

Filter Socket

FilterSocket 软件使用说明书

版本：V1.1

发布日期：2020/5/21

合肥阿巴赛信息科技有限公司

网址: <http://www.abaci3d.cn/>

- - 0 使用之前
 - 0.1 OpenVINO for Windows Detector
 - 0.2 Anaconda环境
 - 0.3 sudo gedit ~/.bashrc
 - 0.4 Conda Env Name
 - 0.5 How to Run
 - 0.5.1 Linux端
 - 0.5.2 Windows端
 - 1 Client 客户端
 - 1.1 图形化界面
 - 1.2 功能说明
 - 1.2.1 深度学习模式
 - 1.2.2 Anaconda环境名
 - 1.2.3 端口
 - 1.2.4 左侧数据流
 - 1.2.5 标定
 - 1.2.5.1 数据集对应的 label 的初始化
 - 1.2.5.2 选择标定数据集
 - 1.2.6 数据管理
 - 1.2.7 训练
 - 1.2.8 测试
 - 2 Server 服务端
 - 2.1 图形化界面

0 使用之前

0.1 OpenVINO for Windows Detector

如果需要使用Windows系统进行测试，请确保在您的Windows系统中安装有 **Intel** 的 **OpenVINO** 软件，该软件在官方网站有详细的安装说明，详见

```
https://docs.openvino toolkit.org/latest/index.html
```

并将

```
openvino\opencv\bin  
openvino\deployment_tools\inference_engine\bin\intel64\Release
```

所对应的路径加入到 **系统环境变量** 的 **Path** 当中

注：如果运行过程中出现问题，可以参照如下步骤更新相关DLL文件

```
运行openvino\deployment_tools\open_model_zoo\demos\build_demos_msvc.bat  
生成Demos.sln  
位于Documents\Intel\OpenVINO\omz_demos_build\Demos.sln
```

```
编译Demos.sln，设置为Release x64，右键ALL_BUILD编译  
获得cpu_extension.lib&&dll  
位于  
Documents\Intel\OpenVINO\omz_demos_build\intel64\Release\cpu_extension.lib&&dll
```

```
依赖的外部dll  
cpu_extension.dll  
位于Documents\Intel\OpenVINO\omz_demos_build\intel64\Release\  
opencv_highgui412.dll  
opencv_videoio412.dll  
opencv_imgcodecs412.dll  
opencv_core412.dll  
opencv_imgproc412.dll  
位于C:\Program Files (x86)\IntelSWTools\openvino_<your version number>\opencv\bin\
```

将上述DLL文件均copy至bin文件夹下，之后 **重新编译** 该项目即可

0.2 Anaconda环境

请确保在您的电脑上安装有Anaconda，并创建了一个conda虚拟环境，该环境需要安装的库包括：

```
os
json
numpy
shutil
pillow
xml
opencv-python
math
codecs
time
random
torch
glob
tensorflow-gpu
scipy
collections
```

