

基于

搭建

3 网络实现图像语义 分割任务



华为技术有限公司

版权所有 华为技术有限公司 2021。 保留一切权利。

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

商标声明



和其他华为商标均为华为技术有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

注意

您购买的产品、服务或特性等应受华为公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，华为公司对本文档内容不做任何明示或暗示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

华为技术有限公司

地址： 深圳市龙岗区坂田华为总部办公楼 邮编： 518129

网址： [://](#)



目录

1 实验介绍	2
1.2 实验目的	3
1.3 实验清单	3
1.4 实验开发环境	3
1.5 开发平台介绍	3
2 基于 搭建 3 网络实现图像语义分割任务	5
2.1 环境准备	5
2.2 项目代码及数据集下载	6
2.2.1 获取项目代码	6
2.2.2 项目代码文件说明	6
2.2.3 获取 数据集和语义边界数据集（ ， ）	8
2.2.4 准备 模型	11
2.3 模型训练集评估	12
2.3.1 打开项目文件	12
2.3.2 数据准备	13
2.3.3 模型训练	21
2.3.4 模型评估	25
2.4 实验小结	28
2.5 思考题	错误！未定义书签。

1

实验介绍

图像分割是计算机视觉中除了分类和检测外的另一项基本任务，它意味着要将图片根据内容分割成不同的块。相比图像分类和检测，分割是一项更精细的工作，因为需要对每个像素点分类。图像分割可以分为两类：语义分割（ ）和实例分割（ ）。语义分割只是简单地对图像中各个像素点分类，但是实例分割更进一步，需要区分开不同物体，这更加困难，从一定意义上来说，实例分割更像是语义分割加检测。这里我们主要关注语义分割。

图像语义分割（ ），从字面意思上理解就是让计算机根据图像的语义来进行分割，例如让计算机在输入下面左图的情况下，能够输出右图。语义在语音识别中指的是语音的意思，在图像领域，语义指的是图像的内容，对图片意思的理解，比如左图的语义就是三个人骑着三辆自行车；分割的意思是从像素的角度分割出图片中的不同对象，对原图中的每个像素都进行标注，比如右图中粉红色代表人，绿色代表自行车。



图 1-1 语义分割

3 相较于之前的 有很大的改进，在 2012 图像语义分割基准上获得了 - - 的性能（论文参考：[://](#) / /1706 05587）。其关键特点：

- 为了解决多尺度目标的分割问题，串行/并行设计了能够捕捉多尺度上下文的模块，模块中采用不同的空洞率。
- 增强了先前提出的空洞空间金字塔池化（ ）模块，增加了图像级特征来编码全局上下文，使得模块可以在多尺度下探测卷积特征。并在没有 作为后处理的情况下显著提升了性能。

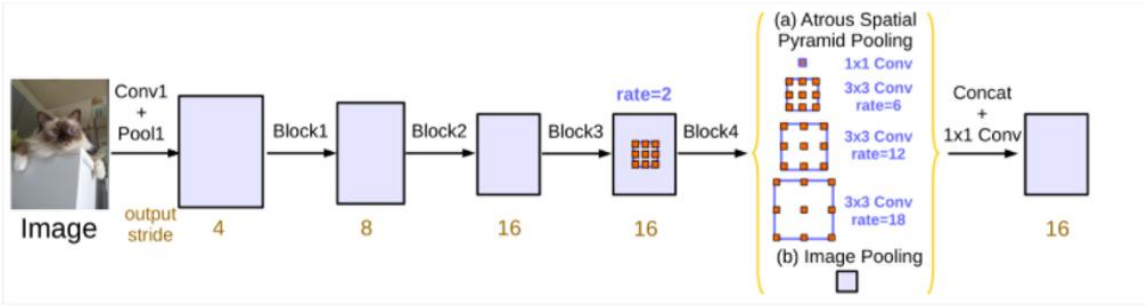


Figure 5. Parallel modules with atrous convolution (ASPP), augmented with image-level features.

图 1-2 3 网络结构

3 使用 作为主干网络。

1 2 实验目的

- 了解语义分割概念。
- 了解 3 网络模型。
- 熟悉并掌握 实现语义分割任务的整体流程。

1 3 实验清单

实验	简述	难度	软件环境	开发环境
基于 搭建 3网络实现 图像语义分割任务	本实验用 2012数据集，实现 在图像语 义分割的应用	中级	3 7 5、 1 5、 3 0 4	机、

1 4 实验开发环境

- 实验平台： 机 运行内存 16 ， 联网；
- 框架： 1 5
- 硬件：

1 5 开发平台介绍

是一种适用于端边云场景的新型开源深度学习训练/推理框架。 提供了友 好的设计和高效的执行，旨在提升数据科学家和算法工程师的开发体验，并为 处理 器提供原生支持，以及软硬件协同优化。

更多参考 [://](#) [/](#)

同时，作为全球开源社区，致力于进一步开发和丰富软硬件应用生态。

提供您在开发所需的一站式开发环境，支持模型开发、算子开发以及应用开发三个主流程中的开发任务，依靠模型可视化、算力测试、本地仿真调试等功能，能够帮助您在一个工具上就能高效便捷地完成应用开发。

采用了插件化扩展机制，开发者可以通过开发插件来扩展已有功能。

更多参考 [://](#) [/](#) [303/](#)

