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	读论文	读论文	读论文	查阅无监督深度估计的论文 阅读论文中的无监督方法
星期二	处理杂事	读论文	预答辩	对无监督与自监督 的区别不是很明晰
星期三	改答辩稿	训练 tip	预答辩	主要准备开题答辩
星期四	改 ppt	改开题报告	处理杂事	主要准备开题答辩
星期五	读论文	开题答辩	休息	看了一下深度估计的 sota
星期六	读论文	读论文	读论文	阅读深度估计的 sota 论文

二、下一步计划（按任务划分）

编号	工作内容	目标	相关配合
1	在更大的数据集上训练 tip	1 月 13 日完成	无
2	阅读一篇无监督的 sota	1 月 16 日完成	无
3	训练深度估计的两个项目	1 月 20 日完成	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：本周主要花在了开题答辩和深度估计的研究现状调研，花在代码上的时间较少
2	课题方面：通过阅读论文，除了 encoder-decoder 在深度估计任务上被广泛应用外，transformer/嵌入不同的空间也可能对涉及彩色图像和深度图像的任务有帮助
3	其他思考：关于深度估计中自监督与无监督的区别不是很明晰

Benchmarks

TREND	DATASET	BEST METHOD	PAPER TITLE	PAPER	CODE	COMPARE
	KITTI Eigen split	AdaBins	AdaBins: Depth Estimation using Adaptive Bins			See all
	NYU-Depth V2	AdaBins	AdaBins: Depth Estimation using Adaptive Bins			See all
	KITTI Eigen split unsupervised	FeatDepth-MS-OR	Feature-metric Loss for Self-supervised Learning of Depth and Egomotion			See all

本周主要对深度估计方向的论文进行了阅读，一篇为无监督的深度估计，一篇为在 kitti 和 nyu depth 两个数据集上取得了 sota 的有监督的单目深度估计。此外还在少量数据集上对 2020-tip 进行了训练。

阅读无监督深度估计的过程中，暂时没有对自监督和无监督比较明确划分的点。几篇中文的深度估计综述类论文主要将单目深度估计划分为有监督/半监督/无监督（按照有无监督），在无监督部分，陈述的论文主要是题目中明确包含无监督的论文（用立体视图/单目视频来做），未提到自监督的论文。但就综述论文中来看，无监督的方法也需要建立图像间的约束。

另一篇有监督的单目深度估计采用了目前在深度估计中广泛使用的 encoder-decoder 作为骨干网络，并在高分辨率的图像上使用 vit（vision transformer）进行优化。

此外，由于本周较多时间用在了开题答辩，花在代码上的时间较少，下周将重点放在训练上。

四、论文总结

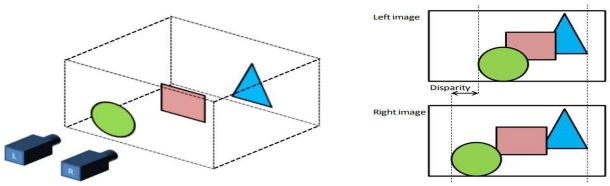
论文标题	Unsupervised Monocular Depth Estimation with Left-Right Consistency
作者及单位	Clement Godard, Oisin Mac Aodha, Gabriel J. Brostow University College London
论文出处	2017-CVPR
创新点提炼	该论文提出的方法在深度估计的结果上有了显著改善，且其方法是在稠密深度图的估计上表现了很快的计算速度，在 gpu 估计一张 512 × 256 的图像约耗费 35 毫秒，主要贡献如下： 1. 提出了端到端的无监督单目深度估计网络，并提出了新的损失函数以迫使网络中的左右深度一致。 2. 对几个训练损失和图像形成模型进行评估，强调了本文方法的有效性。 除了展示具有挑战性的驾驶数据集的最新结果，本文还展示了提出的模型可以推广到三个不同的数据集，包括自己收集的一个新的室外城市数据集，并公开提供该数据集。
个人想法	本文主要基于立体视图进行无监督的单目深度估计，在当时取得了很好的估计结果，但对自己的研究方向感觉帮助不大（从输入数据和后续处理方法来看）。阅这篇论文主要想探究无监督单目深度估计与自监督单目深度估计的区别，但并未从文中发现有明显的界限。

