工作总结与计划

一、 工作总结（按时间划分）

日期	工作内容			完成情况及主要问题
	上午	下午	晚上	
星期一	处理材料	坐火车	收拾东西	无相关情况
星期二	修改开题报告	做开题 PPT	修改开题 PPT	按照上周计划准备开题答辩
星期三	写答辩稿	读代码	读代码	基本完成开题答辩的准备 阅读深度估计的代码
星期四	回顾上周论文	回顾上周论文	休息	回顾上周的论文 解决周报中存在的问题
星期五	整理论文	修电脑	做展示 PPT	对本学期的工作进行总结 准备展示思路和内容
星期六	做展示 PPT	写周报	写周报	总结前期工作 完成周报

二、下一步计划（按任务划分）

编号	工作内容	目标	相关配合
1	进行开题预答辩	1 月 5 日左右完成	无
2	调试代码	1 月 9 日前完成	无
3	将两个任务联合起来	按照约束关系的强弱分阶段完成	无
4			
5			
主要风险	无		

三、个人分析与总结

内容提要	
1	进度方面：本周主要对开题和展示做准备，对之前阅读的论文进行了回顾
2	课题方面：在总结过程中发现目前阅读的深度估计中全监督的方法较少
3	其他思考：PyTorch 目前处于基础水平，使用上还需要通过阅读和运用进行更熟悉
4	

本周主要对开题答辩展示做了准备，对之前阅读的论文进行了回顾，与之前的工作相比，本周工作量较少。在开题答辩的准备工作和开题答辩完成后，重点开始进行代码的调试和编写，近期将通过由弱到强的约束关系将两个任务联合起来。

在总结回顾的过程中，认为基于全监督的深度估计论文阅读较少，之前阅读的部分论文大部分为双目深度估计或以视频序列作为训练的自监督深度估计，后期会在阅读一下单幅图片的全监督单目深度估计论文。此外，结合之前阅读的论文和本周查阅的几篇论文来看，Encoder-Decoder 为近来深度估计中使用较多的结构，且能取得较好的估计结果。

由于本周未详细阅读论文，此次周报主要对上周周报中存在的问题进行总结。

1. Distilling Image Dehazing with Heterogeneous Task Imitation

（1） 个人想法

本篇论文重点解决的任务是图像去雾，对于研究的参考来讲，通过不同任务的教师-学生网络（教师网络进行图像重建，学生网络进行图像去雾）探究了教师教师通过不同任务帮助学生网络训练的可行性及方法。该论文通过将的保真损失函数、感知损失函数和教师与学生的中间特征图的像素级差异图结合起来，利用教师网络学习时产生的中间特征图，驱使学生网络模拟学习这些特征图实现图像的去雾，即面向过程的监督或暗知识。同理，可以学习这一模式应用到深度估计与深度超分辨率重建的联合任务中。

2. Semantic-Guided Representation Enhancement for Self-supervised Monocular Trained Depth Estimation

（1） 个人想法

与深度超分辨率重建中存在的问题相同，精细物体或边缘区域在深度估计中也是较难处理的问题，即要保证类别内部的一致性和边缘的锐利。从这一点来看，深度估计与深度超分辨率重建也是相关联的，可能可以互相促进。此外，本文的思路和方法也可以迁移到深度超分任务中。

（2） 自监督深度估计的监督信息是什么？

在缺乏 ground truth 深度的情况下，一种方法是利用图像重建作为监督信号来训练深度估计模型。模型被给定一组图像作为输入，要么是以立体对的形式，要么是以单目序列的形式。通过对给定图像预测深度，并将其投影到附近的视图中，通过最小化图像重构误差来训练模型。

自监督的一种形式来自立体图像对。这种形式下，同步立体图像对在训练过程中是可用的，通过预测它们之间的像素视差，可以训练一个深度网络在测试时进行单目深度估计。

一种约束较少的自我监督形式是使用单目视频，其中连续的时间帧提供训练信号。在这里，除了预测深度之外，网络还必须估计帧之间的摄像机位姿（预测相机的转换矩阵），这在存在物体运动的情况下是具有挑战性的。这个估计的摄像机位姿只需要在训练中帮助约束深度估计网络。

The term “Ego-Motion” is typically being used in psychology and computer vision applications and refers to the motion of an optical sensor (e.g. camera) in 3D space. Many algorithms have been developed to deduce this information from a sequence of images.