2

基于

搭建

3 网络实现图像语义分割任务

2.1 集成开发环境和 框架准备

该环节操作均在 下进行。

请参考《 环境搭建指南》手册，完成集成开发环境 和 框架 、 相关 依赖库的安装。



MindStudio环境
搭建实验手册.docx

本实验需要电脑连接互联网下安装 到 3.7.5 环境下，请按如下操作进行。

- 进入命令提示符窗口。

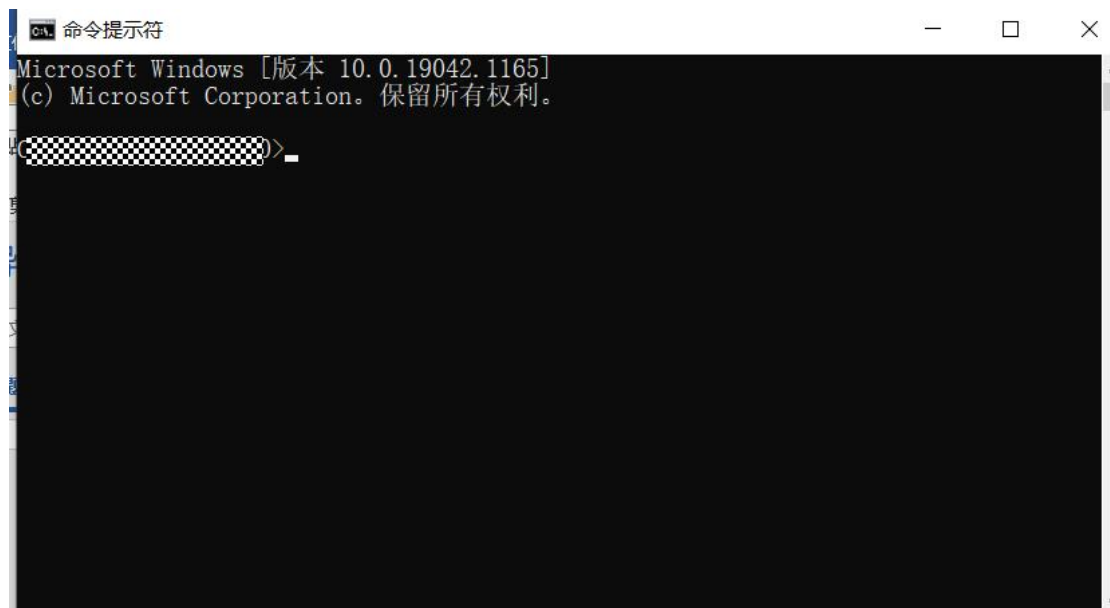


图 2-1 进入命令提示符窗口

- 输入以下命令，安装所需要的包。

```
- - :// /
```

2.2 项目代码及数据集下载

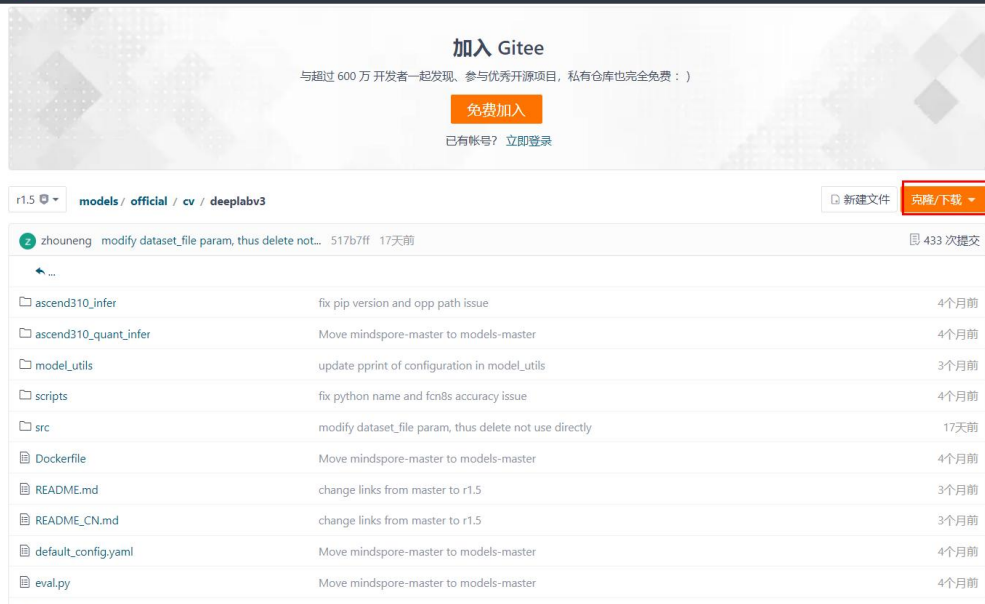
2.2.1 获取项目代码

步骤 1 在本机 盘下，新建文件夹并命名为 目录为 。

步骤 2 下载 1.5 版本的 项目至本机 。

点链接 [://](#) / / / 1.5/ / / 3

下载 1.5 版本的文件，保存并解压缩到 目录下，得到 - 1.5 文件夹。

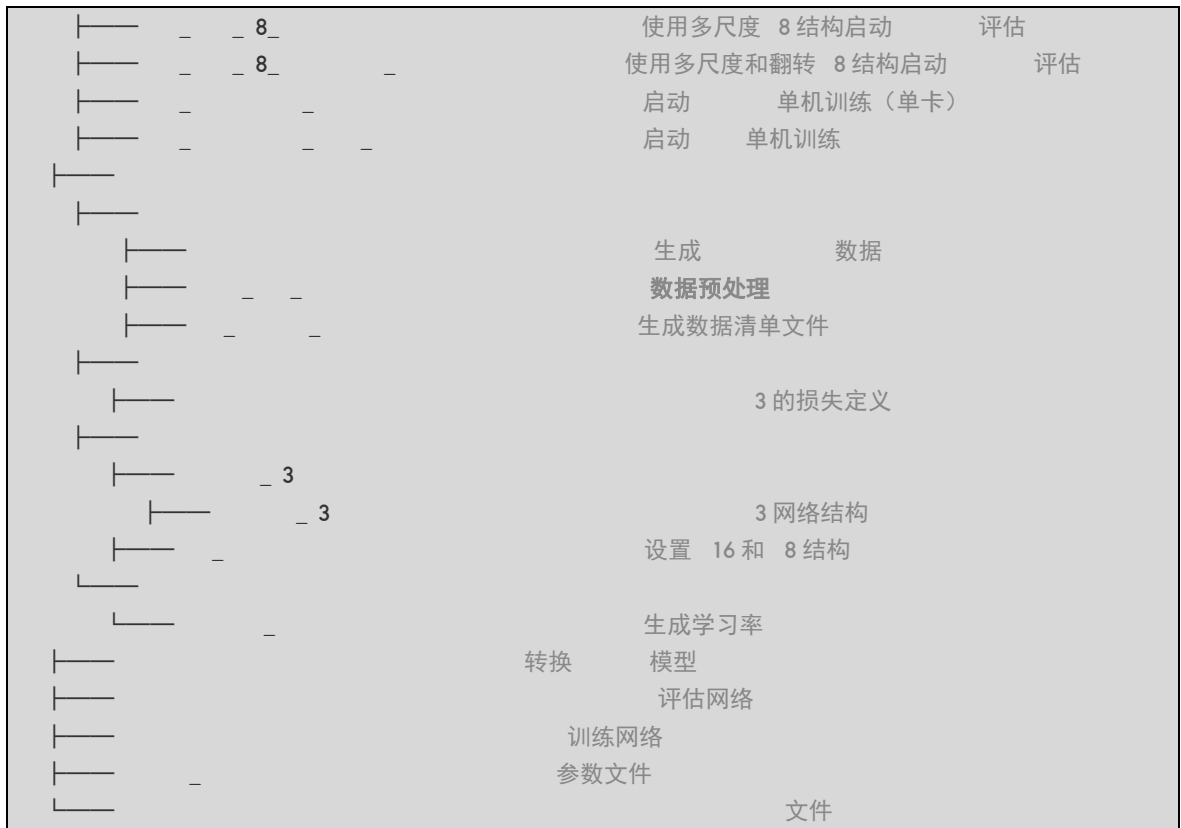


2.2.2 项目代码文件说明

本项目是采用开源项目 位于 [- 1.5/ / / 3](#)，改项目代码涵盖内容较多，如 分布式训练代码、 单卡训练代码、 训练代码，本实验基于 进行训练，所用主要代码均已加粗进行标记。

2.2.2.1 脚本及样例代码

└── 3			
└──			
└──			
└── 将原始数据转换为 数据集			
└── 使用 16 结构的 数据集启动 分布式			
训练 (8 卡)	-	-	16_1
└── 使用 8 结构的 数据集启动 分布式训			
练 (8 卡)	-	-	8_1
└── 使用 8 结构的 数据集启动 分布式			
训练 (8 卡)	-	-	8_2
└── 使用 16 结构启动 评估			
└── 使用 8 结构启动 评估			
└── 8			



2.2.2.2 脚本参数 ()

该文件主要用来进行训练数据路径和超参数的设置，通过修改本文件，可以在一定程度上实现对训练的优化，如增加训练的轮数，学习率的设定等。当然，你也可以采用其他办法实现模型优化的目的。

_: / / /	数据集路径
_:	训练后端类型
_:300	总轮次数
_:32	输入张量的批次大小
_:513	裁剪大小
_:0.08	初始学习率
_:	用于生成学习率的衰减模式
_:0.5	数据增强的最小尺度
_:2.0	数据增强的最大尺度
_:255	忽略标签
_:21	类别数
_: 3_16	选择模型
_: / / /	加载预训练检查点的路径
_:	分布式训练，设置该参数为
_:410	用于保存的迭代间隔
_:	设置该参数 为
_:200	用于保存的最大检查点

2.2.3 获取数据集和语义边界数据集

2.2.3.1 数据集

挑战赛（<http://www.image-net.org/>）是一个世界级的计算机视觉挑战赛，
 全称：ImageNet Challenge，是一个由欧盟资助的网络组织。
 挑战赛主要包括以下几类：图像分类、目标检测、目标分割、行为识别等。

- 图像分类与目标检测任务

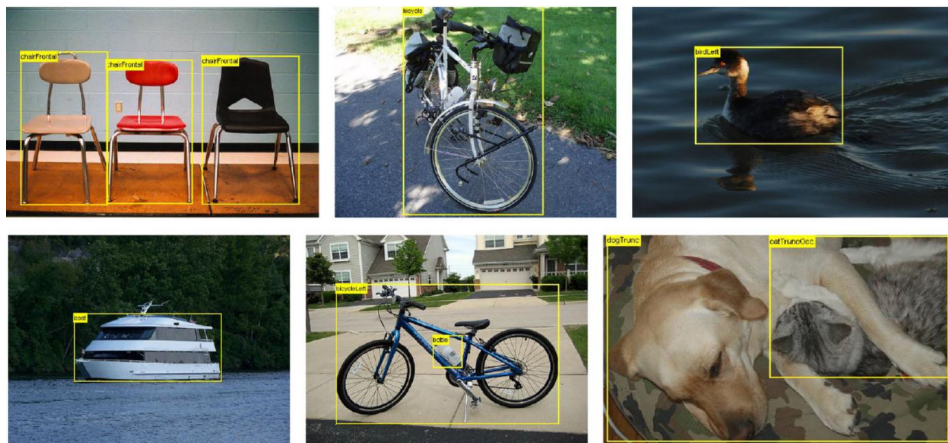


图 2-2 图像分类与目标检测

- 分割任务，注意，图像分割一般包括语义分割、实例分割和全景分割，实例分割是要把每个单独的目标用一种颜色表示（下图中间的图像），而语义分割只是把同一类别的所有目标用同一颜色表示（下图右侧的图片）。

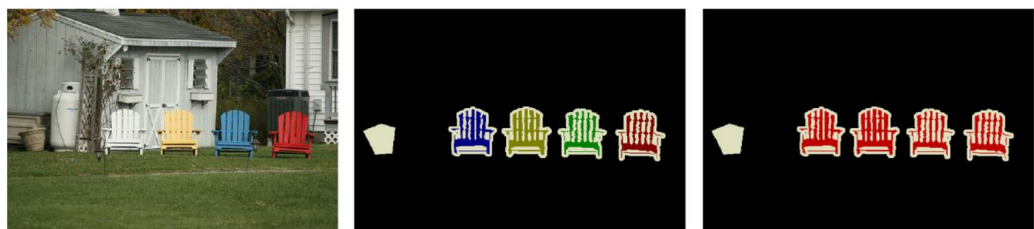


图 2-3 图像分割

- 行为识别任务



图 2-4 行为识别

步骤 2 下载 2012 数据集
<http://www.image-net.org/> / / / 2012/

Development Kit

The development kit consists of the training/validation data, MATLAB code for reading the annotation data, support files, and example

The development kit is now available:

- Download the [training/validation data](#) (2GB tar file)
- Download the [development kit code and documentation](#) (500KB tar file)
- Download the [PDF documentation](#) (500KB PDF)
- Browse the [HTML documentation](#)
- View the [guidelines](#) used for annotating the database (VOC2011)
- View the [action guidelines](#) used for annotating the action task images

步骤 3 把下载后的数据拷贝到 : - 1 5 3 解压，解压后如下



本实验案例是基于 2012/ / 做语义分割任务。
/ , 每一行对应一个图像的索引。根据索引在 文件夹中找到对应的图片。

2007_000032
2007_000039
2007_000063
2007_000068
2007_000121
2007_000170
2007_000241
2007_000243
2007_000250
2007_000256
2007_000333
2007_000363
2007_000364
2007_000392
2007_000480

如 2007_000032 为例，根据索引在

文件夹中可以找到 2007_000323 文件



图 2-5 原图

根据索引在

文件中找到相应的标注图像，2007_000323 文件。

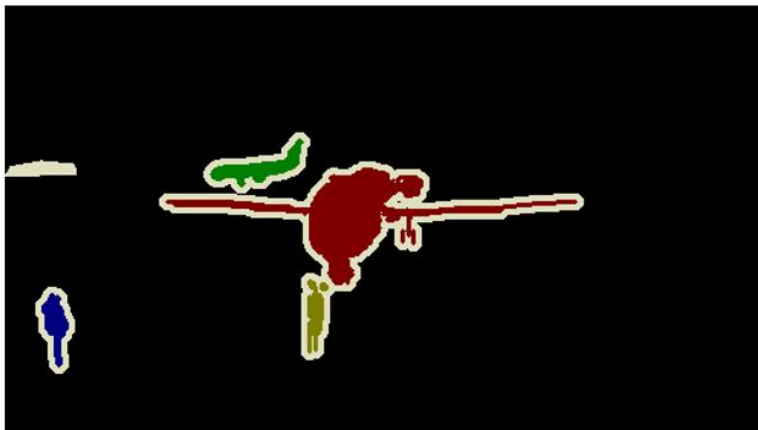
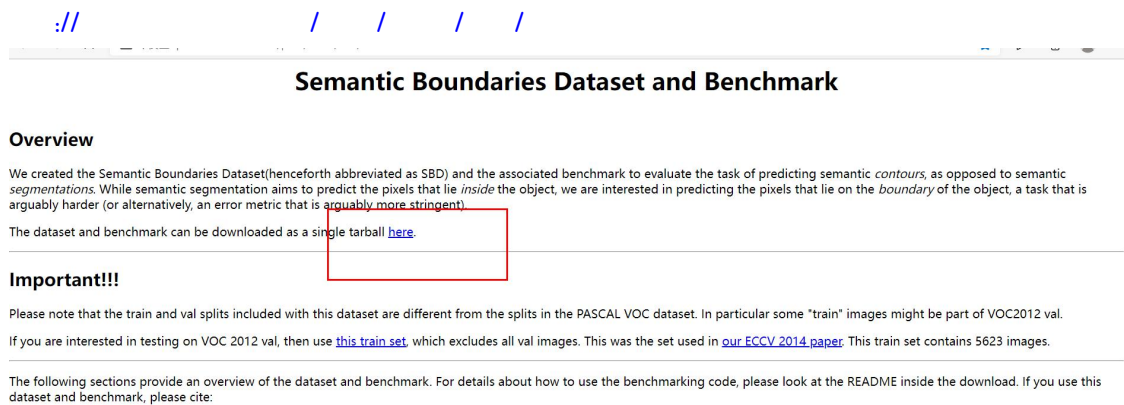


