



## 中国大学生服务外包创新创业大赛

基于非特定类别的图像前景主体分割算法【万兴科技】

团队名称:	Have Hands
队长:	卢林军
队员 1:	崔子航
队员 2:	蔡响
队员 3:	李校青

---

## 目录

一、设计需求分析 .....	2
1.1 作品的市场需求分析 .....	2
二、执行思路 and 软件架构 .....	2
2.1 执行思路 .....	2
2.2 软件架构 .....	3
三、算法介绍 .....	3
3.1 引言 .....	3
3.2 设计思路 .....	4
3.2.1 研究现状 .....	4
3.2.2 现有模型存在的问题 .....	4
3.2.3 网络解决的问题 .....	5
3.3 CONV-BATCHNORM-RELU .....	5
3.4 Residual U-blocks .....	6
3.4.1 RSU-L 的构成 .....	6
3.5 网络整体结构 .....	7
3.6 优化方案 .....	9
3.6.1 Loss 设计 .....	9
3.6.2 门控注意力机制 .....	10
四、项目工程化 .....	11
4.1 推理框架介绍 .....	11
4.2 推理基本流程 .....	11
4.2.1 生成 IE (Infer Engine) .....	11

---

4.2.2 构建推理类 .....	12
4.3 前端界面设计 .....	13
4.4 前端与推理类的交互 .....	17
五、算法模型的优劣 .....	17
六、展望 .....	30

---

## 摘 要

目前,语义分割在人像分割上逐渐成熟并被广泛应用,为了让分割的图像不再局限于人像,并且能够快速准确将图像的主体和背景分离,以提升更广泛的实用性。人类视觉系统在面对自然场景时具有快速搜索和定位感兴趣目标的能力,这种视觉注意机制是人们日常生活中处理视觉信息的重要机制。这种视觉注意力机制被引入到了深度学习中,于是兴起了显著性目标检测算法,其能够实现不区分物体的类别,专注于将人类最能注意到的前景与无关紧要的背景分离而不需要详细区分其类别,提高了从海量图像和视频数据中快速获取重要信息的能力。

为实现非特定类别图像前景主体分割,本团队收集了数据集 DUT-OMRON、DUTS、ECSSD 等显著性目标检测数据集,并使用数据增强的方法对其进行扩展。提出 U<sup>2</sup>GE-Net 显著分割算法,作为本任务的主要分割算法,使用上述数据集对模型进行训练。并在原有模型的基础上添加视觉注意力机制,使得模型对主体部分有更强的专注力。采用 opencv 加速推理框架将其转化成 C++可执行程序,使用 QT 编写应用界面,便于用户操作使用。

本项目较为优秀地将图片的前景和背景分割出来,并生成背景为透明色的背景,满足了用户对图片的需求,且经过训练后的显著性检测模型仅为 7.03MB。对输入图形采用了归一化等预处理方式,在软件上图像前处理运行时长(基于 CPU)为 0.18s;推理过程时长(基于 CPU)为 0.47s;图像后处理时长(基于 CPU)为 0.21s。处理 2k 图片总时长(基于 CPU)约为 1s,处理 4k 图片总时长(基于 cpu)约为 2.5s,其速度较为突出。

关键字: 计算机视觉, 前景主体分割, 显著性目标检测, 注意力机制

---

## 一、设计需求分析

### 1.1 作品的市场需求分析

互联网经济是近年来中国经济的主要增长动力之一。2019 年，中国人均互联网经济占人均国内生产总值的 32.0%。中国拥有一大批不断增长的智能手机用户。在智能手机的热潮推动下，中国移动互联网用户人数从 2014 年的 55680 万人增长至 2019 年的 83070 万人，复合年增长率为 8.3%。

自媒体指用户在自媒体平台上注册的线上账户，用于以文本、图片、音频或视频内容等各种形式向公众发布内容。在 2013 年自媒体出现之前，移动互联网用户主要透过门户网站和移动应用程序获取内容。自 2013 年以来，自媒体快速发展，按 2018 年中国在移动设备上花费的时间计，其已成为主流媒体不可或缺的组成部分。在微信、微博、今日头条和抖音等领先自媒体平台日渐受移动互联网用户关注而普及的推动下，行业客户对自媒体投放的资源 and 营销预算多于传统门户网站。以 2019 年收益计，自媒体营销市场规模约占移动广告市场总额的三分之一。以收益计的自媒体营销市场规模由 2014 年的人民币 162 亿元增加至 2019 年的人民币 1593 亿元，复合年增长率为 57.9%。

为了顺应时代的自媒体行业的发展，大量的影视媒体软件由此诞生。这些软件能更好地帮助自媒体行业的人员发布更好的资源，从而推动这个行业的不断发展。传统的影视媒体软件具备剪辑，美颜等视频图像处理方式，但没有将 AI 的视觉技术融入到影视媒体软件中。本作品面向图像前景和后景分离的方式在影视媒体软件中一直没有很好的表现这一问题开始分析，即使在计算机视觉中的语义分割已经比较成熟的今天，语义分割也仅能较为优秀地对特定类别的物体实现前后景分离，但对于一些可以分类非特定的类别的物体表现一直不佳。优秀的前景分离算法，能够更大程度的满足自媒体行业人的需求，拥有优秀的前景分离算法的影视媒体软件也能扩大自身对自媒体行业人的影响力，提高软件在自媒体行业在地位。

## 二、执行思路 and 软件架构

### 2.1 执行思路

本作品旨在采用高效的显著性目标检测算法模型，对一张普通图片的非特定类型的前景进行分离。本作品能够较为优秀地弥补当今影视媒体软件无法精

准分离非特定类型前景的不足，为更多的自媒体人提供更好的软件体验，推动自媒体的行业的发展，为公众带来更多乐趣，为社会创造更大的价值。

本作品提出了 U<sup>2</sup>GE-Net 显著性分割模型，通过增加视觉注意力机制对非特定类别的主体部分产生更强的注意力，采用推理框架 opencvino 将其部署到 QT 编写的应用软件中。为用户提供简洁的操作界面和优秀的分割算法，满足用户将非特定类别的前景图像分离出来的需求。

