# 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 | 看综述 | 读论文 | 读论文 | 了解单目深度估计的论文<br>并选择进行阅读 |
| 星期二 | 读论文 | 读论文 | 学习 pytorch | 阅读 2014-NIPS 的单目<br>深度估计论文 |
| 星期三 | 读论文 | 读论文 | 形教课考试 | 完成论文阅读 |
| 星期四 | 读论文 | 读论文 | 休息 | 阅读 2015-ICCV 的单目深度估计论文，了解对上一篇论文工作的改进 |
| 星期五 | 写周报 | 写周报 | 学习 pytorch | 对论文进行回顾，完成周报 |
| 星期六 | 写开题报告 | 写开题报告 | 写开题报告 | 开始写开题报告 |

## 二、下一步计划（按任务划分）

| 编号 | 工作内容 | 目标 | 相关配合 |
|---|---|---|---|
| 1 | 读新的论文 | 12 月 19 日前完成 | 无 |
| 2 | 完成开题报告 | 12 月 16 日前完成 | 无 |
| 3 | | | |
| 4 | | | |
| 5 | | | |
| 主要风险 | 无 | | |

## 三、个人分析与总结

| 内容提要 | |
|---|---|
| 1 | 进度方面：本周完成两篇单目深度估计论文阅读，下周尽量阅读两个方向论文各一篇 |
| 2 | 课题方面：第二篇论文为第一篇工作的延伸，第二篇提出了多任务对单目深度估计的辅助作用 |
| 3 | 其他思考：加快 pytorch 的学习，以及可能考虑参考一些多任务进行深度估计的论文 |
| 4 | |

# 四、论文总结

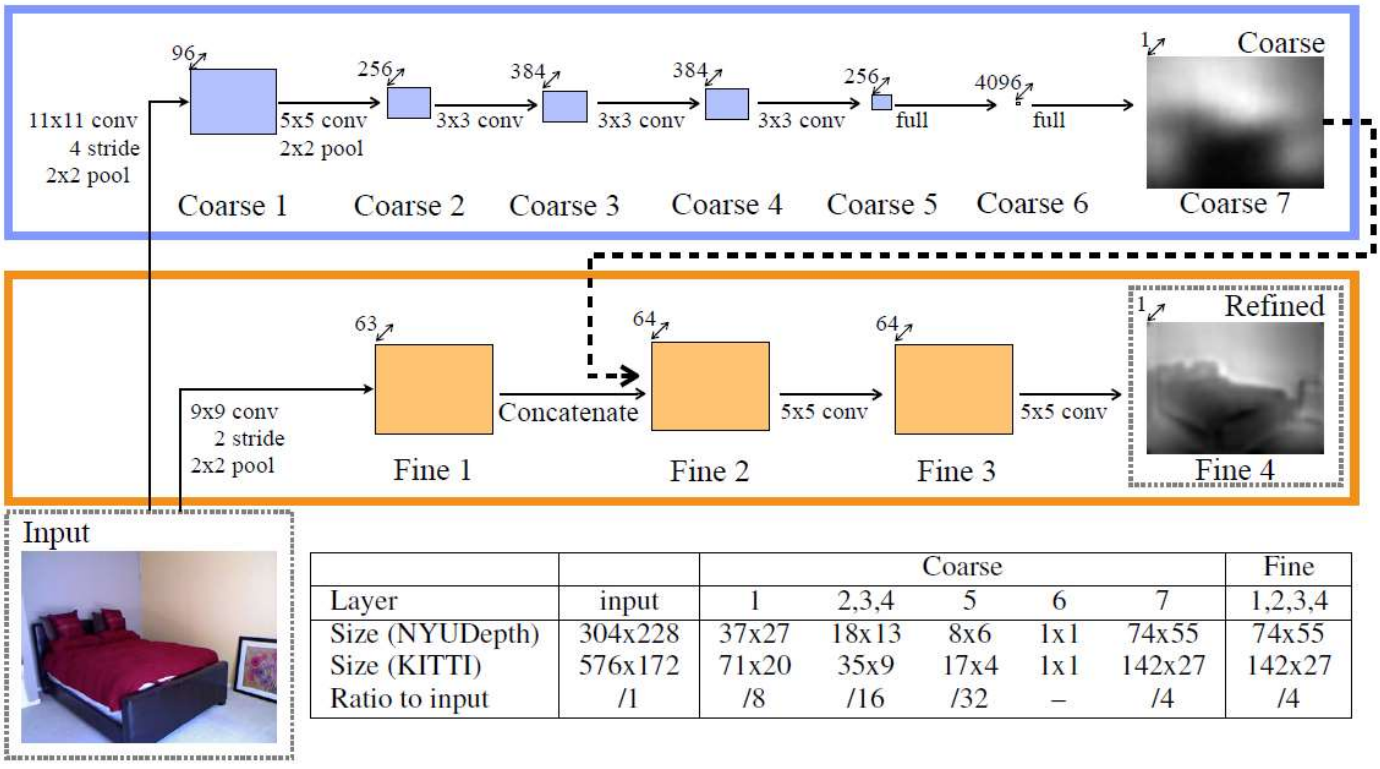
| | |
|---|---|
| 论文标题 | Depth Map Prediciton from a Single Image using a Multi-Scale Deep Network |
| 作者及单位 | David Eigen[1] Christian Puhrsch[1] Rob Fergus[1,2]<br>[1]Dept. of Computer Science, Courant Institute, New York University [2] Facebook AI Research |
| 论文出处 | 2014-NIPS |
| 创新点提炼 | 本文直接用神经网络预测深度。神经网络分为两个 部分：一部分估计场景的全局结构，另一部分用局部信息做精细化。网络用一个损失来训练。这个损失在普遍的尺度相关误差加入尺度不变误差(scale-invariant error)，即考虑像素位置的深度相关性（depth relations between pixel locations）和点的误差(pointwise error)。 |
| 个人想法 | |