图 2-6 标注图

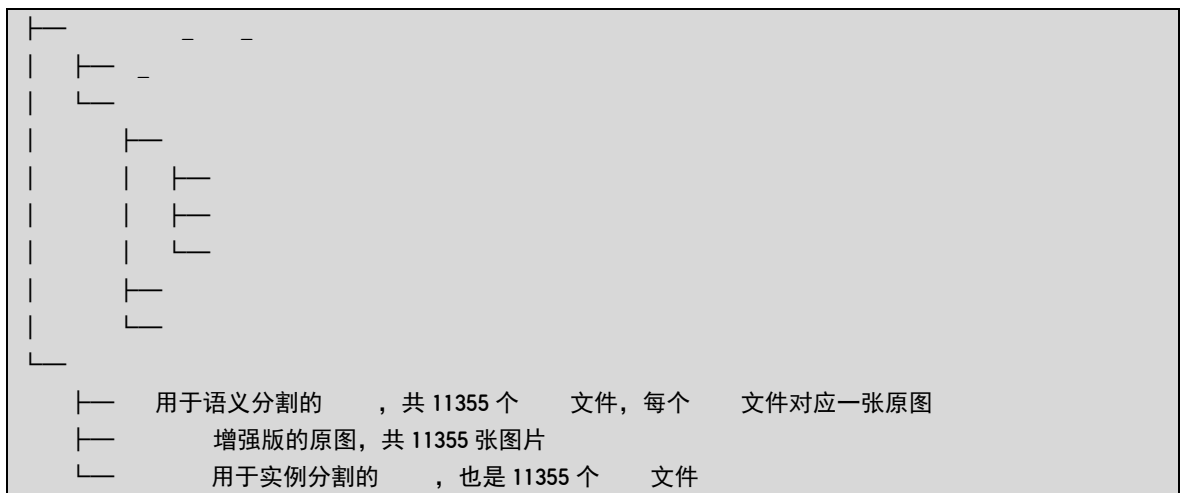
2.2.3.2 语义边界数据集 (Semantic Boundaries Dataset)

2012 数据集的训练或者验证图片的标签图较少，因此我们提供了一个 2012 数据集的增强版，可以综合这两个数据集得到更加多的带标签数据。这个增强版包含来自 2011 数据集的 11355 张图像的注释。这些图像在增强版上进行了注释，分割之间的冲突是手动解决的。对于每张图像，我们提供类别级别和实例级别的分割和边界。提供的分割和边界适用于 2011 挑战中的 20 个对象类别。

步骤 1 下载数据集，点下面链接进行下载



步骤 2 把下载后的数据拷贝到： /data/ - 15 解压 解压后结构如下



大多数实验中使用的数据集是以上两个数据集的融合。

备注：本实验在数据处理时用到了 2012 数据集，生成 2012 数据集，故仍需要下载 2012 数据集，否则会报错。

2.2.4 准备模型

该实验属于 - 训练过程，即对预训练的 3 模型进行微调。

步骤 1 下载预训练模型 准备 101 模型，点下面链接进行下载

步骤 2 在 : - 1 5 3 新建 文件夹, 将下载好的预训练模型拷
到 文件夹下。

2 3 模型训练及评估

2 3 1 打开项目文件

3 项目文件位于 : - 1 5 3 文件夹, 本实验是基于
进行模型训练。

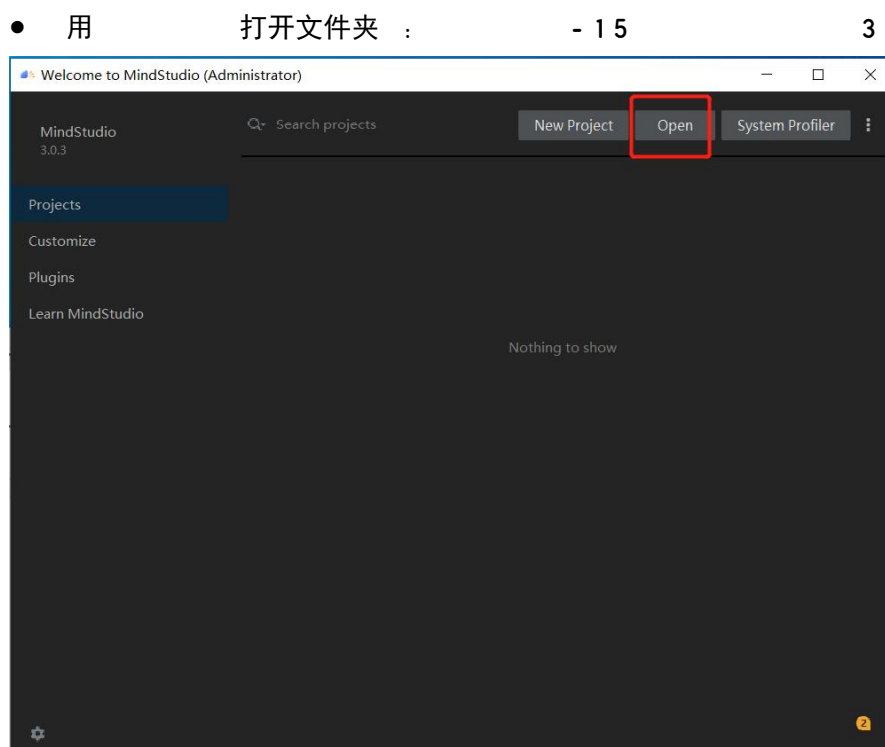
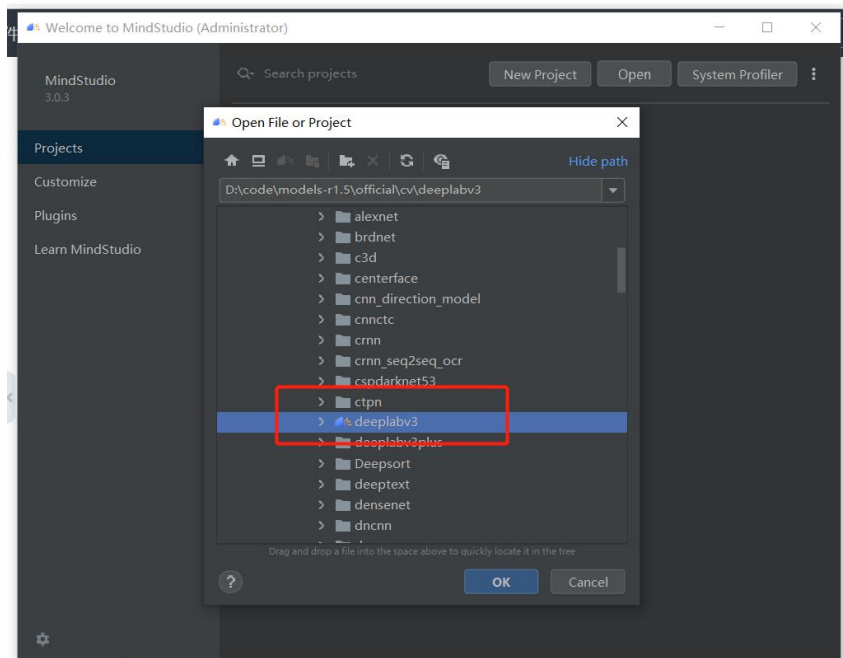