2.2 软件架构

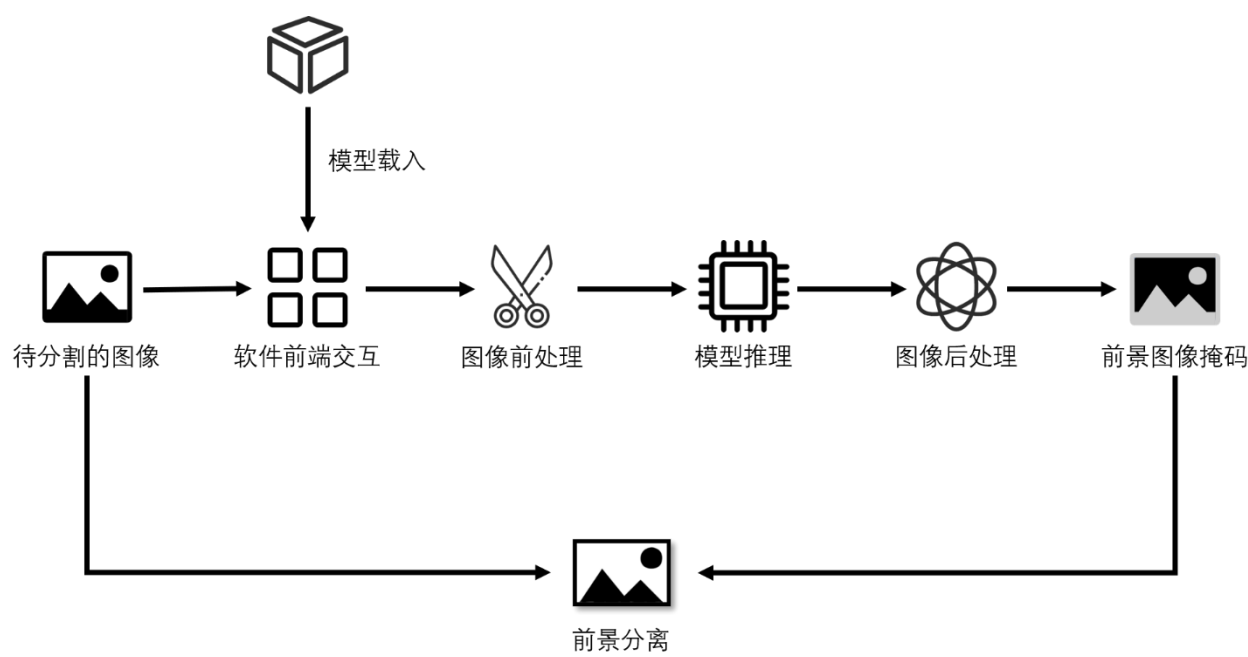


图 2.1 软件架构

本软件主要由 QT 前端界面和非特定类别图像前景分割算法构成。其中非特定类别图像前景分割算法以 U<sup>2</sup>GE-Net 为核心，通过推理框架转化封装成供 QT 前端界面使用的接口。前端界面主要由基本的选择图像，开始执行，输出图像几个基本控件构成，控件中都嵌入了非特定类别图像前景主体分割算法那的接口，使得用户能够更为直接，简便地实现图像的前景分离。

三、算法介绍

3.1 引言

在计算机视觉不断发展的今天，图像分割技术不断成熟。其中语义分割方法在人像分割上表现突出，但是语义分割本质上是对图像的像素级分类，其对

---

于单一类别图像或是几种指定类别的图像表现极佳，但是若类别增加到更大的数量级，语义分割的效果执行的效果并不能令人满意。因此，在本任务中，为实现非特定类别的图像分割，采用了显著性目标检测的方法。经过不断地实践，提出了本任务的核心算法 U<sup>2</sup>GE-Net。

## 3.2 设计思路

### 3.2.1 研究现状

在现在的显著性目标检测算法中，都注重利用现有的主干网络提取图像的深层特征，比如 Alexnet, VGG, ResNet, ResNext, DenseNet 等。这些主干网络不仅被广泛应用在显著性目标检测算法中，也被应用在图像分类，图像分割，传统目标检测等神经网络模型中。

在继承前人实验之后，现在大部分计算机视觉的神经网络模型也都赞同采用这些现有的骨干网络，但是除了显著性目标检测算法，大部分的计算机视觉算法大多数都有实现分类的一个过程，比如在图像分类中就是对整张图片进行分类，在图像分割中就是对像素进行分类，在目标检测中，就是对边界框的物体进行分类。这些算法大部分都采用了现有主干网络，因此这些主干网络最终都是为图像分类任务而设计的，他们提取的特征代表语义，而不是局部细节和全局对比信息，但是局部细节和全局对比信息对显著性目标检测非常重要。

### 3.2.2 现有模型存在的问题

由于在现有的主干提取网络中添加了特征聚合模块的方法，对图像进行多层显著性特征提取，使得传统的显著性目标检测网络变得十分复杂。除此之外现有的主干网络通常通过牺牲高分辨的特征映射来实现更深层次的体系结构，在现有的设备条件比较欠缺的情况下，牺牲分辨率去采用深层结构即满足了当前设备所能支持的极限，同时也能提高网络对图像特征的提取能力，但是图线中原本高分辨率的信息损失也会增大。

因此，现有模型的问题主要是：在研究算法的时候，能不能在保持高分辨率的特征图的同时，以较低的内存，显存和计算成本进行更深层次的研究，使得模型对图像的特征提取力度得到加强。

---

### 3.2.3 网络解决的问题

对于舍弃了主干网络的模型中  $U^2$ -net (CVPR2020) 无疑是一个可靠的存在, 经过  $U^2$ -net 中 RSU-L 的启发, 我们将其改造成可以插入到网络中的模块 GRSU-L, 并继续采用  $U^2$ -net 的原本结构构造了  $U^2$ GE-Net 网络, 通过引入边缘 loss、IoU loss 与门控通道注意力模块 (GCT) (CVPR2020) 来提升对图像的特征提取能力, 使得模型对图像的特征提取更加深入, 提取的特征更具价值。

因此  $U^2$ GE-Net 网络解决的主要有两个问题:

1. 该网络是一个两层嵌套的 U 型结构, 没有使用图像分类的与训练主干网络, 可以从头开始训练。同时也摆脱了图像分类主干网络的局限, 对图像原本的局部细节和全局对比信息的显著性提取得到进一步增强。
2. 在新的体系结构中, 允许网络更加深入地获得高分辨率的特征, 而不会显著地增加内存和计算成本。在底层中新模块 GRSU-L, 能够在不降低特征映射分辨率的情况下提取级内多尺度特征。在顶层, 由于有一个类似于 U-net 的网络结构, 原本 U-net 中的简单卷积就被替换成了 GRSU-L。

本算法的研究重点主要在于: 多层次深层特征集成方法和多尺度特征提取方法。多层次深层特征集成方法主要是指开发更好的多层次特征融合策略, 多尺度特征提取方法是指设计心得模块, 从主干特征提取网络获取的特征中提取局部和全局信息。通过对不同层次的显著性特征信息进行有效集合, 将对显著性目标检测的成果和效率有很大提高。

### 3.3 GCT-CONV-BATCHNORM-RELU

卷积→批量归一化→RELU 激活函数可以使输入  $X$  更趋于非线性分布, 并产生一个更稳定的激活分布, 有利于提升模型性能。因此  $U^2$ -Net 的每一个 RSU-L Block 中, 都使用带填充和空洞的卷积→批量归一化→RELU 激活函数作为基础单元来保证特征提取的准确与稳定。我们在此基础上引入了门控通道注意力模块 GCT, 结构图 3.1 所示:



