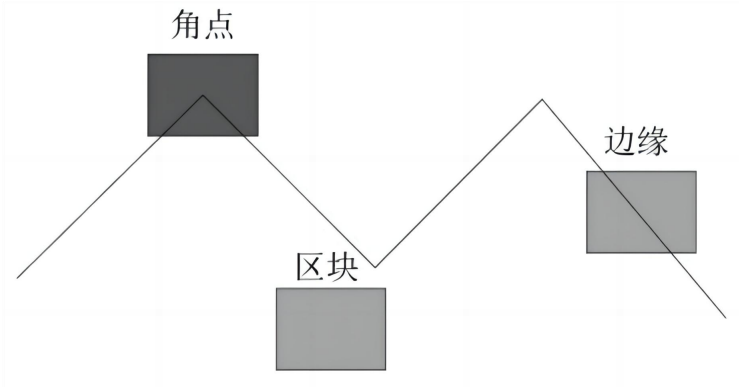


图 2-9 可作为图像特征的部分

SIFT 具有旋转和尺度不变性，但计算量大，不适用于实时处理；SURF 改进了 SIFT，计算量小，比较适合模糊图像的处理；FAST 速度快但精度略低；目前主流的 ORB 特征是在 FAST 基础上改进而来，其提取速度之快，更适用于视觉定位的实时处理。

2.4 本章小结

在本章中，首先介绍了神经网络的各个网络层的功能，然后再介绍了注意力机制的几个常用方向。最后介绍了目前主流的视觉定位的基本框架组成，有前端的视觉里程计，也有后端的优化。通过这些基础理论的铺垫，并在下一章中主要去介绍注意力机制在目标检测中的融合与改进，并提高目标检测的性能。

第 3 章 YOLOv8 目标检测融合注意力

3.1 引 言

注意力机制在显著性目标检测中可以提升性能，为了研究其内在关系以及可优化模型设计。我们在 YOLOv8 基础上提出了一种基于注意力机制 EMA 的卷积神经网络模型 YOLOv8n-EMA 来进行显著性目标检测。

3.2 目标检测

3.2.1 目标检测相关介绍

目标检测算法通常采用滑动窗口或区域生成的方法，在图像中搜索候选区域，并对这些区域进行分类，以确定是否存在目标物体。目标检测算法的性能通常用平均精度 mAP 来衡量，mAP 是指在不同召回率下的平均精度。

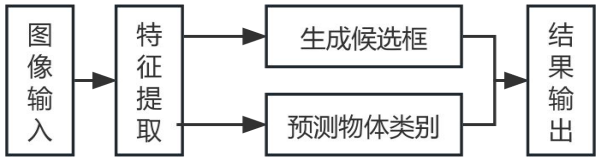


图 3-1 目标识别基本过程

目标检测通常涉及以下几个关键步骤：

特征提取：目标检测算法首先从输入图像中提取特征，以捕捉目标的区别性信息。这些特征可以是手工设计的，如边缘、纹理、颜色等，也可以是通过深度学习模型自动学习的高级特征。

候选框生成：在目标检测中，需要生成一组候选框，这些框可能包含图像中的目标。常见的方法包括滑动窗口和区域提议算法。滑动窗口将一个固定大小的窗口应用于图像的每个位置，并生成一系列不同位置和尺寸的候选框。区域提议算法则通过使用快速区域搜索或基于深度学习的方法，从图像中生成具有高概率包含目标的候选框。

目标分类与边界框回归：对生成的候选框进行目标分类和边界框回归。目标分类是将每个候选框分配给不同的目标类别或背景。通常使用机器学习算法，如

支持向量机 SVM、决策树或深度学习模型（如卷积神经网络）来实现分类。边界框回归则是校正候选框的位置和大小，使其更准确地拟合目标的边界。

非极大值抑制：由于候选框生成和分类可能会产生多个重叠的候选框，需要进行非极大值抑制来剔除重复检测。非极大值抑制通过计算候选框之间的重叠程度，并保留具有最高置信度的候选框，从而得到最终的检测结果。

目标检测在许多应用领域都具有重要的应用，随着深度学习的发展，基于深度学习的目标检测方法，如基于卷积神经网络的 Faster R-CNN、SSD、YOLO 等，取得了很大的进展，实现了更高的检测准确率和实时性能。

3.2.2 YOLOv8n 模型原理

YOLOv8n 算法由 Glenn-Jocher[51]提出，并在 YOLOv5 的基础上加以改进其主要特色为:

(1) 数据预处理。YOLOv8n 采用包括了图像缩放、图像增强、图像归一化等方式，提高模型的性能和鲁棒性。通过对原始图像数据进行预处理，可以使模型更好地学习图像中的目标物体，并提高模型在不同场景下的泛化能力。

(2) 网络结构。原 YOLOv5 的主干网络架构规律是通过每层使用步长为 2 的

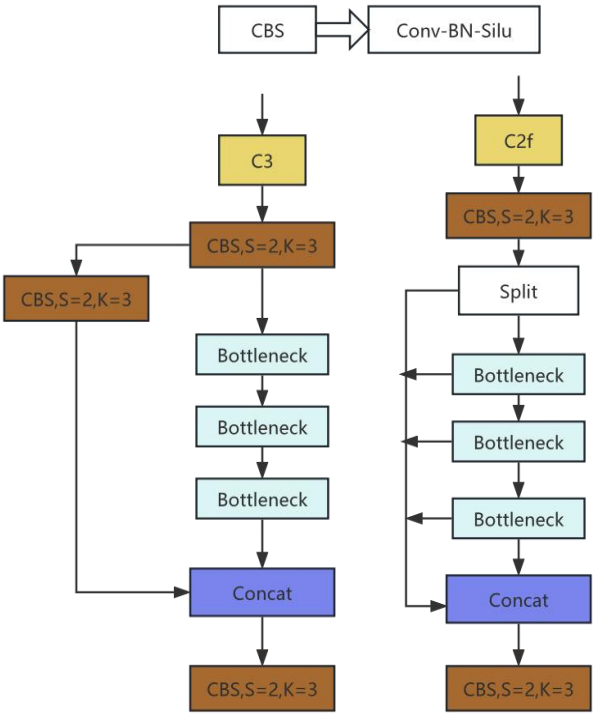


图 3-2 C3 模块和 C2f 模块示意图

3x3 卷积来获得降采样特征图，并借助 C3 模块来增强特征的输出，其深度的参

数可设置为"3/6/9/3"。在 YOLOv8n 中，C3 模块与 C2f 模块的网络模型如 3-2 所示。

(3) FPN-PAN 结构。YOLOv8n 采取 FPN+PAN，实现特征金字塔和多尺度信息融合，并且也使用 C2f 模块来取代结构中的 C3 模块，其他结构保持不变。

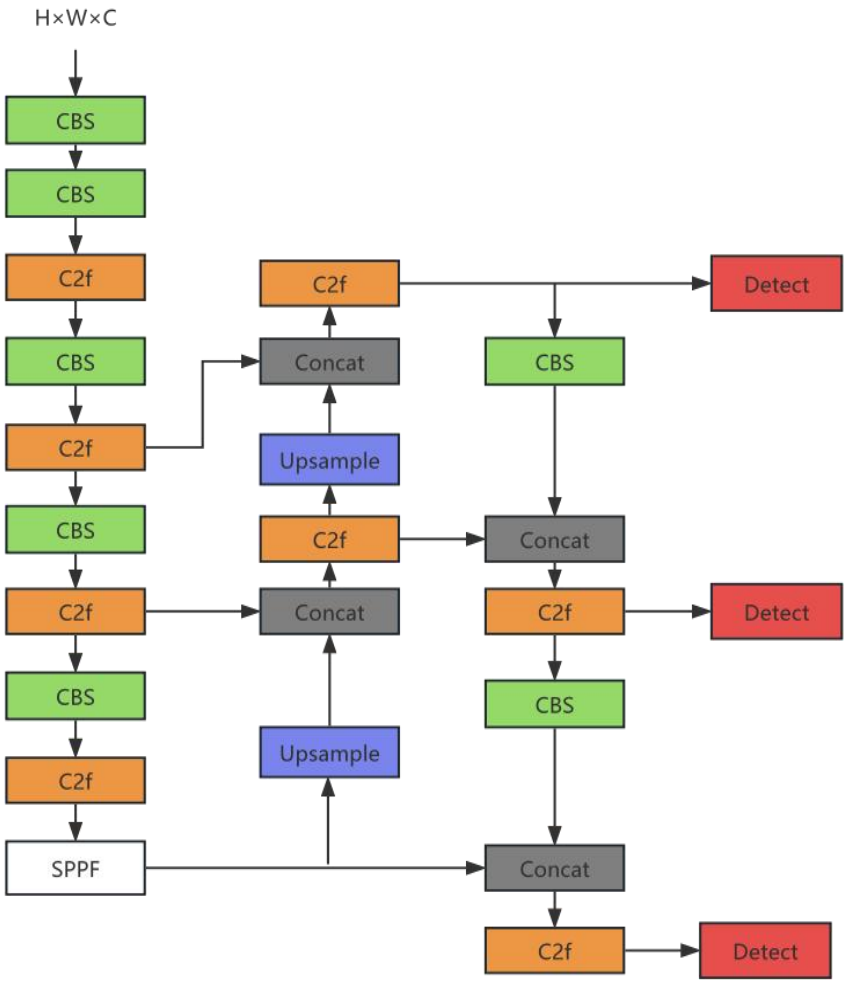


图 3-3 YOLOv8n 网络结构

(4) Detection head 结构。在检测头这里，YOLOX 使用的解耦头是 YOLO 系列的的关键提升，YOLOv3 至 YOLOv5 使用的是耦合头，仅使用卷积操作来处理分类。

(5) 标签分配策略。YOLOv5 引入了自动聚类候选框的功能，但受数据集限制，可能导致候选框与真实物体尺寸比例不匹配，聚类出来的检测框也不够理想。相比之下，YOLOv8n 采用多尺度分配解决正负样本匹配，与 YOLOv6 类似，都

采用动态标签分配策略。所以，YOLOv8n 一共有两种损失函数：类别损失与位置损失。

其中 Varifocal Loss 定义如式 3.1：

$$VFL(p,q)=\begin{cases}-q(q\log(p)+(1-q)\log(1-p)) & q>0\\ -\alpha p^{\gamma}\log(1-p) & q=0\end{cases} \quad \cdots\cdots\cdots (3.1)$$

公式中，p 为类别分数，q 为期望得分，强调高质量的正样本，对正样本使用参数 q 进行加权，选择 CIoU 较高的样本，则对损失的减少更有帮助；同样的，负样本使用参数 p 的γ次幂降低权重，这样会减少其对整体损失的贡献。

3.3 YOLOV8n 融合 EMA 注意力

3.3.1 EMA 注意力模块

EMA 注意力模块主要有两部分组成，一个是采用重协调注意力机制的模块，另一个是使用交叉注意力机制的模块。

如图 3-4 所示，该模块为重协调注意力模块(Revisit Coordinate Attention,CA)，CA 模块使用全局平均池化操作来进行跨通道信息的融合来增强特征，获得空间上的长距离交互，提升感受野。该模块一共三个通道，利用两个通道进行平均池化与卷积操作，获得更多得特征图像细节，第三个通道与这两个通道获得得特征图与权重值进行权重得更新，更新之后得特征图再继续传递至下一层网络当中。

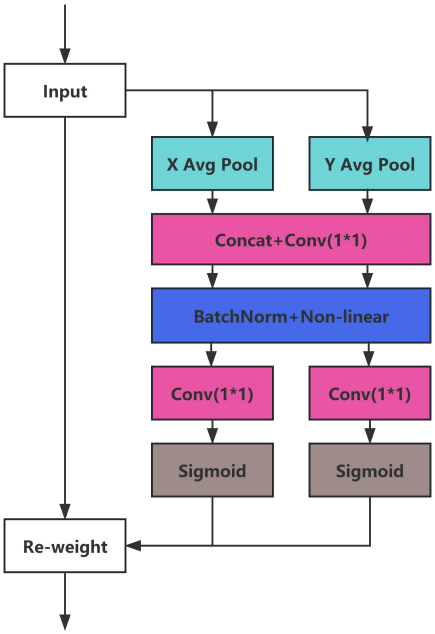


图 3-4 CA 模型网络结构

图 3-5 则为 EMA 注意力模块，其并行结构有助于网络去避免更多的串行处理，只从 CA 模块中挑选出1×1卷积的共享组件，在我们的 EMA 中将其命名为1×1分支。为了聚合多尺度空间结构信息，将3×3内核与1×1分支并行放置以实现快速响应，将其命名为3×3分支。EMA 采用并联的方式来获取特征图的注意力权重描述符，包括两个1×1分支和一个3×3分支，使用跨通道信息交互合并来

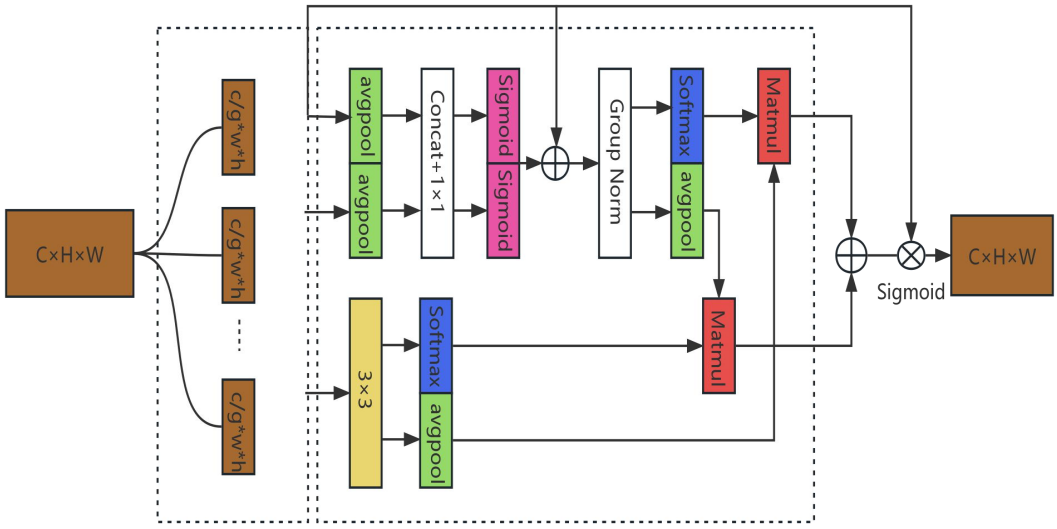


图 3-5 EMA 网络模型图

减轻计算量以及获取通道的依赖关系。详细的解释则是，1×1分支使用全局平均池化编码通道信息，并对通道进行编码，使用3×3分支堆叠单个3×3内核来捕获多尺度特征。

EMA 网络的流程图如图 3-5 所示，首先该模型将整个流程分为三条线路：第一条线路则保留原有特征图组，为后续的权重更新提供初值，保证整个特征不会因池化等降采样而失真严重。第二条线路首先经过 CA 注意力模块进行预处理，与第一条线路的特征图的权重初值进行第一次更新，迭代出新的权重系数。再将更新好的权重系数与特征图一并送入到交叉注意力中，在通过一次正则化操作后，将特征图与第三条路线处理好的特征图进行图融合，准备后续的权重再次更新。第三条路线比较简单，首先使用3×3的卷积进行降采样，首先将降采样后的特征图同步分成两步走，第一步是与第二条线路处理好的特征图进行图融合，第二步则是进行正常的池化与归一化操作，再进行图融合。最后第二条线路与第三条线