论文方法及结论：

**1. 论文提出的问题**

对于立体图像（stereo mages）局部一致足够做估计，但是**从单张图中找到深度相关性是相对而言不直观的**，它需要从各种各样的线索中找到全局信息和局部信息，并将它们结合在一起。此外，这个任务本质上是不明确的，在技术上是不适定问题，**它在所有的尺度中有大量的不确定性**。给出一张图，有无穷多个可能的世界场景能够产生它。因此深度预测是有相当的精度，但是至少有一个主要的不确定存在，即全局尺度。

**2. 解决的方法**

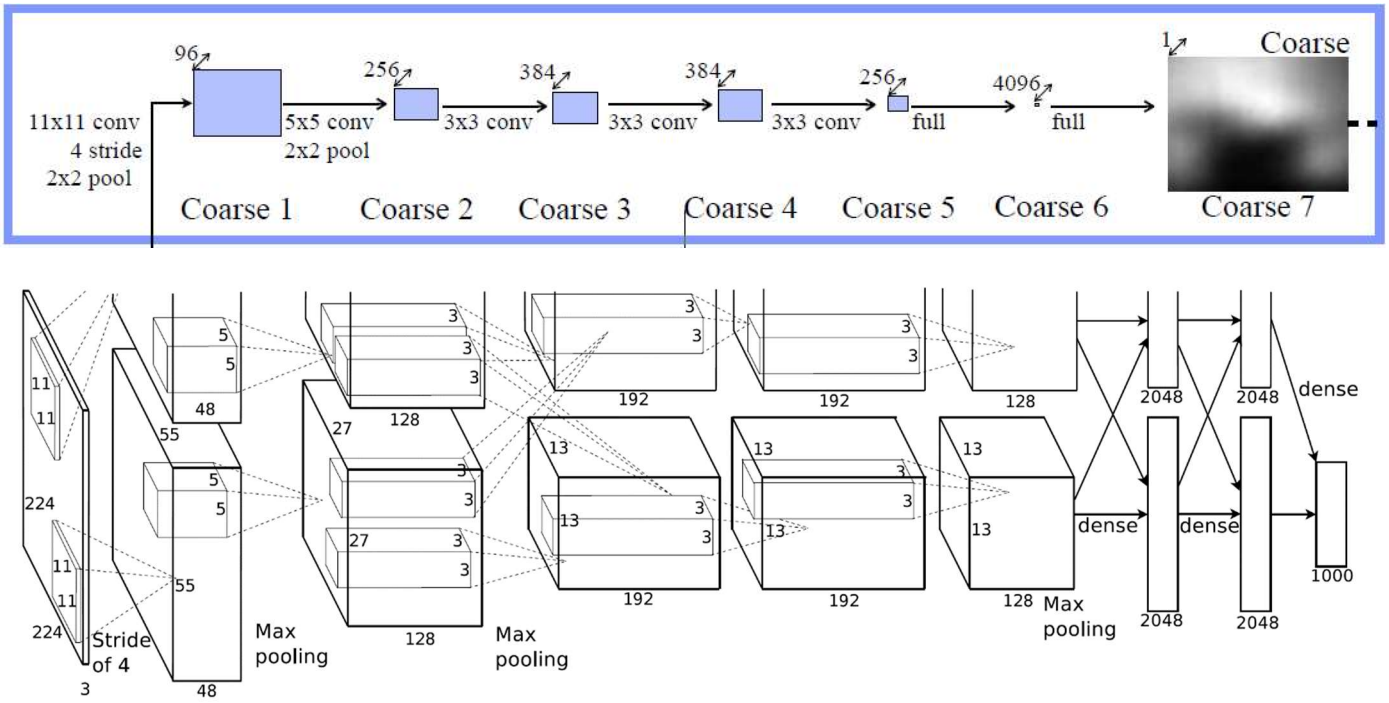



| | | Coarse | | | | | Fine |
|---|---|---|---|---|---|---|---|
| Layer | input | 1 | 2,3,4 | 5 | 6 | 7 | 1,2,3,4 |
| Size (NYUDepth) | 304x228 | 37x27 | 18x13 | 8x6 | 1x1 | 74x55 | 74x55 |
| Size (KITTI) | 576x172 | 71x20 | 35x9 | 17x4 | 1x1 | 142x27 | 142x27 |
| Ratio to input | /1 | /8 | /16 | /32 | – | /4 | /4 |

Figure 1: Model architecture

该网络使用了两个 stacks，一个是 coarse-scale network，也是该模型的第一步，进行对图像深度的 global level 预测，然后 coarse network 的 output 会继续进入第二个 fine-scale network，与原图像输入后 concentrate。这样 fine-scale network 预测的 details 就可以整合到 coarse network 的预测上。

### 2.1 Global Coarse-Scale Network

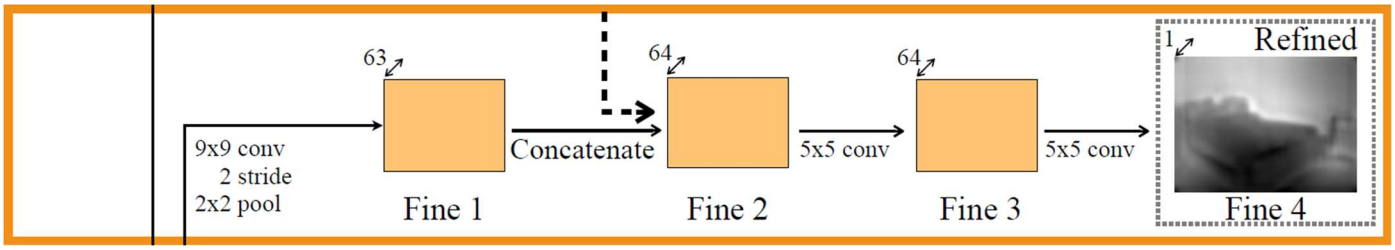



Coarse-Scale Network（AlexNet(NIPS 2012)结构，除了最后一层从 1000 个分类器换做了一个 coarse 的 depth map）的任务是利用全局的场景视野预测出整个场景的深度图结构，上层 layer 是全连接层，包含了整个场景的信息，底层和中层通过最大池化操作来联合图像各个部分的信息。这样网络就可以集合对整个场景的全局理解来预测深度。在单幅图像中，这种对全局的理解需要高效地利用一些线索，如灭点，物体的位置等，而局部地视野是无法注意到这些的。输出的深度图像只是大概的深度信息，对细小边缘响应不强。