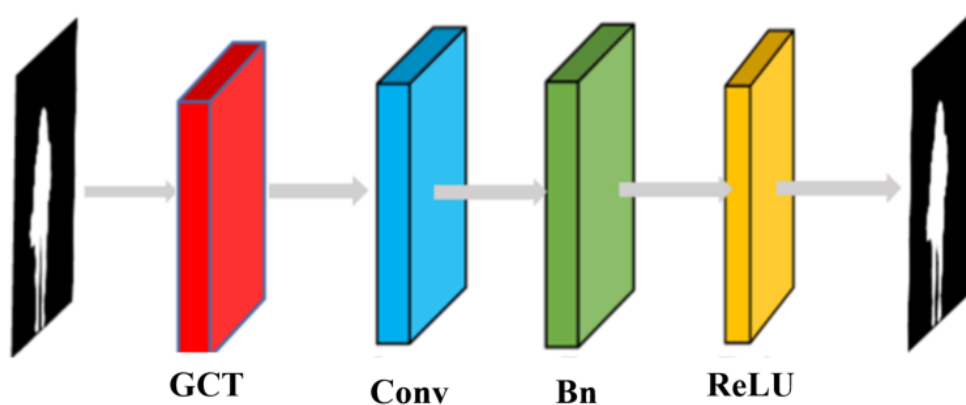


图 3.1 GCT-CONV-BATCHNORM-RELU

### 3.4 Residual U-blocks

对于显著性目标检测和其他分割任务来说，局部和全局上下文信息非常重要。在现有的主干特征提取网络，如 ResNet，DenseNet 等网络， $1\times 1$  或者  $3\times 3$  的小型卷积核是最常用的特征提取元件，因为他们需要较小的存储空间并且计算效率高。

但由于  $1\times 1$  或  $3\times 3$  卷积核的感受野太小而无法捕捉全局信息，因此底层的输出特征图只包含了局部特征。为了在高分辨率浅层特征图上获取更多的全局信息，最直接的想法就是扩大感受野。因此研究人员试图向  $3\times 3$  卷积核添加空洞，通过这样的卷积核来使卷积过程扩大感受野以提取局部特征和非局部特征。然而，在原始分辨率的输入特征图上进行多次拓展卷积（尤其是初始阶段）需要大量的计算和内存资源。

#### 3.4.1 GRSU-L 的构成

使用 GRSU-L (GCT Residual U-blocks Layer) 来捕捉阶段内多尺度特征，结构图如图 3.2:

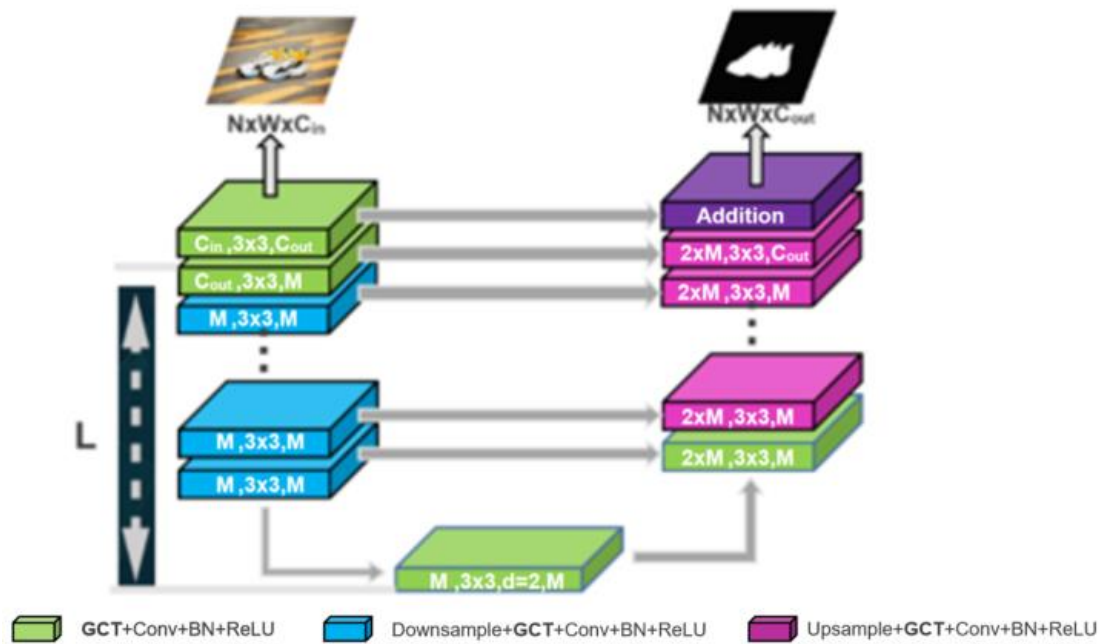


图 3.2 GRSU-L ( $C_{in}$ ,  $M$ ,  $C_{out}$ ) 结构

在 GRSU-L ( $C_{in}$ ,  $M$ ,  $C_{out}$ ) 中,  $L$  是编码器的层数,  $C_{in}$ ,  $C_{out}$  表示的是输入和输出的通道数,  $M$  表示 RSU 内部中的通道数。因此 GRSU 主要由三部分组成:

1. 输入卷积层: 经特征图转化成具有  $C_{out}$  通道数的中间特征图  $F_1(x)$ , 这是提取局部特征的普通卷积核。
2. 中间特征图  $F_1(x)$  为输入, 学习提取和编码多尺度上下文信息  $U(F_1(x))$ 。 $U$  表示上图箭头中  $L=7$  所包括的部分。 $L$  越大, RSU 越深, 池化的操作也越多, 更大的感受野以及更丰富的局部和局部全局特征。通过设置  $L$  可以从具有任意空间分辨率的输入特征图中提取多尺度特征。从梯度降采样特征图中提取多尺度特征, 并通过逐步上采样, 合并和卷积等方法将其编码到高分辨率特征图中。这一过程减少了大尺度直接上采样造成的细节损失。
3. 通过求和:  $F_1(x) + U(F_1(x))$  实现融合局部特征和多尺度特征

### 3.5 网络整体结构

$U^2GE$ -Net 是一个两层嵌套的 U 型结构, 它的顶层是由 11 个 stage 组成, 每个 stage 都由一个配置良好的 GRSU-L 填充, 因此可以更加有效地提取多尺度特征和聚合多层次特征。结构如图 3.3 所示:

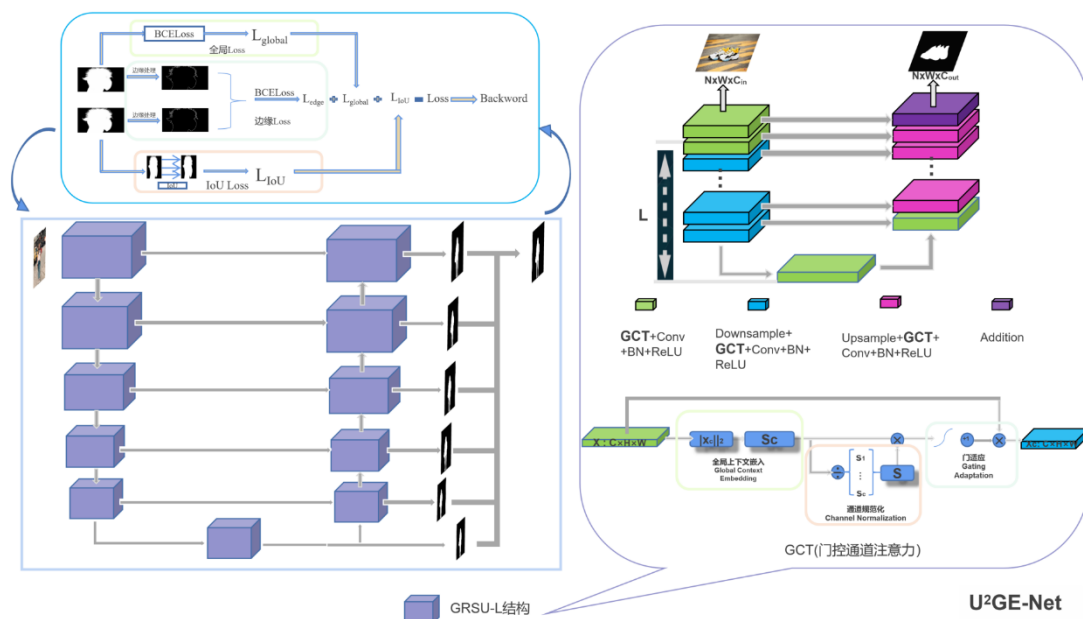


图 3.3 U<sup>2</sup>GE-Net 整体结构

U<sup>2</sup>GE-Net 的主要包括三个部分：

### 1. 编码器：

在编码阶段，GRSU-L 的高度根据输入特征映射的空间分辨率来进行配置；对于高度与宽度较大的特征图，我们使用较大的高度  $L$  来捕捉更大比例尺的信息；在最后两个 GRSU-L 中，特征图的尺寸相对较小，因此不再进行下采样以免造成有效信息丢失。

