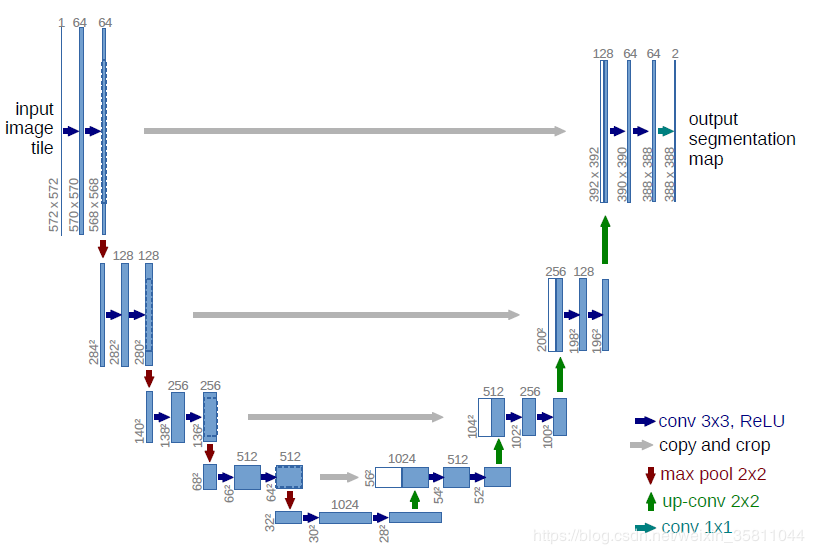
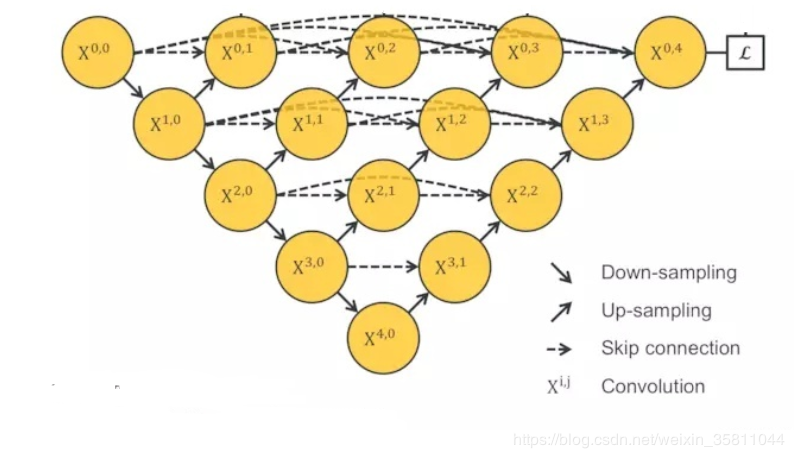
**医学图像算法unet和unet++的darknet开源实现**



**上图是unet的网络模型**



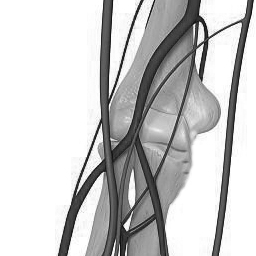
**上图是unet++的网络模型(相当于把残差引入了unet)**

本开源项目是用**移植方便**的**darknet深度学习框架**(纯C语言代码),实现了上面两种模型的算法.并在项目中放入了例图/批处理/执行文件.并配备了多个详细说明文本文件.主要可应用于图像分割/目标滤出/目标定位.

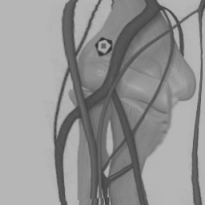
**github下载地址:** https://github.com/lswtzw/zwcode

**百度网盘下载:** https://pan.baidu.com/s/1bE2PZ5Rvu-9QU3\_Ib2heAg 提取码：0r13

下面是开源项目中提供的伪造实验图像:

train训练原图 train训练Label图(二值mask图)

上面的图大约10张,主要是方位和形变的小差别.用于训练,原图和label图的文件名一一对应.

test测试图 test结果图(二值mask图)

**(**能排除**六角形干扰小块**,适应较大形变,并**正确输出血管**的mask图**)**

上面的图大约15张,肯定不同于10张训练图,不仅目标线条有较大变化,而且还增加了明暗变化和干扰背景或干扰杂块,且训练时未用数据增广.