路交叉融合图进行合并，与第一条线路的原特征图与权重初值进行最终更新，输出最终的权重与特征图。

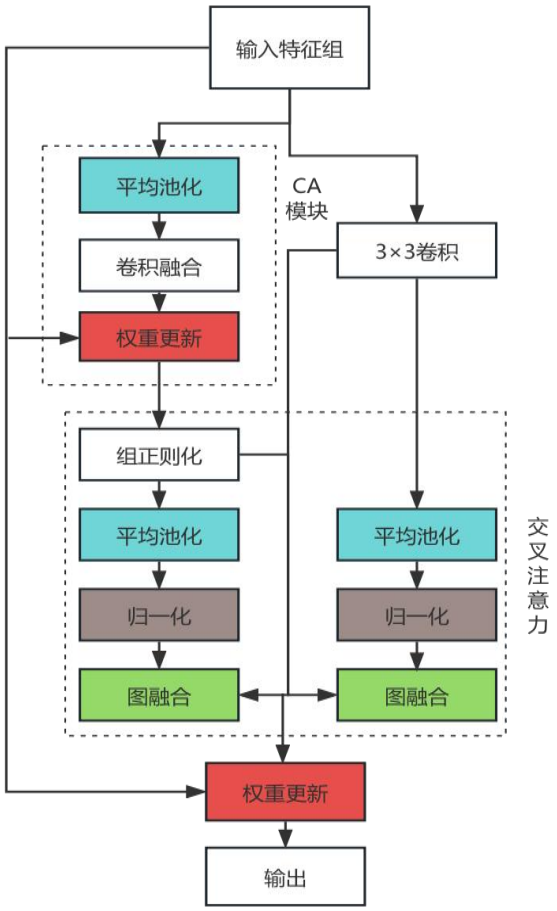


图 3-6 EMA 流程图

3.3.2 C2f 融合 EMA 模块

在目标检测中，生成的边界框用来描述目标在图像中的位置，首先，没有注意力机制的目标检测算法往往只关注整个图像的全局信息，无法有效地聚焦于目标区域，导致对目标的定位不够准确。其次，这种方法对于大尺度和小尺度目标的检测效果可能不理想，无法灵活地适应不同尺度目标的特征表示。此外，没有注意力机制的目标检测在处理密集目标或目标重叠的情况下也容易出现误检测或漏检的问题，而且卷积神经网络在对一张图片数据进行处理时，由于卷积核是固定的，所以同一张图片不同位置的感受野都是相同，且对特征图的不同位置没有额外的依赖与联系，所以加入注意力机制可以是网络具有自适应调整感受野的能力。

添加 EMA 注意力机制旨在增强卷积神经网络对目标物体检测能力，并扩大其感受野。本文将 EMA 注意力模块与 YOLOv8n 的 BACKBONE 与 HEAD 网络结构中的 C2f 模块结合。

首先 C2f 的网络结构在图 3-2 已经给出，本文将在 C2f 网络的输出处接入 EMA 网络结构，融合图如图 3-7 所示，并称该模块为 C2f-EMA 模块。

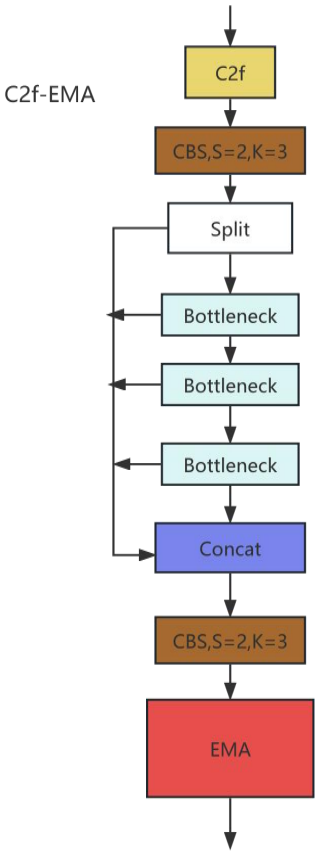


图 3-7　C2f-EMA 模块

其次，复杂的背景以及不规则的目标物体形状都有可能导致传统卷积的目标检测的鲁棒性降低，进行影响模型性能；本文将 C2f-EMA 模块加入在 YOLOv8n 网络模型中，可在需要关注的不规则物体上获得较多的特征权重，提取到更多的特征细节，从而为后续的检测奠定基础，此时改进后的网络模型如图 3-8 所示：

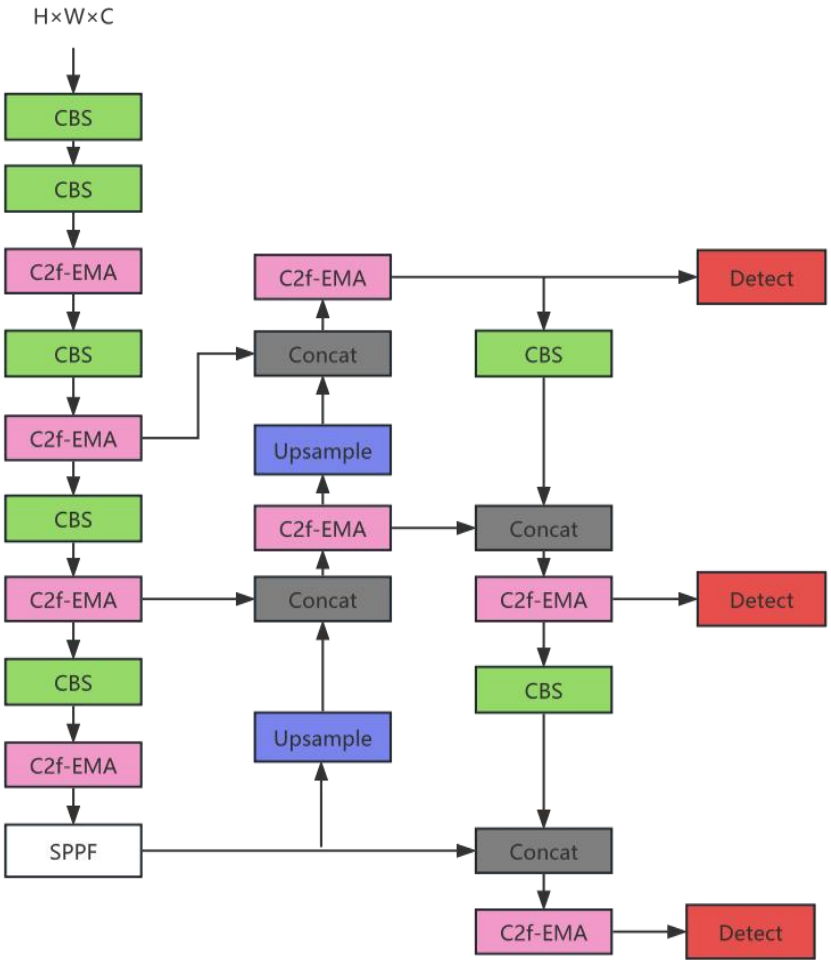


图 3-8　融入 EMA 的 YOLOv8n 模型

3.4 实验结果与分析

3.4.1 实验环境

本实验所使用的实验环境如表 3-1，3-2 所示：

表 3-1 硬件配置

硬件配置	
CPU	Intel Xeon 4210R 2.4GHz
内存	4G
显存	48G
显卡	NVIDIA GeForce A40

表 3-2 软件配置

软件配置	
操作系统	Ubuntu18.04
Python	3.8
Pytorch	1.10.0
CUDA	10.2
cuDNN	7.6.5

3.4.2 数据集介绍

PASCAL VOC 数据集最初由标注了 20 个类别的图像组成，后续版本增加了更多类别，并提供了不同年份的挑战赛数据集，如 PASCAL VOC 2007、PASCAL VOC 2012 等。每个图像都包含了一个或多个目标的标注信息，包括目标类别和边界框的位置。

该数据集广泛应用于目标检测和图像分割算法的评估和比较。它提供了一个标准化的评估框架，包括平均精确度(Mean Average Precision,mAP)等指标，用于衡量算法在不同类别上的检测性能。数据集总共累计有 5200 张数据图像，在测试中使用的一共是 1000 张，然后用其余的 4200 张图像进行训练推理和测验，并且所有照片的标签、大小、位置都有真值，并且存储的格式使用的是 XML，利于读取且格式化效率高。

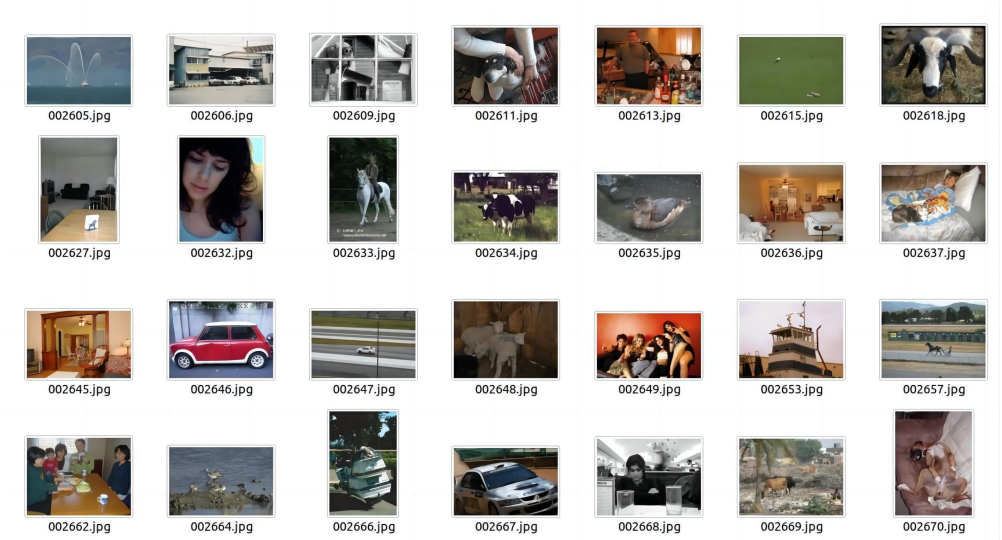


图 3-9 PASCAL VOC 数据集

3.4.3 数据集格式转换

已知 YOLO 系列目标检测训练读取的标签格式为 TXT，所以这里将 PASCAL VOC 数据集的 XML 标签全部转换为 YOLO 的 TXT 格式，并提取类别和坐标信息，并保存在相应的 TXT 文件中，格式转换完成后，再供 YOLO 算法进行训练验证。

3.4.4 实验评价指标

混淆矩阵可以展示与检测分类模型性能的矩阵图，在进行二分类的工作中，混淆矩阵大部分包括四种结果：真正 (True Positive,TP)、假正 (False Positive,FP)、真负 (True Negative,TN) 和假负(False Negative,FN)，如表 4.3 所示。

本研究从混淆矩阵不仅可以得到部分评价指标，亦可借此指标分析模型的不足之处，并以此进行优化。

在目标分类与检测之中，其性能指标总共有以下三种：

(1) 召回率 (Recall) ，Recall=$\frac{TP}{TP+FN}$ 。

(2) 精确率 (Precision) ，Precision=$\frac{TP}{TP+FP}$ 。

(3) 准确率 (Accuracy) ，Accuracy=$\frac{TP+TN}{TP+TN+FP+FN}$ 。

P-R 曲线表示召回率与准确率之间的关系，通常用于评估模型在不同阈值下的性能表现。平均精确度 mAP 则是 P-R 曲线下方的面积，表示模型在所有类别上的性能综合。mAP 是衡量目标检测模型性能的重要指标之一，其计算公式如式 3.2：

$$mAP=\frac{1}{N}\sum_{i=1}^{N}AP_i \quad \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots (3.2)$$

在不同训练配置下，mAP 会呈现不同的变化形式，在目标检测中，mAP@0.5 和 mAP@[.5:.95]是最常见的两个评估指标。

mAP@0.5 指的是目标检测中，当 IoU 等于 0.5 时，取各种类的平均精度，可以正常用于算法性能的评价。而 mAP@[.5:.95] 则表示在目标检测中，IoU 从 0.5 到 0.95 变化时，所获得的各种类的平均精度，这个指标由于范围大，所以考虑所有 IoU 阈值下的精度，反映算法在多个 IoU 阈值下的平均表现。在一些严格的评价中，mAP@[.5:.95]被认为更能评估算法性能。

3.4.5 模型性能评估

为了提升 YOLOv8 的检测能力，本文基于 YOLOv8n 模型进行了 EMA 注意力机制的改进，并把添加了 EMA 模块的 YOLOv8n 称为 YOLOv8n-EMA 模型，最后运用 IoU 计算方法来评测 YOLOv8n-EMA 的性能。为了展示改进 EMA 注意力机制对 YOLOv8 模型的提升，并用验证集的图片去测试模型的表现是否理想。本实验的训练参数配置如表 3-3 所示：

表 3-3 训练参数配置

名称	数值
输入图片分辨率(image size)	640*640*3
迭代次数(epochs)	100
优化器(optimizer)	SGD
工作队列数量(number of workers)	8
学习率(learning rate)	0.01
耐性(patience)	50

3.4.6 收敛性分析

本文通过经过优化后的模型在数据集上的收敛性能曲线对比，我们比较了 YOLOv8n 模型和 YOLOv8n-EMA 模型的收敛情况。这两个模型的边界框损失、置信度损失和类别损失等损失函数的收敛曲线分别展示在图 3-6 和图 3-7 中。

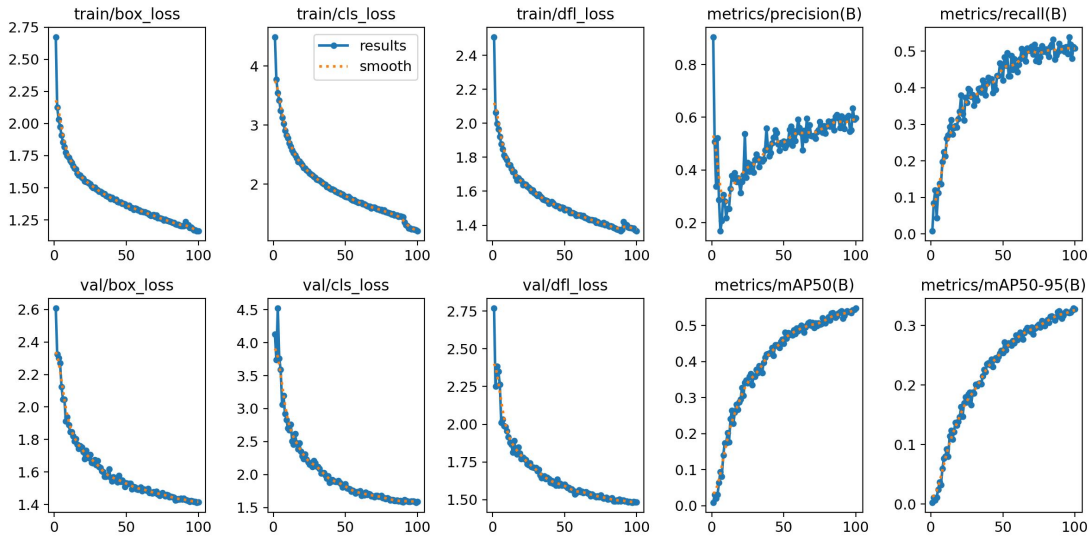


图 3-6 YOLOv8n 在 PASCAL VOC 数据集上的收敛曲线

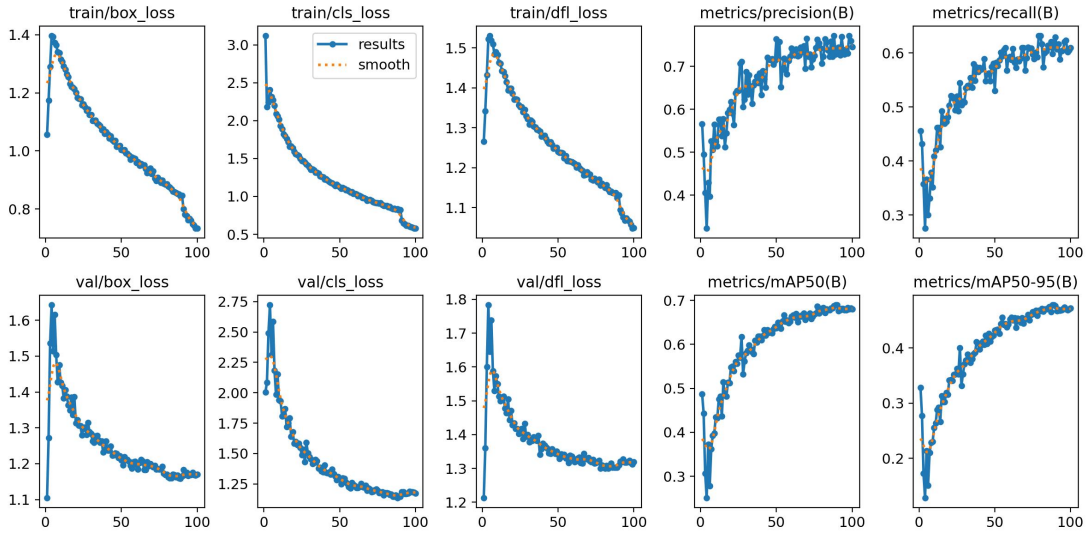


图 3-7　YOLOv8n-EMA 在 PASCAL VOC 数据集上的收敛曲线

通过图 3-6，3-7 可以得出 YOLOv8n 模型与 YOLOV8n-EMA 模型的 loss 值都保持在低状态，但 YOLOv8n 的置信度损失并没有很好的收敛平均 loss 都在 1.5 以上，并且在 epoch 早期出现跳跃的趋势。而其中 YOLOV8n-EMA 分类收敛性均优于 YOLOv8n。在精确度 P、召回率 R 以及 mAP@0.5 和 mAP@[.5:.95]这四个指标的比较中可以看到，YOLOv8n-EMA 模型的 P 与 R 分别达到了在 0.7 和 0.6。