论文方法及结论：

1. 论文提出的问题

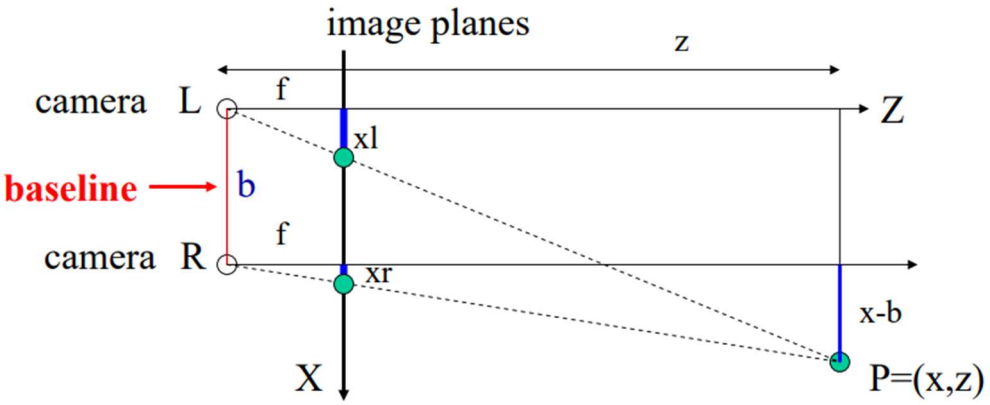
有监督的深度估计方法依赖于大量准确的视差真值数据和立体视图对，这些数据在现实世界中难以获得且花费较高。

2. 解决的方法



一只眼睛看到的图像是二维的，二维的信息是无法用来表示三维的空间的，如上图所示，虽然处于同一水平面上的照相机 L,R 拍摄了同一个物体，两者之间产生的图片是不同的。并且这种不同是不能通过平移生成的图片所消除的。离照相机近的物体偏离的位置比较大，离照相机远的物体偏离的比较少。

这种差异性的存在就是三维空间带来的。同时同一水平线上的两个照相机拍摄到的照片是服从以下物理规律的：



在图中，z 为场景所距离我们的深度，X 为三维场景映射到的二维图像平面，也就是最终我们得到的二维图像所在的平面。f 为相机的焦距。b 为两个相机之间的距离，x_l 和 x_r 分别为相同物体在左右两个不同相机中成像的坐标。根据以上信息，和简单的三角形相似规律我们可以得到：$x-b=\frac{z}{f}*x_r; x=\frac{z}{f}*x_l \Rightarrow (x_l-x_r)=\frac{f*b}{z}$

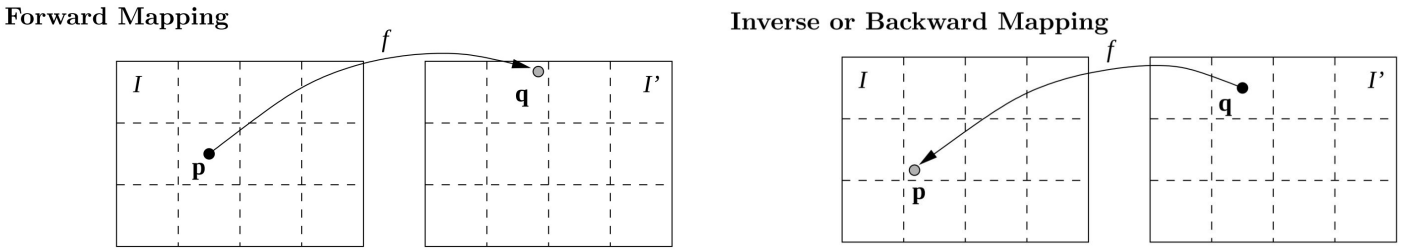
这里 (x_l-x_r) 就是我们常说的视差 d（disparity），代表了 x_r 这个点在照相机 L 和照相机 R 中成像的偏离值。也就是说这个值代表了左照相机中的像素需要通过平移 才能形成右照相机中相应的像素。所以两个视角之间的关系可以写作：$d=x_l-x_r \Rightarrow x_l=d+x_r$

假设有一个很强大的函数 （神经网络）F，使得 $F(I_r)=d$ ，那么就有：$I_r(x_r)=I_l(F(x_l)+x_l)$。

只要以 I_l 作为训练的输入，I_r 作为所对应的参考标准，建立如上关系的神经网络 F，通过大量的双目图片对的训练，得到的神经网络就是一个输入一张图片来预测所对应的时差的函数，这样就将一个没有约束的问题变成了符合如上规律的问题。

但是预测到的视差是连续的浮点数，如果使用 x_l+d 的套路，很有可能会落入到在（整数）像素点的位置，同时由于视差的不同，也有可能有一个 I_r 中的像素点接受多个来自 I_l 的像素点。而有一些点并没有相符合的视差，因为这些点由于视差的原因在原图中可能根本不可见。为了解决这个问题，一般采用 backward mapping 的方法。

这两种方法的区别在于，在 forward mapping 中，得到 I' 中的点可能会落在不是整数像素点的位置，这时只能通过最近原则将原图 I 中的像素点对应到 I' 中去，而在 Inverse mapping 中，从 I' 出发(也就是 x_r-d)，去寻找相对应的原图中的点，这样能够确保 I' 中的每一个点都有赋值不会出现空洞，并且如果 x_r-d 得到的原图中的点不属于（整数）像素点，这时可以通过插值的方法求得所对应非像素点的位置。一般在这里采用双线性



