**基于Torchserve的cropunet模型部署**

**主要分成三部分：环境配置和文件准备、模型打包、模型部署和测试**

一、拉取torchserve官方镜像

地址https://hub.docker.com/r/pytorch/torchserve/tags

拉取了最新版本的镜像：docker pull pytorch/torchserve:latest

二、创建容器、准备文档

基于拉取下来的镜像创建容器：

docker run --rm -it pytorhcserve: latest

创建完后自动进入容器，pwd命令查看当前绝对路径:/home/model-server，ls命令查看当前目录文件：“config.properties model-store tmp”

三、打包深度学习模型，生成mar

1. pth权重文件转pt

把模型文件转换为脚本序列化pt文件（pthTopt.py）

1. 准备打包文件

通过docker命令把打包模型所需文档（模型结构unet.py，模型权重Crop\_unet.pt，执行流程modelhandler.py，附属文件vgg.py,resnet.py）从服务器复制到docker容器中，docker cp文件夹在服务器下地址 容器ID：文件夹。自此model-server目录下多了一个cropunet\_torchserve”文件夹

1. 环境配置

在容器内已经有torchserve环境，只需要在安装程序其他依赖包:

Pip install -c opencv-python

做到这一步已经准备好了环境和配置文件。

1. 打包模型

在容器内到cd 文件夹cropunet中生成模型存放文件：

mrdir model-store

torch-model-archiver --model-name cropunet --version 1.0 --model-file unet.py --serialized-file Crop\_unet.pt --handler modelhandler.py --extra-files "vgg.py,resnet.py" --export-path model-store -f

执行完这个语句后会在model-store文件夹内生成一个新的文件，文件名为cropunet.mar。把更改过的容器commit成新镜像：docker commit pytorch/torchserve:1.X。

四、模型启动部署

基于pytorch/torchserve:1.2镜像创建新的容器并设置端口号

docker run --rm -p 8083:8080 -p 8081:8081 -p 8082:8082 -p 7070:7070 -p7071:7071 pytorch/torchserve:1.X

进入容器，在对应目录下启动torchserve服务：

torchserve --start --ncs --model-store cropunet/model-store --models cropunet=cropunet.mar注意：更换模型前一定要删除外面的logs!!!!!

五、模型推理：

1. 在服务器上root下查看状态：

Ping连接状态：curl http://127.0.0.1:8083/ping

模型加载情况：curl http://127.0.0.1:8081/models

1. 在服务器上root下测试图像：

curl -X POST http://127.0.0.1:8083/predictions/cropunet -T 测试图像地址

1. 在本地测试（windows）下测试：

Ping连接状态：curl http://192.168.8.27:8083/ping

模型加载情况：curl http://192.168.8.27:8081/models