图 2-7 打开



2 3 2 数据准备

该准备工作分两步进行：1 自动生成数据清单文件；2 生成 文件。

2 3 2 1 自动生成数据清单文件

实现语义数据集图像 图转换为灰度图、对数据集划分生成训练集，测试集、增强数据集，按固定格式写入 文件。

打开代码 ： - 1 5 3 _ _

- 配置运行参数 添加运行配置，选择 “ ”。

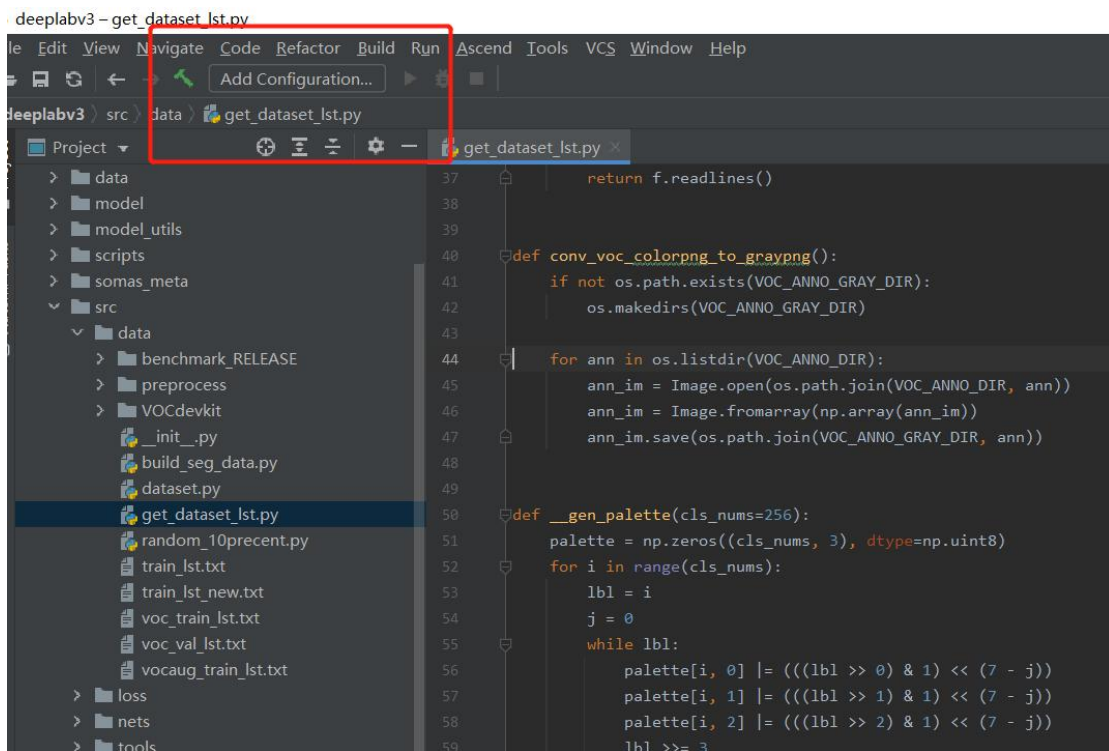


图 2-8

添加运行配置 1

- 进入 “ / ”

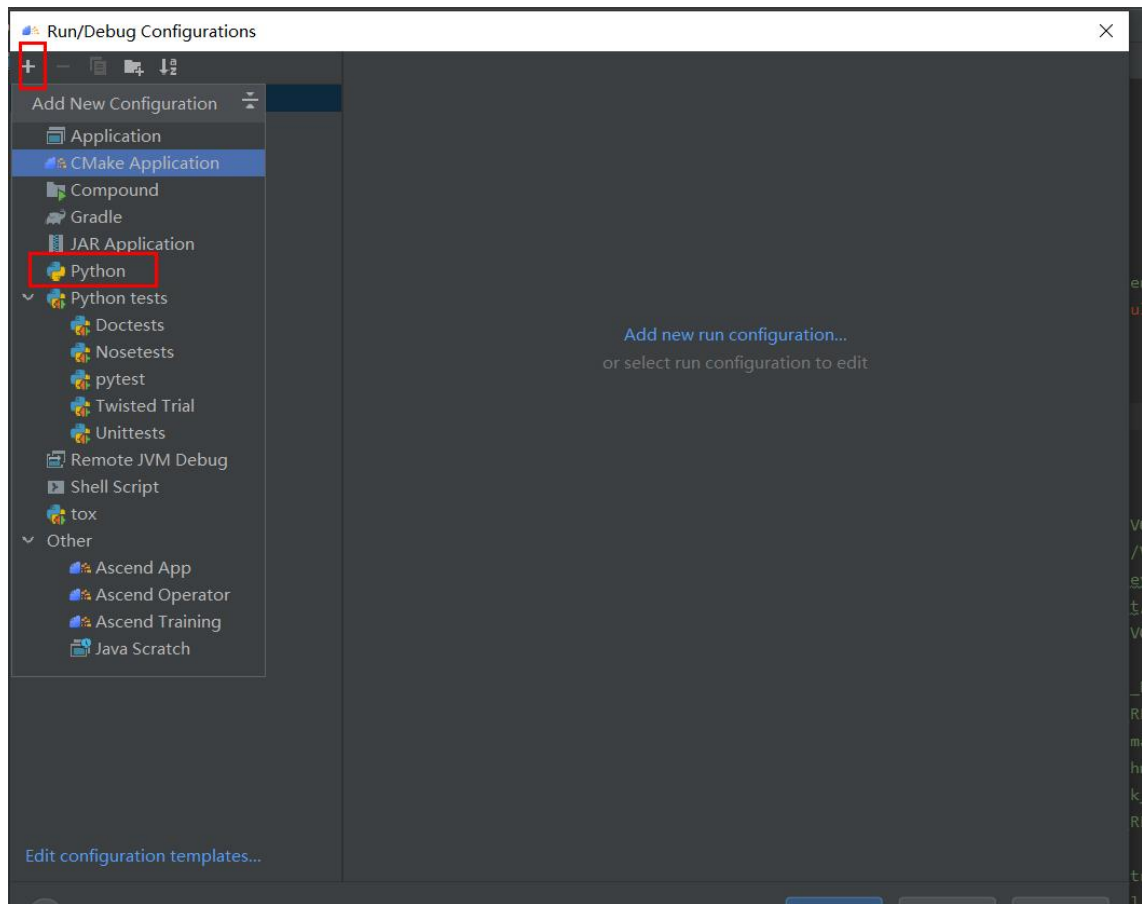


图 2-9 添加运行配置 2

- 添加如下配置，配置完成后，点击“ ”（注意修改脚本和参数文件对应路径）。

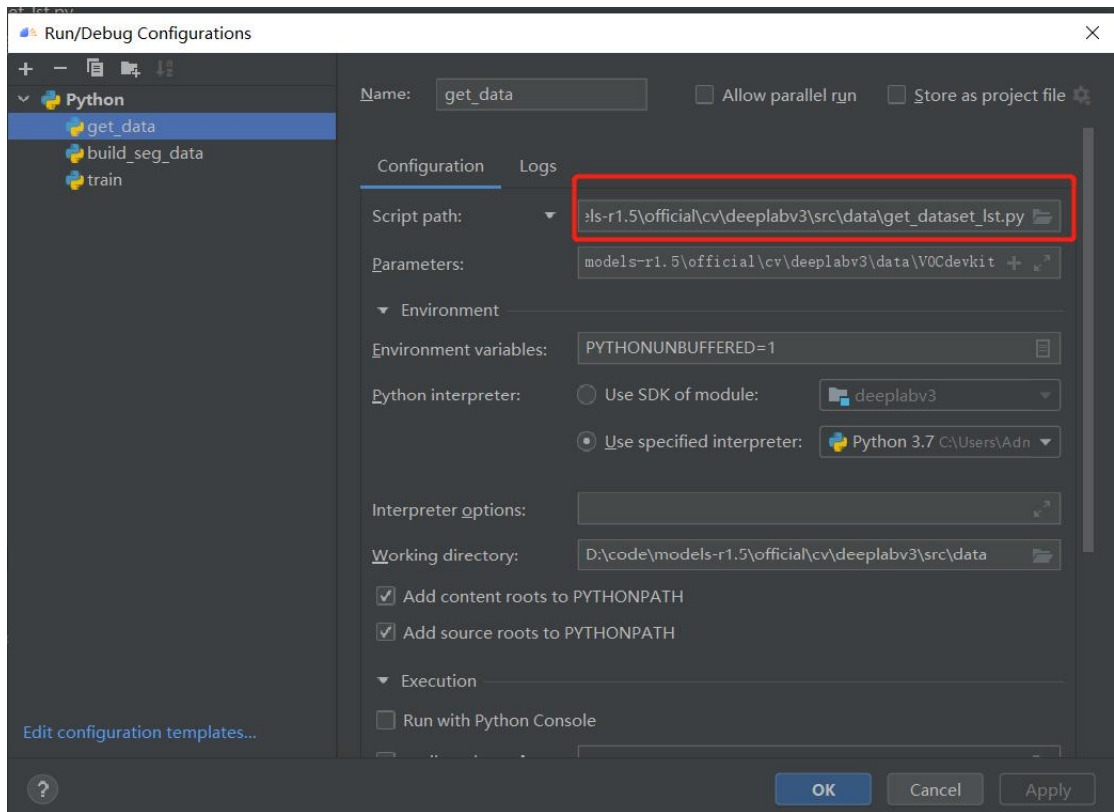


表 2-1 配置表

配置	值
名称	get_data
脚本路径	D:\code\models-r1.5\official\cv\deeplabv3\src\data\get_dataset_lst.py
参数	models-r1.5\official\cv\deeplabv3\data\VOCdevkit
环境变量	PYTHONUNBUFFERED=1
Python 解释器	Python 3.7 C:\Users\Adn
解释器选项	
工作目录	D:\code\models-r1.5\official\cv\deeplabv3\src\data
添加内容根到 PYTHONPATH	<input checked="" type="checkbox"/>
添加源根到 PYTHONPATH	<input checked="" type="checkbox"/>
与 Python 控制台一起运行	<input type="checkbox"/>