3.4.7 二分类精度评估

YOLOv8n 和 YOLOv8n-EMA 这两个由实验产生的混淆矩阵如图 3-8 与图 3-9 所示。

YOLOv8n-EMA 模型相较 YOLOv8n，在分类准确度上有轻微提升，混淆率略微降低。其中，tvmonitor 的准确度提升最显著，达 19%，其他类别提升在 2%至 15%之间。混淆矩阵显示，tvmonitor 分类最高，达 84%，person 次之，为 82%，而 bird 最低，仅 35%，显示模型对不同类别的关注度差异。

针对小目标如 bird 和 dog，在 YOLOv8n-EMA 中分类精度得到改善。在 YOLOv8n 中，bird 和 dog 的准确度分别为 29%和 52%，而在 YOLOv8n-EMA 中分别提升至 35%和 67%，分别增加了 6%和 15%。这表明 YOLOv8n-EMA 能有效提高对狗、鸟这类小型动物的检测精度。

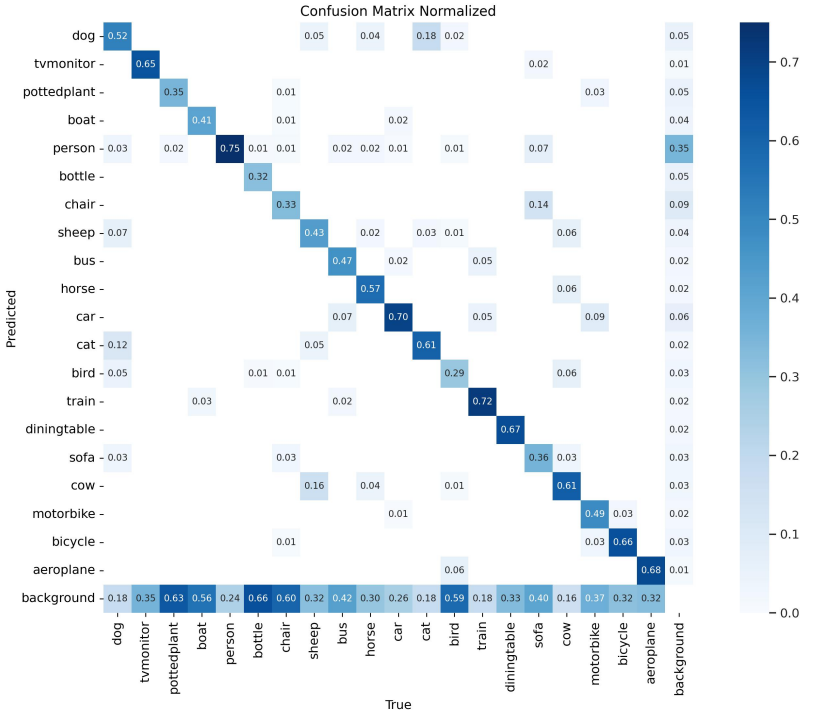


图 3-8　YOLOv8n 在 PASCAL VOC 数据集上的混淆矩阵

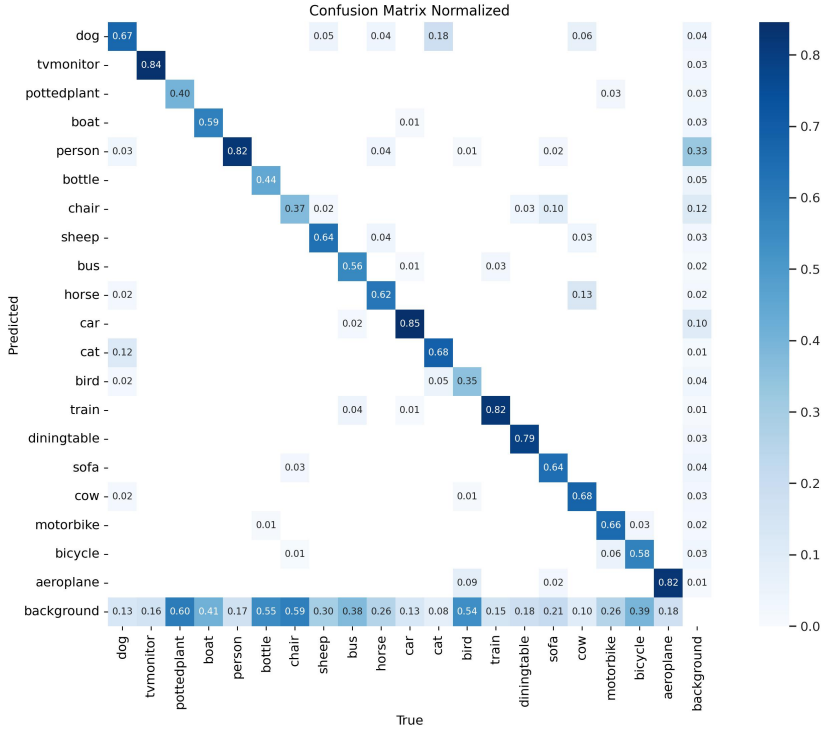


图 3-8　YOLOv8n-EMA 在 PASCAL VOC 数据集上的混淆矩阵

3.4.8 对比实验

为了确认通过添加 EMA 模块优化策略对模型性能带来的增益，本实验在 VOC 数据集对所有类别进行了分别统计与汇总，以及 P-R 曲线的绘制。

如下表 3-4 为 YOLOv8n 在 VOC 数据集下的数据统计表，表 3-5 为 YOLOv8n-EMA 在 VOC 数据集下的实验数据统计。通过对比可知，两个模型在一些较大物体上的精确确率大致相同，例如 aeroplane 的精确率 P 都是 0.7，但是在召回率上 YOLOv8n 模型的性能稍差一点，这也导致了 mAP 指标也随之下降。在一

表 3-4 YOLOv8n 实验数据统计表

Class	Image	Instances	P(%)	R(%)	mAP@.5(%)	mAP@[.5:.95](%)
dog	501	60	0.538	0.583	0.594	0.391
tvmonitor	501	31	0.729	0.581	0.743	0.527
pottedplant	501	60	0.435	0.317	0.274	0.118
boat	501	34	0.378	0.382	0.348	0.134
person	501	541	0.693	0.708	0.749	0.421
bottle	501	88	0.49	0.273	0.31	0.167
chair	501	139	0.454	0.305	0.308	0.156
sheep	501	44	0.472	0.5	0.482	0.275
bus	501	45	0.617	0.511	0.542	0.398
horse	501	47	0.732	0.58	0.644	0.374
car	501	195	0.769	0.665	0.742	0.463
cat	501	38	0.557	0.553	0.558	0.331
bird	501	82	0.543	0.293	0.311	0.161
train	501	39	0.758	0.722	0.772	0.463
diningtable	501	33	0.703	0.646	0.605	0.318
sofa	501	42	0.418	0.357	0.359	0.252
cow	501	31	0.423	0.613	0.556	0.344
motorbike	501	35	0.566	0.486	0.572	0.365
bicycle	501	38	0.676	0.632	0.667	0.451
aeroplane	501	34	0.706	0.676	0.755	0.458

些小的物体，例如 dog、cat 等目标，YOLOv8-EMA 的各项指标明显优于 YOLO v8n 的各项指标。这展现了添加 EMA 注意力机制在提升模型性能方面，尤其在提升小目标的检测性能上更加重要。

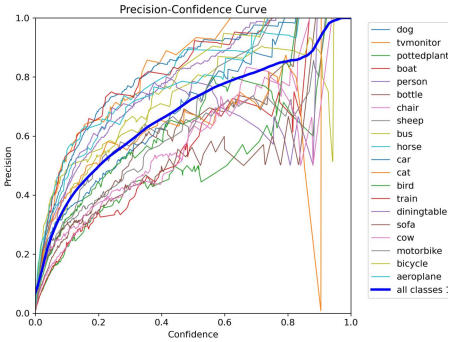
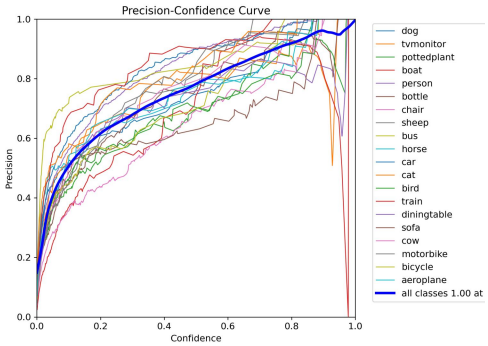
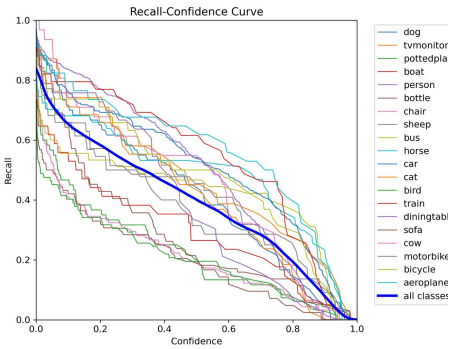
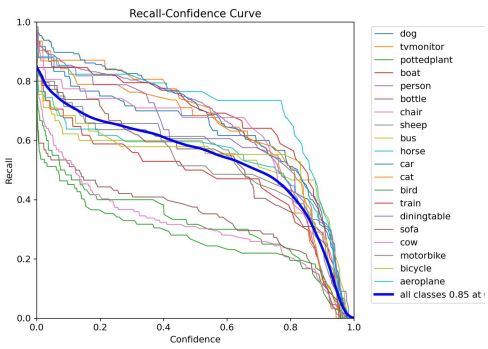
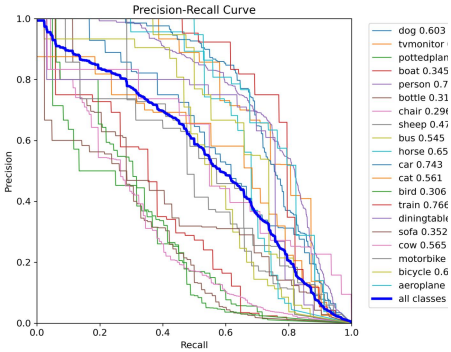
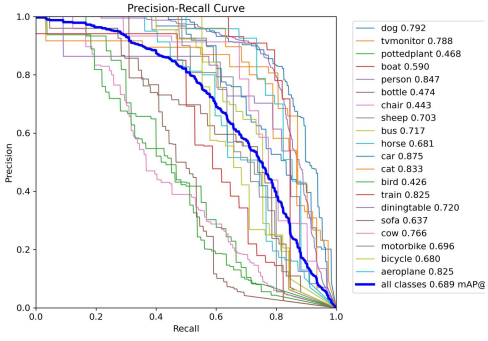
表 3-5 YOLOv8n-EMA 实验数据统计表

Class	Image	Instances	P(%)	R(%)	mAP@.5(%)	mAP@[.5:.95](%)
dog	501	60	0.684	0.7	0.793	0.603
tvmonitor	501	31	0.778	0.806	0.787	0.523
pottedplant	501	60	0.651	0.367	0.465	0.233
boat	501	34	0.638	0.529	0.591	0.364
person	501	541	0.835	0.777	0.846	0.549
bottle	501	88	0.757	0.409	0.479	0.308
chair	501	139	0.604	0.331	0.447	0.264
sheep	501	44	0.754	0.614	0.704	0.430
bus	501	45	0.849	0.6	0.716	0.616
horse	501	47	0.697	0.596	0.685	0.487
car	501	195	0.848	0.769	0.87	0.579
cat	501	38	0.779	0.744	0.831	0.646
bird	501	82	0.673	0.302	0.426	0.262
train	501	39	0.857	0.771	0.822	0.577
diningtable	501	33	0.707	0.659	0.728	0.505
sofa	501	42	0.593	0.595	0.642	0.484
cow	501	31	0.762	0.71	0.767	0.541
motorbike	501	35	0.794	0.6	0.697	0.548
bicycle	501	38	0.668	0.579	0.679	0.548
aeroplane	501	34	0.704	0.794	0.826	0.624

表 3-6 精度平均结果

Method	P(%)	R(%)	mAP@.5(%)	mAP@[.5:.95](%)
YOLOv8n	58.3	51.9	54.5	32.8
YOLOv8n-EMA	73.2	61.3	69.0	48.0

表 3-7 P、R、P-R 曲线对比

Method	YOLOv8n	YOLOv8n-EMA
P 曲线		
R 曲线		
PR 曲线		

从表格 3-6 总体平均下来的可以看出，在 YOLOv8n 的基础上单独增加 EMA 注意力模块分别能提高 14.9%的精准率 9.4%的召回率，并且在 mAP@.5 与 mAP@[.5:.95]指标上分别提升 14.5%与 15.2%，有效地提升了 YOLOv8n 检测模型的性能。

为了更好地展示这 2 个模型实验的对比结果，本文在表 3-7 中绘制了 P 曲线、R 曲线、P-R 曲线，已知，P-R 曲线与坐标轴围成的面积越大，则 AP 值越高，mAP 值就越高，代表模型性能越好。

由表 3-7 显示，相较于 YOLOv8n-EMA 与 YOLOv8n 两个模型，YOLOv8n-EMA 模型虽不是每一类目标检测的 AP 值最高的，但平均来看 YOLOv8n-EMA 模型在每一类目标检测效果都表现较好的，并且整个 P-R 曲线的所占的面积更大，所以可以证明 YOLOv8n-EMA 模型在整体上的表现是优于原有 YOLOv8n 模型的。

