# 语义分割网络贯通开发指导

## 1 项目开发目的和意义

语义分割是计算机视觉中的基本任务，在语义分割中我们需要将视觉输入分为不同的语义可解释类别，「语义的可解释性」即分类类别在真实世界中是有意义的，与图像分类或目标检测相比，语义分割使我们对图像有更加细致的了解。这种了解在诸如自动驾驶、机器人以及图像搜索引擎等许多领域都是非常重要的。

## 2 总体设计

语义分割网络系统可以划分为数据预处理、模型推理、数据后处理三个小模块，各子系统相对独立，但存在数据关联。其中数据预处理包括读取图片，转换为模型需要的格式和大小；模型推理，使用推理接口，对输入的图片数据进行推理；数据后处理完成推理结果的展示。语义分割网络的代码参见 https://gitee.com/Atlas200DK/sample-segmentation-python/

为了说明各模块之间的结构关系，细化的整体结构图如2-1所示。

开始

数据预处理

模型推理

数据后处理

结束

图2-1 系统整体功能结构图

## 3、开发步骤

### 3.1 语义分割网络处理流程

#### 3.1.1图像预处理

分类网络的数据预处理模块，采用Opencv的imread函数读图片，将图片数据发送给推理模块。

#### **3.1.2 模型转换及推理**

推理模块调用jpeg2yuv接口，将BGR图片转为YUV格式。将YUV数据送给推理引擎，推理引擎首先调用hiai.NNTensor 接口创建输入Tensor， 然后调用NNTensorList 讲输入转换未numpy 结构，最后调用GraphInference接口进行推理。

#### **3.1.3 图片后处理**

后处理模块读取推理的结果，输出每个像素点的类别概率。每个像素点选定概率最大的类别作为它的类别。给不同类别设置不同像素颜色，输出图片。

读取推理结果

每个像素点归属类别分析

设置类别的颜色

不同类别输出不同颜色

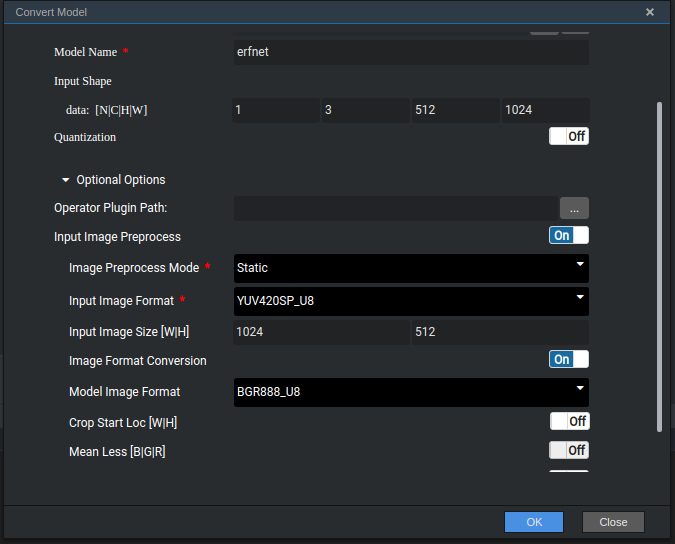
结束

### 3.2 模型说明

从以下路径下载 erfnet 模型，转换为om模型

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 模型名称 | | 权重文件 | 模型文件 | |
| erfnet | [https://obs-model-ascend.obs.cn-east-2.myhuaweicloud.com/erfnet/erfnet\_cityscapes.caffemodel](https://obs-model-ascend.obs.cn-east-2.myhuaweicloud.com/erfnet/erfnet_cityscapes.caffemodel" \o ") | | | [https://obs-model-ascend.obs.cn-east-2.myhuaweicloud.com/erfnet/erfnet.prototxt](https://obs-model-ascend.obs.cn-east-2.myhuaweicloud.com/erfnet/erfnet.prototxt" \o ") |

Erfnet模型转换参数：



## 4 主要代码

### 4.1 预处理关键代码

方法jpeg2yuv 将图片转为yuv

def rgb2nv12(image):

    if image.ndim == 3:

        b = image[:, :, 0]

        g = image[:, :, 1]

        r = image[:, :, 2]

        y = (0.299\*r+0.587\*g+0.114\*b)

        u = (-0.169\*r-0.331\*g+0.5\*b+128)[::2, ::2]

        v = (0.5\*r-0.419\*g-0.081\*b+128)[::2, ::2]

        uv = mergeUV(u, v)

        yuv = np.vstack((y, uv))

        return yuv.astype(np.uint8)

    else:

        print("image is not BGR format")

def mkdirown(path):

    if os.path.exists(path) == False:

        os.makedirs(path)

def jpeg2yuv(src\_name,resize\_w, resize\_h):

    src\_root\_path =  '../jpg'   #'./data/jpg'

    nv12\_root\_path = '../nv12'  #'./data/nv12'

    bgr\_dest\_path =  '../bgr'  #'./data/bgr'

    image\_ori = cv.imread(src\_name)

    image\_ori = cv.resize(image\_ori,(resize\_w, resize\_h))

    yuv = rgb2nv12(image\_ori)

    return yuv

### 4.2 推理关键代码

将YUV信息转为numpy结构，掉用推理结果的到结果resultList

   inputImageTensor = hiai.NNTensor(input\_image,InWidth,InHeight,3,'testImage',DataType.UINT8\_T, InWidth\*InHeight\*3/2)

        nntensorList=hiai.NNTensorList(inputImageTensor)

        resultList = GraphInference(myGraph,nntensorList)

        if resultList is None :

            print "graph inference failed"

            continue

### 4.3 后处理关键代码

从推理结果中获取每个像素点的类别，同一类别填充同种颜色，使用imwrite输出推理结果图片。

resultArray = resultList[0]

        resultArray=resultArray.reshape(19,1024,512)

        prediction = resultArray.argmax(axis=0)

        #print prediction

        prediction = np.squeeze(prediction)

        prediction = np.resize(prediction, (3,512,1024))

        prediction = prediction.transpose(1, 2, 0).astype(np.uint8)

        prediction\_rgb = np.zeros(prediction.shape, dtype=np.uint8)

        cv.LUT(prediction, label\_colours\_bgr, prediction\_rgb)

        input\_path\_ext = child.split(".")[-1]

        input\_image\_name = child.split("/")[-1:][0].replace('.' + input\_path\_ext, '')

        out\_path\_im = dstFileDir + input\_image\_name + '\_erfnet' + '.' + input\_path\_ext

        cv.imwrite(out\_path\_im, prediction\_rgb)  # color images for visualization

## 5 样例运行

样例运行步骤参考

https://gitee.com/Atlas200DK/sample-segmentation-python/blob/master/README.md

## 6 推理结果：

Erfnet（使用cityscape数据集训练）模型推理：



