

**显著性物体检测**

**课题研究个人大作业**

院 系 信息工程学院

专　 　 业 电子信息工程

年 级 2020级

授 课 教 师 刘铁

文 档 作 者  1201002005戴凌慧

完 成 日 期 2023年6月29日

目录

[一、 论文算法基本介绍 3](#_Toc138948814)

[二、 课题研究步骤 4](#_Toc138948815)

[三、 本人工作介绍 9](#_Toc138948816)

[四、 代码说明 11](#_Toc138948817)

[五、 实验结果和项目展示 13](#_Toc138948818)

[六、 个人学习收获 14](#_Toc138948819)

# 论文算法基本介绍

我们在本次的课程任务中选择的显著性物体检测所选参考的论文是于2021年12月14日由李敏锡、申宇锡等人发表的《TRANCER: Extreme Attention Guided Salient Object Tracing Network》一文。文中提出示踪剂：一种极度注意力引导突出对象跟踪网络，通过结合注意力引导跟踪模块来检测具有明确边缘的突出物体。在第一个编码器的末端使用了屏蔽边缘注意力模块，适用快速傅里叶变换将细化的边缘信息传播到下游特征提取。在多级聚合阶段，联合注意力模块识别互补通道和重要空间信息。为了提高解码器的性能和计算效率，适用对象注意力模块最大限度减少解码块的适用。该模块从细化的通道和空间表示中提取未检测到的对象和边缘信息。随后，提出了自适应像素强度损失函数来处理相对重要的像素，这与平等对待所有像素的传统损失函数不同，能够大大提升处理的速度从而提升整个算法的性能。以下是TRANCER算法的框架图（此框架图由小组根据原论文算法调整得到）：