以下我们统称这个环境为“pytorch”，您可以根据自己的实际环境名称进行相应的更改

0.3 sudo gedit ~/.bashrc

在ubuntu系统上，需要打开终端，输入

```
sudo gedit ~/.bashrc
```

并在最后一行加上

```
eval "$RUN_AFTER_BASHRC"
```

然后保存并关闭

0.4 Conda Env Name

下方图形化界面中的顶部有一个名为 **Anaconda环境名** 的文本框

请在完成前述中必要的步骤之后，每次运行此程序进行训练前输入您的conda虚拟环境名

```
Anaconda环境名: pytorch
```

0.5 How to Run

0.5.1 Linux端

打开终端，并将目录切换到

```
/FilterSocket_v1.1/Linux/software/
```

之后运行如下两个可执行文件

```
Client
```

以及

```
Server
```

即可运行此程序

0.5.2 Windows端

进入目录

```
/FilterSocket_v1.1/Windows/software/
```

之后运行如下两个可执行文件

```
Client_boxed.exe
```

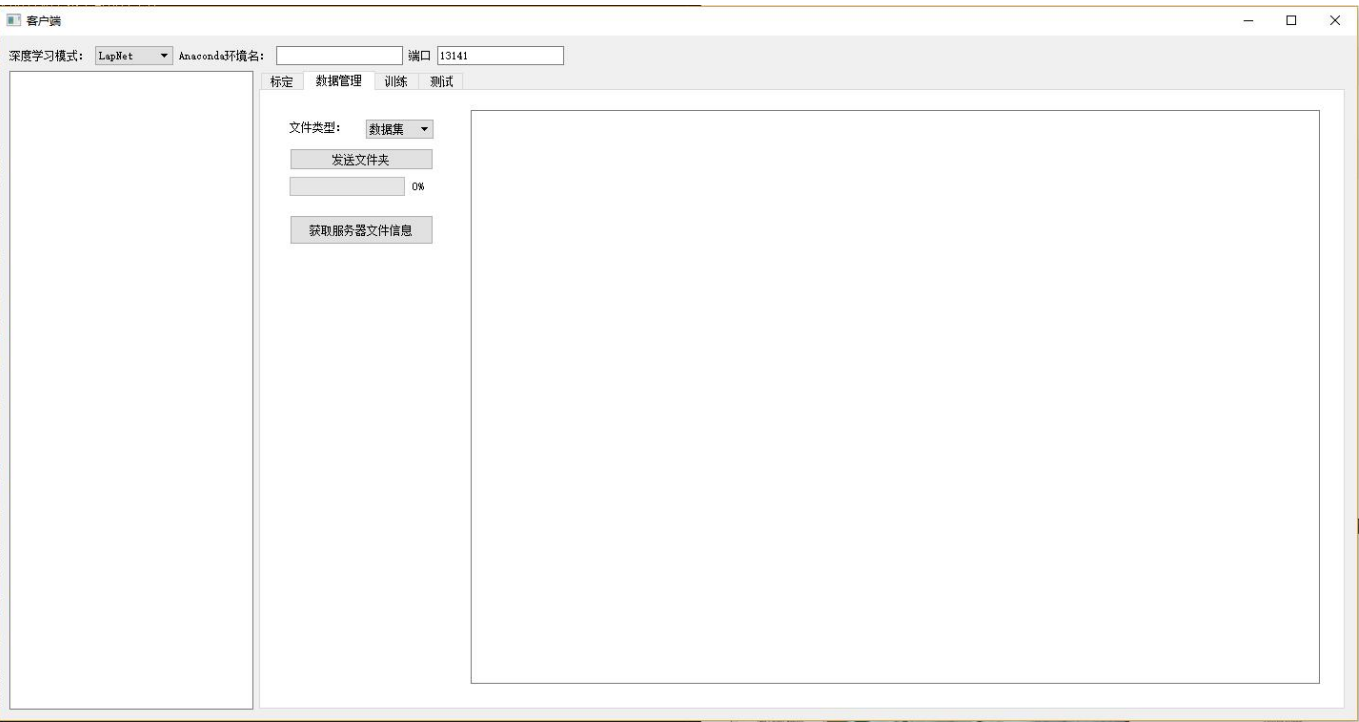
以及

```
Server_boxed.exe
```