插值的方法，而且它在 sub-pixel level 是可导的。所以我们就可以 end-to-end 的来训练网络啦。

$$d = F(I_l)$$

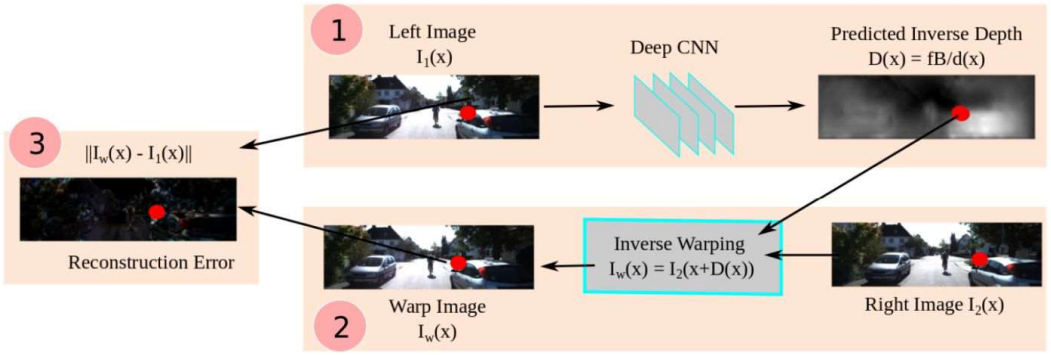
$$I_r^{'} = Mapping(I_l, d)$$

$$argming_\theta Loss(I_r', I_r)$$

在 Unsupervised CNN for Single View Depth Estimation: Geometry to the Rescue（ECCV 2016），采用的方法如下

$$d = F(I_l)$$

$$I_l^{'} = Mapping(I_r, d)$$



$$argming_\theta Loss(I_l', I_l)$$

而本文将两种深度估计的方法结合起来并得到了

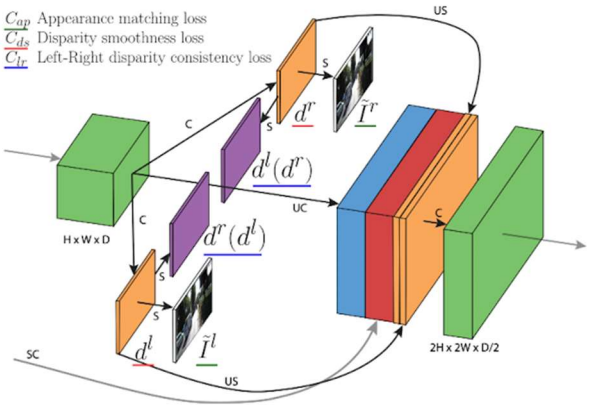


Figure 2. Our loss module outputs left and right disparity maps, d^l and d^r. The loss combines smoothness, reconstruction, and left-right disparity consistency terms. This same module is repeated at each of the four different output scales. C: Convolution, UC: Up-Convolution, S: Bilinear Sampling, US: Up-Sampling, SC: Skip Connection.

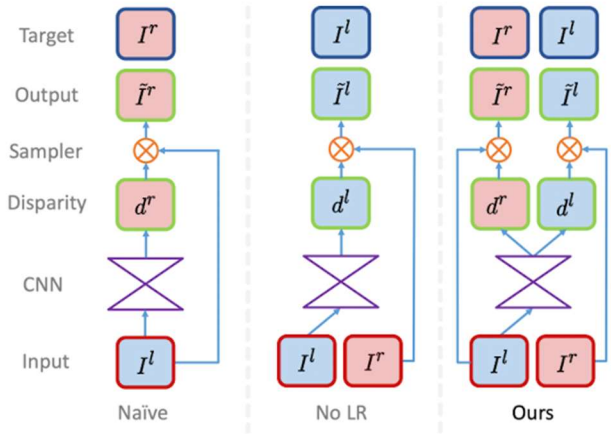


Figure 3. Sampling strategies for backward mapping. With naïve sampling the CNN produces a disparity map aligned with the target instead of the input. No LR corrects for this, but suffers from artifacts. Our approach uses the left image to produce disparities for both images, improving quality by enforcing mutual consistency.

更多的约束：

通过 I_l 进入神经网络可以求得 d_l 和 d_r ， 通过输入不同的参考图片在 mapping 中可以得到相对应的原图 $I_l^{'}$ 和右图 $I_r^{'}$ 。约束为 $C_s = \alpha_{ap}\left(C_{ap}^l + C_{ap}^r\right) + \alpha_{ds}(C_{ds}^l + C_{ds}^r) + \alpha_{lr}(C_{lr}^l + C_{lr}^r)$

第一对 C_{ap} 表示了图片重建的损失函数，在这里，作者采用了一个 SSIM 和 L1 相结合的损失函数，因为 L1 并不能很好的表示真实的图片分布

$$C_{ap}^l = \frac{1}{N}\sum_{i,j}\alpha\frac{1-\text{SSIM}(I_{ij}^l, \tilde{I}_{ij}^l)}{2} + (1-\alpha)\left\|I_{ij}^l - \tilde{I}_{ij}^l\right\|.$$

