

# Monocular Depth Estimation with Augmented Ordinal Depth Relationships

Yuanzhouhan Cao, Tianqi Zhao, Ke Xian, Chunhua Shen, Zhiguo Cao

在预训练好的resNet上训练分类网络，用立体视觉生成相对深度，对相对深度进行分类

**Abstract**—Most existing algorithms for depth estimation from single monocular images need large quantities of metric ground-truth depths for supervised learning. We show that relative depth can be an informative cue for metric depth estimation and can be easily obtained from vast stereo videos. Acquiring metric depths from stereo videos is sometimes impracticable due to the absence of camera parameters. In this paper, we propose to improve the performance of metric depth estimation with relative depths collected from stereo movie videos using existing stereo matching algorithm. We introduce a new “Relative Depth in Stereo” (RDIS) dataset densely labelled with relative depths. We first pretrain a ResNet model on our RDIS dataset. Then we finetune the model on RGB-D datasets with metric ground-truth depths. During our finetuning, we formulate depth estimation as a classification task. This re-formulation scheme enables us to obtain the confidence of a depth prediction in the form of probability distribution. With this confidence, we propose an information gain loss to make use of the predictions that are close to ground-truth. We evaluate our approach on both indoor and outdoor benchmark RGB-D datasets and achieve state-of-the-art performance.

**Index Terms**—Depth estimation, RGB-D dataset, ordinal relationship, deep network

## I. INTRODUCTION

Predicting accurate depths from single monocular images is a fundamental task in computer vision and has been an active research topic for decades. Typical methods formulate depth estimation as a supervised learning task [1], [2], [3]. As a result, large amounts of metric ground-truth depths are needed. However, the acquisition of metric ground-truth depths requires depth sensors, and the collected RGB-D training data is limited in the size as well as the diversity of scenes due to the limitation of depth sensors. For example, the popular Microsoft Kinect can not obtain the depths of far objects in outdoor scenes.

In order to overcome the problem of limited metric ground-truth depths, some recent works manage to predict depths from stereo videos [4], [5], [6] without the supervision of ground-truth depths. Specifically, the model is trained by computing the disparity maps and minimizing an image reconstruction loss between training stereo pairs. The performance is not satisfactory due to the absence of ground-truth depths during training. However, the training stereo videos are easier to

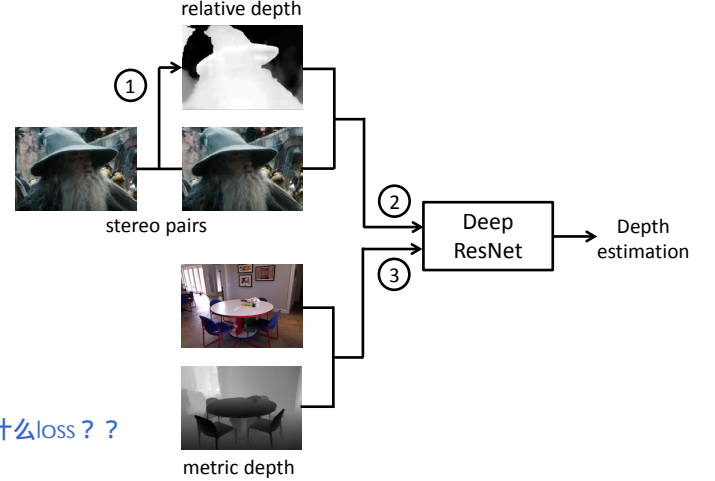


Fig. 1: Overview of our proposed depth estimation method. We first generate relative depths from stereo pairs, then pretrain a deep residual network with the relative depths. Finally, we finetune the network with metric depths for monocular depth estimation.

obtain than metric ground-truth depths and are plenty in terms of amount as well as scene diversity.

Driven by the aforementioned characteristics of recent depth estimation methods, a question arises: Is it possible to acquire large quantities of training data from stereo videos to improve the performance of monocular depth estimation?