- 数据生成

点击运行按钮，在输出窗口查看运行过程和运行结果。

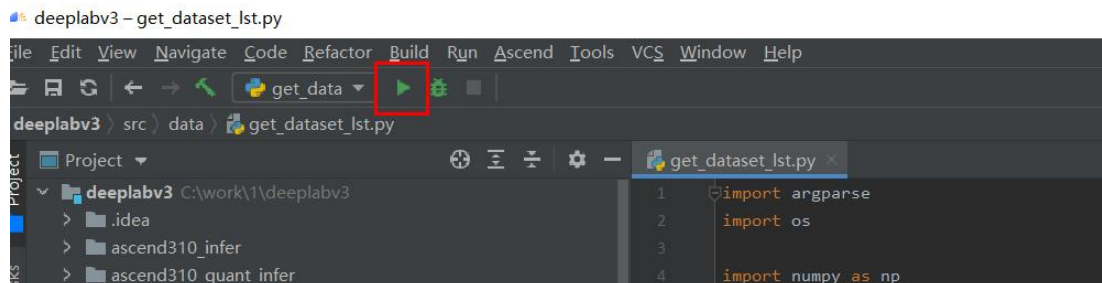


图 2-10 运行按钮

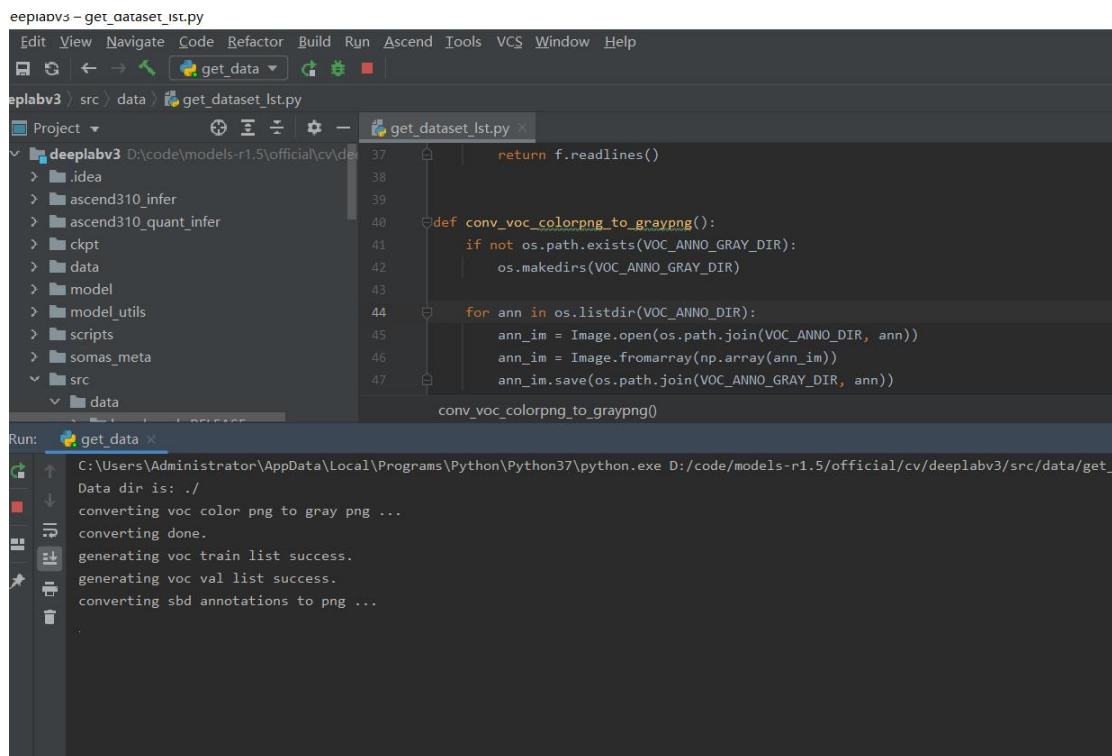


图 2-11 运行状态

- 查看生成文件

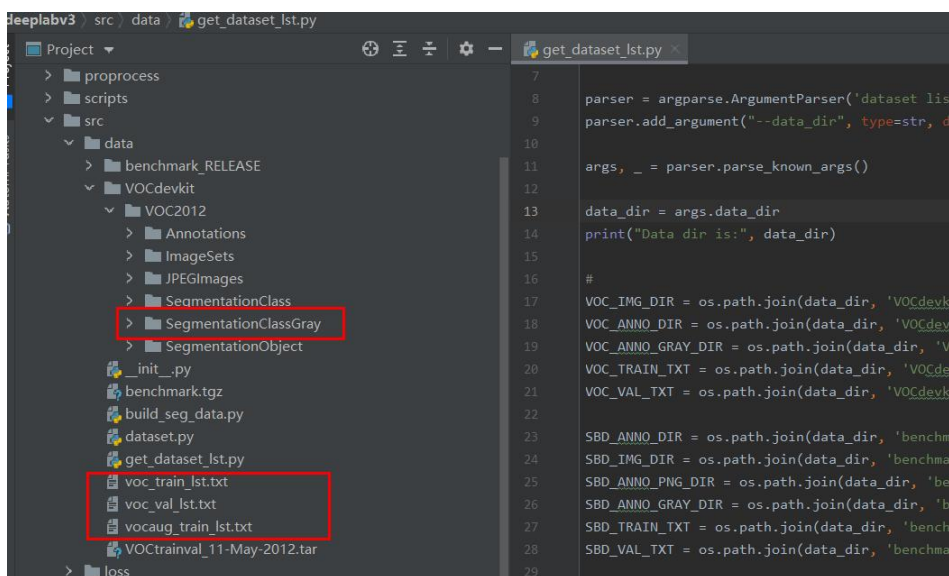


图 2-12 查看生成文件

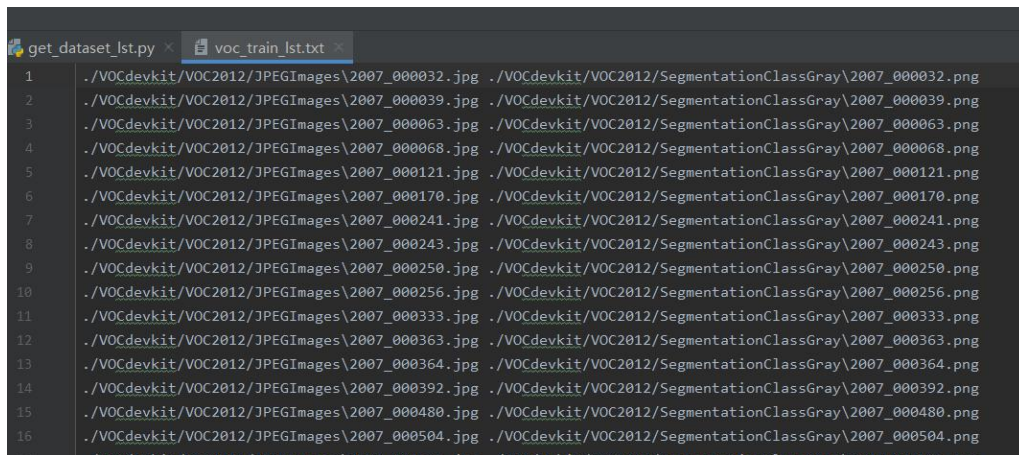


图 2-13 查看 文件

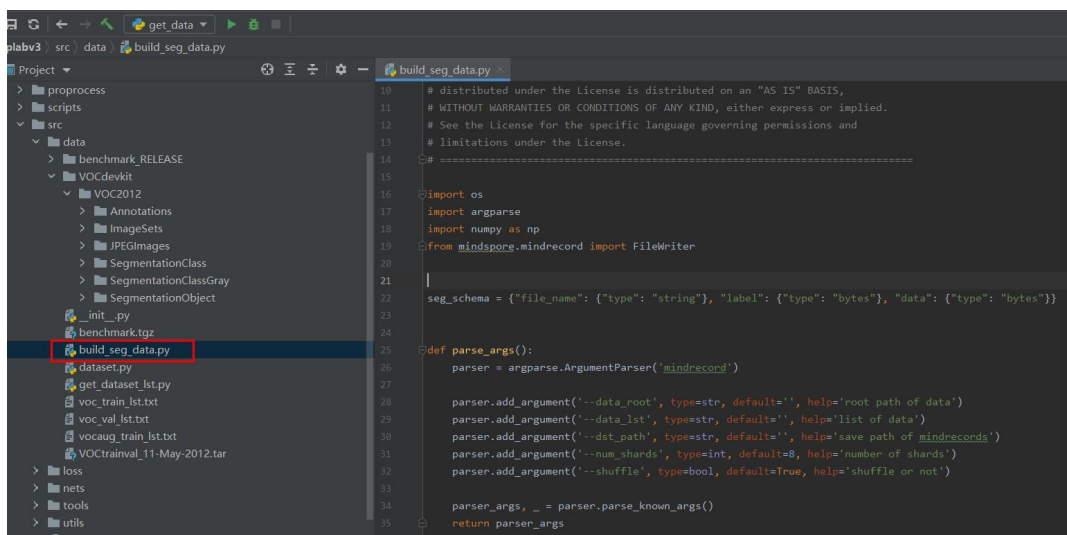
2 3 2 2 将数据集转换为

实现对 2 3 2 1 中生成的数据 转换为 文件。

文件优点：

- 实现多变的用户数据统一存储、访问，训练数据读取更加简便。
- 数据聚合存储，高效读取，且方便管理、移动。
- 高效的数据编解码操作，对用户透明、无感知。
- 可以灵活控制分区的大小，实现分布式训练。

打开 ： - 1 5 3 - -



- 配置运行参数 添加运行配置，选择 “ ”。

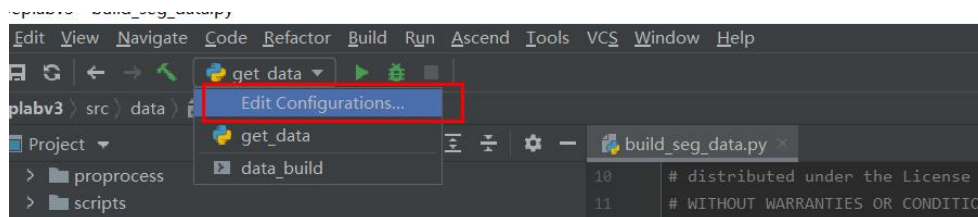


图 2-14 添加运行配置 1

- 进入 “ / ”