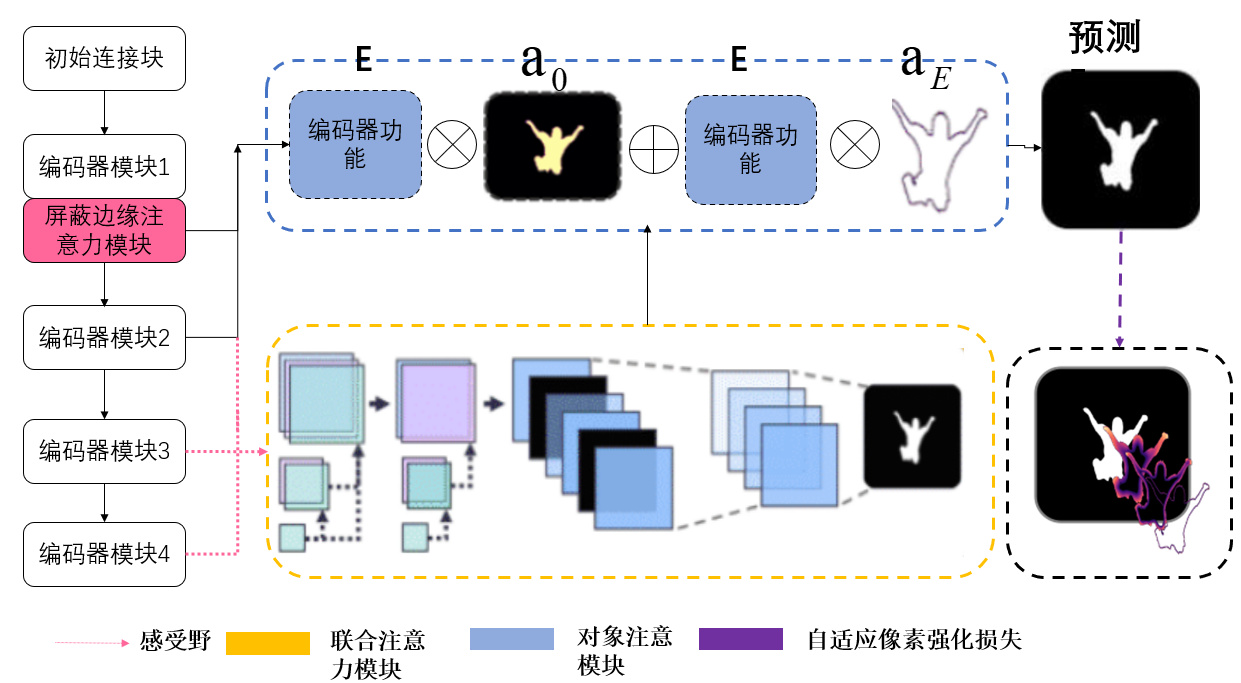


图 1 TRANCER算法框架图

其中的主要模块主要具有以下功能：

* 注意力引导跟踪模块：检测具有边缘的不同物体来提高显著物体检测的性能。主要是用卷积模块，同时通过注意力引导的突出对象跟踪模块跟踪对象和边缘来提高性能，其主要包含屏蔽边缘注意力模块以及联合注意力模块和对象注意模块来达到提升性能的目的。
* 自适应像素强度损失函数：自适应像素强度损失函数主要就是对于像素进行权重划分，由于显著物体检测的过程中，主要是对其中的突出物体进行和背景的分离，所以就划分了一些特殊像素和所谓的背景像素，对于一幅图像，适用自适应像素强度损失函数来进行特殊像素的权重划分，从而在进行后续处理的时候，可以直接找到特殊像素，从而可以提高整个系统的性能，在和其他算法上才具有更好的性能竞争力。

# 课题研究步骤

我们小组在对该算法进行研究的过程中，主要按以下的流程依次进行：

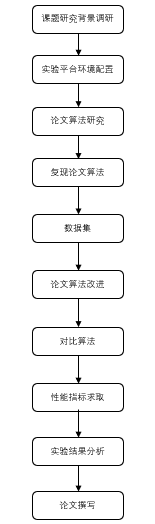


图 2 课题研究步骤

1. 研究问题背景的调研：

我们所研究的问题是显著物体检测，所以我们小组将对显著物体检测的发展过程做一个简要的调研，对于显著物体检测的开始，以及显著物体检测今年来的一个发展情况，所面临的主要挑战是什么？近年来学者研究显著物体检测所使用的方法是什么？在这个过程中，我们基本确定显著性物体检测的发展历程，以及选定小组要研究的论文算法，以及确定一些对比算法的范围。从而进行后续工作的进行。

以下是小组对研究问题背景的调研结果：

显著性物体检测最早可以追溯到1998年，Itti等发表了第一个基于显著度的视觉主义计算模型，使用多种特征的融合获得最终的显著图，视觉注意力机制模型正式引入到计算机视觉领域，从此显著性物体检测算法得到了蓬勃发展。现在显著性物体检测可以分为两种传统的显著性物体检测方法和基于深度学习的显著性物体检测方法。其中传统显著性物体检测算法还可以继续划分为基于对比度的显著性检测算法、基于频域的计算方法、基于稀疏理论的计算方法、基于图模型的计算方法、基于背景和前景先验的计算方法。在这其中例如Cheng等人提出全局对比度算法HC和RC等。在传统的方法使用空域或领域中大量的特征对比或者利用先验信息等进行图像显著性检测，其针对性较强，一般来说只适用于主题明确、物体颜色及结构较为单调的场景。基于深度学习的显著性物体检测方法又可以分为基于传统卷积神经网络的方法和基于完全卷积神经网络的方法。基于传统卷积神经网络的方法是将图像块的区域大小调整到固定的尺寸或进行多尺度的处理，然后进行卷积神经网络的特征提取，最后融合得到显著图。基于完全卷积神经网络的方法主要是利用VGGNet骨干网络架构和ResNet骨干网络架构。在基于深度学习的显著性物体检测方法中主要是利用卷积、下采样和上采样，会丧失显著物体的一些细节，在此基础上基于复杂的特征融合网络结构被提出。为了提高显著目标检测的性能，现有方法可以分为两种：改进边缘表示的方法和减少多级聚合过程中差异的方法。但是由于在显著物体检测的过程中，存在目标背景复杂，边缘细节不够突出，算法程度较高，计算效率低等等的问题，目前所提出的显著物体检测始终还不能很好地接近我们人工判断的理想的显著物体检测目标理想图。