Compared to metric depths, relative depths can be easily obtained from stereo videos using existing stereo matching algorithms [7], [8], [9], [10]. The recent works by Zoran et al. [11] and Chen et al. [12] have revealed that it is possible to predict satisfactory metric depths with only relative ground-truth depths. In this paper, we propose to improve the performance of metric depth estimation with relative depths generated from stereo movie videos. An overview of our approach is illustrated in Fig. 1. Our approach can be broadly divided into 3 steps: We first obtain ground-truth relative depths from stereo movie videos, then we pretrain a deep residual network with our relative ground-truth depths. Finally, we finetune our network on benchmark RGB-D datasets with metric ground-truth depths. Note that, as we exploit 3D movie stereo videos, which do not have the camera parameters and typically are re-calibrated for display, it is impossible to compute the metric depth.

Most existing methods formulate depth estimation as a regression problem due to the continuous property of depths [2], [3], [13]. For human beings, we may find it difficult to tell the

Y. Cao is with The Idiap Research Institute, Switzerland; T. Zhao is with Tsinghua University, China; K. Xian and Z. Cao are with Huazhong University of Science and Technology, China. C. Shen is with The University of Adelaide, Australia.

Corresponding author: C. Shen (e-mail: chunhua.shen@adelaide.edu.au). This work was done when Y. Cao and K. Xian were with The University of Adelaide.

exact distance of a specific point in a natural scene, but we can easily give a rough distance range of that point. Motivated by this, we formulate depth estimation as a pixel-wise classification task by discretizing the continuous depth values into several discrete bins and show that this simple re-formulation scheme performs surprisingly well. More importantly, we can easily obtain the confidence of a depth prediction in the form of probability distribution. With this confidence, we can apply an information gain loss to make use of the predictions that are close to ground-truth during training.

To summarize, we highlight the contributions of our work as follows:

- 1) We formulate depth estimation as a **classification task and propose an information gain loss.**
- 2) We propose a new dataset Relative Depth in Stereo (RDIS) containing images labelled with dense relative depths. The relative depths are generated with very low cost.
- 3) We show that our proposed RDIS dataset can improve the performance of metric depth estimation significantly and we outperform state-of-the-art depth estimation methods on both indoor and outdoor benchmark RGB-D datasets.

## II. RELATED WORK

Traditional depth estimation methods are mainly based on geometric models. For example, the works of [14], [15], [16] rely on box-shaped models and try to fit the box edges to those observed in the image. These methods are limited to only model particular scene structures and therefore are not applicable for general-scene depth estimations. More recently, non-parametric methods [17] are explored. These methods consist of candidate images retrieval, scene alignment and then depth inference using optimizations with smoothness constraints. These methods are based on the assumption that scenes with semantically similar appearances should have similar depth distributions when densely aligned.

Most depth estimation algorithms in recent years achieve outstanding performance by training deep convolutional neural networks (CNN) [18], [19], [20] with fully annotated RGB-D datasets [21], [22], [23]. Liu et al. [24] presented a deep convolutional neural field which jointly learns the unary and pairwise potentials of continuous conditional random fields (CRF) in a unified deep network. Eigen et al. [25] proposed a multi-scale network architecture to predict depths as well as surface normals and semantic labels. Li et al. [26] and Wang et al. [27] formulated depth estimation in a two-layer hierarchical CRF to enforce synergy between global and local predictions. Laina et al. [13] applied the latest deep residual network [28] as well as an up-sampling scheme for depth estimation.

Other recent works managed to train deep CNNs for depth estimation in an unsupervised manner. To name a few, Garg et al. [4] and Clément et al. [5] treated depth estimation as an image reconstruction [29] problem during training and output disparity maps during prediction. In order to construct a fully differentiable training loss, Taylor approximation and bilinear interpolation are applied in [4] and [5] respectively. Since the

network outputs of [4] and [5] are disparity maps, camera parameters are needed to recover the metric depths. Similarly, the Deep3D model [6] also applied an image reconstruction loss during training, where their goal is to predict the right view from the left view of a stereo pair, and the disparity map is generated internally.