3.4.9 检测实验

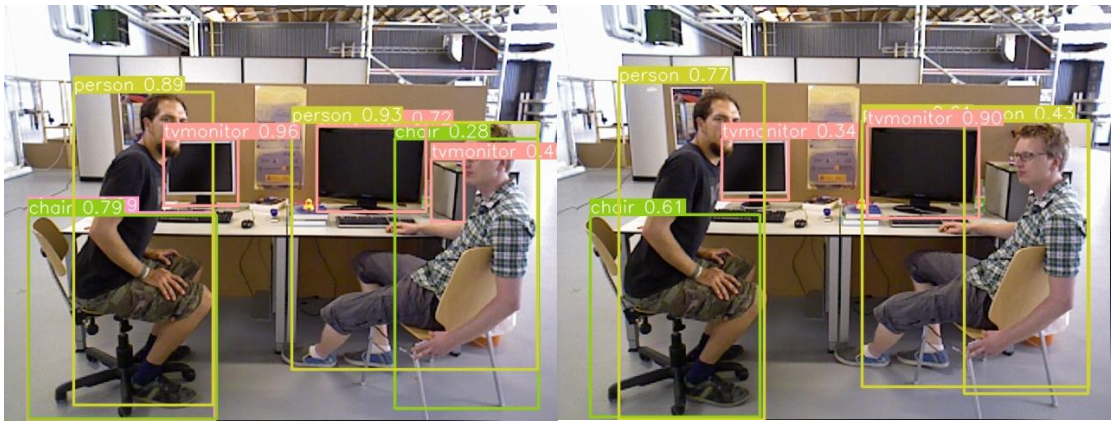
本检测将在 VOC 数据集与 TUM 数据集对 YOLOv8n 网络以及优化过后的 YOLOv8n-EMA 网络进行实例检测。其中图 3-9(a)，图 3-10(a)为 YOLOv8n 的效果图；图 3-9(b)，图 3-10(b)为优化后模型的检测结果。

以图 3-9(a)举例，YOLOv8n 无法检查到较小的目标，特别是当目标出现在背景中且与背景混为一体时，然而添加了 EMA 的 YOLOv8n 则表现更好，基本无漏检，还解决了 YOLOv8n 有时无法检查到与背景混为以一体甚至被遮住的小目标。

(a)YOLOv8n 检测结果　　(b)YOLOv8n-EMA 检测结果

图 3-9　VOC 数据集测试结果

以图 3-10(a)为例，YOLOv8n 模型虽然把人物全部检测出来，但把一些不是显示器的物体错误检测成了显示器，而本文融合 EMA 注意力机制的 YOLOv8n 则检测正确。实验表明本研究的 YOLOv8n-EMA 在检测上性能更优于原 YOLOv8n 网络，如图 3-10(b)所示。



(a)YOLOv8n 检测结果　　(b)YOLOv8n-EMA 检测结果

图 3-10　TUM 数据集测试结果

3.5 本章小结

在本章提出了一种新的目标检测模型 YOLOv8n-EMA，该模型在 YOLOV8 的基础上融合了 EMA 注意力机制。EMA 注意力机制通过并行处理和分组特征图提取。且通过大量实验结果，可以证明 YOLOv8n-EMA 网络比 YOLOv8n 网络拥有更好的检测能力，mAP 平均提升量在 15%，并且为下一章的动态环境下的视觉定位提供较为精确的动态物体的先验信息。

第 4 章 基于注意力的 YOLOV8 移动机器人视觉定位

4.1 引 言

目前，大部分移动机器人视觉定位系统都是只针对静态场景而设计的，而对于动态环境的主流方法都是在已有的视觉定位系统之上进行改进。

通过第二章视觉定位原理的介绍可知，一般移动机器人视觉定位分为前端与后端，前端一般是以视觉里程计为主，本章节致力于在视觉定位的前端进行改进，设计了一种针对高度动态环境的视觉定位系统，其做法是将第三章训练好的 YOLOv8n-EMA 目标检测模型融合进 ORB-SLAM3 系统，在视觉定位系统实时运行的同时，也能够同时识别出动态物体上的特征点，从而达到剔除的效果，从而提高整个视觉定位系统在动态环境下的鲁棒性。

4.2 视觉定位改进框架

本小节主要阐述一个适用于动态环境的移动机器人视觉定位系统的框架流程，主要采用融合 EMA 注意力机制方法的 YOLOv8 目标检测网络，为整个定位

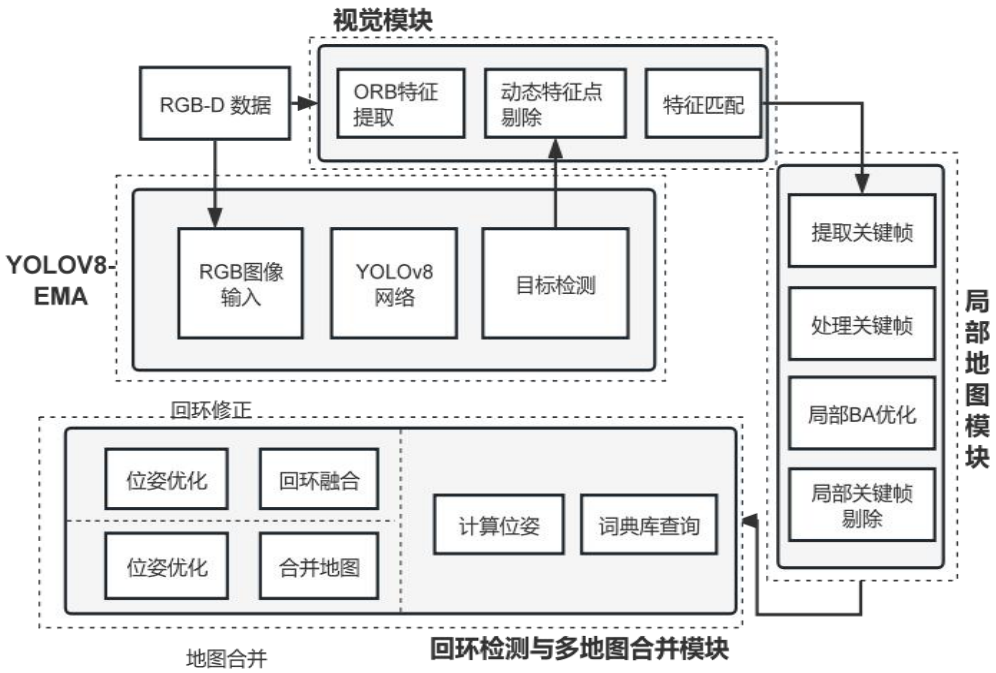


图 4-1 视觉定位数据流图

系统提供语义先验，最后利用语义先验信息来直接去除动态特征点，从而减少动

态物体对系统的影响，提高系统在动态环境下的鲁棒性。

本文视觉定位系统是基于 ORB-SLAM3 系统进行改进的，主要数据流如下：首先 RGB-D 摄像机所采的 RGB 图像会送入 YOLOv8-EMA 目标检测线程进行实时检测运动物体的先验目标框，并将此先验目标框送入视觉模块中进行等待 ORB 特征的提取，而 RGB 图像还会送入视觉模块的线程进行 ORB 特征点和描述子的提取，最后判断 ORB 特征点是否在先验目标框中，若在则剔除，若不在则保留。然后将保留下来的静态特征点进行特征匹配，最后通过关键帧提取策略选出合适的特征匹配图进行进一步的后端优化，最后再利用回环检测算法为整个视觉定位系统计算出的位姿进行修正，得到最终的定位数据，即位姿估计。

4.3 特征提取与匹配

4.3.1 ORB 与 FAST 特征点提取

在视觉定位中，常采用基于特征点的间接法，即在图片上选择具有代表性的点来进行两帧图像的匹配。由第二章介绍了部分特征点及其优劣势，ORB 特征点主要由旋转性 FAST 关键点和 BRIEF 描述子组成， 旋转性 FAST 关键点，即在 FAST 角点提取时为角点添加方向向量，这样图片在旋转时也是不影响特征点的提取的，而 BRIEF 的作用是可直接比较描述子可判断两点是否相同，以供后续的特征匹配进行使用。因此 ORB 由具有明显的速度优势和旋转不变性，成为了主流的视觉定位方法。

FAST 角点提取依赖于灰度值的差值，如果图片中存在单个像素灰度变化比较突出的状况，那么很有可能就是一个角点。其算法过程如下：

1. 通过 RGB 图像转换为灰度图，设定阈值 T，即灰度差值。
2. 在灰度图上遍历像素 P，设其灰度为 G。
3. 并在每个像素周围画半径为 r 的圆，并取出 A 个像素点。
4. 若圆上连续 M 个像素明显大于或小于中心像素 G ± T，则认为该点是角点。
5. 对遍历的每个像素都进行 2-4 步的过程。

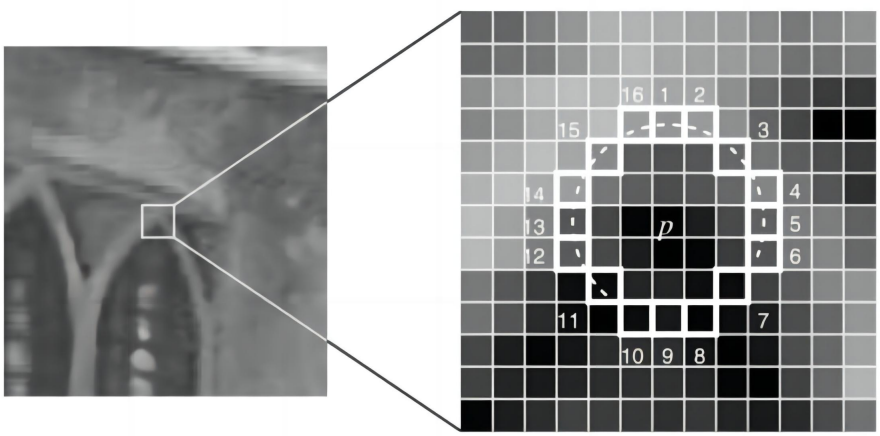


图 4-2　FAST 角点

在一帧图像中，在图像中大多数像素为无用点，为加速检测，可增加一步预检测，选定特定位置的点，只有四个大于 G+T 或者小于 G -T 时，才可成为可能的角点。

FAST 角点存在重复性低、分布不均匀和缺乏方向等问题，可能尺度不一致。ORB 在 FAST 基础上增加了旋转和尺度不变性。通过灰度质心法实现旋转不变性，为角点指定方向，解决尺度问题，算法如下：

1．对于像素块 A，定义 A 的矩为式 4.1：

$$m_{pq} = \sum_{x,y \in A} x^p y^q I(x,y), \quad p,q = \{0,1\} \quad \cdots\cdots\cdots\cdots\cdots\cdots (4.1)$$

2．将式 4.1 的结果再通过式 4.2 来计算 A 的质心：

$$Q = (\frac{m_{10}}{m_{00}}, \frac{m_{01}}{m_{00}}) \quad \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots (4.2)$$

3．根据图 4-3 灰度质心法的原理，连接像素块 A 的中心点 P 和质心 Q 形成向量 $\overrightarrow{PQ}$,该向量的方向即为特征点的方向，表示如式 4.3：

$$\theta = \arctan\left(\frac{\frac{m_{10}}{m_{00}}}{\frac{m_{01}}{m_{00}}}\right) = \arctan\left(\frac{m_{10}}{m_{01}}\right) \quad \cdots\cdots\cdots\cdots\cdots\cdots (4.3)$$

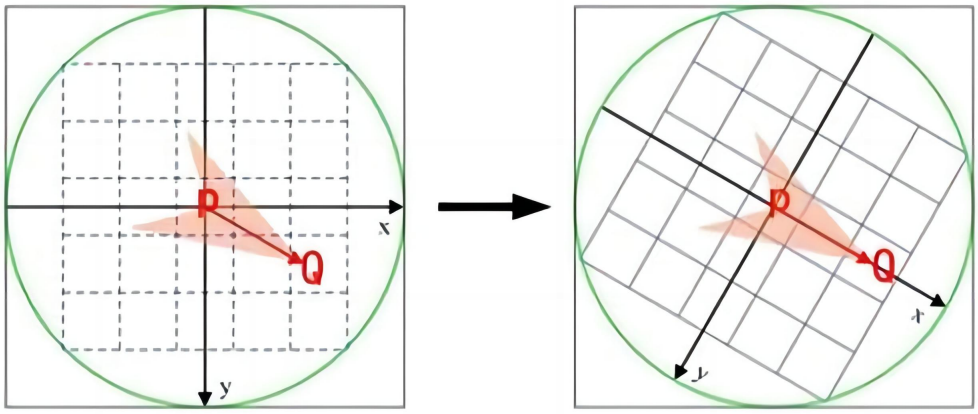


图 4-3 灰度质心法

而尺度不变性可通过图像金字塔法来解决。金字塔包含不同分辨率的图像，因此将不同分辨率的图像进行特征提取，然后可在金字塔的不同分辨率层面进行特征匹配，从而实现了尺度不变性，如图 4-4 所示。

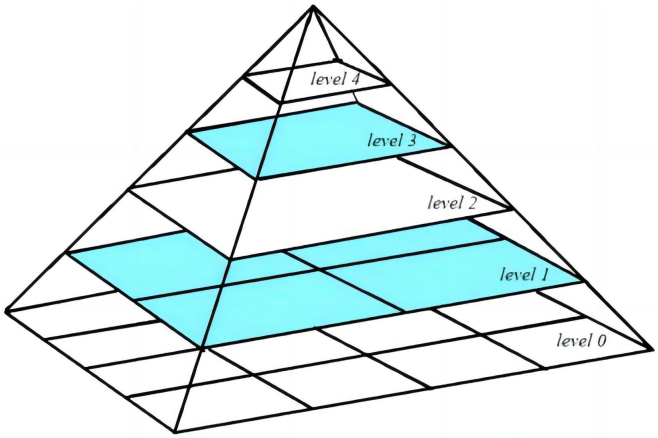


图 4-4 图像金字塔

4.3.2 描述子计算

在获取 Oriented FAST 关键点后，通过计算描述子来加速特征匹配。ORB 描述子改进自 BRIEF，利用的是 ORB 具有旋转不变性，通过计算 FAST 角点方向向量来生成 Steer BRIEF 描述子。采用的是 128 维的向量来表示周围 128 对随机点。而 128 维中的数据是以二进制的表达方式来存储，即以 0 和 1 来表示像素点的特征，在特征匹配的计算时，因为其二进制的表达方式，所以匹配的速度也非常快。

4.3.3 特征点数量分配

若要保证定位系统的视觉尺度不发生变化，则需使用到图像金字塔来处理图像，但是由于随着金字塔梯度升高，越上层的或者说分辨率越小的图片，所能够提出的特征点就越少，所以就更加需要依照平面大小去分发特征点，以此确特征点有着比较大的覆盖范围以及分布的均匀性。

假设金字塔总层数为 m，每层缩放因子为 s，需要提取的特征点数量 N，原始图像即金字塔的 0 层宽为 W，高为 H，其对应的面积 $C=H.W$，则总面积为式 4.4：

$$\begin{aligned} S &= H\cdot W\cdot (s^2)^0 + H\cdot W\cdot (s^2)^1 + \cdots + H\cdot W\cdot (s^2)^{(m-1)} \\ &= H\cdot W\frac{1-(s^2)^m}{1-s^2} = C\frac{1-(s^2)^m}{1-s^2} \end{aligned} \quad \cdots\cdots\cdots (4.4)$$

即均匀化到不同层数的特征点为式 4.5：

$$N_{\text{avg}} = \frac{N}{S} = \frac{N}{C\dfrac{1-(s^2)^m}{1-s^2}} = \frac{N(1-s^2)}{C(1-(s^2)^m)} \quad \cdots\cdots\cdots\cdots\cdots (4.5)$$

4.3.4 特征点均匀化

本节是对特征点均匀化算法进行详细说明，旨在促进特征点在不同金字塔层面的的均匀分布，以此提高系统的性能和精度，解决密集和稀疏特征点分布的问题。本文的特征点的均匀分配流程如下：