（3） 这里用分割的 gt 还是算法得到的分割图？框图画的是用 gt

用算法得到的分割图

（4） 为什么要做点采样？

尽管基于学习的深度估计已经展现出优良的性能，然而这些方法在具有挑战图像区域（如予以边界或者微小物体区域）的精确深度估计上表现欠佳。例如，估计的物体深度通常不能与真实物体的边界对齐，并且具有细小结构的前景物体的深度倾向于被淹没在背景中。我们将这些现象归因于有限的深度表示能力，即（1）当前深度网络无法很好地表示像素级局部深度信息，尤其是在高度模糊的语义边界区域上；（2）当前深度表示无法实现全面描述了深度前景/后轮关系。这些问题导致从真实场景结构进行错误的深度估计，从而阻碍了在实际任务中的进一步应用。

本文通通过语义指导进行自监督的单目深度估计来增强局部和全局深度特征表示。语义分割可进行明确的类别级场景理解并产生对齐良好的对象边界检测。

我们在自监督的框架下使用来自不同领域的多级表示来增强点特征。具体而言，对位于二值语义类别边界上

的一组点位置进行采样，并从编码特征，深度解码特征以及语义解码特征中提取相应边缘位置的点状特征。然后，合并并增强逐点特征，并将其提供给最终的深度解码特征，以促进用于自监督的边缘感知深度推断。