1. 论文算法所用的实验平台的环境配置：

在本次实验的过程中，我们基于的是python3.9平台，使用的是pycharm编程环境，在实验中所要求的库有：Pytorch>=1.8.0，albumentations>=0.5.1，tqdm>=4.54.0，scikit-learn>=0.23.2。在下载的开源代码之后，需要对自己的环境进行配置，配置好所需要的一些库，在进行代码的可行性测试，根据原论文所提供代码的readme文件进行项目文件的修改和添加，大致把整个项目的框架结构梳理好。

1. 论文算法的研究：

我们在初步确定小组要研究的论文算法后，对该论文的算法进行大致的了解，理解算法框架中各个部分的功能和实现方法，对于算法所需要的硬件设备要求，以及对于开源所提供的代码的基本了解。对于算法提出背景的了解。

以下是我们小组对论文算法的研究：

现有的显著性目标检测模型研究由于明确的边缘信息即兴发挥了SOD性能，但它们广泛使用边缘细化模块，这些模块需要全部或部分编码器输出。这些方法增加了开销和计算效率低下的问题。此外，这些研究中的方法没有在下游和高级信息对于确定边缘特征和语义表示是有效的，因此应有选择地确定这两个级别的细化模块的使用，并尽量减少网络效率，所以提出的TRANCER示踪剂网络添加了屏蔽边缘注意力模块和自适应像素强度损失函数来权重划分像素点，从而提高算法性能。

1. 复现论文算法：

在复现论文算法的过程中，主要要做的就是由网上所提供的开源代码，对其代码进行理解，并代码进行改进，从而适应我们本次实验所要达到的实验目的。在本次实验的过程中，我们所要做的是：限制小组所选定的训练数据集、测试数据集、代码运行结果格式输出。需要达到以上的目标，所以在复现代码的过程中，不仅仅需要对其中的问题代码进行修改，还需要对其中代码的数据输入，训练部分代码的修改以及测试训练集修改，和输出进行调整。所以复现论文算法具有非常大的挑战难度。

1. 数据集的采集和整理：

在本次实验的过程中，我们小组所收集的三个基准数据集，分别是：DUTS、ECSSD、DUT\_OMRON，其中我们使用DUTS\_TR数据集进行训练，其余DUTS\_TE、ECSSD、DUT\_OMRON三个数据集作为测试数据集。

以下是三个基准数据集的下载链接以及对三个基准数据集的简单介绍：

其中ECSSD（Yan et al.2013）包含1，000个结构复杂且有意义的场景。DUTS（王等.2017）是SOD最大的基准数据集，它包含10，553张训练图像和5，019张测试图像。DUT-OMRON（Yan等人。2013）有5，168张图像，其中包含一个或多个具有相对复杂背景的突出物体。由于在实际实验的过程中，运行内存的原因，我们将其中的DUTS-TE、DUT-OMRON数据集进行了部分删减，最终所使用的数据集中：ECSSD数据集使用1，000张，DUTS-TE数据集为687张、DUT-OMRON数据集为1，729张。

以下是三个基准数据集的下载链接：

* ECSSD数据集：[扩展的复杂场景显著性数据集 (cuhk.edu.hk)](https://www.cse.cuhk.edu.hk/leojia/projects/hsaliency/dataset.html)
* DUTS数据集：[DUTS 突出目标检测数据集 |卡格尔 (kaggle.com)](https://www.kaggle.com/datasets/balraj98/duts-saliency-detection-dataset)
* DUT-OMRON数据集：[DUT-欧姆龙图像数据集 (saliencydetection.net)](http://saliencydetection.net/dut-omron/)