1．长宽比设为 n，长边 n 等分。若输入为 640×480，则不作处理，初始只有一个节点。

2．若单个个节点拥有大于 1 的特征点，均等 4 分；若为 0 则删除。

3．重复上述步骤，一直到无法再进行分裂或者节点的数量满足条件。

4．从每个单位分割区域中选取质量最高的作为特征点。

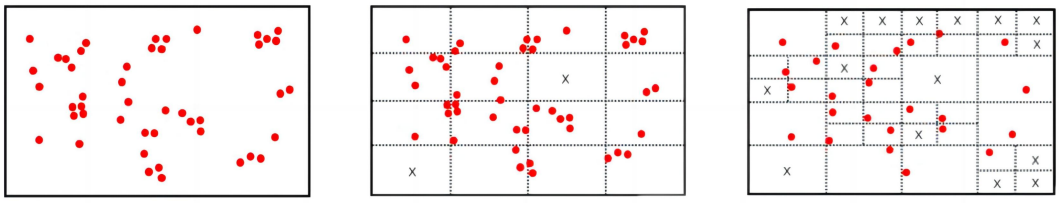


图 4-5 特征点均匀化示意图

4.3.5 动态特征点过滤策略

本文利用融合 EMA 注意力的 YOLOv8n 目标检测网络，并在 VOC 数据集上训练，一共可检测 20 种物体，在常规的室内动态场景基本适用了。

图 4-6 动态特征点剔除

如图 4-6 展示，本文只使用 YOLOV8n-EMA 运行结果中的目标检测框来剔除动态特征点。假定每一个图像上所有提取出来的特征点的集合是 $A=\{p_1,p_2,\cdots,p_3\}$，每个特性点的坐标为 $p_i=(u_i,v_i)$。目标框可以被分类为两种定义：一种把人、车、猫狗类别的目标检测框视为动态目标框 $d_m=(l_m,t_m,r_m,b_m)$，另一种则将显示器、椅子等类别的目标检测框视静态目标框 $d_s=(l_s,t_s,r_s,b_s)$。定义中，下标"m"表示的是运动物体，而下标"s"代表的是静态物体。"l"和"t"分别对应着目标框左侧和顶部水平坐标值，"r"和"b"则是目标框右侧横坐标值与底部边框的坐标值。通过设定目标框架，可以把图像上的关键点划分为两个集群，即位于行人群体内部的关键点构成一组，其余未包含于此或者属于其它实体群体的关键点构成了另一组。对于每一幅画面，都会依次检查所有的关键点并做出以下决策：若点 p_i 的条件符合 $\{(u,v)\,|\,l_m<u<r_m,t_m<v<b_m\}$，则 $p_i\in B$，执行操作，移除此关键点;反之，则保持不动以用于位姿推测。

4.3.6 特征匹配算法

特征匹配是关乎整个视觉里程计的鲁棒性的关键性步骤，主要是用来解决解决一对图像帧之间的特征点的数据关联。特征匹配是通过提取 ORB 特征点时已有描述子来进行特征点之间的快速匹配，匹配的准确性会直接影响到位姿的估计以及后端的优化，所以好的特征匹配方法可以让整个视觉定位系统性能提升。

视觉定位系统提取好特征点后，对于相邻的两张图像，有很多特征点无法一一对应，这会导致无法将相同位置的特征点进行后续的位姿估计，所以最关键的问题是对于两张图上的特征点如何进行匹配。

对此，我们可以使用循环遍历的方法，也就是暴力匹配，对每个特征点都进行距离的运算，选择距离最小的，匹配结果如图 4-8，算法步骤如下：

1. 通过算法来提取图像的特征点。

2. 计算特征点描述子的距离，通常使用汉明距离的方法来计算两个特征点之间的距离，通过比较选出汉明距离最小的特征点对。

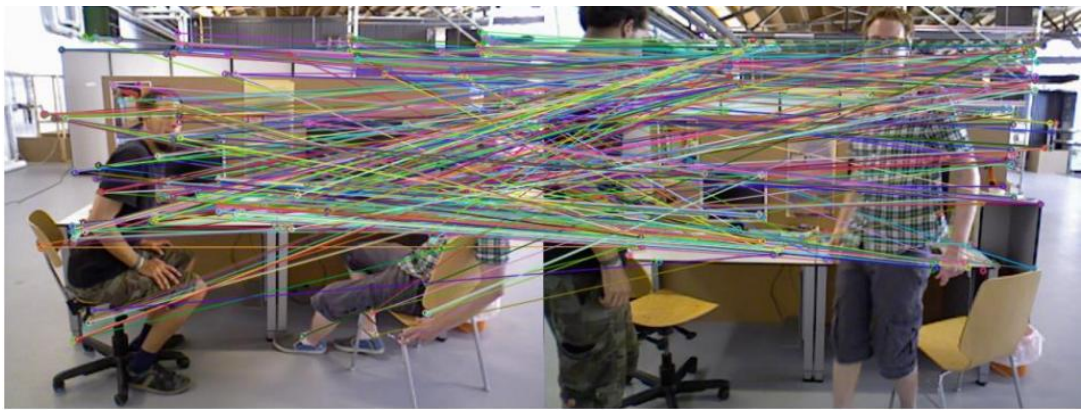


图 4-7 暴力匹配效果图

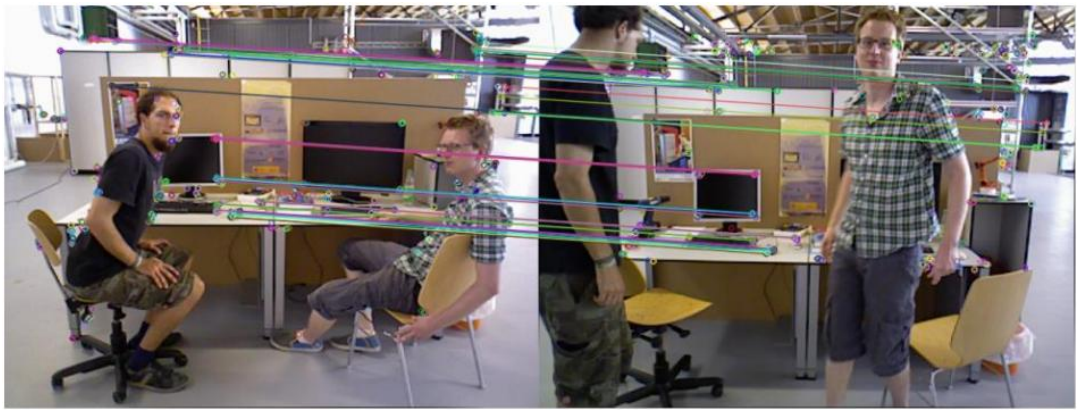


图 4-8 BRIEF 描述子匹配效果图

图 4-7 使用暴力匹配取比对两张图片的特征点，这会导致匹配计算十分耗时且误匹配率也大大提高，所以在视觉定位系统需频繁匹配特征的情况下，应采用 BRIEF 描述子的方法，不仅加速了系统初始化，也加速了后续的特征匹配，相邻帧通过此方法代入位姿估计计算，而后端的回环检测也使用了 BRIEF 描述子进行是否回环的判断。

4.3.7 RANSAC 算法

RANSAC 算法又称随机采样一致性算法，其功能原理是在特征匹配的过程中通过对模型的不断更新取过滤掉产生错误匹配、重复、多余的特征点对，从而将剩下的有用的特征点对在送入后续的位姿计算。其流程如图 4-9：

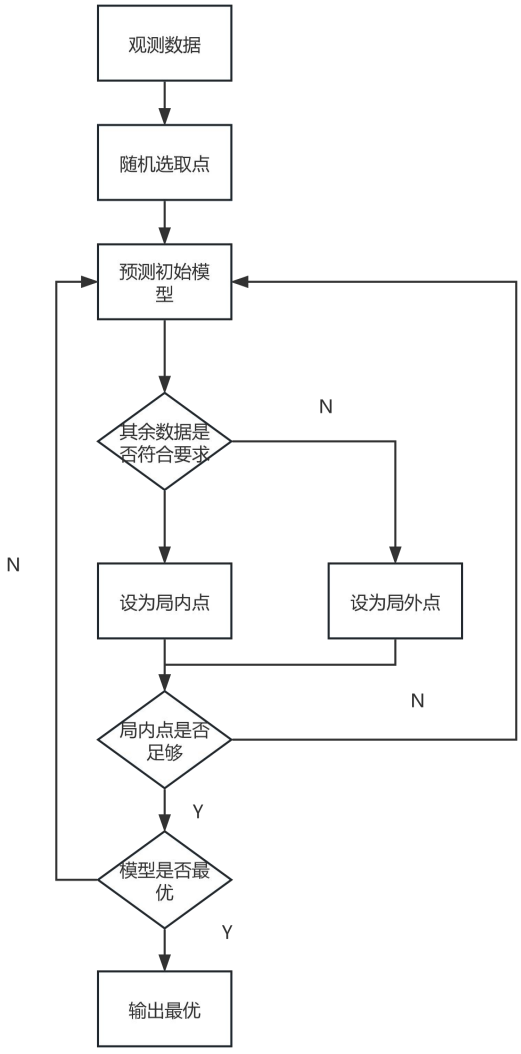


图 4-9 RANSAC 流程图

首先删除特征匹配的误匹配点，继续观测两张图像 I_1 与 I_2 剩余的数据点，并求出两张图像的空间变换矩阵 $H_{3\times3}$，矩阵 H 也叫做单应矩阵，公式推导如式 4.6，步骤如下：

$$\begin{pmatrix} u_2 \\ v_2 \\ 1 \end{pmatrix} = \begin{pmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{pmatrix} \begin{pmatrix} u_1 \\ v_1 \\ 1 \end{pmatrix} \quad \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots (4.6)$$

1. 通过在匹配点数据中随机选取四个，对单应矩阵 H 进行求解。

2. 求解后的单应矩阵 H 中的参数带入到重投影误差公式中，对所有的匹配点再进行一次计算，得出来的值如果小于设定阈值，则将该匹配点加入到集合 p 中，重投影误差公式如下：

$$\sum_{i=1}^{n}((x_{i'} - \frac{h_1x_i + h_2y_i + h_3}{h_7x_i + h_8y_i + h_9})^2 + (y_{i'} - \frac{h_4x_i + h_5y_i + h_6}{h_7x_i + h_8y_i + h_9})^2) \quad \cdots\cdots\cdots (4.7)$$

3.如果 p 中的匹配点数量不够，则会再次重新随机选择四个点进行步骤 1，直到 p 中的数量满足阈值或者迭代次数达到 N 次，才会停止迭代模型。

4.更新结束。

4.4 位姿估计与优化

4.4.1 位姿计算

求解相机的位姿变换可以转换成一个最小化重投影误差问题，首先已知两个匹配图 I_1 与 I_2 图像上的匹配像素点 P_1 P_2，通过过相机的位姿可以求的该像素点在空间位置的坐标为 Q，再通过投影得到 I_2 上 Q 点的投影点 P_3，最终得到 P_2 P_3 之间的误差 e，并对其不断地迭代优化。如图 4-10：

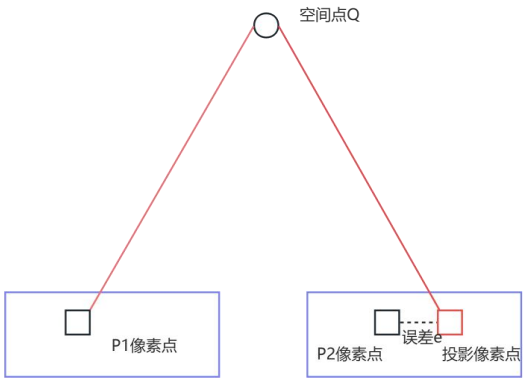


图 4-10　重投影误差图

在测量相机的姿态时，可以使用静止特征点来推算机器人位姿。如果设定满足条件的特征点 $p_i=(u_i,v_i)$ 的空间点 $P_i=(X_i,Y_i,Z_i)$，那么这些特征点与空间点之间的关系就是：

$$s_i\begin{bmatrix}u_i\\v_i\\1\end{bmatrix}=K\exp\left(\xi_i^{\wedge}\right)\begin{bmatrix}X_i\\Y_i\\Z_i\\1\end{bmatrix}+e \quad \cdots\cdots\cdots\cdots\cdots\cdots (4.8)$$

在这里，s 是尺度因子，K 是相机内参矩阵，ξ 是位姿的李代数，$^{\wedge}$ 是将李代数转化为矩阵的计算符，还有就是 e 误差。

将式(4.8)改写为矩阵形式：

$$s_ip_i=K\exp\left(\xi_i^{\wedge}\right)P_i+e_i \quad \cdots\cdots\cdots\cdots\cdots\cdots (4.9)$$

将所有的误差加起来求和，构建出最小二乘问题。

$$\xi^*=\arg\min\frac{1}{2}\sum_{i=1}^{n}\left\|p_i-\frac{1}{s_i}K\exp\left(\xi_i^{\wedge}\right)P_i\right\|_2^2 \quad \cdots\cdots\cdots\cdots (4.10)$$

因为 $p\in C$，所以式(4.10)中的 P_i 均为固定值，即 ξ^* 为所求的相机位姿。

4.4.2 BA 优化

BA 优化(Bundle Adjustment)通过不停迭代计算相机位姿和特征点应该存在的空间位置的最小化误差，从而不断提升相机位姿的精度的一种方法，在 SLAM 系统中，全局 BA 会在最后优化所有的关键帧，进一步提高系统的一致性和准确性，而局部 BA 优化只优化当前帧以及和当前帧共视的关键帧。步骤如下：

1．将三维点 P 通过相机的平移矩阵 t 与旋转矩阵 R，转换至相机坐标系 P'

$$P'=RP+t=[X',Y',Z']^{\mathrm{T}} \quad \cdots\cdots\cdots\cdots\cdots\cdots (4.11)$$

2．对 P' 使用归一化操作：

$$P_c=[u_c,v_c,1]^{\mathrm{T}}=[X'/Z',Y'/Z',1]^{\mathrm{T}} \quad \cdots\cdots\cdots\cdots (4.12)$$

3．由于图像畸变会产生误差，最后得到去除畸变后的像素坐标：

$$\begin{cases}u_{c'}=u_c(1+k_1r_c^{\ 2}+k_2r_c^{\ 4})\\v_c^{\ '}=v_c(1+k_1r_c^{\ 2}+k_2r_c^{\ 4})\end{cases} \quad \cdots\cdots\cdots\cdots\cdots\cdots (4.13)$$

4．根据相机的内参参数，得到坐标：

$$\begin{cases} u_s = f_x u_{c'} + c_x \\ v_s = f_y u_{c'} + c_y \end{cases} \quad \cdots\cdots (4.14)$$

5．式 4.11 到 4.14 完成了将世界坐标系转换为像素坐标系的过程，可以用以下抽象形式来描述：

$$z = h(x, y) \quad \cdots\cdots (4.15)$$

式 4.53 针对观测方程，z 表示当前的观测值，x 表示当前相机的姿态，同样，包括平移和旋转，这次观测的误差可以用以下方式来表示：

$$e = z - h(T, P) \quad \cdots\cdots (4.16)$$

考虑到其他时刻观测的误差，那么整体的代价函数如下：

$$\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{n}\left\|e_{ij}\right\|^2 = \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{n}\left\|z_{ij} - h(T_i, P_j)\right\|^2 \quad \cdots\cdots (4.17)$$

式 4.17 中 $\boldsymbol{z}_{ij}$ 表示位姿为 $\boldsymbol{T}_i$ 时观测路标 $\boldsymbol{P}_j$ 产生的数据，给所有自变量一个增量 $\Delta\mathbf{x}$，目标函数转化为：

$$\frac{1}{2}\left\|f(x+\Delta x)\right\|^2 \approx \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{n}\left\|e_{ij} + F_{ij}\Delta\xi_i + E_{ij}\Delta P_j\right\|^2 \quad \cdots\cdots (4.18)$$