即可运行此程序

1 Client 客户端

1.1 图形化界面



1.2 功能说明

1.2.1 深度学习模式

用来指定当前所采用的网络模型与训练策略

1.2.2 Anaconda环境名

需要输入在 0.1 中创建好的Anaconda环境名称，否则某些训练功能可能不能正常运行

1.2.3 端口

用来与服务端的接收端口进行匹配，当前版本默认为13141，不需要修改，否则可能导致通信失败

1.2.4 左侧数据流

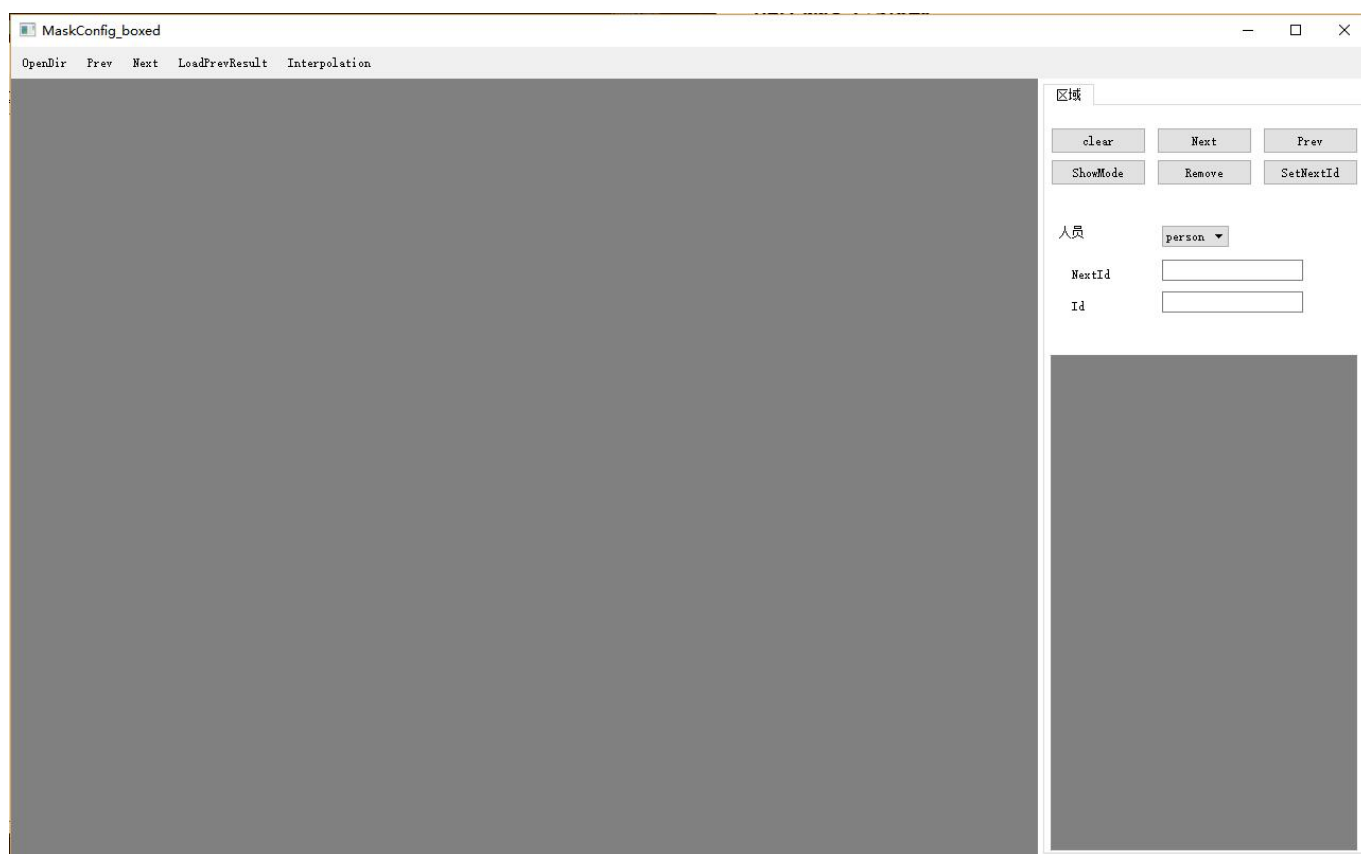
左侧的空白框用来显示当前的连接状态、用时等信息

1.2.5 标定

单击该按钮可以打开标定工具，对现有数据进行标定



标定工具如下图所示



1.2.5.1 数据集对应的 label 的初始化

打开文件

```
/FilterSocket_v1.1/Windows/software/config/config.json
```


根据需要仿照第 21,22 行修改/增加/删除，直到修改为对应当前待标定数据集的 label

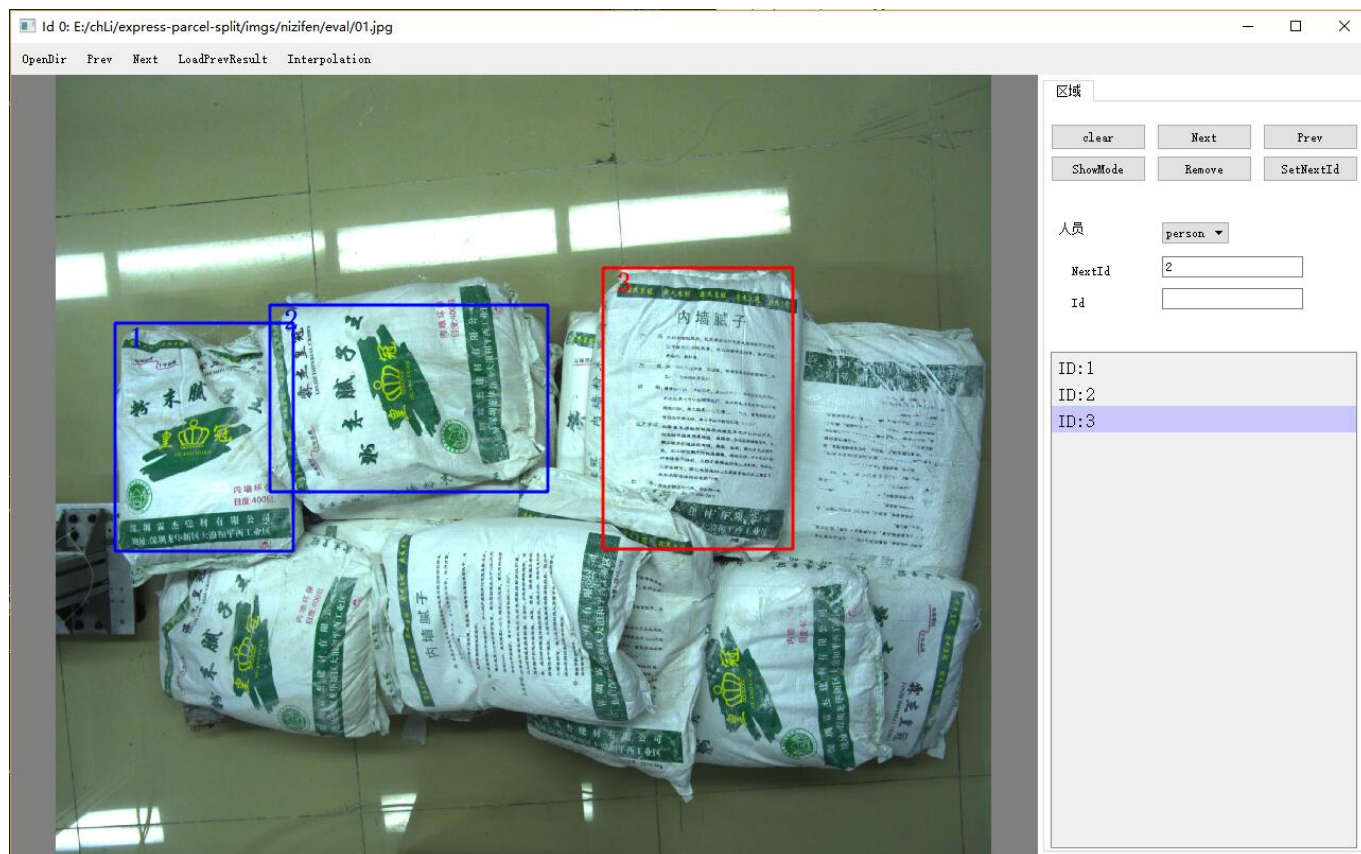


```
1  {
2      "Area": {
3          "category": "",
4          "color": {
5              "b": 255,
6              "g": 0,
7              "r": 0
8          },
9          "color1": {
10             "b": 0,
11             "g": 0,
12             "r": 255
13         },
14         "labels": [
15         ],
16         "labelDict": [
17             {
18                 "name": "cross_type",
19                 "ui_name": "label",
20                 "options": [
21                     ["0", "santongguan_lay"],
22                     ["1", "santongguan_stand"]
23                 ],
24                 "default": "0"
25             }
26         ],
27         "polygons": [
28         ],
29         "type": "bbox",
30         "ui_name": "区域"
31     }
32 }
33
34
```