1. 论文算法的改进：

在对论文算法的改进中，主要要做的就是，修改报错代码，以及修改测试数据集，训练数据集，输出数据集格式和地方以及数据集的获取等等地方，所以在论文算法的改进阶段，还是一个比较复杂的过程，在这个过程中，要去理解代码的含义，能看懂报错代码的意思，如何去修改的等等的问题。以及在实际下载的项目文件下，要适应自己电脑的运行效率和调整项目文件，进行增加或修改等操作。

1. 对比算法的复现：

在本次实验的过程中，我们主要选择了三个算法与TRANCER算法进行比较，分别是：2020年由秦学斌、张子辰等人所提出的《U2NET: Going Deeper with Nested U-Structure for Salient Object Detection》以及2020年所提出的《A2dele: Adaptive and attentive depth distiller for efficient RGB-D salient object detection》两篇论文所提出的U2NET模型和A2dele模型中的CPVR\_DEPTH以及CPVR\_RGB两个模型。以下是对两篇论文算法提出的简单介绍：

在第一个对比算法中，提出的一个简单而强大的深度网络架构，U2NET，用于显著目标检测（SOD），其U2NET是一个两级嵌套的U结构。该设计具有以下优点：（1）由于提出的ReSidual U型块中混合了不同大小的感受野，因此能够从不同尺度捕获更多的上下文信息；（2）由于这些RSU块中使用的池化操作，它增加了整个架构的深度，而不会显著增加计算成本。这种架构使得能够从头开始训练深度网络，而无需使用图像分类任务的主干。

现有的最先进的RGB-D突出目标检测方法依赖于双流架构探索RGB-D数据，其中需要独立的子网来处理深度数据。这不可避免地会产生额外的计算成本和内存消耗，并且在测试过程中使用深度数据可能会阻碍RGB-D显著性的实际应用。为了解决这两个难题，提出了一个深度蒸馏器（A2dele）来探索使用网络预测和注意力作为将深度知识从深度流转移到RGB流的桥梁的方式。首先，通过自适应最小化深度流和RGB流生成的预测之间的差异，实现了对传输到RGB流的像素级知识的所需控制。其次，为了定位知识转移到RGB特征中，鼓励深度流的膨胀预测与RGB流的注意力图之间的一致性。因此，通过嵌入A2dele，在测试时不使用深度数据，实现了轻量级架构。

在复现对比算法的过程中，主要大致看懂代码，对其训练数据集、测试数据集、结果数据集格式输出进行改进，以及对报错代码进行修改和环境进行配置。在实现过程中，具有一定的难度。

1. 性能指标算法的编写：

在选择性能指标时，所选择的性能指标主要有6个，分别是：最大的F-measure（maxf）、平均阈值（fm）、平均绝对误差（MAE）、W-fmeasure、S-measure、E-measure以及评价的时候，还选择了PR曲线来作为另一个性能评估的指标。

以下是对各个指标的一个说明：

* Maxf= max(1.3\*prec.\*rec./(0.3\*prec+rec+eps))

注：其中的prec指的是精确率，rec指的是召回率。

* **F-measure = (2 \* Precision \* Recall) / (Precision + Recall)**
* 

注：其中的指的是真实值，即实验过程中理想的mask，是我们通过算法所预测的结果图。

* W-Fmeasure是综合评价指标F-measure的中的一个评价指标。
* S-measure是计算前景像素和真值的相似度，其计算方法如下：



其中表示按重心位置，然后切割成四个区域，对四个区域按像素占整张图的区域面积作为权重，计算四个区域SSIM的加权平均。最后实验计算需要对所有的sm做一个平均。

* E-measure用来评价预测结果中匹配到正确结果的百分比。
* PR曲线的组成主要是由精确率和召回率进行绘制的，其计算方法如下：



在6个性能指标算法的评价过程中所使用的编程语言是matlab进行，生成数据表格，在绘制PR曲线过程中，使用的python进行绘制。

1. 实验结果的分析：

我们对实验结果的分析主要分为定性和定量两个方面的分析对比，定性分析也就是算法所跑出来的mask图像，对比各个算法所生成的结图像以及结果图和理想maks之间的差距，定量的分析也就是图像质量评价指标进行分析对比，利用选取的6个性能指标以及PR曲线进行定量的实验结果分析。