输出的空间维度大于最顶部卷积 feature map 的维度。但是作者不限制 output 在 feature map 大小上，或者预测传递到最终网络之前依赖 hardcoded 的上采样，而是允许整个顶层学习更大区域上的模板（本质上是让网络自己学习如何从 feature 中学习上采样）。

所有 layer 都采用 ReLU 激活函数，在第六层 layer 使用 dropout。Coarse-Scale Network 的 1-5 层在 ImageNet Classification task 上进行预训练，**作者发现预训练的 performance 会比随机初始化网络好**。

### 2.2 Local Fine-Scale Network




在用全局的角度预测粗糙的深度图之后，作者用了第二个网络（Local Fine-Scale Network）做了局部优化。这个部分的目标是编辑它接收到的之前网络的粗糙预测，使之变得规整（有更多局部细节，像目标和墙边缘）。精细化尺度网络堆只包括卷积层，在第一步边缘特征有池化层。

相对于粗糙网络是“看见”整个场景，在微调网络中，每一个输出单元（output unit）的视觉野对应的是输入的 45×45 像素。卷积层将特征图缩放目标输出的大小尺寸，也就是输入的四分之一分辨率。

粗糙网络的输出是作为一张 low-level 的特征图。根据设计，粗糙预测结果的空间尺寸是和微调网络的第一层输出（池化后）一样的。把它们串联在一起。之后的层利用零填充卷积保持相同的尺寸。所有的隐藏单元用了

Relu，最后一层是线性的，因为它要预测深度。作者先用目标的 ground truth 训练粗糙网络,然后训练精细化尺度的网络（保持粗糙尺度网络固定）。

**2.3 Scale-Invariant Error**

在单目深度估计的问题中，从理论上说单目是无法获得尺度信息的，深度学习可以从大量的数据中学习到场景的尺度信息。但是，如果直接使用 RMSE 的 loss 函数来进行网络的训练，没有对图像尺度进行约束，导致估计得到的深度图像可能元素间相对值是准确的，但是整体深度和 ground truth 给出的深度存在尺度上的差异。因此作者给 RMSE 的 loss 函数添加了一个约束项用以测量场景中各点之间的关系，而不考虑绝对全局尺度（just finding the average scale of the scene accounts for a large fraction of the total error）。

设 $y$ 和 $y*$ 分别是预测得到的深度图和 ground truth，他们均拥有 $n$ 个像素，而 $y_i$ 和 $y_i*$ 分别表示预测深度图和 ground truth 上的第 $i$ 个像素。定义 $log$ 空间下的平均误差如下

$$D(y,y^*)=\frac{1}{n}\sum_{i=1}^{n}\left(\log y_i-\log y_i^*+\alpha(y,y^*)\right)^2$$

其中$\alpha(y,y^*)=\frac{1}{n}\sum_i(\log y_i^*-\log y_i)$是 α对于给定的 $(y,y*)$ 最小化误差的一个值。对于任意预测 $y$ ，$e^{\alpha}$ 是对应 ground truth 最合适的尺度。所有尺度 y 都有同样的错误率，进而使尺度不变（因为 $e^{\alpha}=\prod_{i=1}^{n}\sqrt[n]{\frac{y_i^*}{y_i}}$ 表示的是平均尺度因子，同时 $e^{\alpha}$ 正相关函数，因此对 $e^{\alpha}$ 的最小化可以看作为对 α 的最小化。）。

设定 $d_i=\log y_i-\log y_i^*$，以下与上式等价

$$D(y,y^*)=\frac{1}{2n^2}\sum_{i,j}\left(\left(\log y_i-\log y_j\right)-\left(\log y_i^*-\log y_j^*\right)\right)^2$$

$$=\frac{1}{n}\sum_i d_i^2-\frac{1}{n^2}\sum_{i,j}d_id_j=\frac{1}{n}\sum_i d_i^2-\frac{1}{n^2}\left(\sum_i d_i\right)^2$$

**（附：推导）**

$$D(y,y^*)=\frac{1}{2n}\sum_{i=1}^{n}(logy_i-logy_i^*+\alpha(y,y^*))^2=\frac{1}{2n}\sum_{i=1}^{n}(logy_i-logy_i^*+\frac{1}{n}\sum_j logy_j^*-\frac{1}{n}\sum_j logy_j)^2$$

$$=\frac{1}{2n^2}\sum_{i,j}d_i^2+d_j^2-2d_id_j=\frac{1}{2n^2}(\sum_i d_i^2+\sum_j d_j^2-2\sum_{i,j}d_id_j)$$

$$=\frac{1}{2n}\sum_{i=1}^{n}(\frac{1}{n}\sum_j logy_i-\frac{1}{n}\sum_j logy_i^*+\frac{1}{n}\sum_j logy_j^*-\frac{1}{n}\sum_j logy_j)^2=\frac{1}{2n^2}\sum_{i,j}((logy_i-logy_j)-(logy_i^*-logy_j^*))^2$$

$$=\frac{1}{2n^2}\sum_{i,j}((logy_i-logy_i^*)-(logy_j-logy_j^*))^2=\frac{1}{2n^2}\sum_{i,j}(d_i-d_j)^2$$

$$=\frac{1}{2n}\sum_{i,j}d_i^2+d_j^2-2d_id_j=\frac{1}{2n}(\sum_i d_i^2+\sum_j d_j^2-2\sum_{i,j}d_id_j)=\frac{1}{2n}(2\sum_i d_i-2\sum_{i,j}d_id_j)=\frac{1}{n}\sum_i d_i-\frac{1}{n^2}\sum_{i,j}d_id_j=\frac{1}{n}\sum_i d_i-\frac{1}{n^2}(\sum_i d_i)^2$$

等价一通过在输出比较像素对 $i,j$ 的相对关系表达误差：为了获得小误差，在预测中的每一对像素在深度上要作差，并且要累计与对应 ground truth 里面的像素对的相似性。