1.2.5.2 选择标定数据集

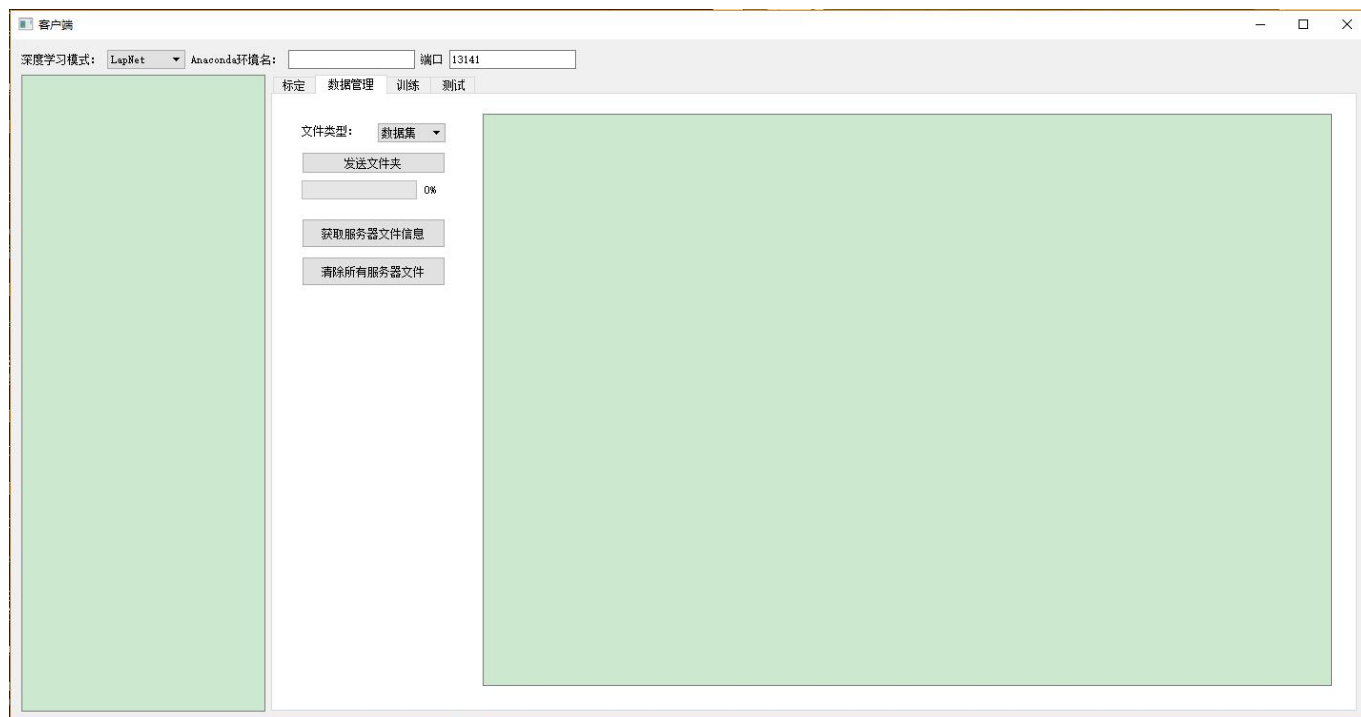
点击标定工具左上角的 **OpenDir** 按钮，选择待标定数据所在根目录，即可开始进行标定