式 4.16 中 $\boldsymbol{\xi}$ 为相机位姿的李代数，分别将相机位姿变量 $[\xi_1, \xi_2, \cdots, \xi_m]^{\mathrm{T}} \in \mathbb{R}^{6n}$ 记为 $\boldsymbol{x}_c$ 和空间点变量 $[P_1, P_2, \cdots, P_m]^{\mathrm{T}} \in \mathbb{R}^{3n}$ 记为 x_p，式子如下：

$$\frac{1}{2}\left\|f(x+\Delta x)\right\|^2 = \frac{1}{2}\left\|e + F\Delta x_c + E\Delta x_p\right\|^2 \quad \cdots\cdots (4.19)$$

E 和 F 分为是本质矩阵与基础矩阵，是一个由每个误差项 F_{ij} 和 E_{ij} 组成的矩阵。对于式 4.19 ，我们可以采用高斯-牛顿法进行求解，最终要解决增量线性方程:

$$H\Delta x = g \quad \cdots\cdots (4.20)$$

最后定位系统系统中的 H 矩阵可采用高效方法求解，降低计算成本，提高系统运行效率。

$$\mathbf{H} = \mathbf{J}^{\mathrm{T}}\mathbf{J} = \begin{bmatrix} \mathbf{F}^{\mathrm{T}}\mathbf{F} & \mathbf{F}^{\mathrm{T}}\mathbf{E} \\ \mathbf{E}^{\mathrm{T}}\mathbf{F} & \mathbf{E}^{\mathrm{T}}\mathbf{E} \end{bmatrix} \quad \cdots\cdots (4.21)$$

4.4.3 回环检测

系统使用回环检测判断相机是否回到之前经过的地方，若检测到回环，则会利用回环修正来消除视觉定位形成的累积误差，从而提升定位精度，回环检测的判断一般是利用 BoW2 词袋模型来提升回环检测的匹配率的。

4.4.4 词袋模型

视觉定位十分依赖于图像特征点的匹配，同样的，词袋模型也是依赖于两个“单词”之间的匹配，通常将图像中人、电脑、猫等视为词袋模型的“单词”。词袋模型是使用低级语义信息计算图像特征的相似性来缩短匹配时间的。

图像的“单词”类似于汉字，需要字典来查找，然后构成图像的词袋。最后形成的字典是通过离线的训练而生成的，可以把这个过程看作一个类似聚类的问题，采用算法则是 K-means++，同样，为加速词袋查找，字典的数据存储结构为树形结构的，使用 K 叉树字典生成算法：

1. 在根节点上，使用聚类算法把样本分成一个个类。

2. 通过第一层聚类获得的节点，再执行同样操作，使得每个节点再次聚成 k 类。

3．最后依次聚类到叶子节点层，即叶子节点的内容为单词，形成树形字典。

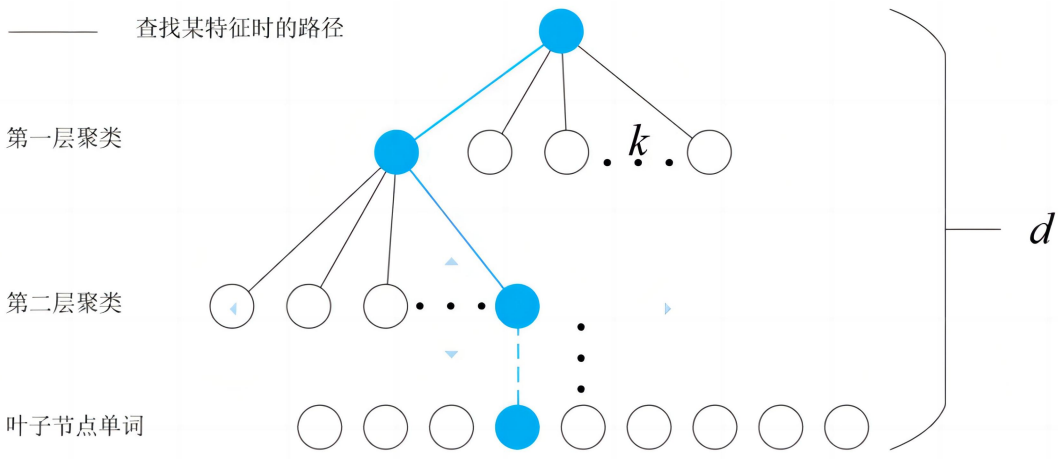


图 4-11　K 叉树字典示意图

视觉定位系统在回环检测的过程中，借助字典的方法就可以快速检查两帧图像的相似度，以此来确认是否有回环产生。

4.5 实验结果与分析

4.5.1 实验环境

本文使用计算机配置如表 4-1：

表 4-1 硬件配置

硬件配置	
操作系统	Ubuntu18.04
CPU	Intel i5 12400
内存	8G
显卡	NVIDIA RTX 2080ti

本文使用软件版本配置如表 4-2：

表 4-2 软件配置

软件配置	
图像处理	OpenCV4.2
矩阵计算	Eigen3
可视化操作	Pangolin
BA 优化	g2o
词袋模型	Bow
点云处理	PCL
精度评价	Evo
CUDA Care	CUDA 11.7/cudnn8.0.4
深度学习框架	pytorch

4.5.2 移动机器人数据集

本文所使用的数据主要来自于 TUM 数据集，数据主要是以移动机器人上装有的微软的 Kinect 相机进行采集，为定位系统提供了标定好的内参，也包含了多种场景（室内外、户外等），可以用于测试、动态物体、三维重建等不同的任务。它能够提供相机的真实位置值，从而协助研究人员评价视觉定位系统的表现。本文使用该数据集的 4 种高度动态数据集(fr3_w_halfsphere, fr3_w_static, fr3_w_xyz，fr3_w_rpy)、4 种低动态数据集(fr3_s_halfsphere, fr3_s_static, fr3_s_xyz，fr3_s_rpy)以及 3 种静态数据集(fr1_desk,fr1_room,fr1_rpy)来进行实验。

数据集文件内容如图 4-12 所示，RGB 图和深度图可以通过时间戳对其来同步使用，方法是借助 accelerometer.txt 来对齐名称与时间戳。groundtruth.txt 为则是该数据集的真实位姿，在视觉系统生成位姿后，使用 EVO 等工具来与真实位姿轨迹做对比，以此来评估系统的性能。

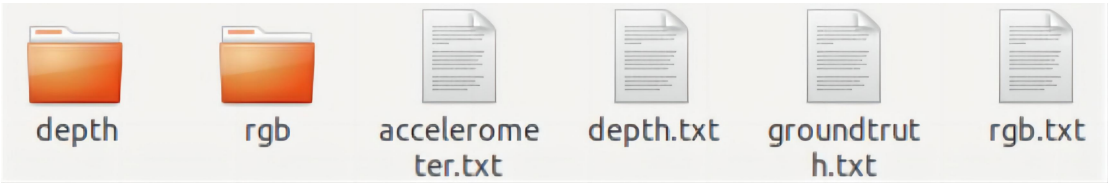


图 4-12　TUM 数据集格式

4.5.3 评价指标

在本研究里采用的是绝对轨迹误差 ATE(Absolute Track Errors)为评估标准来衡量相机实际位置状态同视觉定位系统预测的位置状况间的差异程度。

通过时间戳，将视觉定位系统产生的轨迹与数据集真实轨迹直接通过最小二乘问题求解变换矩阵 $S \in SE(3)$，定义第 i 帧的 ATE(Absolute Track Errors）如下：

$$F_i := Q_i^{-1} S P_i \quad \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots (4.22)$$

使用均方根误差（RMSE）反应 ATE，$trans(\boldsymbol{F}_i)$代表平移部分:

$$\mathrm{RMSE}(\mathbf{F}_{1:n}) := \left(\frac{1}{n}\sum_{i=1}^{n} \| \, trans(\mathbf{F}_i) \|^2\right)^{\frac{1}{2}} \quad \cdots\cdots\cdots\cdots\cdots (4.23)$$

使用 RMSE 的平均值 Mean、中位数 Median、标准差 S.D 等作为额外的性能检测指标。

4.5.4 动态特征点滤除实验

观察图 4-13 (a)可知，一些特征点分布在人身上，这些特征点可能会干扰到机器人的定位精度，从而降低其准确性。而通过对比图 4-13 (b)的显示结果，发现所有被移除的特征点均集中于静止对象上，这样能够显著减少运动目标对于机器人姿态估算产生的负面影响。

(a)剔除前　　(b)剔除后

图 4-13 动态特征点剔除

4.5.5 视觉定位实验

在本研究里采用的是绝对轨迹误差 ATE 为评估标准来衡量移动的相机实际位置状态同定位系统预测的位置状况间的差异程度 ，其中提升率的计算方法为:

$$A_{improvement}=\frac{A_{origin}-A_{our}}{A_{origin}}\times 100\% \quad \cdots\cdots\cdots\cdots\cdots\cdots \quad (4.24)$$

式(4.24)中的 $A_{improvement}$、A_{origin}、A_{our} 依次代表在 ATE 下的提升率、原始值、本文算法得出的值。

本研究使用上面提到的 TUM 中 11 个不同数据集，并且在具有明显差异的场景下使用融合了 YOLOv8n-EMA 的视觉定位系统进行实验。为保证实验严谨性，分别进行多次实验并取平均值,且表格中 A_{RMSE}、A_{MEAN} 以及 A_{STD} 依次代表在 ATE 下的均方根误差、平均值以及标准差，两者的绝对轨迹误差（ATE）对比如表 4-3 所示。

表 4-3 显示了视觉定位系统在不同环境下，ORB-SLAM3 和本文改进的系统的绝对轨迹误差。通过表 4-3 可以直观地看出，本文改进的算法在搞动态环境下对原系统的提升率很大，本文改进的系统位姿定位提升达到了 91.56%，最大值为 98.1%，但随着环境逐渐从高动态变为静态时，本文改进的系统的优势就会越来越小。因此得出结论：本文所改进的系统是针对高动态环境的十分有效的系统，在低动态环境性能差异减小，是由于 ORB-SLAM3 中 BA 优化所导致的。

表 4-3 绝对轨迹误差对比

数据集		ORB_SLAM3/m			改进算法/m			提升率/%		
		A_{RMSE}	A_{MEAN}	A_{STD}	A_{RMSE}	A_{MEAN}	A_{STD}	A_{RMSE}	A_{MEAN}	A_{STD}
高动态环境	fr3_w_xyz	0.677	0.586	0.340	0.016	0.014	0.007	**97.6**	**97.6**	**97.9**
	fr3_w_rpy	0.767	0.719	0.283	0.219	0.194	0.102	**71.4**	**73.0**	**63.9**
	fr3_w_half	0.395	0.359	0.346	0.026	0.022	0.013	**93.4**	**93.8**	**96.2**
	fr3_w_static	0.354	0.148	0.165	0.007	0.006	0.003	**98.0**	**95.9**	**98.1**
低动态环境	fr3_s_xyz	0.017	0.009	0.007	0.010	0.007	0.004	**58.8**	**22.2**	**42.8**
	fr3_s_rpy	0.041	0.033	0.025	0.027	0.021	0.017	**31.7**	**33.3**	**32.0**
	fr3_s_half	0.115	0.109	0.046	0.060	0.040	0.040	**47.8**	**63.3**	**13.0**
	fr3_s_static	0.011	0.010	0.005	0.008	0.007	0.004	**27.2**	**30.0**	**20.0**
静态环境	fr1_desk	0.018	0.015	0.010	0.016	0.014	0.009	**11.1**	**6.6**	**10.0**
	fr1_room	0.078	0.065	0.042	0.080	0.053	0.045	-2.5	**18.4**	**7.1**
	fr1_rpy	0.024	0.019	0.014	0.023	0.018	0.013	**4.1**	**5.2**	**7.1**

通过 EVO 工具可绘制定位系统的轨迹与真实轨迹的对比图，黑色虚线为真是轨迹，蓝色实线为定位轨迹。表 4-4 展示了 ORB-SLAM3 和本文改进系统在高动态环境中的轨迹与真实轨迹的对比情况。可以看出，本文系统在高动态环境下精度有了大幅度提升。

表 4-5 与表 4-6 分别是高动态环境下轨迹误差的分布对比图与高动态环境下的旋转角对比。可见的本文改进的算法相对于原系统更加贴合真值曲线，定位精度更高。

表 4-4 高动态环境轨迹图展示

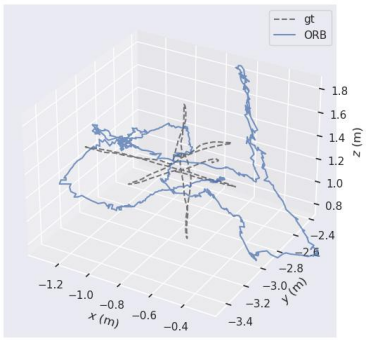
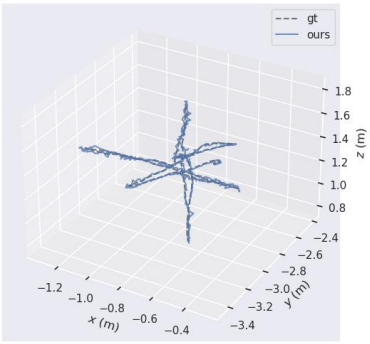
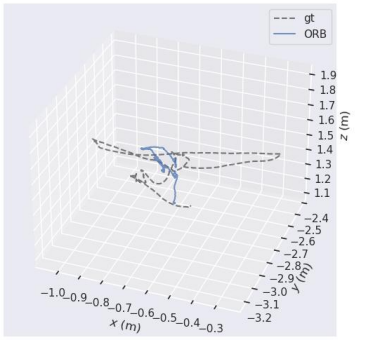
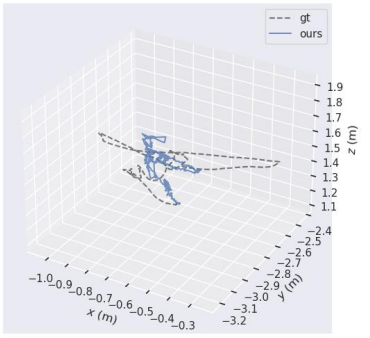
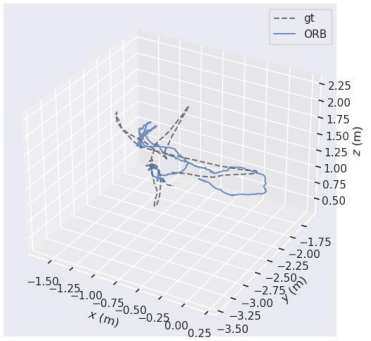
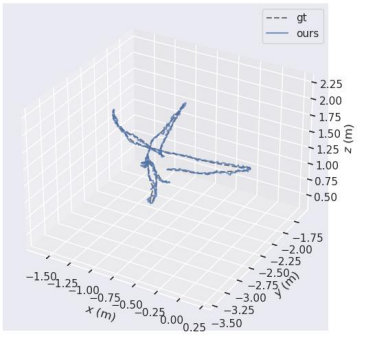
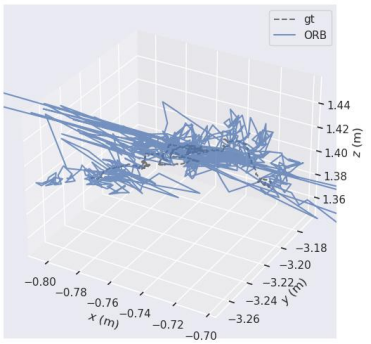
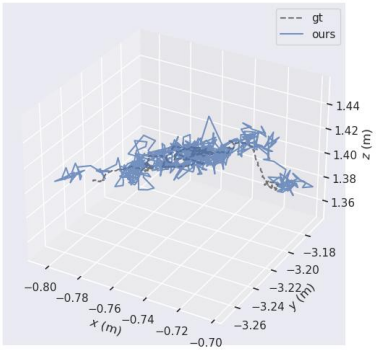
高动态环境	ORB_SLAM3	改进算法
fr3_w_xyz		
fr3_w_rpy		
fr3_w_half		
fr3_w_static		