1. 论文的撰写：

论文的撰写主要就是讲清楚课题研究的背景、算法介绍、各部分模块的分析以及实验部分，对之前的各个阶段的过程结果进行整理，完成本次论文的撰写。

# 本人工作介绍

在本次课题研究的过程中，我主要负责的部分是对比算法的复现、数据集的采集和整理、论文算法步骤的梳理和PPT部分内容的编辑、课题论文的撰写、性能指标算法以及PR曲线绘制算法的编写部分。

其中对比算法的复现：我主要进行的就是基于pycharm编程环境，对于从网上下载的源代码进行修改，配置环境、库的下载以及原始环境包的升级。在实际复现过程中，其实还是遇到了很多困难，比如一些包安装不成功的问题，由于版本不匹配的原因或者是其他原因导致一些库安装不成功，所以在实际操作过程中，还是在网上查询了不少相关的资料去解决一些包安装不成功的问题。并且在实际运行中，代码中的路径要实际符合自己电脑上的路径，所以在运行之前，就要看懂整个文件的具体框架以及运行主程序的代码框架，对其中的路径进行修改，修改完之后，再对项目文件中的数据进行增加或修改，从而开始试着运行代码，一步一步修改其中错误的部分。最终达到整个算法的可执行性。以下是对比算法所参考的论文链接：[U$^2$-net：使用嵌套 U 结构更深入地进行突出对象检测 |带代码的论文 (paperswithcode.com)](https://paperswithcode.com/paper/u-2-net-going-deeper-with-nested-u-structure)+[A2dele：自适应和专注深度蒸馏器，用于高效的RGB-D突出物体检测 |带代码的论文 (paperswithcode.com)](https://paperswithcode.com/paper/a2dele-adaptive-and-attentive-depth-distiller)。

数据集的采集和整理：在数据集的采集中，我主要选取的是一些数据量相对而言不是很多也不是很少的数据集，但是其中所选取的训练数据集还是选取的比较热门的DUTS数据集进行模型的训练，在实际运行过程中，由于运行内存的原因，我又对部分数据集进行了删减。

论文算法步骤梳理：主要是在课程汇报阶段，自己对论文的算法进行了一个简单的梳理，对于其中的重要方法和公式进行了整理，并且在原论文所提供的算法框架图的基础上进行了一定的修改生成我们本论文所使用的算法框架图。

PPT部分内容的编写：主要是在每一次汇报之前对于PPT的内容的书写，我们小组内所分配的部分进行编写。

课程论文的撰写：我在撰写本次课程论文的过程中，主要参考的是原论文的算法介绍，在实验验证阶段的时候，主要是自己进行说明和撰写，由于我们在本次实验的过程中，所选用的训练数据集和测试数据集以及生成数据集都是我们小组自己拟定的，所以在实验验证阶段，与原论文有所不同。我主要介绍的是：课题研究背景，目前所买面临的挑战、算法介绍、实验结果分析的说明。

性能指标算法的和PR曲线的绘制：在进行定量分析的过程中，我主要使用了matlab语言进行了性能指标算法的编写，选取6个性能指标进行计算，在绘制PR曲线的过程中，我主要对各个数据集各种方法所对应的精确率和召回率用python绘制了一个PR曲线对实验结果进行分析。以下是在计算性能指标所使用到的数据：（每个算法下有对应三个基准数据集的生成结果图，从而对实验结果图与理想图之间进行质量评价）