### 2. 解码器：

解码器与编码器具有相似的结构，并且每个 stage 将前一级的更新采样特征映射与对称编码器的 stage 的特征映射拼接起来作为输入。

### 3. 显著映射融合模块：

显著映射融合模块用于生成最终显著图，它将解码器的每一个侧输出拼接起来，通过  $1 \times 1$  的卷积得到最终的显著图。

相较于 U<sup>2</sup>Net，我们在每一个 GRSU-L Block 中的每一个卷积层前添加了 GCT 门控通道注意力模块，来获得更好的特征提取能力和对前景主体的关注

---

## 3.6 优化方案

### 3.6.1 Loss 设计

由于本项目只需要区分前景和背景,这就相当于是一个在像素级上的二分类问题,因此在模型构建初期采用的是二分类交叉熵损失函数(Binary Cross Entropy Loss)来对全局的 mask 和 ground truth 进行损失计算,通过计算出来的损失值,经过反向传播来不断优化网络结构模型。但是由于这样的损失函数更多地关注于整张 mask 和 ground truth 的主体,经过试验证实,这使得模型生成的 mask 在边缘上会相对粗糙,于是,本项目提出了使用边缘检测 Loss, IoU\_Loss 全局 Loss 相结合的方式引导模型进行优化。

BCE 计算方式如下:

$$L = -\sum_{i=1}^N (y^{(i)} \log \hat{y}^{(i)} + (1 - y^{(i)}) \log (1 - \hat{y}^{(i)})) \dots\dots\dots (1)$$

#### 1. 边缘 Loss

在边缘检测 Loss 中,首先采用 canny 算子,分别对模型生成的 mask 和 ground truth 进行边缘检测,分别生成出 mask 和 ground truth 的边缘信息,再对这两个边缘信息使用二分类交叉熵损失函数。

#### 2. IoU Loss

IoU loss 可以使网络关注更多的前景信息,在 IoU Loss 中,我们首先计算模型输出的 mask 与 ground truth 之间的 IoU,并用  $1-\text{IoU}$  作为 IoU Loss 的损失值,通过与全局 Loss,边缘 Loss 相加来指导网络的优化。

本项目在训练模型最终的损失函数设计如图 3.4 所示。

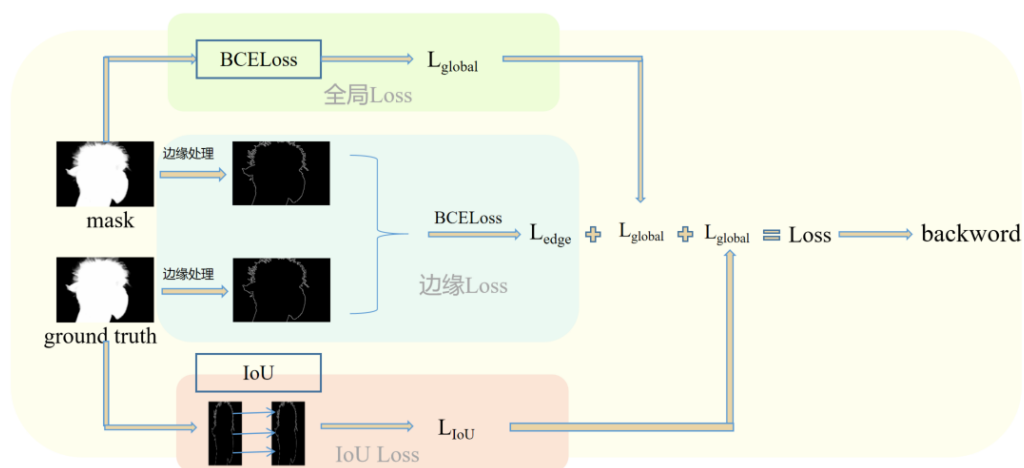


图 3.4 损失函数设计

### 3.6.2 门控注意力机制

《深度学习》一书中解释道：“卷积等效于使用傅里叶变化将输入与核都转换到频域、执行两个信号的逐点相乘，再使用傅里叶逆变换转换回时域。”，即通过卷积可将特征图的信息分解给不同卷积核上的信号分量。考虑到特征图过多的通道数对于关键信息的贡献不同，我们引入了门控通道注意力模块（GCT）。该模块可置于卷积层之前，卷积层与激活函数之间，以及激活函数之后，经过实验，我们最终将该模块置于卷积层之前。

GCT 是一种普遍适用的基于深度卷积神经网络的视觉识别转换单元。该转换用可解释的控制变量明确地为通道关系建模。这些变量决定了神经元的竞争或合作行为，并与卷积权值共同优化，使识别更加准确。GCT 结构如图 3.5：

