预处理

Key1：

I removed the alpha channel from the annotations and masks.

RGB的每个通道在各个像素上都拥有具体的强度或值。每个通道的强度决定图像中像素的颜色。通过添加第四种 alpha 通道，文件可以指定每个像素的透明度或不透明度。alpha 的值为 0 表示透明，alpha 的值为 255 则表示不透明，在此范围之间的值表示半透明。Alpha通道的内容代表的不是图像的颜色，而是选择区域，其中的白色表示完全选取区域，黑色为非选取区域，不同层次的灰度代表不同的选取百分率，最多可有256级灰阶。

-----去除alpha通道，即去除透明度（cell边缘）

<https://blog.csdn.net/u012581760/article/details/78683195> -apple电脑软件操作

I closed small holes in objects. I believe I used a (1, 1) structuring element. I also spot checked every object to ensure it was a mistake (e.g. or it seemed like a mistake to my eye).

----填补空洞，开闭、扩张腐蚀操作 --代码get

I also supplemented the training set in an attempt to standardize annotation disagreements:

I computed a distance transform on each of the masks (not the flattened label matrix) and found the maximum peak.

图像的距离变换实现了像素与图像区域的距离变换，使得最后生成的图像在该自己元素位置处的像素为0，临近的背景的像素具有较小的值，且随着距离的增大它的的数值也就越大。对于距离图像来说，图像中的每个像素的灰度值为该像素与距离其最近的背景像素间的距离，也就是说，给每个像素赋值为离它最近的背景像素点与其距离，一幅二值图像的距离变换可以提供每个像素到最近的非零像素的距离。

<https://blog.csdn.net/qq_34784753/article/details/68951918> ---c++，cv

I computed a watershed segmentation for each masked image (that was dilated with a (1, 1) structuring element) using the peaks as markers. I added this new set of masks to the existing set.

分水岭法来操作mask，并加入到mask集合中

<https://blog.csdn.net/tengfei461807914/article/details/77620216> py，cv实现

I took the centroid of each mask and used the random walker algorithm to compute a third set of masks using the aforementioned masked images. I then used the same watershed process from before.

在之前的mask之后增加第三批mask—使用随机游走算法来进行图像分割，然后再使用分水岭算法

<https://blog.csdn.net/menjiawan/article/details/46862115> c++实现

Finally, I created a fourth and fifth set of annotations by using a geodesic active contour with two different edge preserved responses (from a median filter and an inverse Gaussian filter). I then used the same watershed process from before.

geodesic active contour-- 该类方法主要指的是活动轮廓模型（active contour model）以及在其基础上发展出来的算法，其基本思想是使用连续曲线来表达目标边缘，并定义一个能量泛函使得其自变量包括边缘曲线，因此分割过程就转变为求解能量泛函的最小值的过程，一般可通过求解函数对应的欧拉(Euler．Lagrange)方程来实现，能量达到最小时的曲线位置就是目标的轮廓所在。

<http://www.pudn.com/Download/item/id/2775962.html> --- matlab实现

I used the supplemental data like this:

I took the initial set of training data and made a 80%/20% split. I repeated this process five times to create five training sets and five validation sets.

将train数据集4:1分开，重复五次—》五份训练集五分验证集

其中五份训练集分别做 （1）原始train数据 （2）分水岭算法（3）随机游走+分水岭 （4）基于中值滤波的geodesic active contour （5）基于反高斯滤波的geodesic active contour

Each epoch I would randomly swap each annotation for one of the five buckets (the real training data, the watershed data, the random walker and watershed data, and the two buckets of geodesic and watershed data) so each epoch would only see one image once.

每一个epoch都随机选择上述中五份train的其中一个，所以一个epoch只能可视化一张图片

This is an area that I believe can be improved further with more time.

I augmented the data like this:

random scaled down version of the annotation and image (with nearest neighb。。。。。。。。。。。。。。。。。。。。。。。)