注：在完成一张图片的标定之后，程序会在切换到下一张图片时保存当前的标定结果，因此标定完最后一张时需要向前切换一张图片，最后一张才会保存

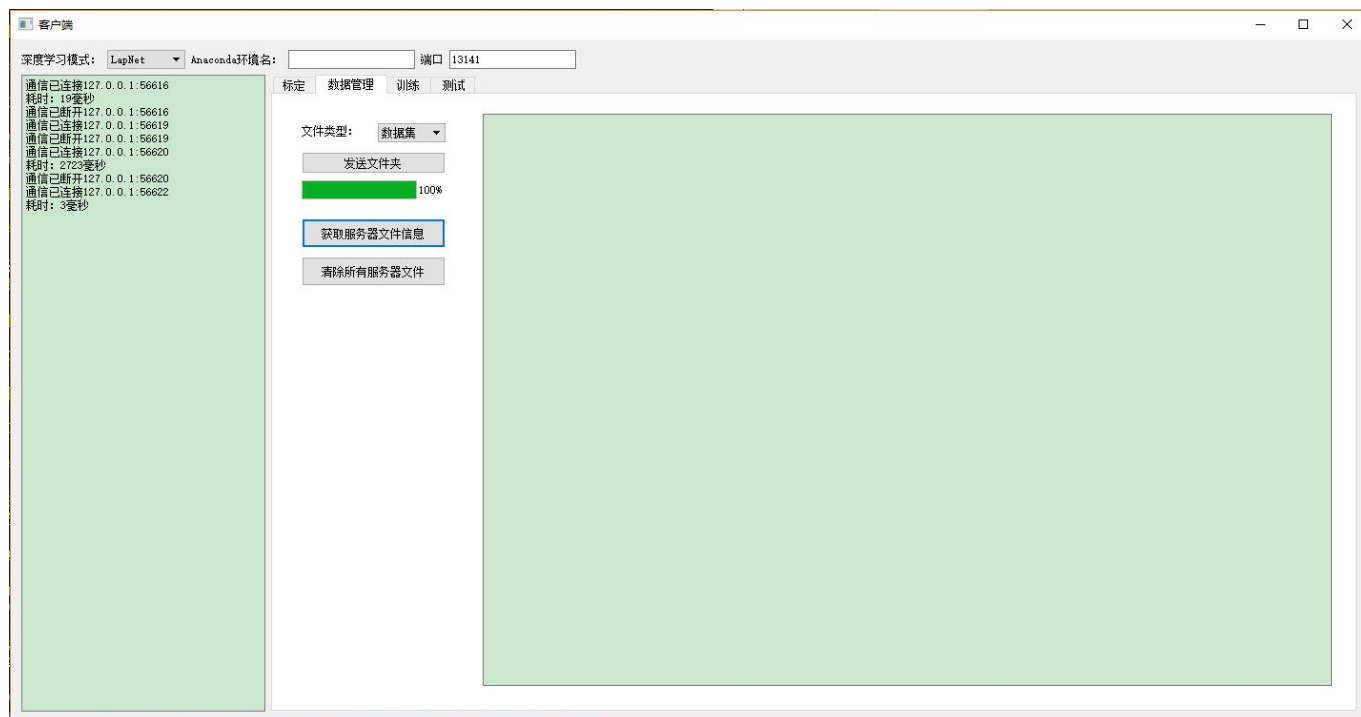


1.2.6 数据管理

该部分可以选择发送 **标定好的数据集** 或发送 **相关的 OBJ文件**，服务端会根据 **当前的深度学习模式** 来确定保存路径，不同模式下发送的文件不会共享