3. Deep Color Guided Coarse-to-Fine Convolutional Network Cascade for Depth Image Super-Resolution

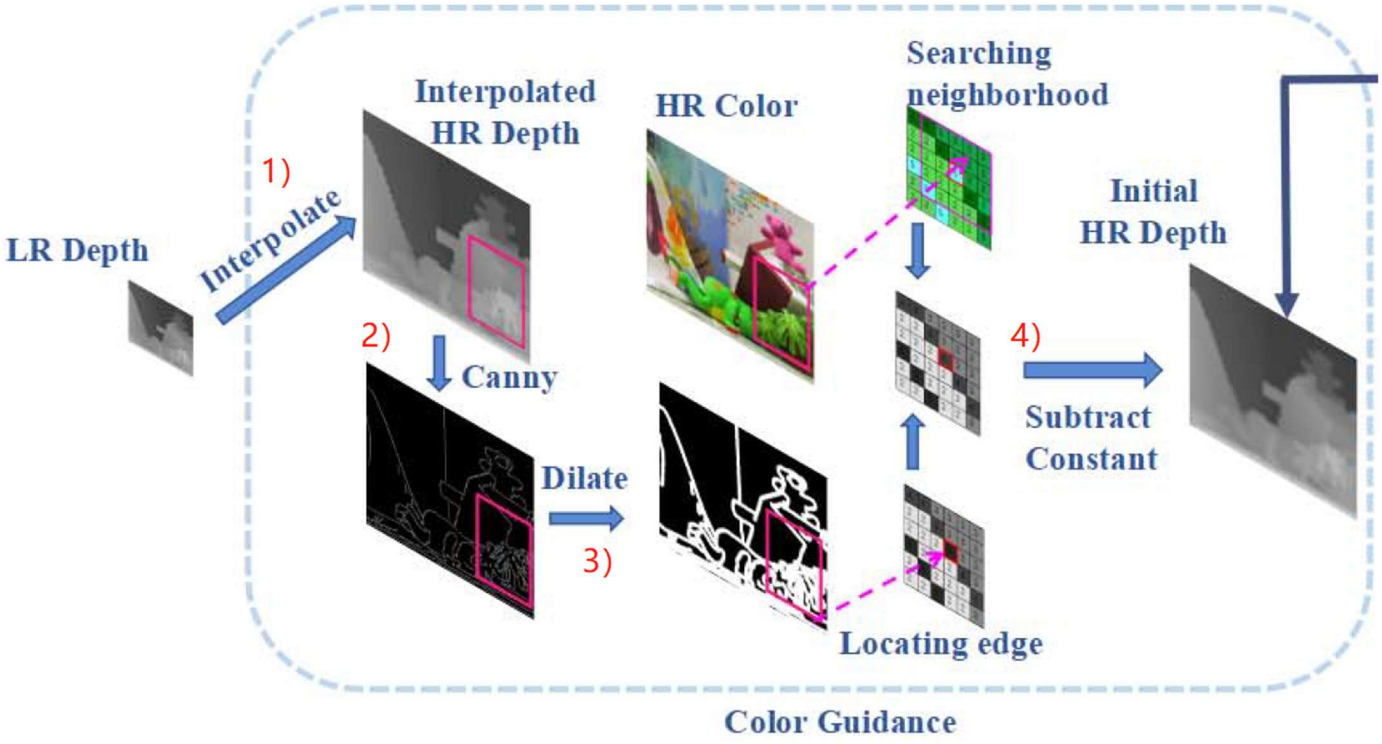
（1） 个人想法

本文主要以基于滤波的方法进行深度图像的超分辨率重建，与以往不同的是，滤波器是通过神经网络训练学习得到的，与之前阅读的深度估计通过学习获得损失函数等有相似之处。个人想法可以在这种方向进行借鉴。

（2） 什么作用？物理意义是什么？

（3） 这部分是在干啥？

问题（2）（3）均出现在修改插值后的边缘区域不确定的插值深度值。



1） 利用双三次插值方法从 LR 深度图像中获取初始的 HR 深度图像

2） 主要利用 canny 算子检测初始 HR 深度图像的边缘

3） 对 canny 算子检测到的边缘进行膨胀，确定边缘区域

4） 对于初始 HR 深度图像的边缘区域中以像素 p 为中心的邻域，我们在 HR 彩色图像的对应区域中找到一组颜色相似的像素，并确定与像素 p 颜色最相似的像素 q 的位置，最后找到与像素 q 在 LR 深度图像中的位置对应的像素，d_p 为像素 p 的深度值

首先为像素 p 找到一系列候选点的集合 $\widetilde{L_p}$

$$q_{\downarrow} \in \widetilde{L_p}, \quad \text{if } |q_{\downarrow}, p_{\downarrow}| \leq 2$$

其中，$p_{\downarrow}$ 表示像素 p 在低分辨率深度图 L 中的对应坐标，$\widetilde{L_p}$ 有 $p_{\downarrow}$ 周围的像素构成。