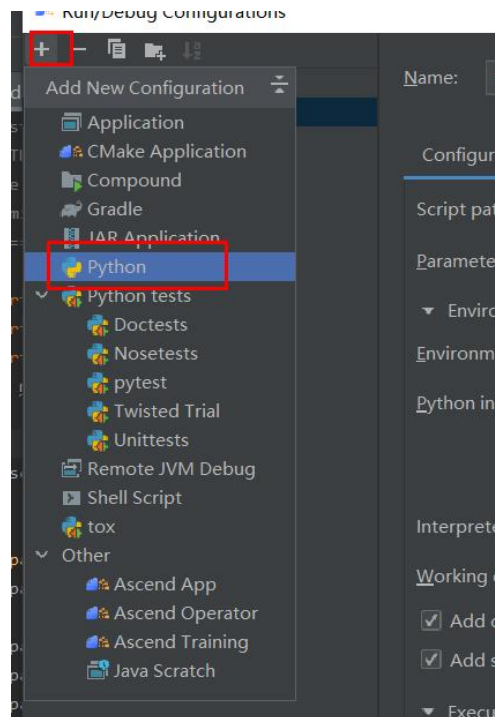


图 2-15 添加运行配置 2

- 添加如下配置，配置完成后，点击 “ ” （注意修改脚本和参数文件对应路径）。

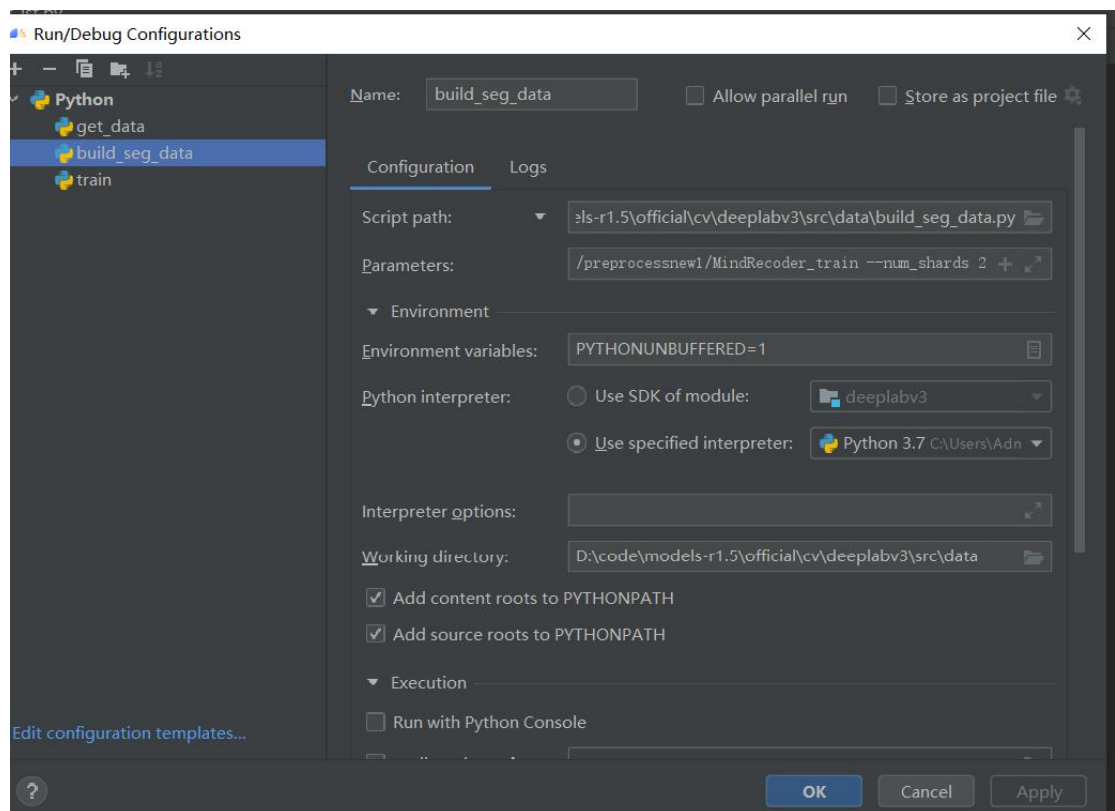


表 2-2 配置表

配置	值
	- -
	: - 1 5 3 - -
	-- - 训练数据的根路径
	-- - / - - 数据清单文件 上步生成的3个
	-- - / / - 存放 所在路
	径
	-- - 8 的分片数
	-- 是否混洗

- 数据生成

点击运行按钮开始运行，在输出窗口查看运行过程和运行结果。

deeplabv3 - build_seg_data.py

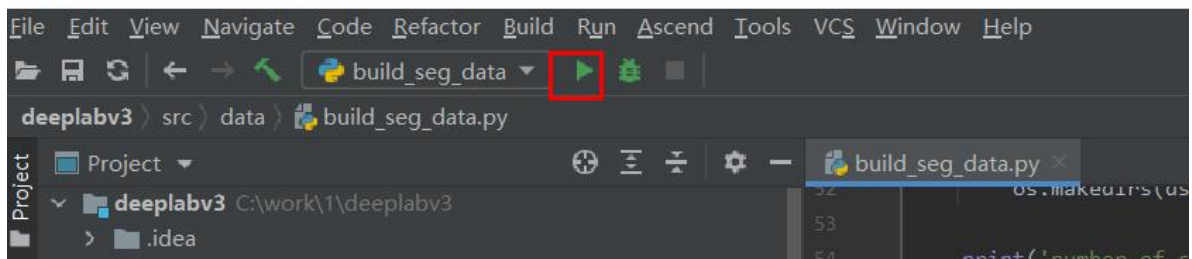


图 2-16 运行按钮

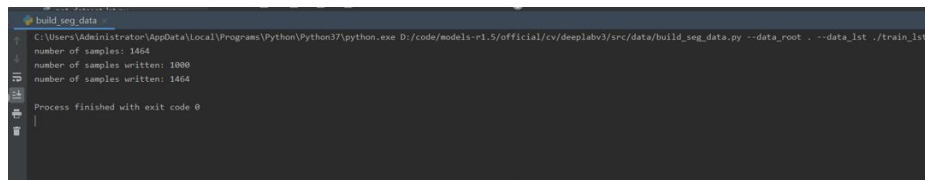


图 2-17 运行完成

- 查看生成文件

工具栏刷新按钮进行刷新下，查看生成文件。

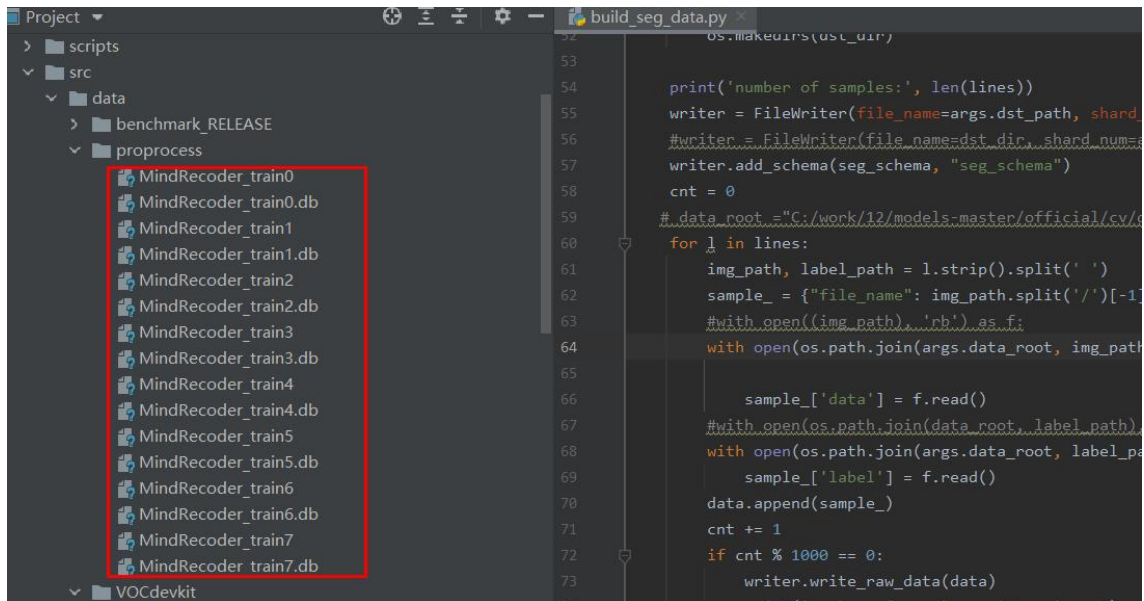


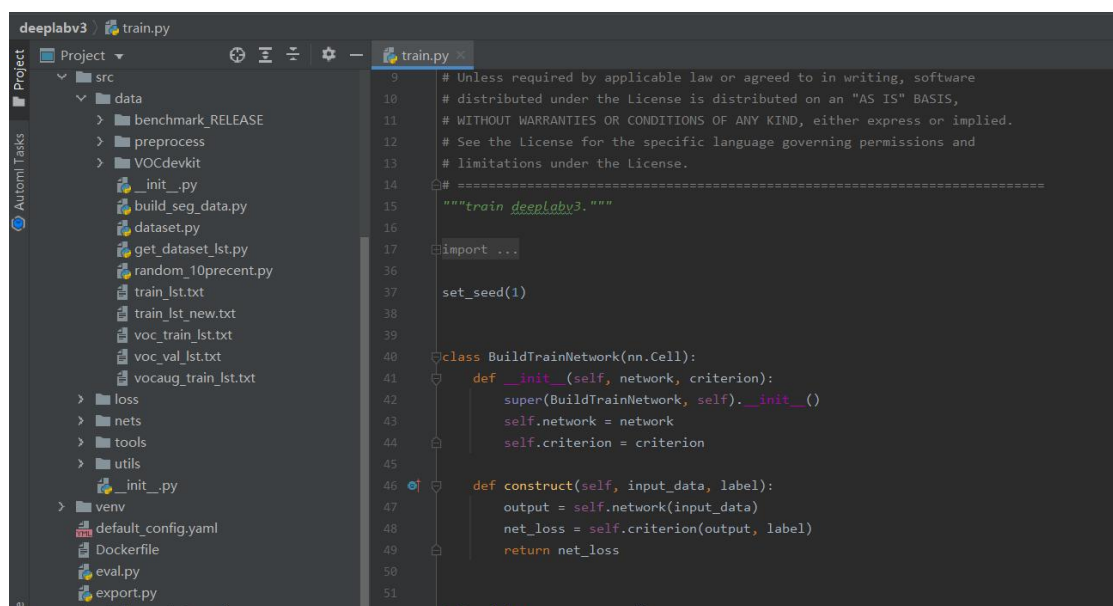
图 2-18 查看生成文件

2 3 3 模型训练

打开：

- 15

3



- 配置运行参数。添加运行配置，选择 “ ”。

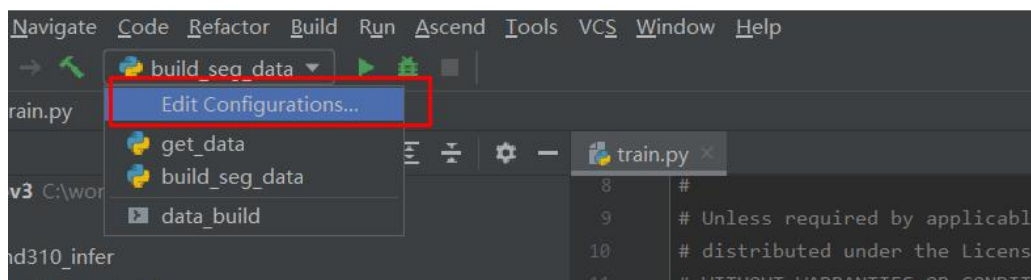


图 2-19 添加运行配置 1

- 进入 “ / ”