等价二将距离衡量与 L2 联系起来，但是加了一个项 $-\frac{1}{n^2}\sum_{i,j}d_id_j$ ，如果它们错误是同一方向的，则这一项奖励它们；如果它们错误是反方向的，这一项将惩罚它们。这样一个不是最完美的估计会有一个更低的误差，当它们的错误方向一致。

为了做性能估计，作者也尝试了尺度不变误差作为训练误差。作者设置每个样本训练损失为

$$L(y, y^*) = \frac{1}{n}\sum_i d_i^2 - \frac{\lambda}{n^2}\left(\sum_i d_i\right)^2$$

当 $\lambda = 0$，得到的只是逐像素误差。

当 $\lambda = 1$，得到的是尺度不变误差。

当 $\lambda = 0.5$，得到的是较好的绝对尺度估计的结果。

在训练的时候，大多数的目标深度有一些值是没有的，特别是目标边界，窗口和镜面。作者简单的 mask 那些不可用的点。

## 3. 实验结果结论

| | Mean | Make3D | Ladicky&*al* | Karsch&*al* | Coarse | Coarse + Fine | |
|---|---|---|---|---|---|---|---|
| threshold $\delta < 1.25$ | 0.418 | 0.447 | 0.542 | – | **0.618** | 0.611 | higher |
| threshold $\delta < 1.25^2$ | 0.711 | 0.745 | 0.829 | – | **0.891** | 0.887 | is |
| threshold $\delta < 1.25^3$ | 0.874 | 0.897 | 0.940 | – | 0.969 | **0.971** | better |
| abs relative difference | 0.408 | 0.349 | – | 0.350 | 0.228 | **0.215** | |
| sqr relative difference | 0.581 | 0.492 | – | – | 0.223 | **0.212** | lower |
| RMSE (linear) | 1.244 | 1.214 | – | 1.2 | **0.871** | 0.907 | is |
| RMSE (log) | 0.430 | 0.409 | – | – | **0.283** | 0.285 | better |
| RMSE (log, scale inv.) | 0.304 | 0.325 | – | – | 0.221 | **0.219** | |

Table 1: Comparison on the NYUDepth dataset

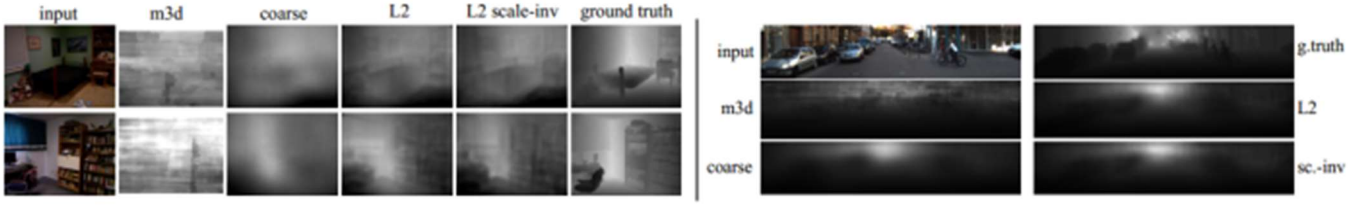



Figure 3: Qualitative comparison of Make3D, our method trained with $l_2$ loss ($\lambda = 0$), and our method trained with both $l_2$ and scale-invariant loss ($\lambda = 0.5$).

| | Mean | Make3D | Coarse | Coarse + Fine | |
|---|---|---|---|---|---|
| threshold $\delta < 1.25$ | 0.556 | 0.601 | 0.679 | **0.692** | higher |
| threshold $\delta < 1.25^2$ | 0.752 | 0.820 | 0.897 | **0.899** | is |
| threshold $\delta < 1.25^3$ | 0.870 | 0.926 | **0.967** | **0.967** | better |
| abs relative difference | 0.412 | 0.280 | 0.194 | **0.190** | |
| sqr relative difference | 5.712 | 3.012 | 1.531 | **1.515** | lower |
| RMSE (linear) | 9.635 | 8.734 | 7.216 | **7.156** | is |
| RMSE (log) | 0.444 | 0.361 | 0.273 | **0.270** | better |
| RMSE (log, scale inv.) | 0.359 | 0.327 | 0.248 | **0.246** | |

Table 2: Comparison on the KITTI dataset.

通过结合全局和局部视野的信息，它可以表现的不错。作者用两个深度网络：一个估计全局深度结构；另一个在更好的分辨率上在它的局部精细化。在 NYU Depth 和 KITTI 上取得 state-of-the-art。

## 4. 存在的问题

暂无

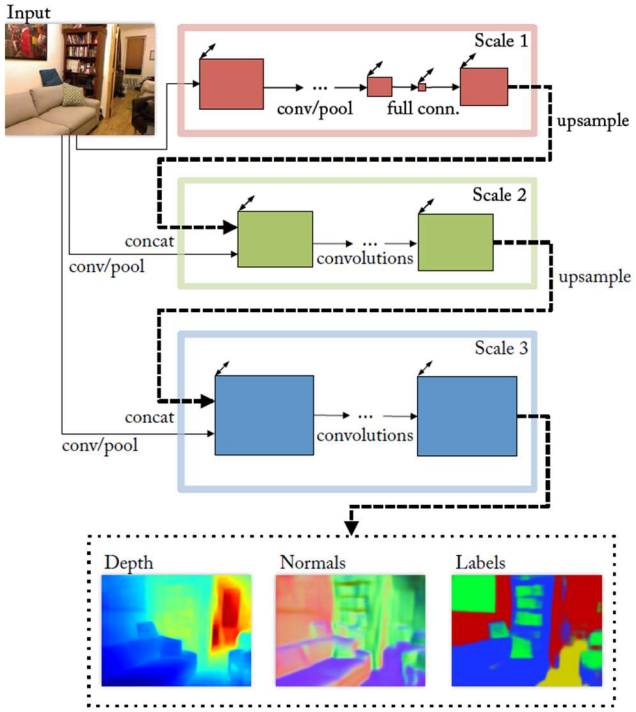
| | |
|---|---|
| 论文标题 | Predicting Depth, Surface Normals and Semantic Labels with a Common Multi-Scale Convolutional Architecture |
| 作者及单位 | David Eigen[1] Rob Fergus[1,2]<br>[1]Dept. of Computer Science, Courant Institute, New York University [2] Facebook AI Research |
| 论文出处 | 2015-ICCV |
| 创新点提炼 | （1） scale 1 上将 AlexNet 换成了更深的 VGG16；<br>（2） 增加了第三尺度网络，为了获取更高的输出分辨率，最终的输出分辨率为输入图片的一半，相比之前工作中的 1/4 有了提高；<br>（3） 第一尺度的输出并未采用直接的预测图，而是选用多通道的特征图（之前工作第一尺度输出的是一个模糊的全局深度图），并将其上采样作为第二尺度的输入。目的是想两个尺度的一起训练，从而简化训练步骤且提升了最终的效果。 |
| 个人想法 | |