点击 **获取服务器文件信息** 按钮可以得到当前存储在服务器的所有文件的路径信息



点击 **清除所有服务器文件** 按钮可以清空所有保存在服务器或在服务器生成的文件

1.2.7 训练

在该板块，可以在 **发送过OBJ文件夹之后** 设置需要生成的数据量，进行自动的数据生成，但目前仍处于测试阶段，**生成结果并不能保证达到理想效果**

数据增强与数据转换则可以稳定生成所需的数据集及对应的处理结果，其中

(1) 数据增强会将当前已有的数据进行随机放缩、旋转和平移，生成的新数据集的数量取决于设置的 **数据增强系数**，如果 **数据增强系数=0**，则会默认只 **搬运数据**，而不生成新数据

(2) 数据转换会将现有的 **搬运好/数据增强得到的数据集** 转换为 **标准COCO数据集**，以便能够进行接下来的训练

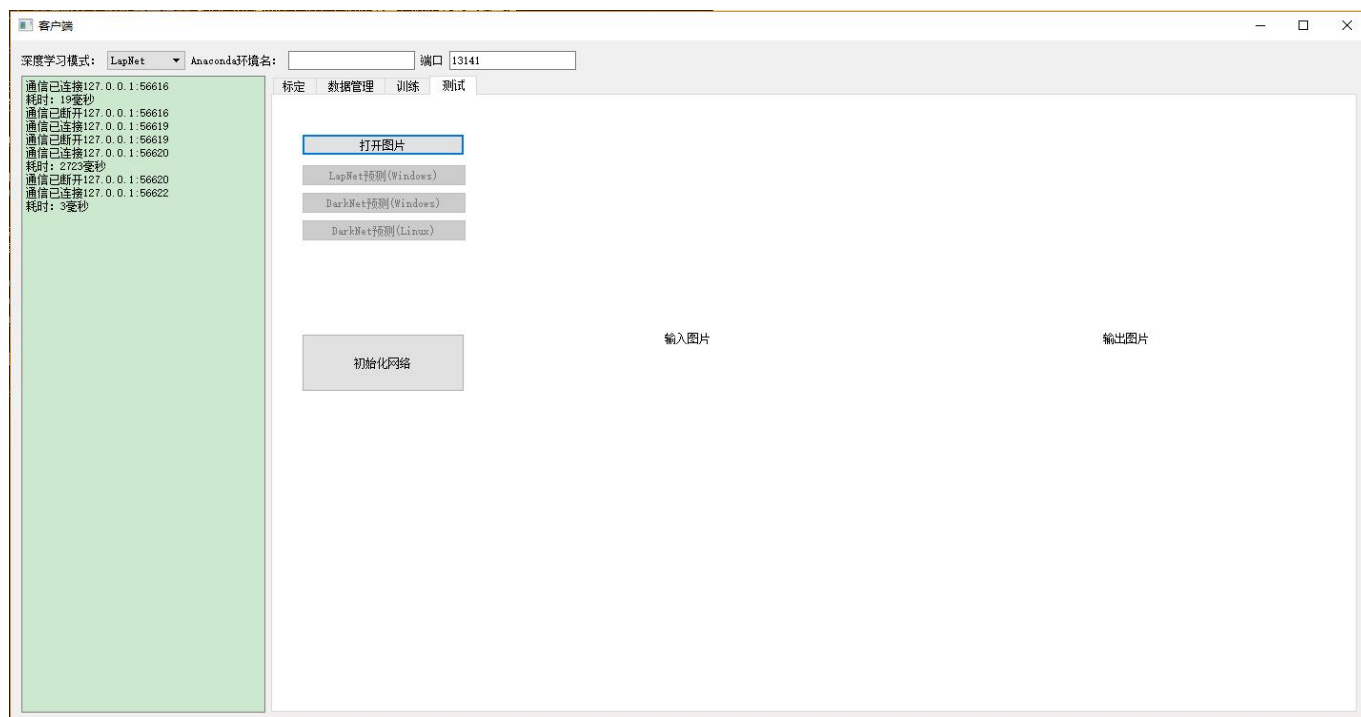
注1：数据生成、数据增强、数据转换暂时仅支持处理DarkNet数据集 注2：如果仅使用自己标定好的数据集进行训练，则需要发送数据集之后，设置数据增强系数=0，并单击开始数据增强以进行数据搬运