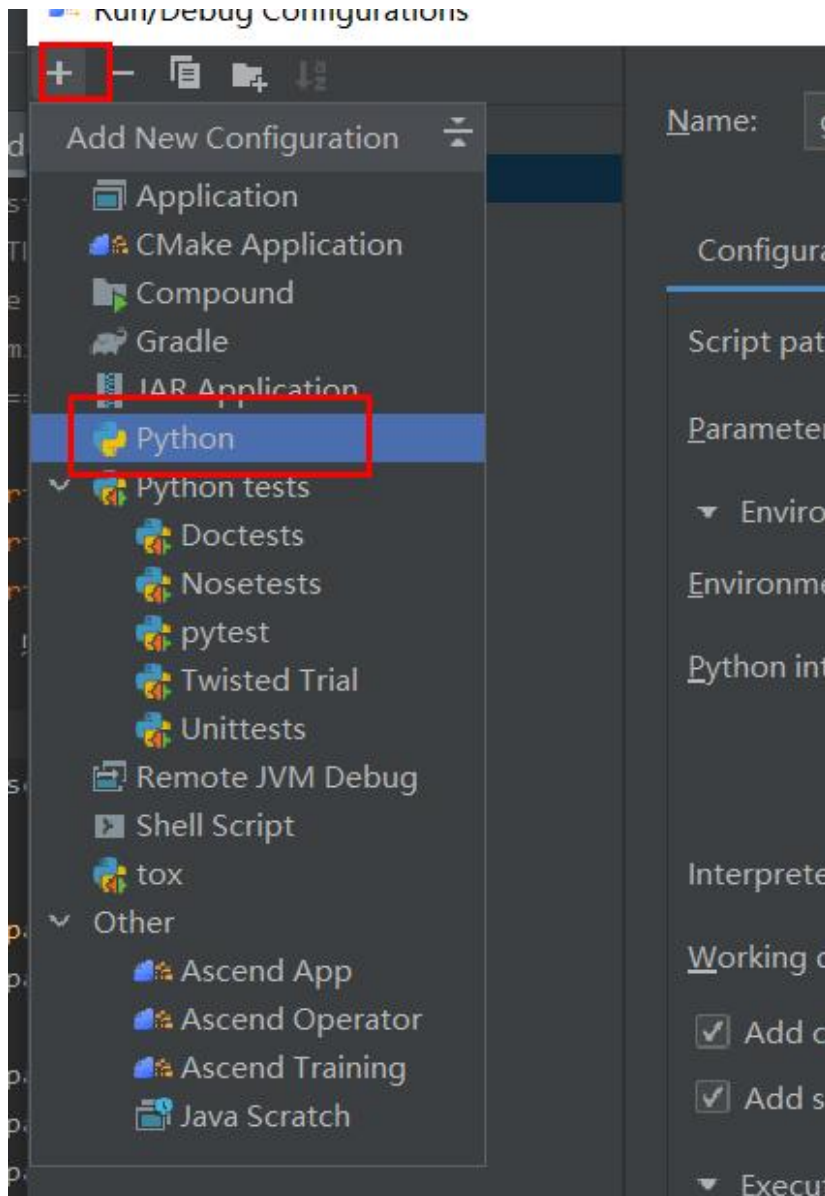


图 2-20 添加运行配置 2

- 添加如下配置，配置完成后，点击 “ ”（注意修改脚本和参数文件对应路径）。

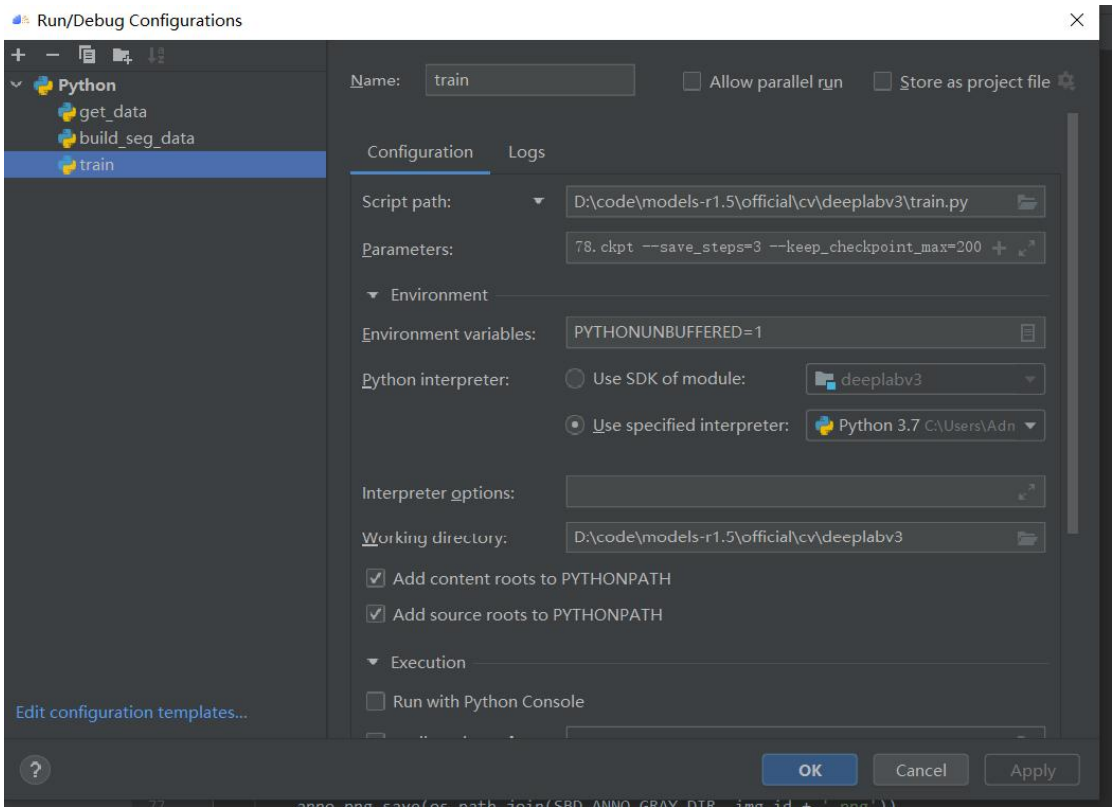
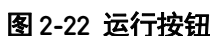


图 2-21 添加配置

表 2-3 配置表

配置	值
	: - 1 5 3
	-- - : - 1 5 3 - 0 修改自己的数据路径
	-- - 使用芯片类型，可以选
	-- - / 存储模型文件路径
	-- - 20 训练轮数
	-- - 32 根据自己电脑配置进行修改，16，8等
	-- - 513 图片尺寸
	-- - 0 015 学习率
	-- -
	-- - 0 5
	-- - 2 0
	-- - 255
	-- - 21 类别数
	-- - _ 3_ 16 模型类型

- 点击运行按钮开始运行，在输出窗口查看运行过程和运行结果。



- ```

[WARNING] ME(19468:19520,MainProcess):2022-01-07 09:57:43.386.588 [mindspore/train/serialization.py:570] network.aspp.conv.aspp_conv.1.beta is not loaded.
[WARNING] ME(19468:19520,MainProcess):2022-01-07 09:57:43.386.588 [mindspore/train/serialization.py:570] network.aspp.aspp_pooling.conv.0.weight is not loaded.
[WARNING] ME(19468:19520,MainProcess):2022-01-07 09:57:43.386.588 [mindspore/train/serialization.py:570] network.aspp.aspp_pooling.conv.1.moving_mean is not loaded.
[WARNING] ME(19468:19520,MainProcess):2022-01-07 09:57:43.386.588 [mindspore/train/serialization.py:570] network.aspp.aspp_pooling.conv.1.moving_variance is not loaded.
[WARNING] ME(19468:19520,MainProcess):2022-01-07 09:57:43.386.588 [mindspore/train/serialization.py:570] network.aspp.aspp_pooling.conv.1.gamma is not loaded.
[WARNING] ME(19468:19520,MainProcess):2022-01-07 09:57:43.402.381 [mindspore/train/serialization.py:570] network.aspp.aspp_pooling.conv.1.beta is not loaded.
[WARNING] ME(19468:19520,MainProcess):2022-01-07 09:57:43.402.381 [mindspore/train/serialization.py:570] network.aspp.conv1.weight is not loaded.
[WARNING] ME(19468:19520,MainProcess):2022-01-07 09:57:43.402.381 [mindspore/train/serialization.py:570] network.aspp.bn1.moving_mean is not loaded.
[WARNING] ME(19468:19520,MainProcess):2022-01-07 09:57:43.402.381 [mindspore/train/serialization.py:570] network.aspp.bn1.moving_variance is not loaded.
load_model ./model/resnet101_ascend_v100_inagenet2021_official_cv_b332_acc78.cpkt success
[WARNING] ME(19468:19520,MainProcess):2022-01-07 09:57:43.402.381 [mindspore/train/serialization.py:570] network.aspp.bn1.gamma is not loaded.
[WARNING] ME(19468:19520,MainProcess):2022-01-07 09:57:43.402.381 [mindspore/train/serialization.py:570] network.aspp.bn1.beta is not loaded.
[WARNING] ME(19468:19520,MainProcess):2022-01-07 09:57:43.402.381 [mindspore/train/serialization.py:570] network.aspp.conv2.weight is not loaded.
[WARNING] ME(19468:19520,MainProcess):2022-01-07 09:57:43.402.880 [mindspore/train/serialization.py:570] network.aspp.conv2.bias is not loaded.