表 4-5 高动态环境轨迹误差分布对比

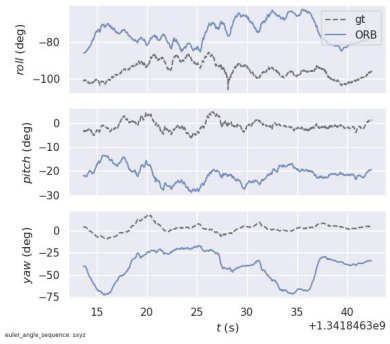
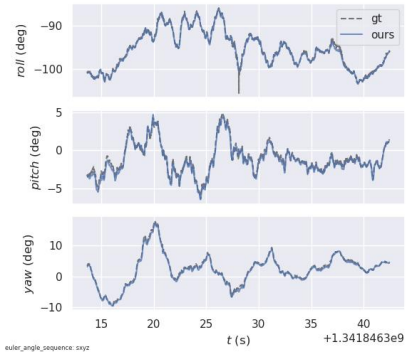
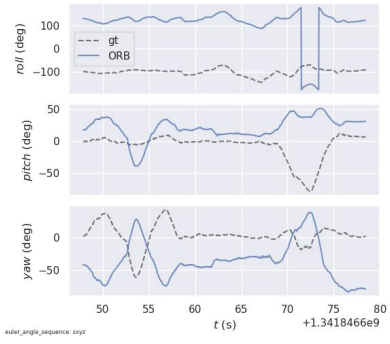
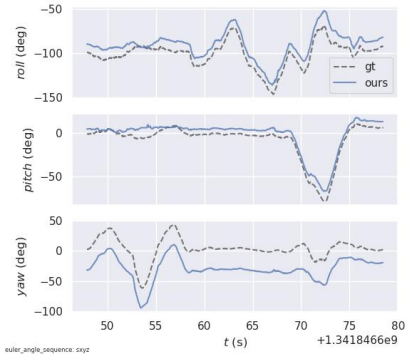
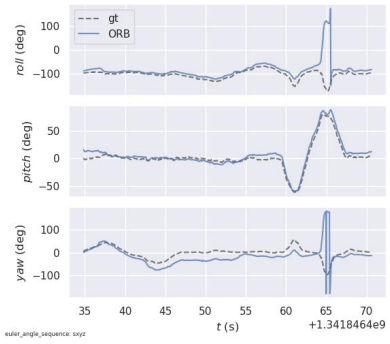
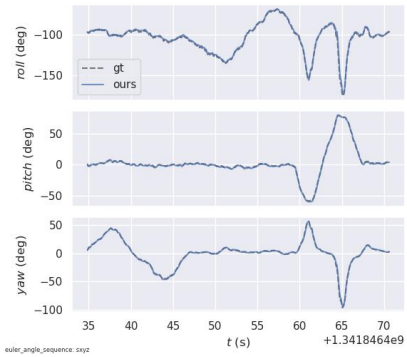
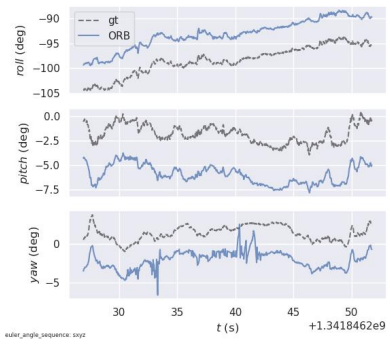
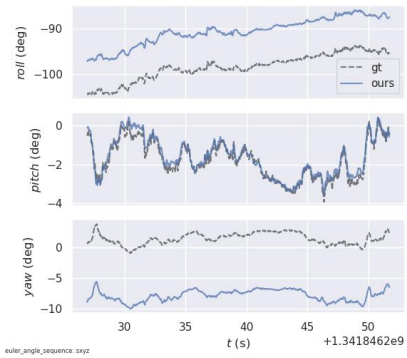
高动态环境	ORB_SLAM3	改进算法
fr3_w_xyz		
fr3_w_rpy		
fr3_w_half		
fr3_w_static		

表 4-6 高动态环境旋转角对比

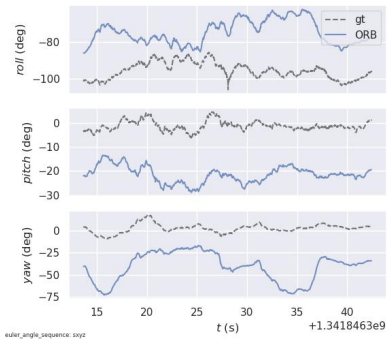
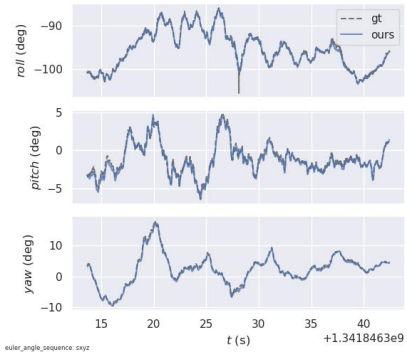
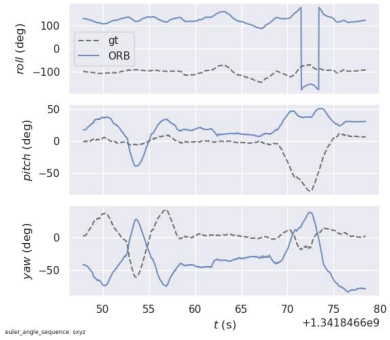
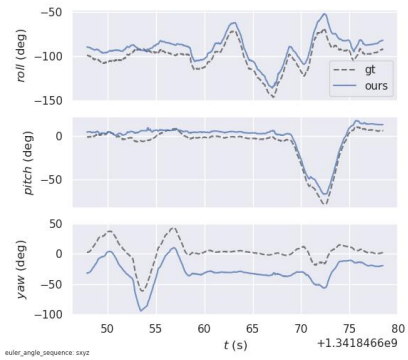
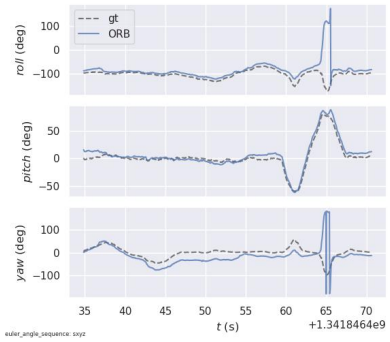
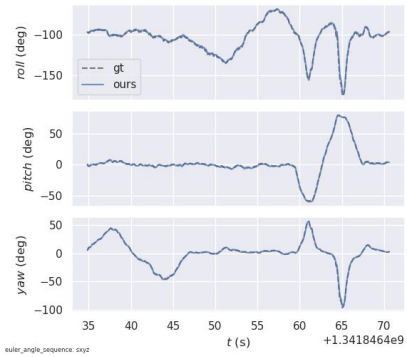
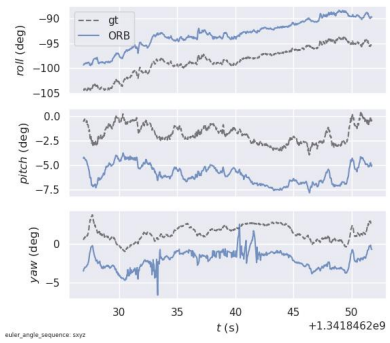
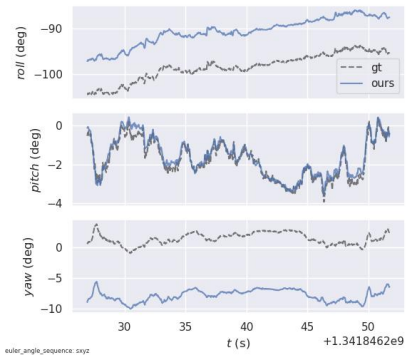
高动态环境	ORB_SLAM3	改进算法
fr3_w_xyz		
fr3_w_rpy		
fr3_w_half		
fr3_w_static		

表 4-7 低动态环境下绝对轨迹误差对比

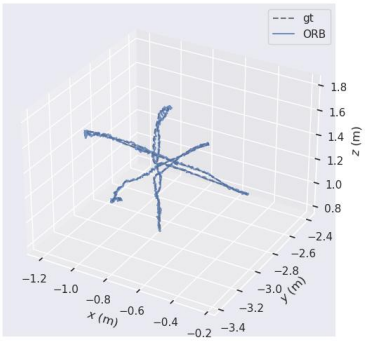
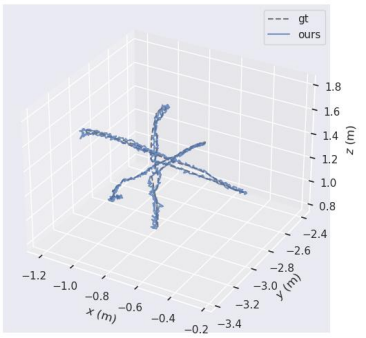
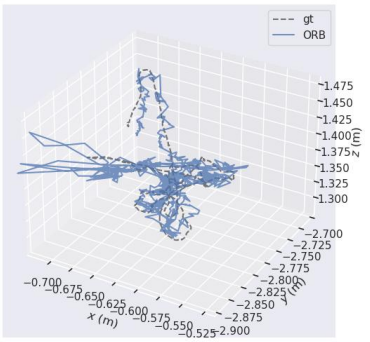
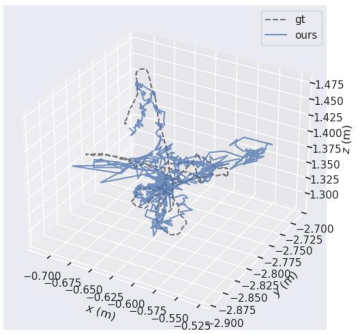
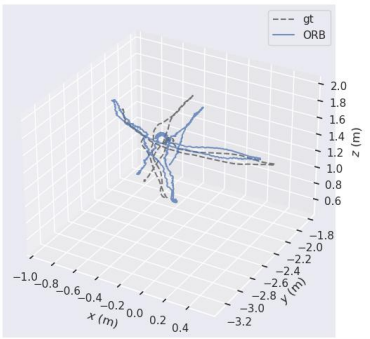
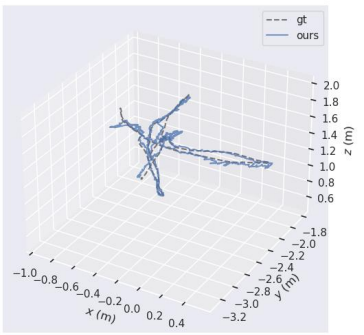
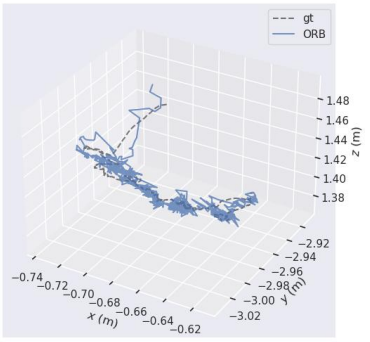
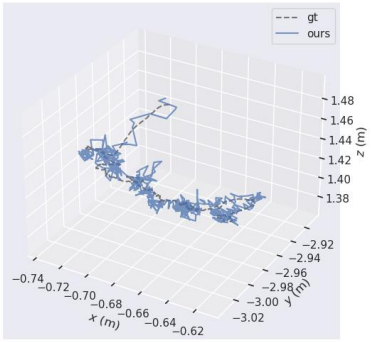
低动态环境	ORB_SLAM3	改进算法
fr3_s_xyz		
fr3_s_rpy		
fr3_s_half		
fr3_s_static		

表 4-7 展示了 ORB-SLAM3 和本文改进系统在低动态环境中的轨迹与真实轨迹的对比情况，可以看出，低动态环境下，本文改进的视觉定位系统略优于 ORB-SLAM3 系统，但差距没有在高动态环境下明显，说明在环境动态物体较少的情况下，动态物体对视觉定位的影响也会随之减小。

本文只展示了高动态与低动态环境的轨迹图，因从表 4-3 的绝对轨迹误差可知，静态环境下 ORB-SLAM3 和本文改进系统在静态环境中的误差几乎相同，所以两者的轨迹图也近乎相同，由此可以得出，两者其实在静态环境下都有很好的精度，且精度差距越来越小。

4.5.6 消融实验

本节为研究 YOLOv8n 视觉定位在动态环境下添加与不添加 EMA 注意力机制之间的区别，特做此消融实验，该实验则在高动态环境进行的，该实验提升率的对照则是 ORB-SLAM3 在该环境下所实验的得出的绝对轨迹误差。实验对比 YOLOv8n 视觉定位与改进了 EMA 注意力机制的视觉定位在动态环境下相对于 ORB-SLAM3 的提升率。

表 4-8 YOLOv8n 视觉定位精度

数据集		YOLOv8n-定位/m	
		A_{RMSE}	A_{STD}
高动态环境	fr3_w_xyz	0.021	0.009
	fr3_w_rpy	0.231	0.110
	fr3_w_half	0.027	0.014
	fr3_w_static	0.006	0.004

表 4-8 为不添加 EMA 注意力机制的视觉定位系统在高动态环境下的绝对轨迹误差，可知使用先验目标框的视觉定位系统在高动态环境下的效果都是十分理想的。

表 4-9 添加了 EMA 注意力定位提升量

数据集		Yolov8n 提升率		本文算法提升率		提升量	
		A_{RMSE}	A_{STD}	A_{RMSE}	A_{STD}	A_{RMSE}	A_{STD}
高动态环境	fr3_w_xyz	92.3%	92.9%	**97.6%**	**97.9%**	**5.3%**	**5.0%**
	fr3_w_rpy	66.2%	59.9%	**71.4%**	**63.9%**	**5.2%**	**4.0%**
	fr3_w_half	90.4%	91.2%	**93.4%**	**96.2%**	**3.0%**	**5.0%**
	fr3_w_static	**98.6%**	**98.6%**	98.0%	98.1%	-0.6%	-0.5%

表 4-9 消融实验数据展现了 YOLOv8n 视觉定位系统融合 EMA 注意力机制比不添加 EMA 注意力机制在高动态环境提升约 5%的定位精度，但在低动态环境提升率减少，而在静态环境两者性能几乎不变，由此可推断出，融合 EMA 注意力机制在更复杂的动态环境会有更好的效果。

4.5.7 与其他系统性能对比实验

表 4-10 则展示了本文的方法与其他同类算法在 TUM 高动态场景数据集的实验实验结果对比，在尾缀为 xyz，与尾缀为 half 两个数据集中，本文算法表现更优，其根本原因在于在数据集训练的时候很少使用旋转后的图片对模型进行训练，导致 YOLOv8 在摄像头旋转比较大的场景里检测效果较差。

表 4-10 同类算法精度对比

数据集	DS-SLAM/m	DynaSLAM/m	RDS-SLAM/m	本文算法
	A_{RMSE}	A_{RMSE}	A_{RMSE}	A_{RMSE}
Walking_xyz	0.0247	0.0164	0.0565	**0.0161**
Walking_rpy	0.4442	0.0354	**0.0128**	0.219
Walking_half	0.0303	0.0296	0.0291	**0.0261**
Walking_static	0.0081	0.0068	**0.0039**	0.0072

