1. 高斯模糊和边缘检测一样，都是用了卷积的原理。
2. 卷积是什么，卷积操作指的就是用一个卷积和对一张图的每个像素进行一些列操作。卷积核一般是一个方形区域，然后每个格都有一个权重值。对图像素进行遍历的时候，会把卷积核中心放在像素上，核翻转（这里不需要，在数字图像的卷积中，只为了提取图像的特征，其实只是借鉴了“加权求和”的特点。）之后再依次计算每个元素和其覆盖的图像像素值的乘积并求和。得到新的值就是该位置的像素值
3. 边缘检测顾名思义就是把一张图的边缘给检测出来，那么认为一个图上有一个边，说明这个像素跟他相邻像素之间应该是由明显的颜色，亮度，纹理间有较大差别，有颜色梯度的变化才会有一个边。过程大致就是通过vertex shader中获得像素周围的像素点的uv坐标，在fragment shder里面用水平和竖直方向两个卷积核去计算该像素点的周围卷积结果并且相加，最后把卷积结果插值到自己定义的边缘颜色，再把前一步的插值结果，插值到整个图片上。
4. 高斯模糊我理解为把像素周围的图的颜色按照一定的权值混到当前像素的颜色中，距离越近影响越大，融合后形成一种模糊的感觉。在《unityshader入门精要》里面，为了减少计算，把N\*N的高斯图像卷积核分解为两个1\*N的高斯图像卷积核并且通过两个pass计算结果叠加。不过通过比较两个1\*N的卷积核会发现，他们只是方向上不同，所以计算两个Pass可以直接写成一个Pass，只需要在计算第二次模糊时，把卷积核倒置一下即可。记录一下模糊过程。但是性能是相仿的，因为两个Pass的shader也只是顺序调用每一个pass。