三次以上的实测发现,unet网络有很好的泛化能力,能正确提取出想要的目标,**能正确过滤掉比目标淡和比目标浓的大小干扰色块或背景,适应较大程度的形变,从而清晰地过滤出目标并定位目标.**墙裂建议在训练时就把各种干扰色块和背景弄到train图像上去,但在train的对应mask图中却只保留要的目标.

* **运行环境和运行条件:**

windows系统 -----(最好是有4GB显存以上的独立显卡,并安装好了CUDA开发环境,未安装只能用CPU版本);

本项目需要先安装opencv320-----(最好是opencv320,与开发机保持一致,其它版本估计也行);

最后:请把opencv320下的opencv\_world320.dll复制到可执行文件目录下(darknet/build/darknet/x64/);

opencv320在上面提到的百度网盘中可以下载到;

首次使用建议先跑通darknet\_no\_gpu.exe这个版本(CPU版本,配置要求低).

* **发行版本说明:**

本开源项目是**基础于AlexeyAB大神的Windows版的darknet框架加改的**,所以如果你的电脑已配置好了AlexeyAB大神的版本,只要替换掉:

darknet.h,darknet.c,data.h,data.c,parser.c,并增加unet.h,unet.c这几个文件就可以了.用户可按需修改主文件unet.c,

darknet框架的windows版本安装方法请百度查CSDN网文:"win10下配置yolov3","window下安装配置darknet"等.

只要**这个配置好了,能编译出darknet\_no\_gpu.exe,就可以继续下面的步骤了.**

另外,本开源项目也参考了网友:**阳光玻璃杯 的"图片语义分割"的博文**,并在他的git关键代码和配置的基础上改出了本开源代码,他用的是segnet网络模型.

* **使用说明:**

因为darknet.exe是在我的电脑上编译出的gpu版本,可能在你的电脑上启不动.需要重新编译源码.

如果你想先快速上手试试,可以先用:darknet\_no\_gpu.exe这个可执行文件.

把下载好的darknet和unet\_data两个目复制到你的电脑c:根目录下,可以少些路径配置.

测试的话可以先执行unet\_data目录下的几个批处理文件.

**一.文件和目录配置：**

1.darknet目录是项目源码,用vs2015或以上打开它的.sln工程文件,在build子目录下.

2.unet\_data目录是**调用批处理**/配置文件/图像集/权重模型文件的存放主目录,核心是data.txt:

这个文件是各种路径的配置文件,注意修改对应的盘符!默认初始值是在C盘根目录下.

上面这个和darknet框架上的传统data配置文件差不多,用法略有改动.

3.color目录下放的是彩色训练图像和对应配置文件.

4.backup目录一般用于存放训练出的模型文件.

**二.网络模型配置：**

unet\_cfg.txt/unet\_cfg++.txt这两个文本文件就是网络模型配置文件

(完全同于darknet格式的cfg),注意:不要另存为UTF-8编码(有时改了会出错),

就用windows记事本默认的ANSI编码来保存.(windows回车换行模式)

配置文件可以自由使用各种网络模型,如segnet也是可以的.这个我放在了darknet的cfg目录下.

**三.关于数据增广:**

因为label(蒙版二值图像)和image是按文件名称,进行的一一对应关系,如果产生平移或挤压等变化时,

两者要同步进行,所以须要另外想办法用其它工具软件或程序实现,结果图直接加到训练模板中.

**四.结束训练的时机:**

根据笔者的经验,avg\_loss的值,训练到100以下,就差不多了可以停了.