其次，理想情况下， d_l 和 d_r 之间也存在着与原图相同的视差关系，所以当预测的深度达到最有时，以下损失函数达到最小值，即本文提出的 left-right Consistency：

$$C_{lr}^l = \frac{1}{N}\sum_{i,j}\left|d_{ij}^l - d_{ij+d_{ij}^l}^r\right|.$$

关于损失的 C_{ds} ，作者提出 edge preserving 损失函数，其原因是深度的不连续性往往发生在边缘附近。提出了这个损失函数用来保证所得到的深度图的光滑性与图像梯度一致。

$$C_{ds}^l = \frac{1}{N}\sum_{i,j}|\partial_x d_{ij}^l|e^{-\|\partial_x I_{ij}^l\|} + |\partial_y d_{ij}^l|e^{-\|\partial_y I_{ij}^l\|}$$

这篇文章所使用的网络和之前的方法类似，均采用了 FCN 的方法进行训练，不同的是在 decoder 部分的最外面 4 层，作者都估计了当前特征大小所对应的视差的值，并且将它上采样后传递给了 decoder 的下层，这样能确保每一层都在做提取 disparity 这件事，同时也相当于做了一个 coarse-to-fine 的深度预测，同时由于我们采用了双线性差值，梯度的范围始终来自于周围的 4 个坐标点，coarse-to-fine 的预测能够让梯度来自于离当前位置更远的坐标点。

3. 实验结果结论

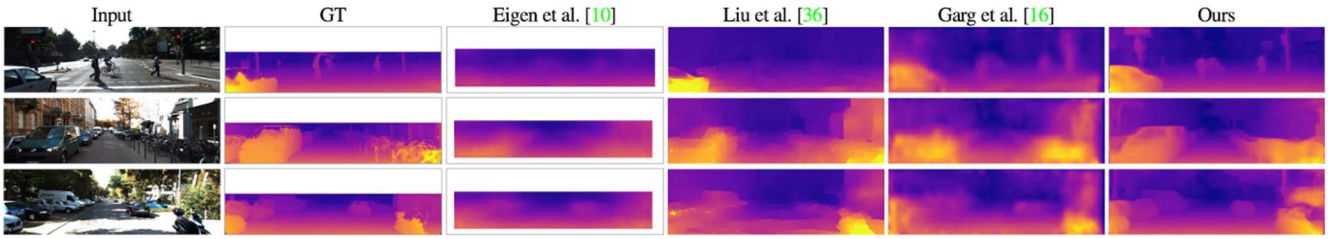


Figure 4. Qualitative results on the KITTI Eigen Split. The ground truth velodyne depth being very sparse, we interpolate it for visualization purposes. Our method does better at resolving small objects such as the pedestrians and poles.

Method	Supervised	Dataset	Abs Rel	Sq Rel	RMSE	RMSE log	$\delta<1.25$	$\delta<1.25^2$	$\delta<1.25^3$
Train set mean	No	K	0.361	4.826	8.102	0.377	0.638	0.804	0.894
Eigen et al. [10] Coarse °	Yes	K	0.214	1.605	6.563	0.292	0.673	0.884	0.957
Eigen et al. [10] Fine °	Yes	K	0.203	1.548	6.307	0.282	0.702	0.890	0.958
Liu et al. [36] DCNF-FCSP FT *	Yes	K	0.201	1.584	6.471	0.273	0.68	0.898	0.967
Ours No LR	No	K	0.152	1.528	6.098	0.252	0.801	0.922	0.963
Ours	No	K	0.148	1.344	5.927	0.247	0.803	0.922	0.964
Ours	No	CS + K	0.124	1.076	5.311	0.219	0.847	0.942	0.973
Ours pp	No	CS + K	0.118	0.923	5.015	0.210	0.854	0.947	**0.976**
Ours resnet pp	No	CS + K	**0.114**	**0.898**	**4.935**	**0.206**	**0.861**	**0.949**	**0.976**
Garg et al. [16] L12 Aug 8× cap 50m	No	K	0.169	1.080	5.104	0.273	0.740	0.904	0.962
Ours cap 50m	No	K	0.140	0.976	4.471	0.232	0.818	0.931	0.969
Ours cap 50m	No	CS + K	0.117	0.762	3.972	0.206	0.860	0.948	0.976
Ours pp cap 50m	No	CS + K	0.112	0.680	3.810	0.198	0.866	0.953	**0.979**
Ours resnet pp cap 50m	No	CS + K	**0.108**	**0.657**	**3.729**	**0.194**	**0.873**	**0.954**	**0.979**
Our pp uncropped	No	CS + K	0.134	1.261	5.336	0.230	0.835	0.938	0.971
Ours resnet pp uncropped	No	CS + K	0.130	1.197	5.222	0.226	0.843	0.940	0.971

Lower is better

Higher is better

Table 2. Results on KITTI 2015 [17] using the split of Eigen et al. [10]. For training, K is the KITTI dataset [17] and CS is Cityscapes [8]. The predictions of Liu et al. [36]* are generated on a mix of the left and right images instead of just the left input images. For a fair comparison, we compute their results relative to the correct image. As in the provided source code, Eigen et al. [10]° results are computed relative to the velodyne instead of the camera. Garg et al. [16] results are taken directly from their paper. All results, except [10], use the crop from [16]. We also show our results with the same crop and maximum evaluation distance. The last two rows are computed on the uncropped ground truth.