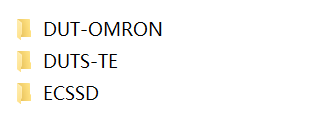


图 3 测试数据集

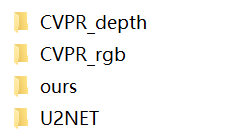


图 4 各个算法实验结果图

# 代码说明

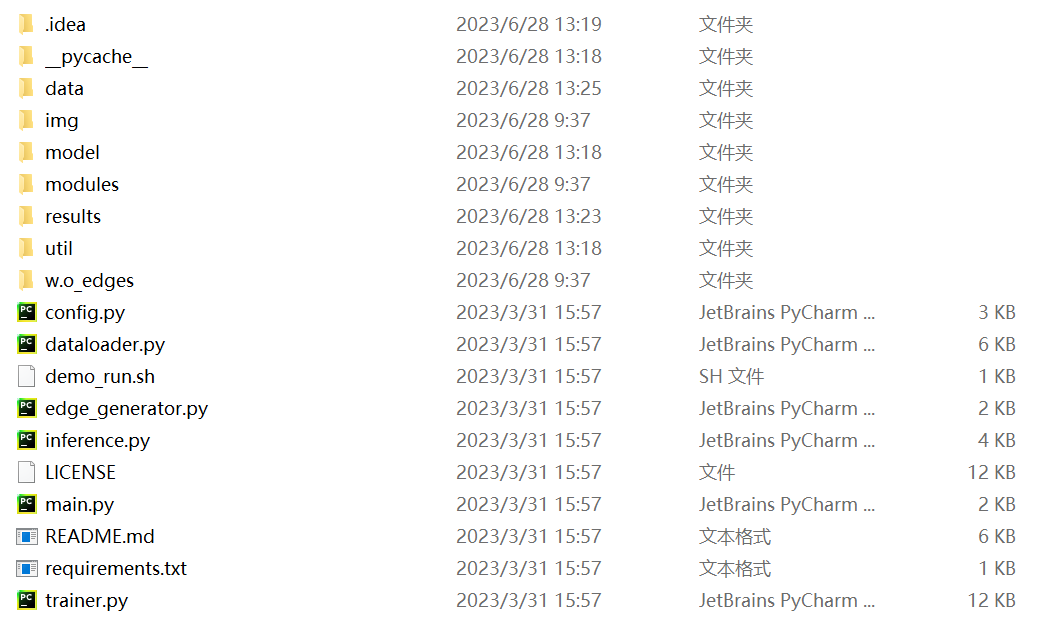


图 5 项目文件

在代码运行的过程中，主要的项目文件如上图所示，其中我们主要运行的就是main.py文件，其中包含了模型的训练，数据集的测试两个部分。其中的data所含的文件主要是我们的训练的数据集，results用来存放我们算法的结果图。其中的traner提供了单独的训练代码，可以先对其进行训练，生成预训练模型将其放入对应的文件夹models下。

在代码的运行过程中，我们将使用预先训练的TE7版本发布初始化TRANCER。运行main.py脚本。其中main.py代码的框架结构如下：

**TRACER**

├── data

│ ├── custom\_dataset

│ │ ├── sample\_image1.png

│ │ ├── sample\_image2.png

.

.

.

其中所使用的训练数据集为DUTS-TR，测试数据集为DUTS-TE、ECSSD、DUT-OMRON其数据集的结构如下：

**TRACER**

├── data

│ ├── DUTS

│ │ ├── Train

│ │ │ ├── images

│ │ │ ├── masks

│ │ │ ├── edges

│ │ ├── Test

│ │ │ ├── images

│ │ │ ├── masks

│ ├── ECSSD

│ │ │ ├── images

│ │ │ ├── masks

│ ├──DUT-OMRON

│ │ │ ├── images

│ │ │ ├── masks

运行完训练部分之后，所生成的预训练模型将其放入results/DUTS/TE7/best\_model.pth中。接着就是开始测试阶段，用所训练好的模型进行数据集的测试将结果输出到results中，完成代码的运行。