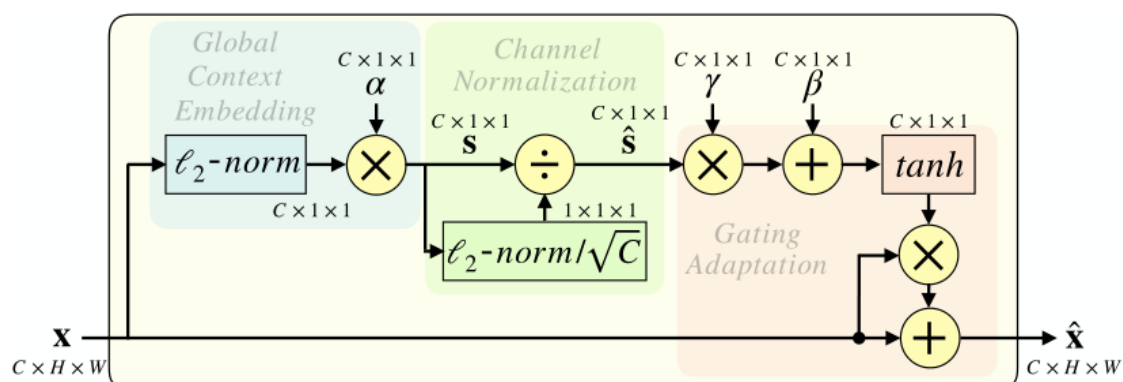


图 3.5 GCT 模块结构图

GCT (Gated Channel Transformation) 主要由三个部分组成：

- 
1. 全局上下文嵌入 (Global Context Embedding): GCT 使用了一个全局上下文嵌入模块来聚合各个通道中的全局上下文信息。该模块可以利用卷积层的小接收域之外的全局上下文信息。不同于经典网络 Squeeze-and-Excitation(SE) Networks 中通道关系由全连接层隐式学习, GCT 使用  $l_2$  规范化来减少参数数量和计算复杂度; 同时引入嵌入权重  $\alpha$ , 负责对嵌入输出进行调整。
  2. 通道归一化 (Channel Normalization): 与全局上下文操作类似, GCT 依旧使用了一个  $l_2$  规范化来进行规范化, 与经典神经网络的全连接层相比时间复杂度更低。
  3. 自适应门控 (Gating Adaptation): 不同于 LRN 只能从神经元的竞争中获益, GCT 将标准化方法与门控机制结合, 可以对不同通道间的竞争与合作关系进行建模。通过引入门控权重  $\gamma$  与门控偏置  $\beta$ , 学习控制门控的激活。当门控权重  $\alpha$  被积极激活时, GCT 鼓励该通道与其他通道合作; 当  $\alpha$  被消极激活时, GCT 鼓励该通道与其他通道竞争。

## 四、项目工程化

### 4.1 推理框架介绍

本作品采用了 `openvino` 推理框架, `OpenVINO` 是英特尔推出的一款全面的工具套件, 用于快速部署应用和解决方案, 支持计算机视觉的 CNN 网络结构超过 150 余种。当模型训练结束后, 上线部署时, 就会遇到各种问题, 比如, 模型性能是否满足线上要求, 模型如何嵌入到原有工程系统, 推理线程的并发路数是否满足, 这些问题决定着投入产出比。只有深入且准确的理解深度学习框架, 才能更好的完成这些任务, 满足上线要求。实际情况是, 新的算法模型和所用框架在不停的变化, 这个时候恨不得工程师什么框架都熟练掌握, 令人失望的是, 这种人才目前是稀缺的。`OpenVINO` 是一个 Pipeline 工具集, 同时可以兼容各种开源框架训练好的模型, 拥有算法模型上线部署的各种能力, 只要掌握了该工具, 你可以轻松的将预训练模型在 Intel 的 CPU 上快速部署起来。

### 4.2 推理基本流程

#### 4.2.1 生成 IE (Infer Engine)

由于本作品的模型训练使用的是 python 的 `pytorch` 深度学习框架, 其模型在 `openvino` 上尚不支持, 但是 `openvino` 支持使用 `onnx` 模型来生成 `InferEngine`,

---

于是经过不断地查找资料。使用 pytorch 的 api, 在设定好输入数据的维度之后, 将其转化成 onnx 模型。接着使用 openvino 工具箱中的 model optimizer 工具, 在设定好模型的输入维度, 数据类型, 生成名称后, 使用工具成功将其转化成 IE (Infer Engine)。

#### 4.2.2 构建推理类

首先构建需要明确推理类需要的属性, 其中包含: 用户选择的图片地址 (用于与前端交互的接口); IE 存放的路径; 生成的 mask 存储位置; 前景分割结果存放的位置, 整个推理流程所需时间。

在定义好上述的 5 个基本属性后, 需要定义一个运行方法, 此方法用于对用户输入的图片, 经过 IE 的推理过程, 并生成推理结果的 mask 和前景分割的结果。这个方法的工作流程主要有以下几个步骤:

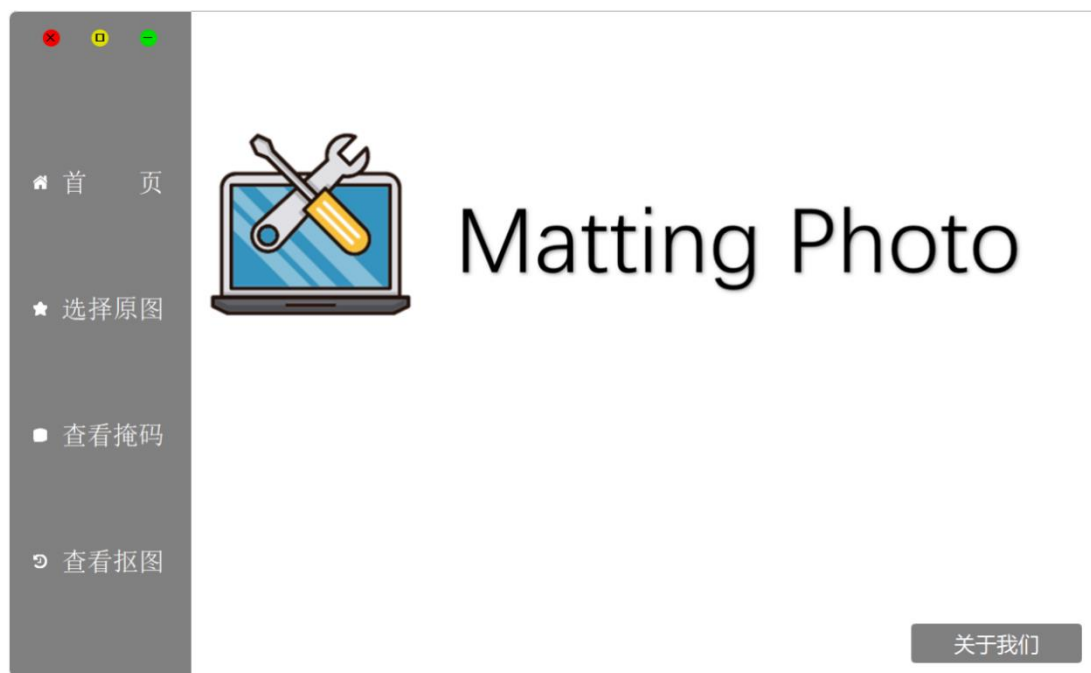
1. 加载 openvino 的 IE 核心库
2. 读取原模型转化成的 Infer Engine。
3. 将 Infer Engine 加载到硬件设备中。(显卡, CPU 等)
4. 获取输入层和输出层的信息。(用于确认输入层应该输入的维度和找到模型多输出结果中应该保留的输出)
5. 图像前处理。(确保图像推理过程中, 与训练过程的图像处理过程一致, 避免因为图像前处理不当造成的推理失败)
6. 将前处理后的图像输入到 Infer Engine 中, 执行推理。
7. 通过输出层信息, 获取最终推理结果, 并对推理图像进行后处理, 生成用户输入图像的 mask。
8. 将用户输入的原图与 mask 结合, 完成图像的前景主体分割。

其中图像前处理, 图像后处理, 以及最后的前景主体分割也分别设计了不同的方法来实现功能。

### 4.3 前端界面设计

本作品从用户的角度出发，考虑了用户的观感与操作性，通过相对简洁的按键说明，使得用户具有良好的体验感和观感。在前端中，用户可以在软件首页获取本团队作品的简易介绍与本团队的联系方式，同时用简洁的图标和简练语言表现了本作品的核心内容。

图 4.1 前端首页



用户首先需要点击选择原图按钮，进入选择要分割的图像界面，在选择好图像后，图像会自适应的限制在前端的最佳位置，若用户不小心点错了图片，可以通过再次点击选择图像实现图像更换。





图 4.2 选择图像页面

用户可以用过点击“选择图像”按钮旁的“开始推理”，对本图片进行前景主体分割。在推理结束后，会有消息框弹出，显示用户选择图像推理成功，并且显示出这张图片的推理时长。