论文标题	Unsupervised Monocular Depth Estimation with Left-Right Consistency
作者及单位	Clement Godard, Oisin Mac Aodha, Gabriel J. Brostow University College London
论文出处	2017-CVPR
创新点提炼	该论文提出的方法在深度估计的结果上有了显著改善，且其方法是在稠密深度图的估计上表现了很快的计算速度，在 gpu 估计一张 512 × 256 的图像约耗费 35 毫秒，主要贡献如下： 3. 提出了端到端的无监督单目深度估计网络，并提出了新的损失函数以迫使网络中的左右深度一致。 4. 对几个训练损失和图像形成模型进行评估，强调了本文方法的有效性。 除了展示具有挑战性的驾驶数据集的最新结果，本文还展示了提出的模型可以推广到三个不同的数据集，包括自己收集的一个新的室外城市数据集，并公开提供该数据集。
个人想法	本文主要基于立体视图进行无监督的单目深度估计，在当时取得了很好的估计结果，但对自己的研究方向感觉帮助不大（从输入数据和后续处理方法来看）。阅这篇论文主要想探究无监督单目深度估计与自监督单目深度估计的区别，但并未从文中发现有明显的界限。

论文方法及结论：

1. 论文提出的问题

我们工作的动机是推测当前的架构没有对输出值进行足够的全局分析。卷积层的一个缺点是，只有当张量在瓶颈处或瓶颈附近达到非常低的空间分辨率时，它们才处理全局信息。然而，我们认为，在高分辨率下进行全局处理要强大得多。

2. 解决的方法

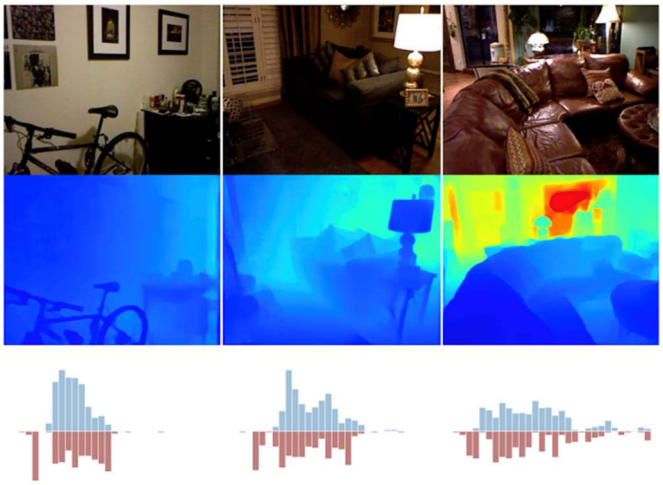


Figure 1: Illustration of AdaBins: **Top**: input RGB images. **Middle**: depth predicted by our model. **Bottom**: histogram of depth values of the ground truth (blue) and histogram of the predicted adaptive depth-bin-centers (red) with depth values increasing from left to right. Note that the predicted bin-centers are focused near smaller depth values for closeup images but are widely distributed for images with a wider range of depth values.

本文的总体想法是对传统的编码器-解码器架构的输出进行全局统计分析，并使用以最高分辨率工作的学习后处理构建模块来优化输出。对应不同 RGB 输入的深度分布变化很大。有些图像的大部分对象位于非常小的深度值范围内。例如，家具的特写将包含大多数靠近照相机的像素，而其他图像可能具有分布在更宽范围内的深度值，例如走廊，其中深度值的范围从小值到网络支持的最大深度。深度分布的这种变化使得以端到端的方式进行深度回归变得更加困难。最近的工作已经提出利用关于室内环境的假设，例如平面性约束来引导网络，这对于真实世界的环境可能成立，也可能不成立，尤其是对于室外场景。我们研究的方法不是强加这样的假设，而是网络学会自适应地聚焦于更可能出现在输入图像场景中的深度范围的区域。

近年来，编码器-解码器这种体系结构的使用在深度估计问题的监督和非监督设置中都显示出巨大的成功。这种方法通常使用一个或多个编码器-解码器网络作为其较大网络的子部分。在本文中，我们采用了下图的结构。