Ordinal relationships and rankings have also been exploited for mid-level vision tasks including depth estimation in recent years. Zoran et al. [11] learned the ordinal relationships between pairs of points using a classification loss, then they solved a constrained quadratic optimization problem to map the ordinal estimates to metric values. Chen et al. [12] proposed to learn ordinal relationships through a ranking loss [30] and retrieve the metric depth values by simple normalization. Notably, [12] also proposed a new dataset Depth in the Wild (DIW) consisting of images in the wild labelled with relative depths.

Our work is mainly inspired by Chen et al.'s single-image depth perception in the wild [12]. However, our approach is different in three distinct aspects. First, instead of manually labelling pixels with relative relationships, we acquire relative depths using existing stereo matching algorithm from stereo movie videos and thus can obtain large amount of training data with low cost. Second, instead of labelling only one pair of points per image with relative relationships, we generate dense relative depth maps. Finally, in order to retrieve metric depth predictions, they arbitrarily normalize the predicted relative depth maps such that the mean and standard deviation are the same with the metric ground-truth depths of training set, while we finetune our pretrained network with metric ground-truth depths for better performance.

## III. PROPOSED METHOD

In this section, we elaborate our proposed method for monocular depth estimation. We first present the stereo matching algorithm that we used to generate relative ground-truth depth. Then we introduce our network architecture, followed by the introduction of our loss functions.

### A. Relative depth generation

The first step of our approach is to generate relative ground-truth depth from stereo videos using **existing stereo matching algorithm**. Stereo matching algorithms rely on computing matching costs to measure the similarities of stereo pairs. In this paper, we choose the commonly-used absolute difference (AD) matching cost combined with a background subtraction by bilateral filtering (BilSub) which has been proven to perform well by Hirschmuller et al. [31]. **For a pixel  $p$  in the left image, its corresponding pixel in the right image is represented as  $p - d$ , where  $d$  is the disparity.** The absolute difference is represented as:

$$C_{ad}(p, d) = |I_L(p) - I_R(p - d)|, \quad (1)$$

where  $I_L$  and  $I_R$  are left and right images respectively. We sum the costs of all three channels of color images. The bilateral filtering effectively removes a local offset without

用的是立体视觉匹配

选左图和右图中对应点RGB值的绝对距离，求和，用于生成相对距离



Fig. 2: Some examples of our Relative Depth in Stereo (RDIS) dataset. The first row are RGB images, the second row are disparity maps directly generated by stereo algorithm. The last row are post-processed disparity maps, which are used as ground-truths.

blurring high contrast texture differences that may correspond to depth discontinuities.

As for the stereo algorithm, we use the semi-global matching (SGM) method [32]. It aims to minimize a global 2D energy function by solving a large number of 1D minimization problems. The energy function is: 怎么找左右图, 优化绝对距离

$$E(D) = \sum_p (C(p, D_p) + \sum_{q \in N_p} P_1 T[|D_p - D_q| = 1]) + \sum_{q \in N_p} P_2 T[|D_p - D_q| > 1]), \quad (2)$$

后边两项允许有一定的差别

where the first term calculates the sum of a pixel-wise matching cost for all pixels at their disparities  $D_p$ . The second term adds a constant penalty  $P_1$  for all pixels  $q$  in the neighborhood  $N_p$  of  $p$ , for which the disparity changes a little bit (i.e., 1 pixel). Similarly, the third term adds a larger constant penalty  $P_2$ , for all larger disparity changes. The SGM calculates  $E(D)$  along 1D paths from 8 directions towards each pixel of interest using dynamic programming. The costs of all paths are summed for each pixel and disparity. The disparity is then determined by winner-takes-all. During training, We label the pairs of points with ordinal relationships (farther, closer, equal) according to their disparities. Since the disparity values of two points can not be exactly the same, we apply a relaxed definition of equality. The ordinal relationship of a pair of points is equal if the disparity difference is smaller than a fixed threshold.