图 4.3 图像推理成功

在推理结束后，用户可以点击查看掩码和查看抠图按钮，对输入图像的 mask 和前景主体分割结果进行查看。与显示原图一致，图像也会通过自适应的方式显示在前端界面的最佳位置。



图 4.4 查看掩码页面



图 4.5 查看抠图页面

用户还可以在不同的页面中选择“导出当前图片的按钮”，软件会自动定位当前图片的是 mask 还是前景主体分割结果，并打开用户的资源管理器，让用户选择图像应该保存的位置，由于前景主题分割结果的背景是透明的，所以建议用户在保存时以.png 的形式保存，否则可能会造成原本图像的透明背景消失。

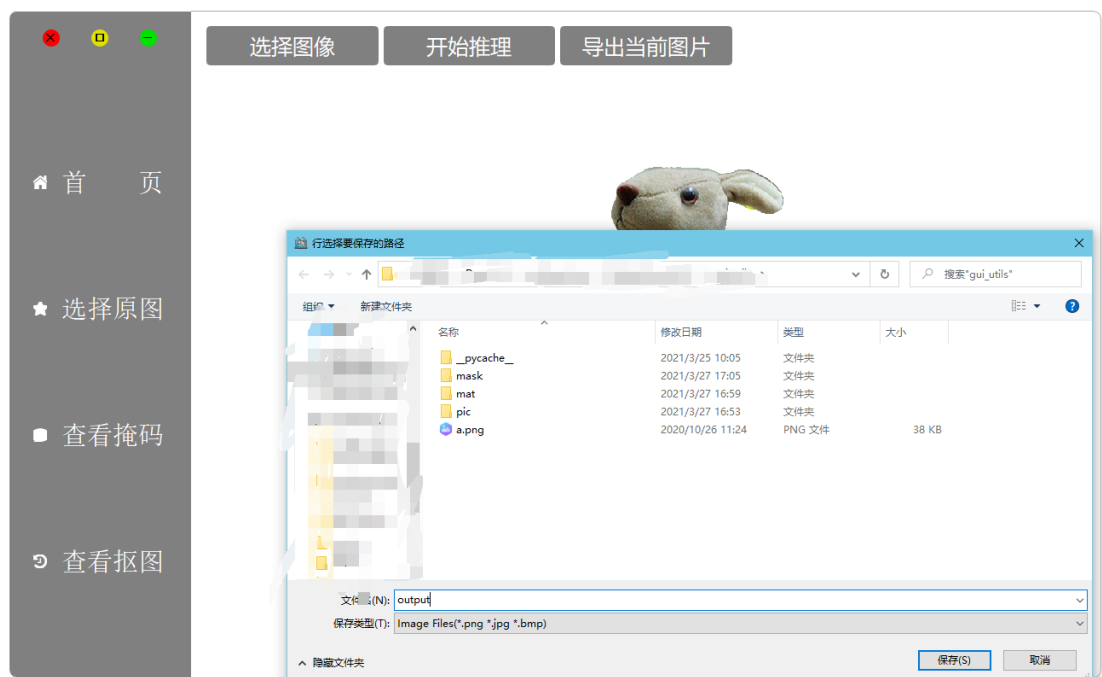


图 4.6 导出当前图片

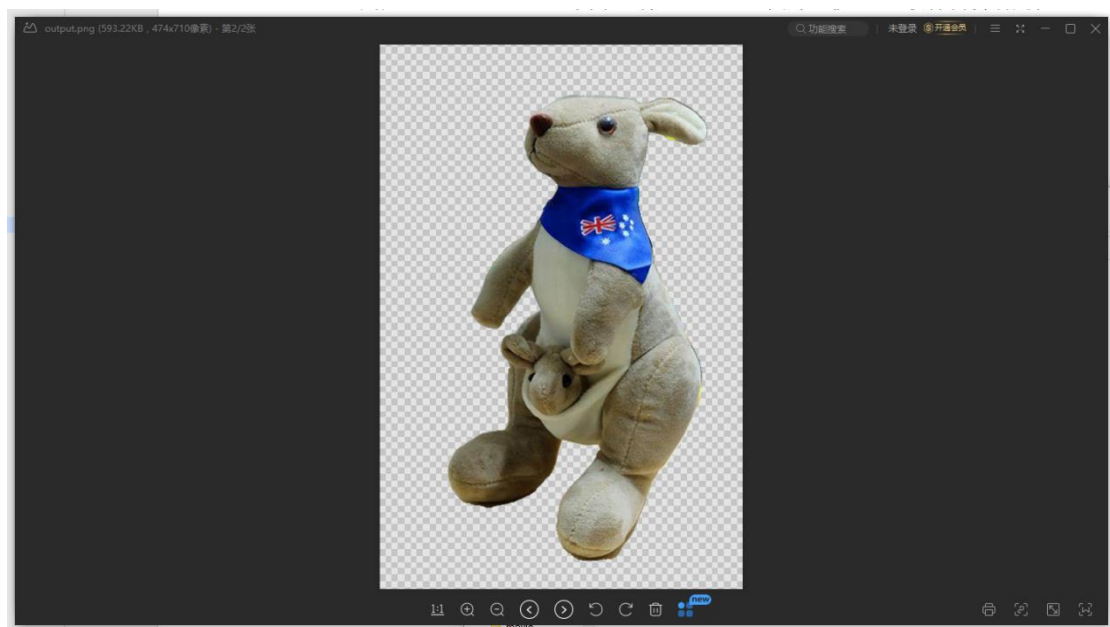


图 4.7 图片导出结果

#### 4.4 前端与推理类的交互

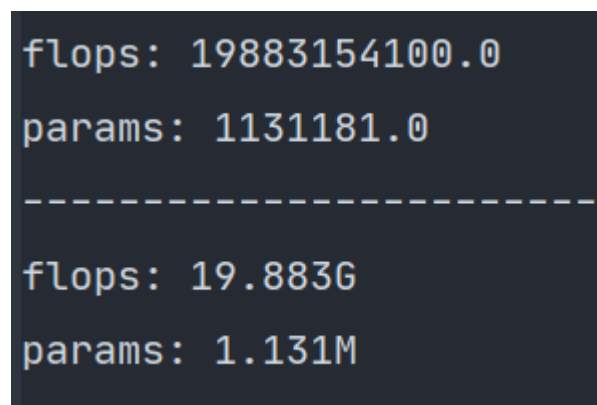
首先在前端启动时，自动创建推理类的实例对象。其中大部分的展示过程全部由前端进行，推理类的实例对象被链接在选择图像页面的开始推理按钮中。点击开始推理按钮后，前端中存储的原图路径会通过实例对象，传递到对象的中的路径属性，这样就能为推理类的实例对象获取到原图。

接着推理开始对原图进行读取，进入推理类的前处理，推理，后处理过程中，最后生成的图片被临时存放在，本作品的指定文件夹中。并将 mask 图像和前景主体分割结果图像的路径传递回前端属性中，当用户点击查看掩码和查看抠图的选项时，便可以直接通过 mask 和前景主体分割的图像地址，向用户展示。