表 4-10 同类算法耗时对比

方法	GPU	网络模型	模型推理时间/ms	跟踪每帧的时间/ms
ORB-SLAM3	GeForce RTX 2080Ti	-	-	21.3
DS-SLAM	P4000	SegNet	37.57	59.40
DynaSLAM	Tesla M40 GPU	Mask R-CNN	195.0+0	>300
RDS-SLAM	GeForce RTX 2080Ti	Mask R-CNN	200.00	50~65
本文方法	GeForce RTX 2080Ti	YOLOv8n	**22.1**	**42.1**
本文方法	GeForce RTX 2080Ti	YOLOv8n-EMA	23.12	43.1

表4-10给出了在TUM数据集测试下最终本文方法和同类方法的每帧模型推理和跟踪时间，本文方法采用了两组模型，并在 RTX 2080Ti 显卡下进行测试，第一组 YOLO 模型不融合 EMA 注意力机制，第二组 YOLO 模型融合 EMA 注意力机制，结果，第一组改进后的系统每帧跟踪时间最短，仅为 42.1ms，第二组系统每帧跟踪时间为 43.1ms，并且两组模型的推理时间也是同类系统中最优的，试验结果表明：在满足实时性的要求下，本文采用 YOLOv8 融合 EMA 注意力机制的视觉定位系统相比于同类动态视觉定位系统在速度上有明显提高，由于添加了 EMA 注意力机制模块，所以相比于原生 YOLOV8 模型，推理时间略微增加，导致跟踪时间延长。

4.6 本章小结

本章中主要介绍了一种基于 YOLOV8n-EMA 的视觉定位系统，该系统可以过滤动态物体，提高视觉定位系统的精度。

该系统首先使用YOLOV8融了EMA注意力机制的目标检测网络检测图像中的运动物体，然后将检测框内的特征点标记为动态特征点，并将其从后续的匹配和位姿估计中去除。为了提高特征点的均匀分布，该系统还使用了特征点均匀化算法，将图像划分为多个区域，并从每个区域使用算法提取特征点。在特征匹配

阶段，该系统使用了暴力匹配算法，对两幅图像之间的所有特征点进行匹配，并根据距离筛选出最优匹配点。在位姿计算阶段，该系统使用了最小二乘法估计相机的位姿，并通过 BA 优化进一步优化位姿和路标点的精度，最后使用了词袋模型来提升回环修正的效率。

最后使用 ATE 作为定位精度的评价指标来对改进的系统进行评估，并绘制了轨迹，相对于 ORB-SLAM3 系统，本文改进系统在高动态环境下的的 ATE 的平均提升率为 98%，在低动态环境下的平均提升率为 25%，在静态环境下的平均提升率为 8%。而对于同类系统，本文改进系统在模型推理时间与跟踪时间上达到了 23ms 与 43ms，优于其他同类系统。

第 5 章 总结与展望

5.1 工作总结

视觉定位系统是智能移动机器人的核心技术，该目前市面上开源系统在面对有大量动态物体干扰的环境时，往往会导致定位失效的结果。本文研究为解决传统的视觉定位系统不能应对高动态环境的难题，做了以下两点工作：

1.提出了一种新的目标检测模型 YOLOv8n-EMA，该模型在 YOLOV8 的基础上融合了 EMA 注意力机制。EMA 注意力机制通过并行处理和分组特征图提取。且通过大量实验结果，可以证明 YOLOv8n-EMA 网络比 YOLOv8n 网络拥有更好的检测能力，mAP 平均提升量为 15%，并且为下一章的动态环境下的视觉定位提供较为精确的动态物体的先验信息。

2.提了出一种融合 YOLOv8-EMA 的视觉定位系统，然后将 YOLOv8-EMA 提供的检测框内的特征点标记为动态特征点，并将其从后续的匹配和位姿估计中去除。为了提高特征点的均匀分布，本文系统还使用了特征点均匀化算法，将图像划分为多个区域，并从每个区域使用算法提取特征点。在特征匹配阶段，该系统使用了暴力匹配算法，对两幅图像之间的所有特征点进行匹配，并根据距离筛选出最优匹配点。在位姿计算阶段，该系统使用了最小二乘法估计相机的位姿，并通过 BA 优化进一步优化位姿和路标点的精度，最后使用了词袋模型来提升回环修正的效率，并通过回环矫正消除累计误差，提高视觉定位的精度。相对于 ORB-SLAM3 系统，本文改进系统在高动态环境下的的 ATE 的平均提升率为 98%，在低动态环境下的平均提升率为 25%，在静态环境下的平均提升率为 8%。而对于同类系统，本文改进系统在模型推理时间与跟踪时间上达到了 23ms 与 43ms，优于其他系统。

5.2 进一步工作和展望

当面对有摄像机转动的动态环境时，视觉定位系统的稳定性能会有所下降。为使得系统更加鲁棒，未来将通过增加大量经过旋转处理的数据图像到神经网络的训练中，用来提高该网络模型在不同动态环境中的准确率，同时在视觉几何上添加极线几何约束来进一步排除动态物体影响。在实时性方面，未来将继续轻量

化 YOLOv8n-EMA 网络来提高系统的速度。

参考文献

[1] Zhang Y, Wang Q, Shen Y, et al. An online path planning algorithm for autonomous marine geomorphological surveys based on AUV[J]. Engineering Applications of Artificial Intelligence, 2023, 118: 105548.

[2] da Fonseca I M. Space robotics and associated space applications[C]//Vibration Engineering and Technology of Machinery: Proceedings of VETOMAC XV 2019. Springer International Publishing, 2021: 151-170.

[3] Huang S, Huang H Z, Zeng Q, et al. A Robust 2D Lidar SLAM Method in Complex Environment[J]. Photonic Sensors, 2022, 12(4): 220416.

[4] Kazerouni I A, Fitzgerald L, Dooly G, et al. A survey of state-of-the-art on visual SLAM[J]. Expert Systems with Applications, 2022, 205: 117734.

[5] Campos C, Elvira R, Rodríguez J J G, et al. Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam[J]. IEEE Transactions on Robotics, 2021, 37(6): 1874-1890.

[6] Engel J, Schöps T, Cremers D. LSD-SLAM: Large-scale direct monocular SLAM[C]//European conference on computer vision. Cham: Springer International Publishing, 2014: 834-849.

[7] Smith R C, Cheeseman P. On the representation and estimation of spatial uncertainty[J]. The international journal of Robotics Research, 1986, 5(4): 56-68.

[8] Davison A J, Reid I D, Molton N D, et al. MonoSLAM: Real-time single camera SLAM[J]. IEEE transactions on pattern analysis and machine intelligence, 2007, 29(6): 1052-1067.

[9] Shi J. Good features to track[C]//1994 Proceedings of IEEE conference on computer vision and pattern recognition. IEEE, 1994: 593-600.

[10] 石杏喜, 赵春霞, 郭剑辉. 基于 PF/CUKF/EKF 的移动机器人 SLAM 框架算法[J]. 电子学报, 2009, 37(8): 1865.

[11] Klein G, Murray D. Parallel tracking and mapping for small AR workspaces[C]//2007 6th IEEE and ACM international symposium on mixed and augmented reality. IEEE, 2007: 225-234.

[12] Rosten E, Porter R, Drummond T. Faster and better: A machine learning approach to corner detection[J]. IEEE transactions on pattern analysis and machine intelligence, 2008, 32(1): 105-119.

[13] Endres F, Hess J, Sturm J, et al. 3-D mapping with an RGB-D camera[J]. IEEE transactions on robotics, 2013, 30(1): 177-187.

[14] Rublee E, Rabaud V, Konolige K, et al. ORB: An efficient alternative to SIFT or SURF[C]//2011 International conference on computer vision. Ieee, 2011: 2564-2571.

[15] Mur-Artal R, Montiel J M M, Tardos J D. ORB-SLAM: a versatile and accurate monocular SLAM system[J]. IEEE transactions on robotics, 2015, 31(5): 1147-1163.

[16] Campos C, Elvira R, Rodríguez J J G, et al. Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam[J]. IEEE Transactions on Robotics, 2021, 37(6): 1874-1890.

[17] Wang Y, Cai S, Li S J, et al. CubemapSLAM: A piecewise-pinhole monocular fisheye SLAM system[C]//Asian Conference on Computer Vision. Cham: Springer International Publishing, 2018: 34-49.

[18] Urban S, Hinz S. Multicol-slam-a modular real-time multi-camera slam system[J]. arXiv preprint arXiv:1610.07336, 2016.

[19] Kerl C, Sturm J, Cremers D. Dense visual SLAM for RGB-D cameras[C]//2013 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, 2013: 2100-2106.

[20] Forster C, Pizzoli M, Scaramuzza D. SVO: Fast semi-direct monocular visual odometry[C]//2014 IEEE international conference on robotics and automation (ICRA). IEEE, 2014: 15-22.

[21] Engel J, Stückler J, Cremers D. Large-scale direct SLAM with stereo ca

meras[C]//2015 IEEE/RSJ international conference on intelligent robots an d systems (IROS). IEEE, 2015: 1935-1942.

[22] Engel J, Koltun V, Cremers D. Direct sparse odometry[J]. IEEE transactio ns on pattern analysis and machine intelligence, 2017, 40(3): 611-625.

[23] Kim D H, Han S B, Kim J H. Visual odometry algorithm using an RGB -D sensor and IMU in a highly dynamic environment[C]//Robot Intelligen ce Technology and Applications 3: Results from the 3rd International Con ference on Robot Intelligence Technology and Applications. Springer Inter national Publishing, 2015: 11-26.

[24] Bloesch M, Burri M, Omari S, et al. Iterated extended Kalman filter base d visual-inertial odometry using direct photometric feedback[J]. The Intern ational Journal of Robotics Research, 2017, 36(10): 1053-1072.

[25] 张慧娟, 方灶军, 杨桂林. 动态环境下基于线特征的 RGB-D 视觉里程计 [J]. 机器人, 2019, 41(1): 75-82.

[26] Sun Y, Liu M, Meng M Q H. Improving RGB-D SLAM in dynamic env ironments: A motion removal approach[J]. Robotics and Autonomous Syst ems, 2017, 89: 110-122.

[27] Liu H, Zhang G, Bao H. Robust keyframe-based monocular SLAM for a ugmented reality[C]//2016 IEEE International Symposium on Mixed and A ugmented Reality (ISMAR). IEEE, 2016: 1-10.

[28] Wang R, Wan W, Wang Y, et al. A new RGB-D SLAM method with mo ving object detection for dynamic indoor scenes[J]. Remote Sensing, 2019, 11(10): 1143.

[29] Bescos B, Fácil J M, Civera J, et al. DynaSLAM: Tracking, mapping, an d inpainting in dynamic scenes[J]. IEEE Robotics and Automation Letters, 2018, 3(4): 4076-4083.

[30] He K, Gkioxari G, Dollár P, et al. Mask r-cnn[C]//Proceedings of the IE EE international conference on computer vision. 2017: 2961-2969.

[31] Yu C, Liu Z, Liu X J, et al. DS-SLAM: A semantic visual SLAM towar

ds dynamic environments[C]//2018 IEEE/RSJ international conference on i ntelligent robots and systems (IROS). IEEE, 2018: 1168-1174.

[32] Badrinarayanan V, Kendall A, Cipolla R. Segnet: A deep convolutional en coder-decoder architecture for image segmentation[J]. IEEE transactions on pattern analysis and machine intelligence, 2017, 39(12): 2481-2495.

[33] Fan Y, Zhang Q, Tang Y, et al. Blitz-SLAM: A semantic SLAM in dyna mic environments[J]. Pattern Recognition, 2022, 121: 108225.

[34] Xiao L, Wang J, Qiu X, et al. Dynamic-SLAM: Semantic monocular visu al localization and mapping based on deep learning in dynamic environm ent[J]. Robotics and Autonomous Systems, 2019, 117: 1-16.

[35] Zhang J, Henein M, Mahony R, et al. VDO-SLAM: a visual dynamic ob ject-aware SLAM system[J]. arXiv preprint arXiv:2005.11052, 2020.

[36] Xing Z, Zhu X, Dong D. DE-SLAM: SLAM for highly dynamic environ ment[J]. Journal of Field Robotics, 2022, 39(5): 528-542.

[37] Chang J, Dong N, Li D. A real-time dynamic object segmentation frame work for SLAM system in dynamic scenes[J]. IEEE Transactions on Instr umentation and Measurement, 2021, 70: 1-9.

[38] Wu W, Guo L, Gao H, et al. YOLO-SLAM: A semantic SLAM system t owards dynamic environment with geometric constraint[J]. Neural Computi ng and Applications, 2022: 1-16.

[39] Guo M H, Xu T X, Liu J J, et al. Attention mechanisms in computer vi sion: A survey[J]. Computational visual media, 2022, 8(3): 331-368.

[40] Hu J, Shen L, Sun G. Squeeze-and-excitation networks[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2018: 7132-7141.

[41] Jaderberg M, Simonyan K, Zisserman A. Spatial transformer networks[J]. Advances in neural information processing systems, 2015, 28.

[42] Zhu X, Hu H, Lin S, et al. Deformable convnets v2: More deformable, b etter results[C]//Proceedings of the IEEE/CVF conference on computer visi

on and pattern recognition. 2019: 9308-9316.

[43] Dosovitskiy A, Beyer L, Kolesnikov A, et al. An image is worth 16x16 words: Transformers for image recognition at scale[J]. arXiv preprint arXiv:2010.11929, 2020.

[44] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[J]. Advances in neural information processing systems, 2017, 30.

[45] Carion N, Massa F, Synnaeve G, et al. End-to-end object detection with transformers[C]//European conference on computer vision. Cham: Springer International Publishing, 2020: 213-229.

[46] Ouyang D, He S, Zhang G, et al. Efficient Multi-Scale Attention Module with Cross-Spatial Learning[C]//ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2023: 1-5.

[47] Hubel D H, Wiesel T N. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex[J]. The Journal of physiology, 1962, 160(1): 106.

[48] LeCun Y, Bengio Y. Convolutional networks for images, speech, and time series[J]. The handbook of brain theory and neural networks, 1995, 3361 (10): 1995.

[49] Yu F, Koltun V. Multi-scale context aggregation by dilated convolutions[J]. arXiv preprint arXiv:1511.07122, 2015.

[50] Guo M H, Lu C Z, Hou Q, et al. Segnext: Rethinking convolutional attention design for semantic segmentation[J]. Advances in Neural Information Processing Systems, 2022, 35: 1140-1156.

[51] Jocher G, Chaurasia A, Stoken A, et al. ultralytics/yolov5: v6. 1-TensorRT, TensorFlow edge TPU and OpenVINO export and inference[J]. Zenodo, 2022.

攻读学位期间发表的学术论文目录

[1]杨海波,曹雏清.基于YOLOv8同步动态检测与局部语义视觉SLAM[J/OL].重庆工商大学学报(自然科学版),1-8

致 谢

对于我的二十岁人生来说，是在不停地思考，不停地摸索一个人生真相的一个过程，始终围绕着“我到底应该去做什么，如何去达到理想生活”这个中心主旨。

毕业是一个存档点，回顾二十岁的人生，一路上是没有人去告诉你，做什么是正确的，这个年龄做出的很多决策都是被自己现有思想左右的结果，所以，我一直奉行一句原则：“不要用战术上的勤奋去掩盖战略上的懒惰。”，想要破局，除了努力争取外，一定要多去收集信息，多去请教过来人，多去思考，只有方向正确，你的努力才会有指数型的效率，所以，宁可成为光想不做的人，也不要成为光做不想的人。

最后感谢朱怡晴的一路陪伴，感谢这研究生三年给我平台的导师曹雏清，感谢这三年路过我、影响过我的人，希望我们都可以达到自己理想的彼岸。