这些候选点与 像素 p 之间的颜色差异被用于寻找最合适的 d_p

$$d_p = L_{arg\min_{i \downarrow \in \widetilde{L_p}} |G_i - G_p|}$$

其中，G 是高分辨率彩色图片。

4. TIP 运行结果

训练：

```
===> Epoch[82](1/3): Loss: 0.0268 || Timer: 4.6000 sec.
===> Epoch[82](2/3): Loss: 0.0147 || Timer: 4.4930 sec.
===> Epoch[82](3/3): Loss: 0.0152 || Timer: 4.7419 sec.
===> Epoch 82 Complete: Avg. Loss: 0.0189
===> Epoch[83](1/3): Loss: 0.0266 || Timer: 4.7550 sec.
===> Epoch[83](2/3): Loss: 0.0169 || Timer: 4.8560 sec.
===> Epoch[83](3/3): Loss: 0.0137 || Timer: 4.8235 sec.
===> Epoch 83 Complete: Avg. Loss: 0.0191
===> Epoch[84](1/3): Loss: 0.0256 || Timer: 4.8784 sec.
===> Epoch[84](2/3): Loss: 0.0125 || Timer: 5.3047 sec.
===> Epoch[84](3/3): Loss: 0.0133 || Timer: 5.4761 sec.
===> Epoch 84 Complete: Avg. Loss: 0.0171
===> Epoch[85](1/3): Loss: 0.0273 || Timer: 4.7491 sec.
===> Epoch[85](2/3): Loss: 0.0131 || Timer: 4.8066 sec.
===> Epoch[85](3/3): Loss: 0.0135 || Timer: 4.9030 sec.
===> Epoch 85 Complete: Avg. Loss: 0.0180
===> Epoch[86](1/3): Loss: 0.0131 || Timer: 4.7781 sec.
===> Epoch[86](2/3): Loss: 0.0131 || Timer: 4.7384 sec.
===> Epoch[86](3/3): Loss: 0.0255 || Timer: 4.9528 sec.
===> Epoch 86 Complete: Avg. Loss: 0.0172
===> Epoch[87](1/3): Loss: 0.0137 || Timer: 5.0060 sec.
===> Epoch[87](2/3): Loss: 0.0128 || Timer: 5.0809 sec.
===> Epoch[87](3/3): Loss: 0.0259 || Timer: 4.8181 sec.
===> Epoch 87 Complete: Avg. Loss: 0.0175
===> Epoch[88](1/3): Loss: 0.0260 || Timer: 4.8689 sec.
===> Epoch[88](2/3): Loss: 0.0133 || Timer: 4.8410 sec.
===> Epoch[88](3/3): Loss: 0.0177 || Timer: 4.7212 sec.
===> Epoch 88 Complete: Avg. Loss: 0.0190
===> Epoch[95](1/3): Loss: 0.0323 || Timer: 4.4840 sec.
===> Epoch[95](2/3): Loss: 0.0152 || Timer: 4.4900 sec.
===> Epoch[95](3/3): Loss: 0.0262 || Timer: 4.7075 sec.
===> Epoch 95 Complete: Avg. Loss: 0.0246
===> Epoch[96](1/3): Loss: 0.0160 || Timer: 4.5990 sec.
===> Epoch[96](2/3): Loss: 0.0335 || Timer: 4.7125 sec.
===> Epoch[96](3/3): Loss: 0.0274 || Timer: 4.4950 sec.
===> Epoch 96 Complete: Avg. Loss: 0.0256
===> Epoch[97](1/3): Loss: 0.0289 || Timer: 4.6030 sec.
===> Epoch[97](2/3): Loss: 0.0243 || Timer: 4.4850 sec.
===> Epoch[97](3/3): Loss: 0.0138 || Timer: 4.5040 sec.
===> Epoch 97 Complete: Avg. Loss: 0.0223
===> Epoch[98](1/3): Loss: 0.0298 || Timer: 4.4930 sec.
===> Epoch[98](2/3): Loss: 0.0207 || Timer: 4.4950 sec.
===> Epoch[98](3/3): Loss: 0.0175 || Timer: 4.4940 sec.
===> Epoch 98 Complete: Avg. Loss: 0.0227
===> Epoch[99](1/3): Loss: 0.0202 || Timer: 4.9262 sec.
===> Epoch[99](2/3): Loss: 0.0278 || Timer: 5.0418 sec.
===> Epoch[99](3/3): Loss: 0.0244 || Timer: 4.8395 sec.
===> Epoch 99 Complete: Avg. Loss: 0.0241
Learning rate decay: lr=1e-05
===> Epoch[100](1/3): Loss: 0.0284 || Timer: 4.7270 sec.
===> Epoch[100](2/3): Loss: 0.0187 || Timer: 4.7300 sec.
===> Epoch[100](3/3): Loss: 0.0152 || Timer: 4.4810 sec.
===> Epoch 100 Complete: Avg. Loss: 0.0208
```

测试：

上周主要是在本地对少量图片进行小 batch 训练，调整代码，下周计划在服务器上尝试训练。