论文方法及结论：

**1. 论文提出的问题**

2014-NIPS 中指出，未来工作中将结合更多 3D 结合信息，如表面法向量，这将进一步改善深度估计和法线估计的总体性能。此外，作者也希望通过多次应用 Local Fine-Scale Network 使得深度图像达到输入图像的分辨率。

**2. 解决的方法**

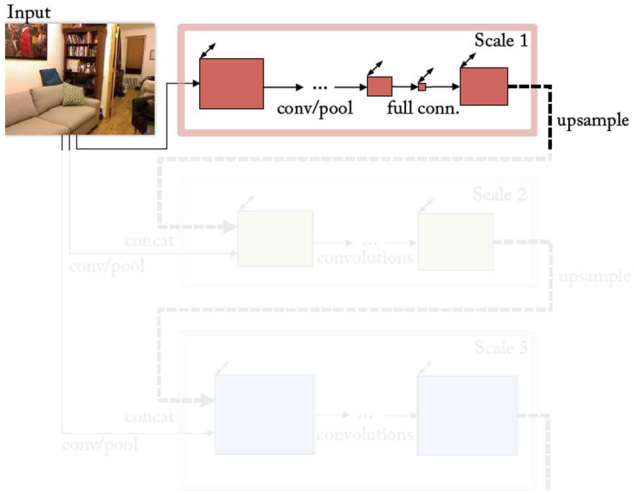



| | Layer | 1.1 | 1.2 | 1.3 | 1.4 | 1.5 | 1.6 | 1.7 | upsamp |
|---|---|---|---|---|---|---|---|---|---|
| Scale 1 (AlexNet) | Size | 37x27 | 18x13 | 18x13 | 18x13 | 8x6 | 1x1 | 19x14 | 74x55 |
| | #convs | 1 | 1 | 1 | 1 | 1 | – | – | – |
| | #chan | 96 | 256 | 384 | 384 | 256 | 4096 | 64 | 64 |
| | ker. sz | 11x11 | 5x5 | 3x3 | 3x3 | 3x3 | – | – | – |
| | Ratio | /8 | /16 | /16 | /16 | /32 | – | /16 | /4 |
| | l.rate | 0.001 | 0.001 | 0.001 | 0.001 | 0.001 | see text | | |
| | Layer | 1.1 | 1.2 | 1.3 | 1.4 | 1.5 | 1.6 | 1.7 | upsamp |
| Scale 1 (VGG) | Size | 150x112 | 75x56 | 37x28 | 18x14 | 9x7 | 1x1 | 19x14 | 74x55 |
| | #convs | 2 | 2 | 3 | 3 | 3 | – | – | – |
| | #chan | 64 | 128 | 256 | 512 | 512 | 4096 | 64 | 64 |
| | ker. sz | 3x3 | 3x3 | 3x3 | 3x3 | 3x3 | – | – | – |
| | Ratio | /2 | /4 | /8 | /16 | /32 | – | /16 | /4 |
| | l.rate | 0.001 | 0.001 | 0.001 | 0.001 | 0.001 | see text | | |
| | Layer | 2.1 | 2.2 | 2.3 | 2.4 | 2.5 | | | upsamp |
| Scale 2 | Size | 74x55 | 74x55 | 74x55 | 74x55 | 74x55 | | | 147x109 |
| | #chan | 96+64 | 64 | 64 | 64 | $C$ | | | $C$ |
| | ker. sz | 9x9 | 5x5 | 5x5 | 5x5 | 5x5 | | | – |
| | Ratio | /4 | /4 | /4 | /4 | /4 | | | /2 |
| | l.rate | 0.001 | 0.01 | 0.01 | 0.01 | 0.001 | | | |
| | Layer | 3.1 | 3.2 | 3.3 | 3.4 | | | | final |
| Scale 3 | Size | 147x109 | 147x109 | 147x109 | 147x109 | | | | 147x109 |
| | #chan | 96+$C$ | 64 | 64 | $C$ | | | | $C$ |
| | ker. sz | 9x9 | 5x5 | 5x5 | 5x5 | | | | – |
| | Ratio | /2 | /2 | /2 | /2 | | | | /2 |
| | l.rate | 0.001 | 0.01 | 0.01 | 0.001 | | | | |

Figure 1. Model architecture. $C$ is the number of output channels in the final prediction, which depends on the task. The input to the network is 320x240.