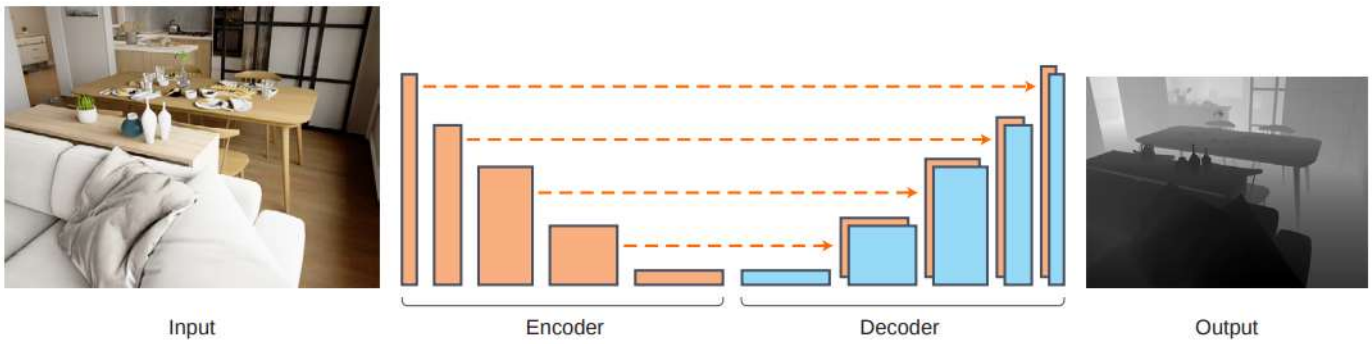


Figure 2. **Overview of our network architecture.** We employ a straightforward encoder-decoder architecture with skip connections. The encoder part is a pre-trained truncated DenseNet-169 [17] with no additional modifications. The decoder is composed of basic blocks of convolutional layers applied on the concatenation of the 2× bilinear upsampling of the previous block with the block in the encoder with the same spatial size after upsampling.

Transformer 网络作为一种可行的构建模块，在传统的自然语言处理任务和计算机视觉任务之外，正受到越来越多的关注。随着最近将 CNN 与 Transformer 相结合的趋势取得成功，我们提议利用 Transformer 作为 CNN 出的非本地处理的构建模块。

本文的思想可以看作是通过有序回归网络进行深度估计的推广。 Fu 等人观察到，如果深度回归任务被转换成分类任务，则可以实现性能改善。他们建议将深度范围划分为固定数量的预定宽度的箱（bin）。而本文首先建议根据输入场景的特征动态变化地计算自适应箱。第二，分类方法导致深度值的离散化，这导致具有明显尖锐深度不连续性的差的视觉质量。就标准评估指标而言，这可能仍会产生良好的结果，但它可能会对下游应用(如计算

摄影或3D重建)带来挑战。因此，我们建议将最终深度值预测为箱中心的线性组合。这使我们能够将分类的优势与深度图回归的优势结合起来。最后，与其他体系结构(如DAV)相比，我们以高分辨率全局计算信息，而不是以低分辨率主要在瓶颈部分计算信息。

2.1 AdaBins design

首先，为了说明我们将深度区间划分为箱的想法，我们希望将我们的最终解决方案与我们评估的其他三种可能的设计选择进行对比:

- 具有统一箱柜宽度的固定箱：深度间隔 D 被分成 N 个相同大小的箱。
- 具有 log 尺度箱宽度的固定箱：深度间隔 D 被分成 log 尺度的相等尺寸的箱。
- 可训练的箱宽度：箱宽度是自适应的，可以针对特定数据集进行学习。虽然箱宽度是通用的，但是所有图像最终共享深度间隔 d 的相同箱细分。
- AdaBins：自适应计算每个图像的 bin 宽度 b。

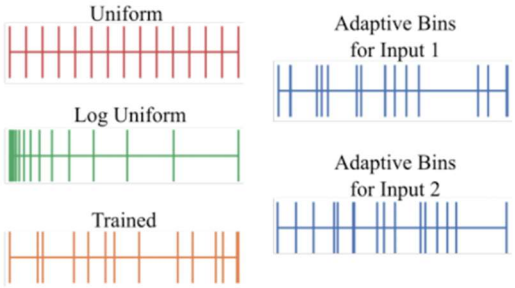


Figure 3: Choices for bin widths. Uniform and Log-uniform bins are pre-determined. 'Trained bins' vary from one dataset to another. Adaptive bins vary for each input image.

第二，将深度间隔 D 离散化为箱并将每个像素分配给一个箱会导致深度离散化伪影。因此，我们将最终深度预测为箱中心的线性组合，使模型能够估计平滑变化的深度值。

第三，几个先前的体系结构提出使用注意力来执行全局处理，以在体系结构中的编码器块之后处理信息，当前最先进的深度估计使用这种策略。这种体系结构由三个按顺序排列的块组成:编码器、注意力，然后是解码器。我们最初遵循这种方法，但注意到在空间分辨率更高的张量上使用注意力可以获得更好的结果。因此，我们提出了一个架构，也有这三个块，但排序如下:编码器，解码器，注意力。

第四，我们希望建立在最简单的架构上，以隔离我们新提出的 Adabin 概念的影响。因此，我们构建了一个现代的编码器-解码器，使用 EfficientNet B5 作为编码器的主干。