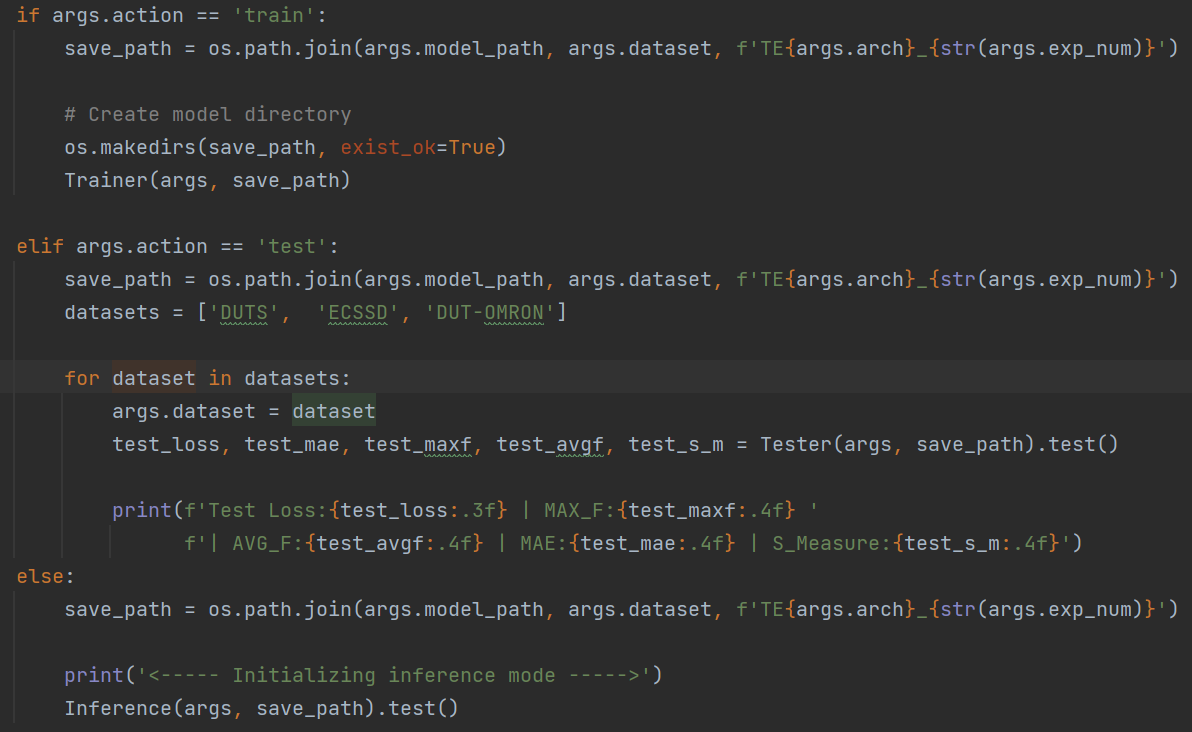


图 6 main.py主要部分代码

# 实验结果和项目展示

定性分析结果：

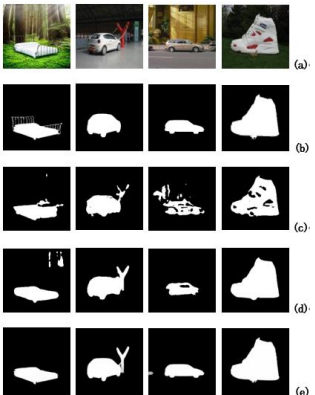
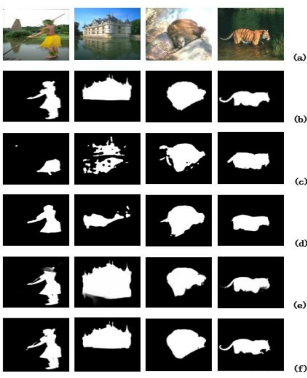
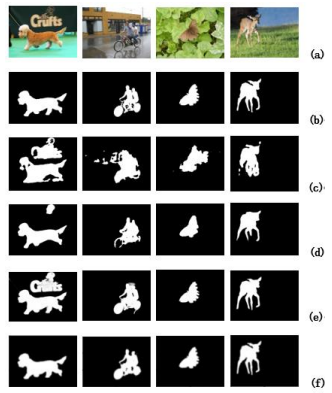