本文设计了一种通用的多尺度网络，仅需要通过少量的修改就能适用于三个不同的计算机视觉任务：深度估计，表面法向量估计，语义分割。给定输入图片，网络能够直接回归出输出图，如深度图、法向量图、分类图。网络结构在之前工作上加以改进，堆叠了 3 层卷积神经网络，从低分辨率逐渐升到高分辨率，进一步改善了图像细节。测试中，网络的输出是实时的，达到了 30Hz 左右，同时生成的结果在三项任务中都获得了当前最佳性能，证明了网络的多面性。

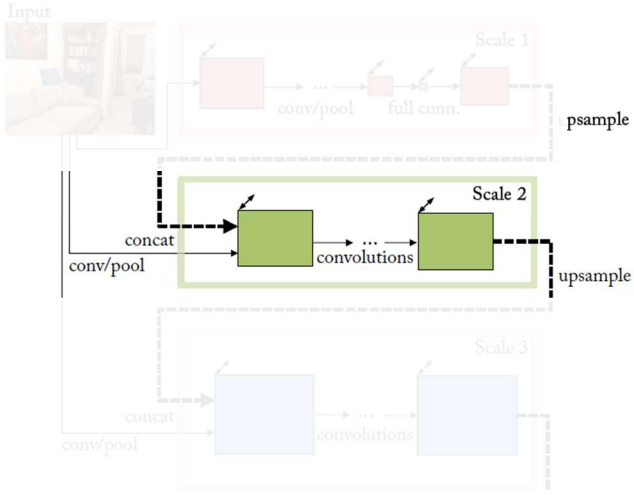
### 2.1 scale 1：Full-Image View




- Coarsest scale
- Full-image field of view at the coarst scale (1/16 scale)
- AlexNet (smaller model size) and VGG
- Upsamping to be the same size as the input for the next scale
- Important for producing smooth output
- VGG version significantly outperforms the smaller AlexNet version

第一尺度网络是整个网络最重要的一环，后面实验部分也证明了这点。它预测出的是一个模糊但是反映图像空间变换的全局特征图，因为后接了两个全连接层，因此其感受野是全局的，最终的输出尺度是原始图像的 1/16，但是有 64 个特征通道，之后上采样到 1/4 作为第二尺度网络的输入。

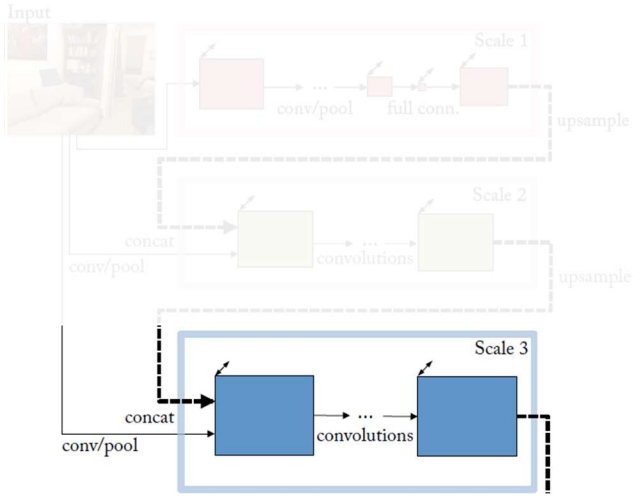
### 2.2 scale 2：Predictions




- Prediction at scale 2 (1/4 scale)
- 5 layers
- Concatenate the output of previous scale with those from a single layer of convolution and pooling
- Output has the same size, with the number of channel depending on the task

第二尺度网络生成中等分辨率的预测图（比上文的特征粗提取网络的视角虽然狭窄了，但是会提供了更多的细节）。其输入是由原图经过简单的卷积加池化形成的特征图以及第一尺度网络输出上采样后的特征图连接而成，最终输出的是原始图片 1/4 分辨率大小的预测图（深度图等）。

### 2.3 scale 3：Higher Resolution




- Refines to higher resolution at scale 3（1/2 scale）
- 4 layers
- Concatenate the output of scale 2
- Provide details while maintaning the spatially coherent structure from previous scales

第三尺度网络进一步改善预测结果，使其拥有更高的分辨率，其输入与第二层类似，最终的输出分辨率为原图的 1/2，且获得了更多的细节信息。

对于深度估计、表面法向量估计以及语义分割，作者使用了不同的损失函数以及相应的训练数据。

**1) 深度估计、**

设 $D, D*$ 别为估计出的对数深度图和真实的对数深度图（每个深度值取 log），$d = D - D*$ 为两者差值，定义最终的损失函数为

$$L_{depth}(D, D^*) = \frac{1}{\mathrm{n}} \sum_{\mathrm{i}} d_i^2 - \frac{1}{2\mathrm{n}^2} \left( \sum_{\mathrm{i}} d_i \right)^2 + \frac{1}{n} \sum_i \left[ (\nabla_x d_i)^2 + \left( \nabla_y d_i \right)^2 \right]$$

其中所有计算损失的像素点均为有效点（有 ground truth 的像素点），$\nabla_x d_i$ 和 $\nabla_y d_i$ 分别是深度差值的水平和垂直梯度。这里定义的损失函数与之前工作中提出的损失函数相似，仅仅是在原来的基础上加入了梯度项 $(\nabla_x d_i)^2 + \left(\nabla_y d_i\right)^2$，这一项直观的理解就是希望深度差值能够尽量一致（若 $d_i$ 均为 1，则该项为 0），即保证相对误差小。论文中也说到这一项使最终预测结果不仅与真实梯度值相近，同时也有相似的局部结构。

**2) 表面法向量**

表面法向量是三维的，因此最终的输出通道要从一个通道改成三个，最终输出的法向量在计算损失函数之前要先归一化，定义损失函数如下

$$L_{\text{normals}}(N, N^*) = -\frac{1}{\mathrm{n}} \sum_i N_i \cdot N_i^* = -\frac{1}{n} N \cdot N^*$$

其中 $N, N^*$ 是经过归一化后的估计表面法向量图和真实表面法向量图，同样这里只针对有效点计算损失。只看损失函数，其目的是希望估计的法向量和真实的法向量点乘结果大，那么两者角度就小，相似性就高。对于真实的表面法向量，作者用 Silberman 的方法，从实际深度值得到真实的表面法向量。