2.2 Architecture description

本文的架构由两个主要组件组成:1)建立在预先训练的 EfficientNet B5 编码器和标准特征上采样解码器上的编码器-解码器块；2)我们提出的称为 AdaBins 的自适应 bin 宽度估计器块。解码器的输出为张量 $x_d \in R^{h\times w\times C_d}$,我们将其称之为 解码特征（decoded features），并作为第二部分 AdaBins 模块的输入，而 AdaBins 模块的输出为大小为 $h \times w \times 1$ 的张量。由于 GPU 和内存的限制，$h = \frac{H}{2}, w = \frac{W}{2}$，最后的深度图通过简单的双边上采样来计算得到与输入图像相同大小。

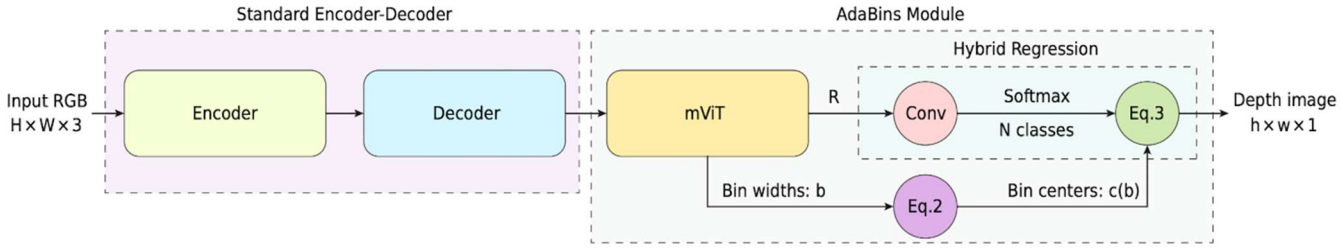


Figure 2: Overview of our proposed network architecture. Our architecture consists of two major components: an encoder-decoder block and our proposed adaptive bin-width estimator block called AdaBins. The input to our network is an RGB image of spatial dimensions H and W, and the output is a single channel $h \times w$ depth image (e.g., half the spatial resolution).

AdaBins 模块中的第一个块叫做 mini-ViT，这是最近提出的一种使用 Transformer 进行图像识别的技术的简化版本，mini-ViT 有两个输出:1)bin 宽度的向量 b，它定义了如何为输入图像划分深度间隔 D，以及 2)大小为 h × w × C 的 范围-注意力-地图 R，它包含用于像素级深度计算的有用信息。

对于给定的图像，估计深度范围 D 内更可能出现的子间隔需要同时结合局部结构信息和全局分布信息。我们建议使用全局关注来计算每个输入图像的 bin 宽度向量 b。就内存和计算复杂性而言，全局关注都是昂贵的，尤其是在较高的分辨率下。然而，最近 Transformer 的快速发展提供了一些有效的替代方案。我们在设计带

Transformer 的 AdaBins 模块时，从视觉 Transformer ViT 中获得了灵感。因为我们的数据集更小，使用了一个更小版本的 Transformer，在下面的描述中，我们称这个转换器为 mini-ViT 或 mViT。

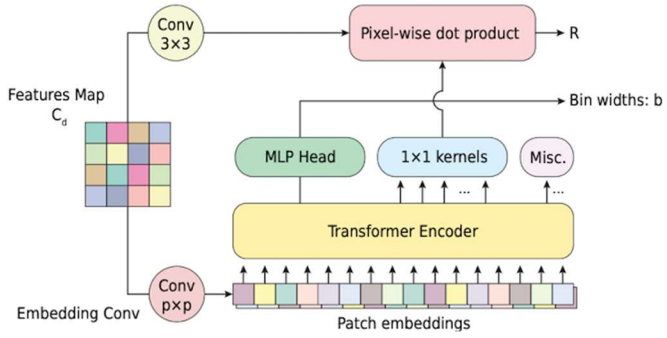


Figure 4: An overview of the mini-ViT block. The input to the block is a multi-channel feature map of the input image. The block includes a Transformer encoder that is applied on patch embeddings of the input for the purpose of learning to estimate bin widths b and a set of convolutional kernels needed to compute our Range-Attention-Maps R.

Patch size (p)	E	Layers	num heads	C	MLP Size	Params
16	128	4	4	128	1024	5.8 M

Table 1: Mini-ViT architecture details.

Bin-widths. mViT 的输入是解码特征 $x_d \in R^{h\times w\times C_d}$ 的张量。然而，Transformer 采用一系列固定大小的向量作为输入。我们首先将解码后的特征通过一个称为嵌入 Conv 的卷积块，其核大小为 p × p，步长为 p，输出通道数为 E。因此，该卷积的结果是大小为 h/p × w/p × E 的张量(假设 h 和 w 都可以被 p 整除)。结果被重新整形为空间平坦张量 $x_p \in R^{S\times E}, S=\frac{hw}{p^2}$ 作为 Transformer 的有效序列长度。我们称这个序列的 E 维向量为 patch 嵌入。