图 7 从左到右依次为DUTS-TE、ECSSD、DUT-OMRON数据集，其中(a)原图（b）理想mask（c）CVPR-DEPTH（d）CVPR-RGB（e）U2NET（f）ours

定量分析结果：

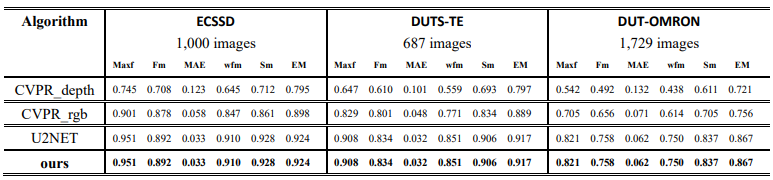


图 8 6个性能指标结果

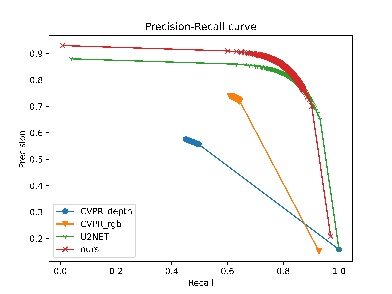
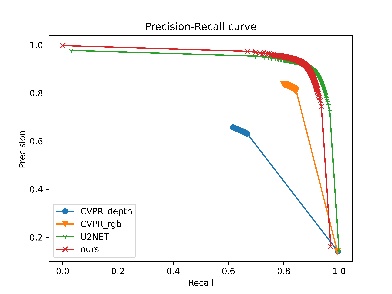
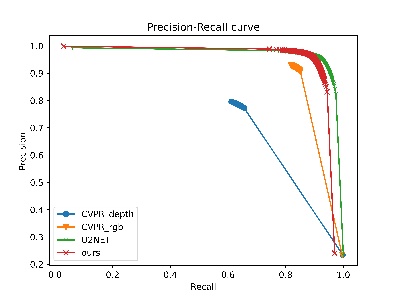


图 9 PR曲线 从左到右依次为ECSSD、DUTS-TE、DUT-OMRON数据集

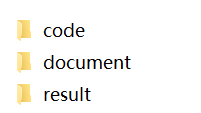


图 10 最终上传项目文件

# 个人学习收获

经过本次对显著物体检测课题的研究，自己从中收获还是挺大的，在这样的一个课程项目实践的过程中，了解到了做一个项目的大致流程是什么，需要去准备些什么，以及在这个过程中可能遇到的一些问题。在这一次的显著物体检测的算法实现的过程中，自己也积极承当了其中很多比较困难的任务，尤其是对于对比算法的复现，由于在这个过程中，不仅仅要看懂别人的代码，以及要考虑环境的配置，以及电脑的运行效率等问题，所以在实际操作过程中，遇到了不少难题，但是在自己查阅相关的知识，以及自己对于代码的理解从而对其中的报错代码或者是配置问题、运行内存问题、参数设置问题不断地进行修改，从而达到最终的实验目的。并且，在本次实验的过程中，自己对于一些算法的理解之后，发现，其实很多算法中所包含的一些方法很多都是一些基本的操作，只是这样一部分一部分结合在一起，来完成整个项目的设计过程，其中很多图片特征的提取都离不开卷积操作又或是池化操作。所以，在本次的课题研究项目的设计过程中，自己还是学习到很多有用的知识，以及学会使用了一些新的开源网站，在其中，自己也学习到不少有关显著性物体检测的方法。希望在今后的学习生活中，自己能够多去学习一些课外的知识，不断扩充自己的实力。以上就是我本次设计显著物体检测课题研究的个人感悟。