**3) 语义分割**

语义分割中每个像素点对应一个类别，因此网络的输出通道数与类别数一致，损失函数使用交叉熵损失，如下所示

$$L_{\text{semantic}}(C, C^*) = -\frac{1}{\mathrm{n}} \sum_i C_i^* \log(C_i)$$

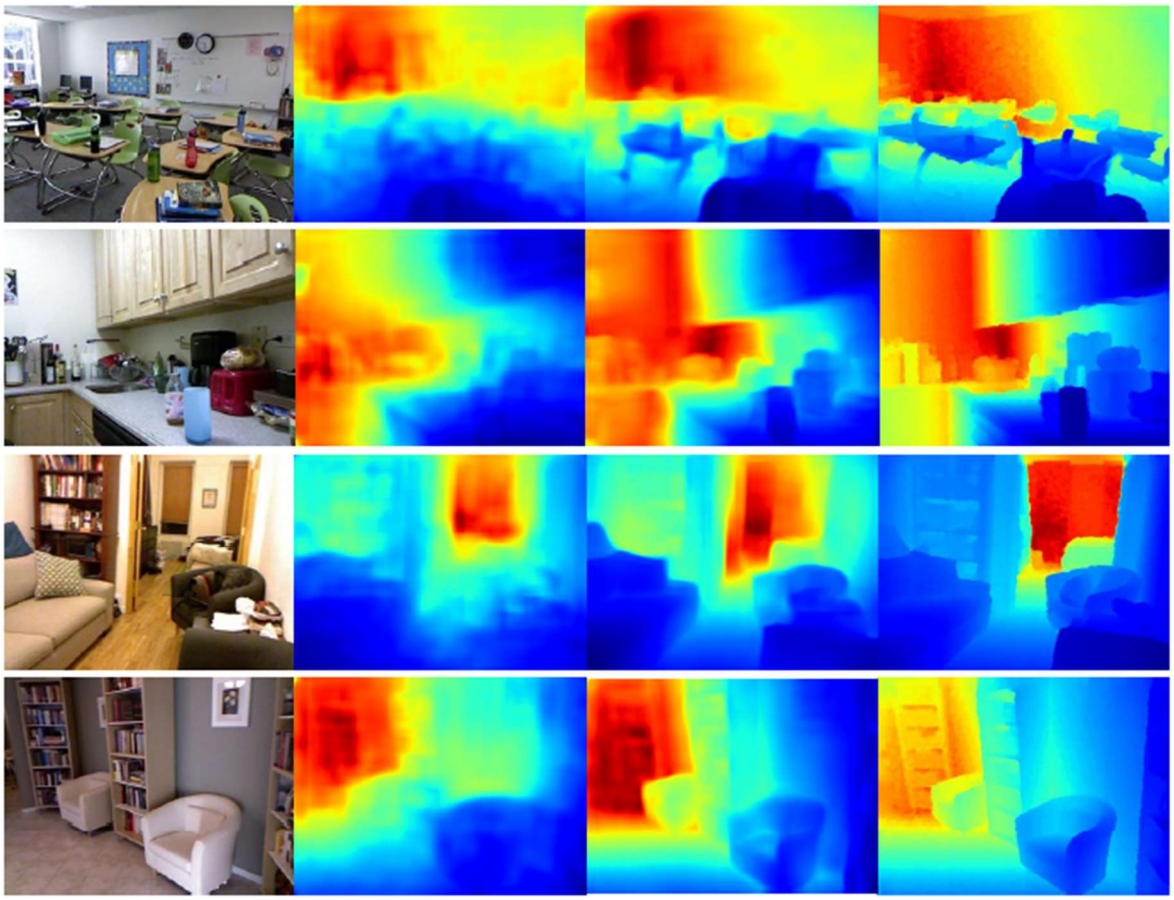
其中 $C_i = e^{z_i} / \sum_c e^{zi,c}$ 是像素点 $i$ 的类别预测结果。在进行语义分割时，作者将真实的深度图及表面法向量图一起作为网络的输入。具体来讲，在第一尺度网络仅使用 RGB 图作为输入，在第二尺度网络在原有网络输入基础上新增深度图以及表面法向量图，但是三者并非直接通道相连作为网络输入，而是分别一个网络进行训练，最后将三个网络的输出通道相连，作为第二尺度网络的输出。这么做是为了保证三者在低层次特征上保持相对独立性。

训练模型的过程中，使用随机梯度下降法 SGD，先联合训练第一尺度和第二尺度网络，然后固定这两部分的参数训练第三尺度网络，为了加快训练速度，作者在实际训练第三尺度网络时，将输入的特征图和原图随机裁剪成 74 * 55 大小（NYUDepth 数据集上，原始输入的一半），在不太影响结果精度的情况下提高了 3 倍左右的速度。

为了减小计算量，作者将深度估计和表面法向量估计这两个任务合并在一起训练，即输入一张 RGB 图能同

时得到其深度图和表面法向量图。实际操作中两者共用第一尺度网络，而后面两个网络则完全独立，我认为作者这么做的原因还是考虑到两个任务的关联性并不那么大，因此仅共用底层特征。合并后的网络训练速度是两个完全独立网络训练速度的 1.6 倍左右。

### 3. 实验结果结论



(a) (b) (c) (d)

Figure 2. Example depth results. (a) RGB input; (b) result of [8]; (c) our result; (d) ground truth. Note the color range of each image is individually scaled.

| Depth Prediction | | | | | | | |
|---|---|---|---|---|---|---|---|
| | Ladicky[20] | Karsch[18] | Baig [1] | Liu [23] | Eigen[8] | Ours(A) | Ours(VGG) |
| $\delta < 1.25$ | 0.542 | – | 0.597 | 0.614 | 0.614 | 0.697 | **0.769** |
| $\delta < 1.25^2$ | 0.829 | – | – | 0.883 | 0.888 | 0.912 | **0.950** |
| $\delta < 1.25^3$ | 0.940 | – | – | 0.971 | 0.972 | 0.977 | **0.988** |
| abs rel | – | 0.350 | 0.259 | 0.230 | 0.214 | 0.198 | **0.158** |
| sqr rel | – | – | – | – | 0.204 | 0.180 | **0.121** |
| RMS(lin) | – | 1.2 | 0.839 | 0.824 | 0.877 | 0.753 | **0.641** |
| RMS(log) | – | – | – | – | 0.283 | 0.255 | **0.214** |
| sc-inv. | – | – | 0.242 | – | 0.219 | 0.202 | **0.171** |

Table 1. Depth estimation measurements. Note higher is better for top rows of the table, while lower is better for the bottom section.