由于 mask 和前景主体分割的图像只是临时存储，因此当用户选择另一张图片时，两张图片就会被删除，可以通过点击导出当前图片的方式对图片进行保存。

### 五、算法模型的优劣

本次使用算法模型的模型大小为 7.03MB，可以支持在硬件设备上运行，移植性较强，同时 7.03MB 的模型也可以给用户减轻软件带来的占用负担。在模型较小的同时，还具备不错的前景主体分割效果。模型的参数量和计算量如图 5.1 所示：



```
flops: 19883154100.0
params: 1131181.0
-----
flops: 19.883G
params: 1.131M
```

图 5.1 参数量(Params)和计算量(Flops)

在 CPU 上对本作品的性能进行测试，测试结果为：图像前处理的平均时间约为:0.18s，图像推理的平均时间约为:0.47s，图像后处理的平均时间约为:0.21s。对于一张图片的处理过程达到了较快的处理速度，约为 1s。测试结果如图 5.2：

```
[INFO]load the information in input
[INFO] get information about the network of input and output
[INFO] preprocess finished!
[INFO] preprocess finish
[INFO] Preprocess work has used the time about 0.1807176999999882 seconds 图像前处理时间
[INFO] begin to infer the input
[INFO] handle the result of infer
infer work has used the time about 0.4690801999999934s 模型推理时间
[INFO] gui_utils/mat/mat_04.png save successful!
[INFO] last work has used the time about 0.20721620000000485 seconds 图像后处理时间
[INFO] This work has used the time about 0.8581847999999894 seconds 总推理时间
```

图 5.2 推理时间测试

同时本算法对透明玻璃杯的处理也有较好的表现，如图 5.2，5.3 所示：



图 5.3 透明玻璃杯处理



图 5.4 液体+玻璃杯图像对比

本算法对多对象主体也有优秀的效果，如图 5.5、图 5.6、图 5.7：



图 5.5 多对象主体（1）





图 5.6 多对象主体 (2)



图 5.7 多对象主体 (3)

发丝与类发丝的对象，也有较好的效果，如图 5.8、图 5.9、图 5.10。



图 5.8 人类发丝



图 5.9 动物毛发 (1)



图 5.10 动物毛发 (2)

在面对分割主体被遮挡时，本算法不会“脑补”遮挡部分，如图 5.11、图 5.12。



图 5.11 主体被遮挡 (1)





---

图 5.12 主体被遮挡 (2)

对于对比度低的图片，本算法效果较为不错，如图 5.13、图 5.14、图 5.15。



图 5.13 对比度较低 (1)



图 5.14 对比度较低 (2)



---

图 5.15 对比度较低 (2)

对于背景简单的主体带有镂空的效果如，图 5.16、图 5.17、图 5.18：



图 5.16 主体镂空 (1)



图 5.17 主体镂空 (2)



图 5.18 主体镂空 (3)

我们的模型还针对低照度图片做了特别优化，如图 5.19 和图 5.20：



图 5.19 低照度优化前的原图和前景

原图经过低照度增强后：



图 5.19 低照度优化后的原图和前景

对于较小的主体，本算法的效果如图 5.20、图 5.21、图 5.22、图 5.23：

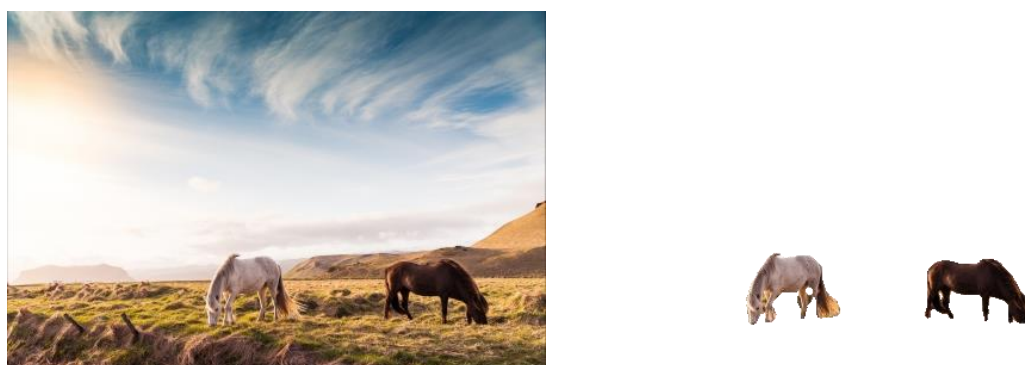


图 5.20 小目标主体（1）



图 5.21 小目标主体 (2)



图 5.22 小目标主体 (3)



图 5.23 小目标主体 (4)