总之我的GT750显卡4GB显存,10张模板用了4个小时以上,训练unet++时间比unet要长得多.

如果训练24小时都不行,一般是模板图像有问题,存在矛盾模板.

根据笔者的经验,GPU识别的速度一般在2秒以内.视图像大小和网络复杂度而定.

启动训练后,可以用nvidia-smi查看显存占用,如果余得多,可以增大cfg的[net]的batch的值或图像宽高.

**五.调用接口:**

为方便C#和JAVA等的弱耦合调用,提供了一次性载入模型和多次性按需识别的调用接口,请参见项目中包含的说明文档: <<关于C#或java的调用说明.txt>>

darknet.exe unet <train/test/preload> <data filename> <cfg filename> [weights filename] [image file name]

上面是调用的传入参数: train是训练模型,test是识别,preload是预装模型,data filename是路径配置文件,cfg filename是模型配置文件,后面是权重(模型)文件

**六.简单调用示例:**(unet\_data目录下提供了各种调用方式的.bat文件范例)

**1.训练train**(第一个是cpu版) (需要在data.txt中指定train图像的路径)

*"../darknet/build/darknet/x64/darknet\_no\_gpu.exe" unet train data.txt unet\_cfg.txt*

*"../darknet/build/darknet/x64/darknet.exe" unet train data.txt unet\_cfg.txt*

训练的明细文件名称列表在train.list.txt中

**2.继续以前的模型做再训练**

*"../darknet/build/darknet/x64/darknet.exe" unet train data.txt unet\_cfg.txt backup/unet\_cfg.backup*

把unet\_cfg.txt换成unet++\_cfg.txt就是用unet++算法模型.

**3.预测/识别**

*darknet.exe unet test data.txt unet\_cfg.txt backup/unet\_cfg.backup test/10.jpg*

**4.批量预测/识别**(需要在data.txt中指定test图像的路径)

*darknet.exe unet test data.txt unet\_cfg.txt backup/unet\_cfg.backup*

将会对指定test路径下的图像10.jpg,11.jpg,12.jpg,13.jpg.....进行批量顺序识别.

5.**预装模型与后期多次按需调用**(详细调用请参见说明书)

*darknet.exe unet preload data.txt unet\_cfg.txt backup/unet\_cfg.backup #预装模形入内存后待机*

*wdarknet\_call.exe 1 #按需控制darknet做一次批量识别*

**结束语:** 个人感叹,用KERAS训练UNET非常快,编的程简洁易懂,是深度学习的第一选择,实验室的好帮手;但DARKNET是低级的,C语言原子级编码操作,可以自个儿改写dropout算子的具体实现等,这里可以让你真正吃透神经网络的底层,因为用了BN吧,训练非常慢,个人测试识别效果却貌似更好些,缺点是感觉其更新慢支持少还很山寨;tensorflow作为中间语言,作出的行业开天辟地贡献不可磨灭,其作者创造了一个晦涩编码的高尚哲学体系,个人主观认为既不正宗也不亲民,不推荐思想传统的人学用.

**bug修订补充:**

1.实测unet,unet++两种算法各有优劣,最后输出mask结果图轮廓,若颜色都偏淡少,可对两种算法的结果图做加法合成,输出一张较浓多的结果图.相反,如果都偏浓多,则可做交集输出.

2.实测发现unet,尤其是unet++模型,对小面积目标的单独定位(比如上图大骨上的小六边形图块)效果不好,这种情况可以用减法实现,就是用两套(train)label图像,都有大面积背景轮廓线,但其一套上面不保留小目标的mask二值图,另一套上保留,这样训练出两套权重模型,识别同一套test图像,输出的两套mask结果图,相减,就能得到小目标的正确mask定位图.如果还要对小目标做分类,可以再选一套比alexnet更成熟的网络模型,在深度学习领域内采用多模机制进行图像识别.