```

图 2-23 训练过程



- 查看生成文件

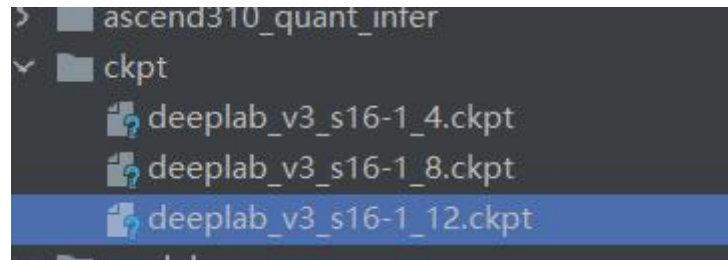


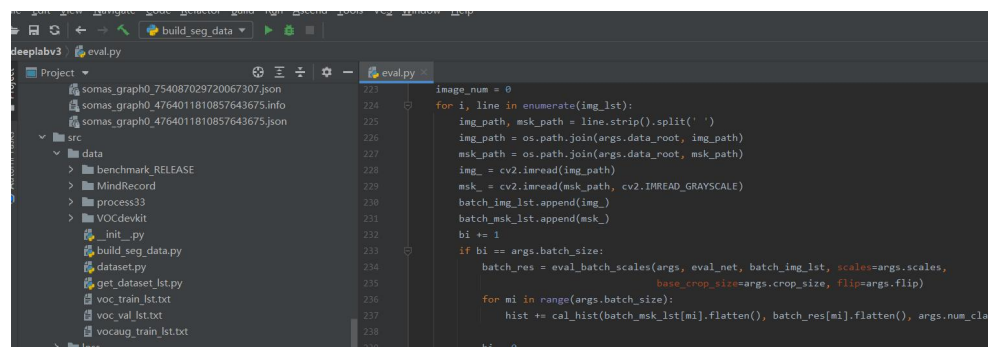
图 2-25 模型文件

## 2 3 4 模型评估

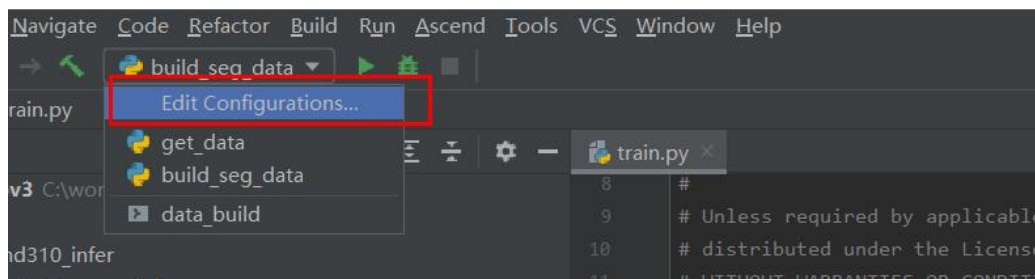
打开文件：

- 1 5

3



- 添加运行配置，选择 “ ”。



- 进入 “ / ”

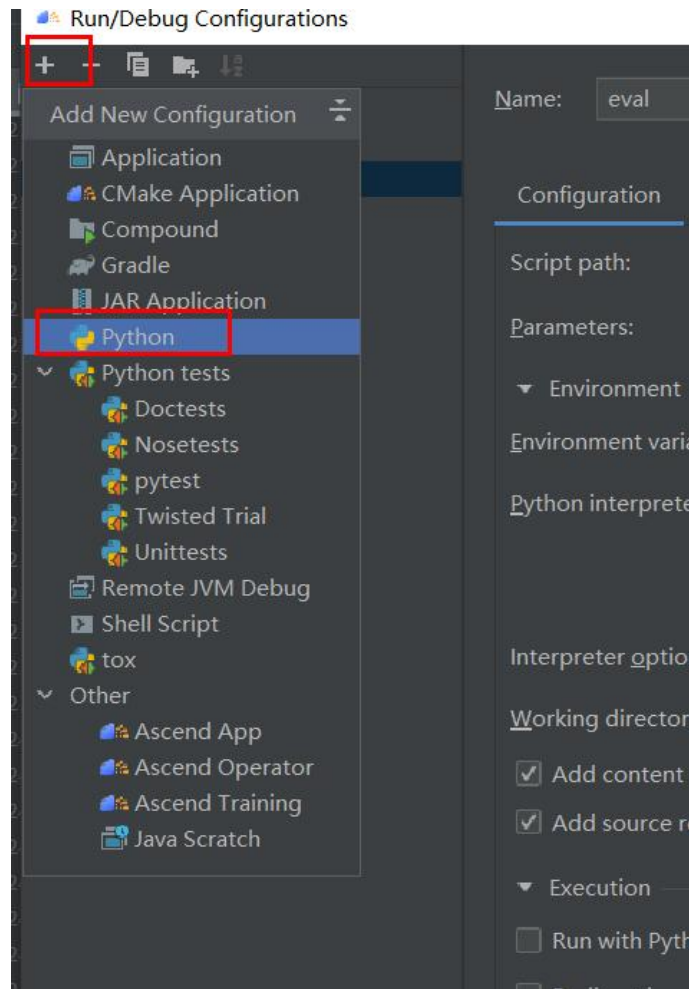


图 2-26 配置界面

- 添加配置，配置完成后，点击 “ ”（注意修改自己的文件路径）。

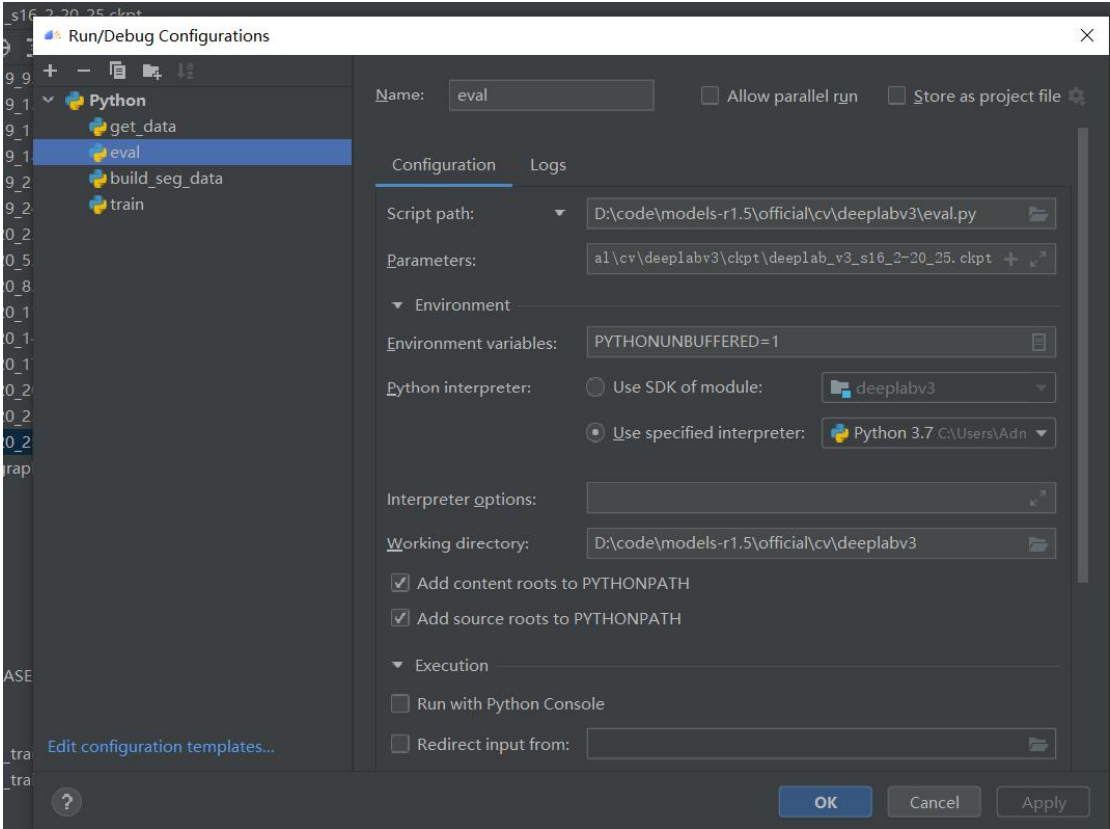


图 2-27 配置界面

表 2-4 配置表

按如下配置填写：

| 配置 | 值                    |
|----|----------------------|
|    |                      |
|    | : - 1 5 3            |
|    | -- _ 数据路径注意修改        |
|    | -- _ : - 1 5 3 _ _   |
|    | 验证数据集路径              |
|    | -- _ 16              |
|    | -- _                 |
|    | -- _ 513             |
|    | -- _ 255             |
|    | -- _ 21              |
|    | -- _ _ 3_ 8          |
|    | -- _ :               |
|    | - 1 5 3 _ 3_ 16-1_12 |

- 点击运行按钮开始，在输出窗口查看评估结果。

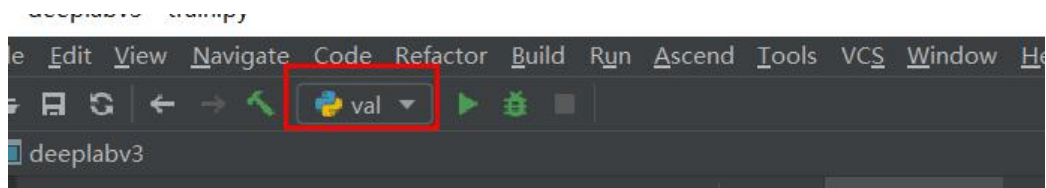


图 2-28 点击

- 等待运行完成

评估语义分割算法性能的标准指标是平均 IoU (Intersection over Union, 交并比)

：模型对某一类别预测结果和真实值的交集与并集的比值。

```
per-class IoU [7.68437585e-01 6.75741353e-06 0.00000000e+00 4.83292261e-05
0.00000000e+00 8.37724013e-05 2.83086935e-01 1.52888961e-01
1.61670649e-01 0.00000000e+00 6.09668144e-04 4.08412601e-05
5.15321817e-03 2.61653012e-04 1.16498575e-01 2.91688347e-01
0.00000000e+00 5.17382768e-07 4.19494182e-05 3.00354935e-03
0.00000000e+00]
mean IoU 0.08492958607079139
```

图 2-29 模型评估

## 2.4 实验小结

本案例使主要运行开源项目工程文件，实现 3 模型对图像语义分割任务，从数据集处理到模型训练和评估整个实验流程，让学生通过本次实验体验数据处理、参数修改，体验在 下模型的训练。