按照通常的做法，我们将学习到的位置编码添加到 patch 嵌入中，然后将它们馈送给 Transformer 并输出一系列输出嵌入 $x_0 \in R^{S\times E}$。我们在第一个输出嵌入上使用 MLP 头。MLP 头使用 ReLU 激活并输出一个 N 维向量 b_0。最后，我们对向量 b_0 进行归一化，使其总和为 1，从而获得 bin 宽向量 $b_i=\frac{b_i^{'}+\epsilon}{\sum_{j=1}^{N}\left(b_j^{'}+\epsilon\right)}, \epsilon=10^{-3}$。归一化引入了 bin 宽度之间的竞争，并在概念上迫使网络通过预测感兴趣区域的较小 bin 宽度来关注 D 内的子间隔。

Range attention maps. 解码特征表示高分辨率和局部像素级信息，而 Transformer 输出嵌入有效地包含更多的全局信息。来自 Transformer 的输出嵌入 2 到 C + 1 被用作一组 1 × 1 卷积核，并与解码特征(在 3 × 3 卷积层之后)卷积，以获得范围-关注图。这相当于计算被视为“key”的像素特征和被视为“query”的 Transformer 输出嵌入之间的点积关注权重。这种使用输出嵌入作为卷积核的简单设计使得网络将自适应全局 Transformer 集成到解码特征的局部信息中。R 和 b 一起用来获得最终的深度图。

距离-注意力图 R 通过 1 × 1 卷积层获得 N 个信道，随后是 Softmax。我们解读 N Softmax 分数 $p_k, k=1,\cdots,N$，在每个像素处作为在 N 个深度 bin 中心上的概率 $c(b):=\{c(b_1),c(b_2),\cdots,c(b_N)\}$ 根据 bin 宽向量 b 计算，

$$c(b_i)=d_{min}+(d_{max}-d_{min})\left(b_i/2+\sum_{j=1}^{i-1}b_j\right)$$

最后，在每个像素处，最终深度值由该像素处的 Softmax 分数和深度 bin 中心 c(b) 的线性组合计算得出

$$\tilde{d}=\sum_{k=1}^{N}c(b_k)p_k$$

3. 实验结果和结论

Method	$\delta_1\uparrow$	$\delta_2\uparrow$	$\delta_3\uparrow$	REL $\downarrow$	RMS $\downarrow$	$log_{10}\downarrow$
Eigen *et al.* [8]	0.769	0.950	0.988	0.158	0.641	–
Laina *et al.* [25]	0.811	0.953	0.988	0.127	0.573	0.055
Hao *et al.* [16]	0.841	0.966	0.991	0.127	0.555	0.053
Lee *et al.* [27]	0.837	0.971	0.994	0.131	0.538	–
Fu *et al.* [11]	0.828	0.965	0.992	0.115	0.509	0.051
SharpNet [34]	0.836	0.966	0.993	0.139	0.502	0.047
Hu *et al.* [19]	0.866	0.975	0.993	0.115	0.530	0.050
Chen *et al.* [4]	0.878	0.977	0.994	0.111	0.514	0.048
Yin *et al.* [47]	0.875	0.976	0.994	0.108	0.416	0.048
BTS [26]	0.885	0.978	0.994	0.110	0.392	0.047
DAV [22]	0.882	0.980	0.996	0.108	0.412	–
AdaBins (Ours)	**0.903**	**0.984**	**0.997**	**0.103**	**0.364**	**0.044**

Table 2: Comparison of performances on the NYU-Depth-v2 dataset. The reported numbers are from the corresponding original papers. Best results are in bold, second best are underlined.

Method	$\delta_1\uparrow$	$\delta_2\uparrow$	$\delta_3\uparrow$	REL $\downarrow$	Sq Rel $\downarrow$	RMS $\downarrow$	RMS log $\downarrow$
Saxena *et al.* [36]	0.601	0.820	0.926	0.280	3.012	8.734	0.361
Eigen *et al.* [8]	0.702	0.898	0.967	0.203	1.548	6.307	0.282
Liu *et al.* [29]	0.680	0.898	0.967	0.201	1.584	6.471	0.273
Godard *et al.* [15]	0.861	0.949	0.976	0.114	0.898	4.935	0.206
Kuznietsov *et al.* [24]	0.862	0.960	0.986	0.113	0.741	4.621	0.189
Gan *et al.* [12]	0.890	0.964	0.985	0.098	0.666	3.933	0.173
Fu *et al.* [11]	0.932	0.984	0.994	0.072	0.307	2.727	0.120
Yin *et al.* [47]	0.938	0.990	0.998	0.072	–	3.258	0.117
BTS[26]	0.956	0.993	0.998	0.059	0.245	2.756	0.096
AdaBins (Ours)	**0.964**	**0.995**	**0.999**	**0.058**	**0.190**	**2.360**	**0.088**

Table 3: Comparison of performances on the KITTI dataset. We compare our network against the state-of-the-art on this dataset. The reported numbers are from the corresponding original papers. Measurements are made for the depth range from $0m$ to $80m$. Best results are in bold, second best are underlined.