对于 4k 及以上图像，在 cpu（测试 cpu: Intel i5 10210U）上，我们也有不错的速度，如图 5.24、图 5.25、图 5.26



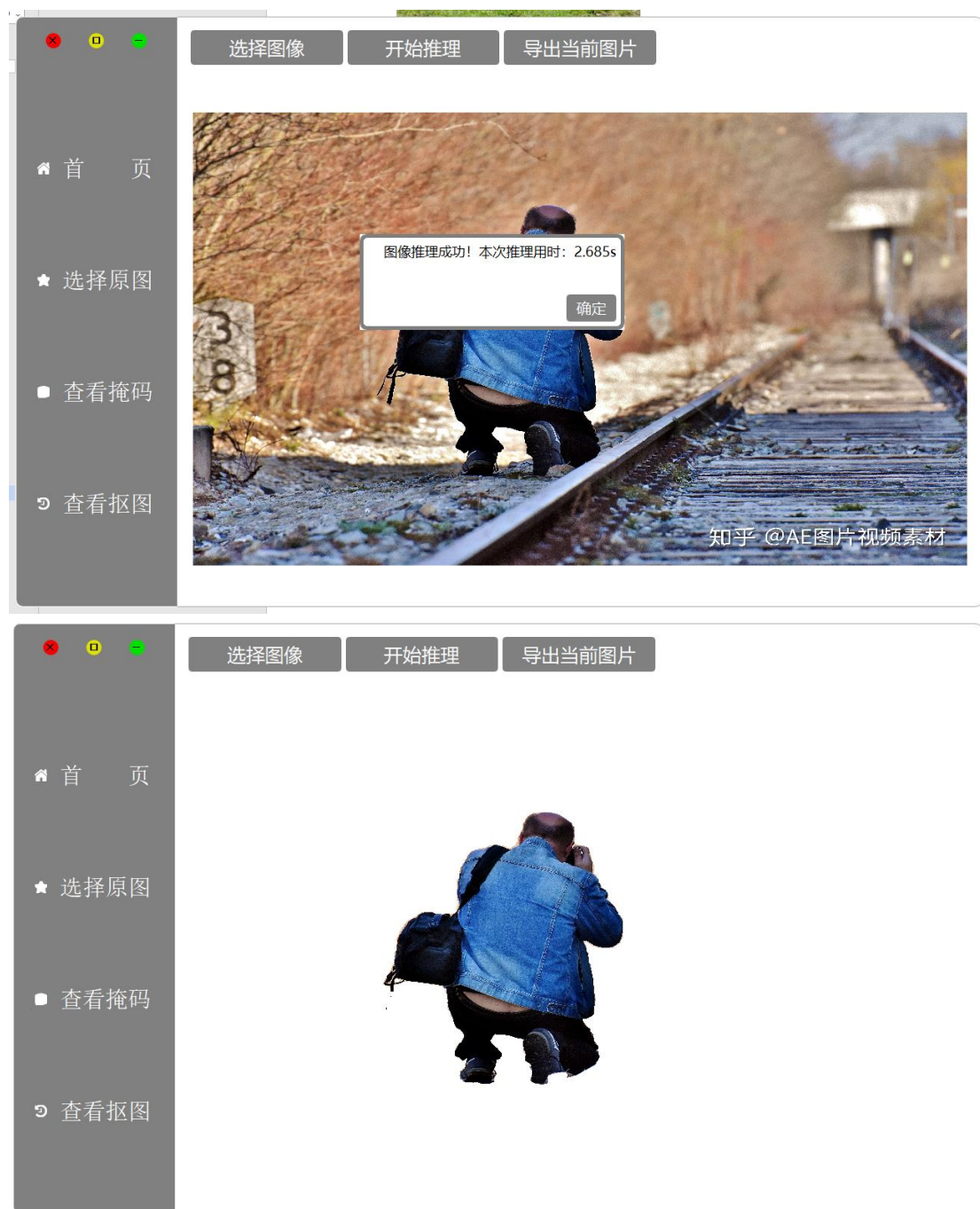


图 5.24 6k 图像 (6000\*4000)



图 5.25 5k 图像 (5616\*3744)

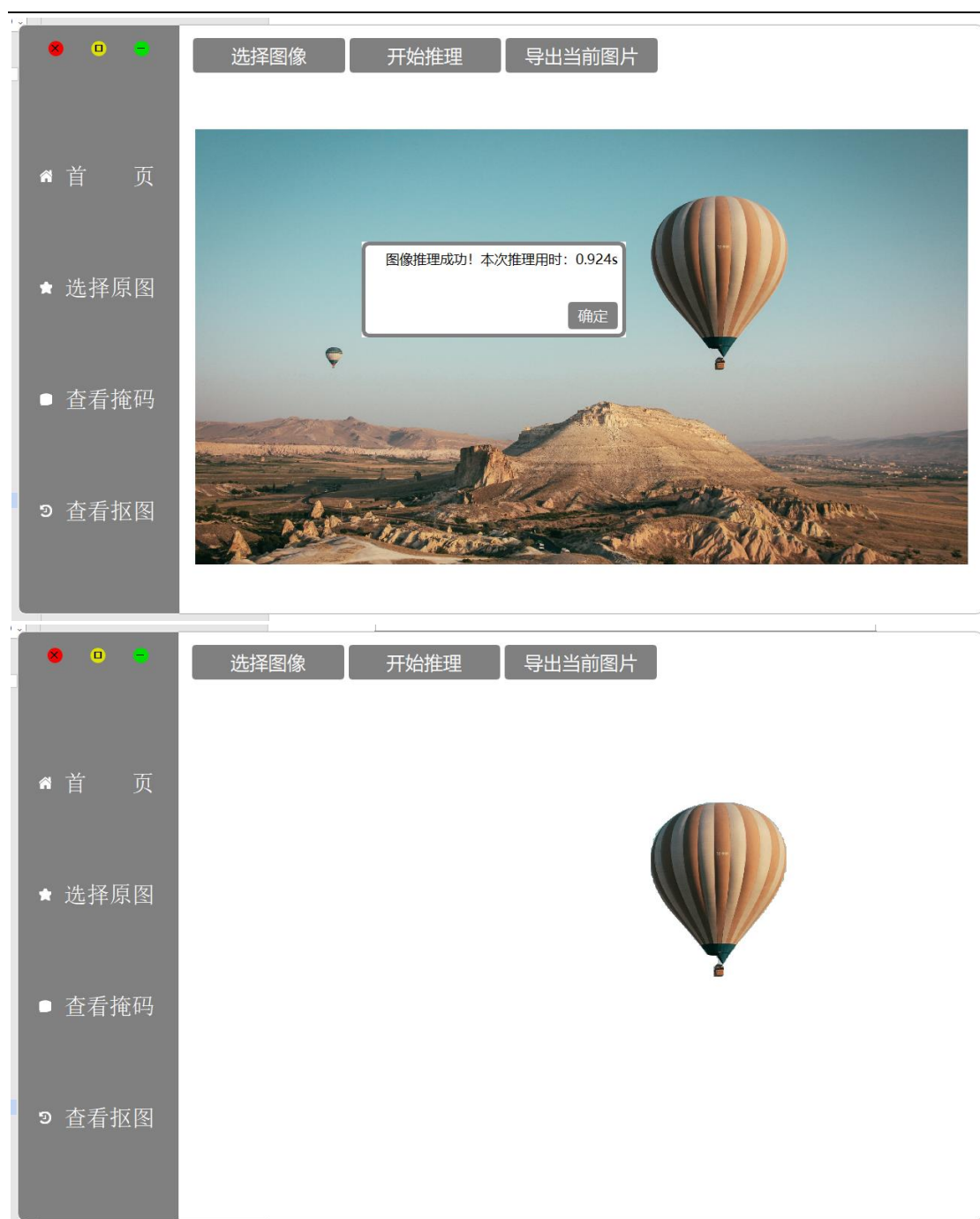


图 5.26 4k 图像 (3840\*2160)

本模型以公开数据集的 DUTS 的测试集作为这个模型的测试数据集，在测试过程中计算了模型的三个指标  $miou$ ,  $F_\beta$ ,  $Mae$  具体数值为见表 5.1，从指标数值上看，本模型在这个数据集上的测试集上的表现极佳。在原本算法提供的模型经过本项目选择的数据集训练后指标计算结果同见表 5.1，对比于  $U^2-Netp$  本项目的指标在每一项上都有所提高。

表 5.1 在 DUTS 上的指标测试结果

模型	Miou	$F_{\beta}$	Mae
U <sup>2</sup> GE-Net	0.794	0.869	0.03
U <sup>2</sup> -Netp	0.737	0.818	0.049

在本团队实验选择的所有非特定类型图像前景主体分割算法中，我们所选择的算法无疑是最优的，但是他仍然存在一定的问题，比如当主体内容与周围景色接近过度，就像动物的保护色一样，对于这样的图片，本算法可能不会有较好的表现；由于推理框架的限制，对于非 Intel 的设备可能会对本软件的兼容性产生一定影响，不过也可以使用不同的推理框架去实现推理过程；同时由于模型转换也会产生一定微小的精度损失，如图 5.27，图 5.28：



图 5.27



图 5.28



---

## 六、展望

在人工智能不断发展的今天，计算机视觉无疑是发展的最快的技术，目标检测，行人检测，图像分割，人脸识别等等，不断被使用在我们的日常生活中。人们因为实例分割对图像分割受到类别的影响的局限性，显著性目标检测（SOD）就不断发展起来，为了解决这个问题，研究人员不断向更深的语义信息，更高效的语义特征融合进发。在本作品中，也选择了本团队觉得最佳的模型，然而他仍然存在不能在颜色较为相近的部分有较好的分割效果，并且对于复杂背景主体带有镂空面积较大或者主体被杂草遮挡较小的图片，不能对边缘和镂空部分进行精细分割。

希望在未来的学习与研究中，能够找到能区分颜色过于相近的主体与背景算法，或处理模块，嵌入到本模型当中，为其实现真正的，比较完善的图像前景主体分割算法。