1.2.8 测试

基于 **Windows** 的测试采用了 **IR** 网络模型格式，使用 **OpenVINO** 引擎进行推理，因此暂时仅支持安装有 **OpenVINO** 的 **Windows** 系统

基于 **Linux** 的测试使用了 **Darknet** 框架，因此暂时仅支持安装有 **Darknet** 的 **Linux** 系统





单击 **初始化网络** 按钮可以对网络模型进行初始化和更新，其中

(1) 在 **Windows** 系统中，网络模型保存在

```
/FilterSocket_v1.1/Windows/models
```

需要将最新的模型转化为 **IR** 模型，并 **同名替换** 掉上述路径中的模型，然后初始化网络即可使用新模型进行测试，转化方式如下

```
pth转onnx  
修改并运行 FilterSocketv1.1/pth2onnx.py
```

```
onnx转ir  
"C:\Program Files (x86)\IntelSWTools\openvino_2019.3.334\bin\setupvars.bat"  
python <path-to-your-openvino\deployment_tools\model_optimizer\mo.py> \  
--input_model <path-to-your-onnx> \  
--input_shape <your-model-input-shape>
```

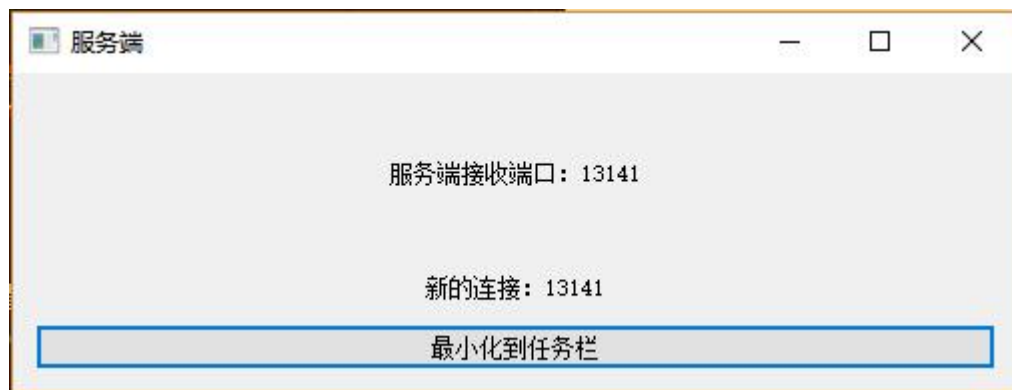
(2) 在 **Linux** 系统中，直接单击初始化网络，即可更新为最新的网络模型进行测试，如果出现问题可以选择尝试 **重新编译darknet-master**，编译方法为 **直接在 darknet-master 目录下执行 make**

注：在 **Linux** 系统中，如果多次点击初始化网络按钮，可能会导致服务端退出，原因为之前申请的 **CUDA Memory** 需要在程序退出后才会清空，只要再次重新启动服务端即可继续使用

2 Server 服务端

2.1 图形化界面

主界面，会提示新的接入；单击 **最小化到任务栏** 按钮可以将服务端隐藏到任务栏



任务栏图标



右键任务栏图标，选择显示主窗口可以呼出服务端主窗口

