1. orb-slam的单目帧间匹配利用的是匀速模型，在下一帧图像中对上一帧图像的特征点划定一块固定的区域进行特征点关联。如果对上一帧的特征点利用语义分割贴上标签之后，在下一帧中进行特征点的语义分割，与上一帧进行匹配，速度更快，也不容易丢帧

mask-rcnn语义分割何凯明，把每个特征点标上语义信息  
两帧之间进行单帧匹配（优先级高），重定位

现在的语义slam

1. 将float描述符转换为二进制，直接的转换精度下降很快，可以试试神经网络
2. 如果用语义slam的话，得学习比较多概率的东西，有很多的公式，可能最后比较难，如果用二进制描述符的话，可能需要的相关知识少一些
3. 描述子的trick：

光照不敏感：灰度梯度（SIFT）

噪声不敏感：直方图（SIFT）

亮度影响：特征归一化（SIFT）

描述子占用空间大：二进制描述子（BRIEF、ORB）

尺度不变：图像金字塔

视角不变性：特征点的主方向（ORB）

1. 应该花两三天对所有出现的特征点和描述子进行总结归纳，了解每个的优缺点，实现方法
2. 孪生网络训练方式借鉴，可以考虑结合语义之类，用Harris或其他角点检测方法，提取的还是低维度信息，而且通篇都是利用3D投影变换进行的学习，不是单纯的像Superpoint着眼于特征点提取上，尽管在outdoor数据上也取得了较好的效果，但其他泛化效果有待考察（利用RGB-D局限于室内尺度），但任务驱动型思想很好。

GCNv2实现了在TX2上的ORB-SLAM架构的实时slam

1. GNN注意力机制
2. 需要学习的东西：

传统的特征提取（Harris、SIFT、SURF），

图像预处理的方法，包括图像的滤波、分裂、归并、分割以及形态学处理等等，也可以与深度学习方法相结合，例如dilated CNN就可以看做形态学操作与CNN的结合

相关的深度学习架构：LSTM、RNN、GNN、GAN

ORB-SLAM2的代码

马太原学长的代码跑起来、理解

思考的方向：基于马太原学长的论文，写一下深度学习生成二进制描述符，最后最好能达到在TX2上运行，也不一定用马太原学长论文里面的损失函数，自己可以结合其他的论文进行损失函数的设计

1. 马太原学长论文：论文里面已经突出的优点在我的论文里面就一笔带过，主要实验就比较二进制，突出实时性，内存占用，帧率，最好能在tx2上跑。

有篇论文实验显示深度学习描述符不如传统描述符，但是只描述了现象，没有分析原因，学长给出的原因是slam的匹配机制显示深度学习方法，在代价函数上有理论缺陷