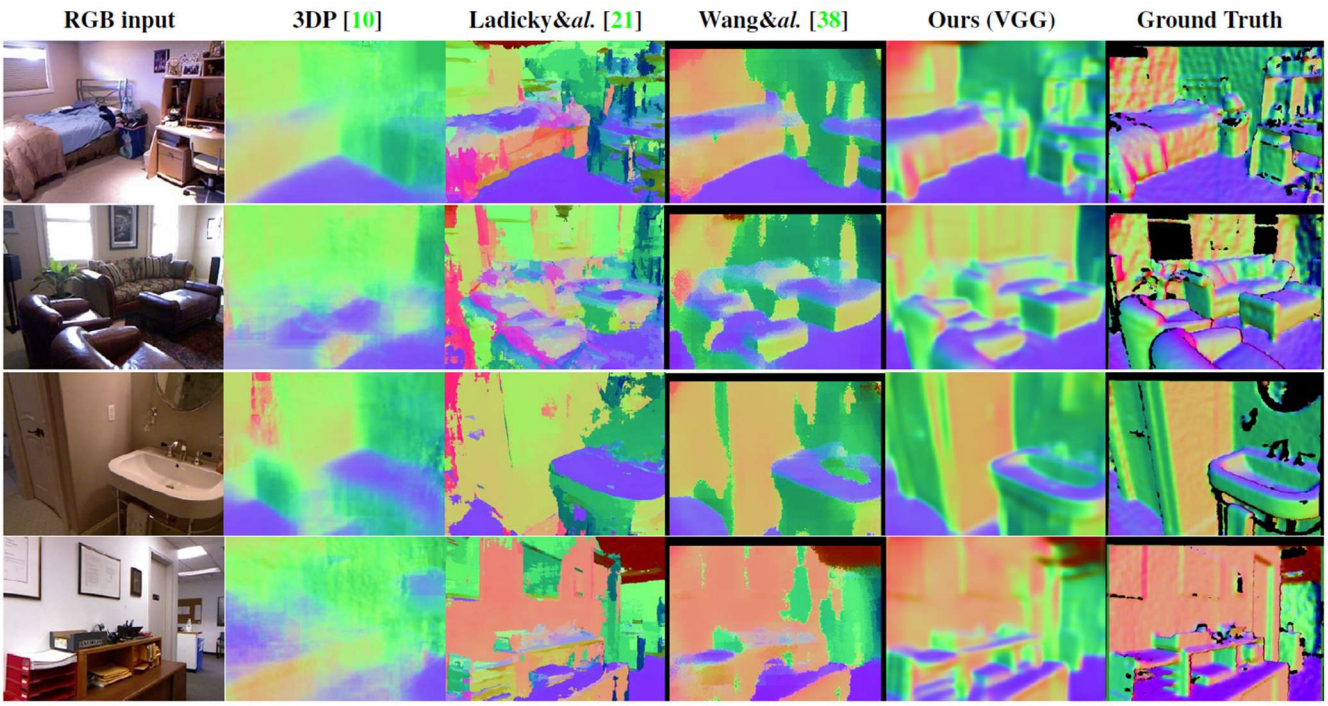



Figure 3. Comparison of surface normal maps.

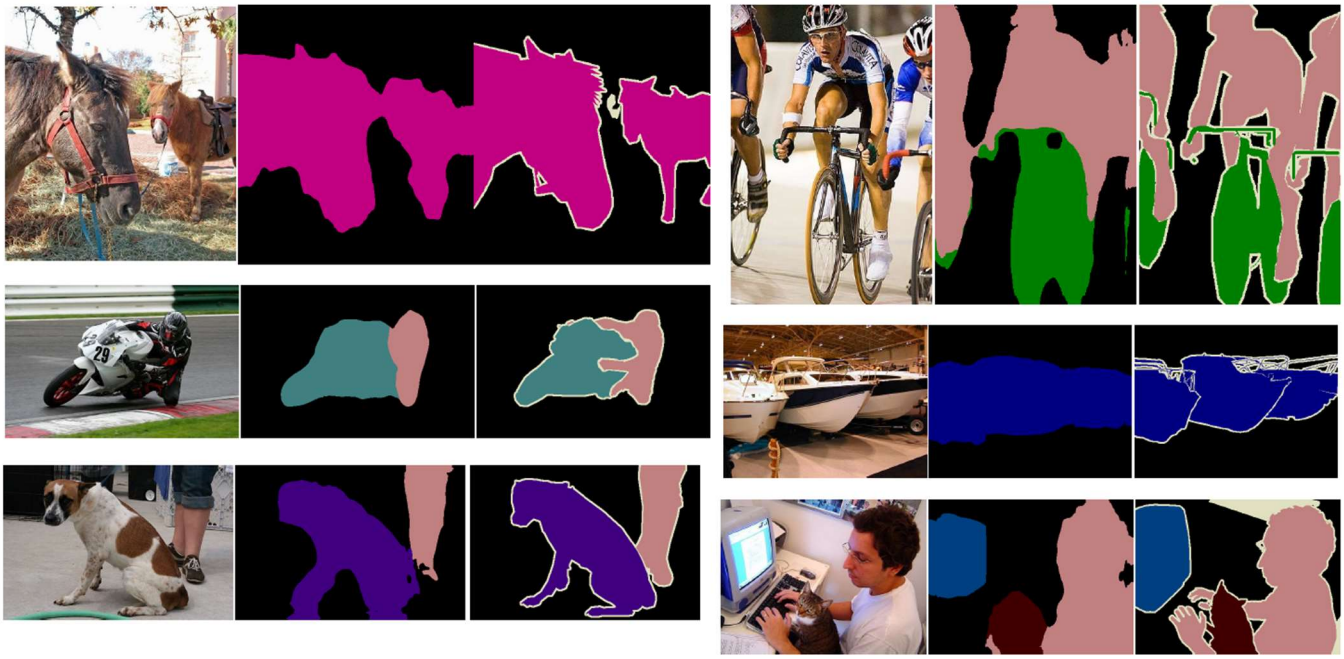


Figure 5. Example semantic labeling results for Pascal VOC 2011. For each image, we show RGB input, our prediction, and ground truth.

## 4. 存在的问题

暂无