The direct output of stereo matching algorithm is a dense disparity map with the left image treated as the reference image. This disparity map can not be directly used for training due to the defects such as noise, discontinuities or incorrect values. Some examples are shown in Fig. 2. As a result, we need to post-process the disparity maps generated by stereo algorithm. The post-processing is done by experienced workers from movie production company using professional movie production software. Specifically, we first correct the vague or missing boundaries of objects using B-splines, then we smooth the disparity values within objects and background. It takes a median of 90 seconds to post-process an image of our dataset. Although the labelling of our dataset takes longer time than the DIW dataset, our dataset is densely labelled

and contains more ordinal relationships than the DIW dataset. In terms of single ordinal relationship labelling our method is much more efficient. After post-processing, each disparity map is visually checked by two workers according to the intensities. The workers are required to assign “overall correct”, “contain mislabelled parts” or “not sure” to each disparity map. We only keep the disparity maps which both workers assigned as “overall correct”.

We also test several other stereo matching algorithms including the deep learning based. Although the qualities of these direct output disparities are different, they are all very coarse, furthermore the difference becomes negligible after human post-processing. So we pick the simplest stereo matching method.

We collect 70 3D movies produced in recent years. Since the stereo matching algorithm requires the stereo videos to be rectified, we only use 3D movies created by post-production instead of movies taken with stereo cameras. In order to avoid similar frames, we only select roughly 1500 frames in each movie. With the selected frames, we generate a new dataset Relative Depth in Stereo (RDIS) containing 97652 training images labelled with dense relative ground-truth depths. Notably, we can not obtain the metric depths from the relative depths because we do not have the camera parameters of these 3D movies. Our dataset has no test images because: 1) The goal of our dataset is to improve the performance of metric depth estimation. 2) Our ground-truth disparities are obtained through stereo matching algorithm and inevitably contain noisy points.

### B. Network architecture

Recently, a deep residual learning framework has been introduced by He et al. [28], [33] and showing compelling accuracy and nice convergence behaviours. In our work, we follow the deep residual network architecture proposed by Wu et al. [34] which contains fewer layers but outperforms the deep residual network with 152 layers in [28].

Instead of directly learning the underlying mapping of a few stacked layer, the deep residual network learns the residual mapping through building blocks. We consider two types of

building blocks in our network architecture. The first is defined as:

$$\mathbf{y} = F(\mathbf{x}, \{W_i\}) + \mathbf{x}, \quad (3)$$

where  $\mathbf{x}$  and  $\mathbf{y}$  are the input and output matrices of stacked layers respectively. The function  $F(\mathbf{x}, \{W_i\})$  is the residual mapping that need to be learned. The dimensions of  $\mathbf{x}$  and  $F$  need to be equal since the addition is element-wise. If this is not the case, we apply another building block defined as:

$$\mathbf{y} = F(\mathbf{x}, \{W_i\}) + W_s \mathbf{x}. \quad (4)$$

Comparing to the shortcut connection in Eq. (3), a linear projection  $W_s$  is applied to match the dimensions of  $\mathbf{x}$  and  $F$ .

We illustrate our detailed network structure in Fig. 3. Generally, it is composed of 6 convolution blocks. Each convolution block starts with a building block with linear projection followed by different numbers of building blocks with identity mapping. Two max pooling layers with stride of 2 are applied before the first and the second convolution blocks. The first convolutional layers of block 3, 4 and 5 have a stride of 2. The dilations of the first convolutional layers of block 4 and 5 are 2 and 4 respectively. As a result, our network takes as inputs of arbitrarily sized images and downsamples by a factor of 8. We apply the bilinear interpolation to upsample the network output. Batch normalizations (BNs) [35] and ReLUs are applied before weight layers. We initialize the layers up to block 6 with our model pretrained on the ImageNet [36] and Places365 [37] datasets. After block 6, we add 3 convolutional layers with randomly initialized weights. The channels of the first and second added convolutional layers are 1024 and 512 respectively. The channel number of last convolutional layer is determined by the loss function. The channel number is 1 for the pretraining using ranking loss. As for the finetuning, we discretize the continuous metric depths into several bins and formulate depth estimation as a classification task, the channel number is equal to the bin number. We give more details about the loss functions below.

### C. Loss function

Our proposed approach for depth estimation contains two training stages: pretraining with relative depths and finetuning with metric depths. For the pretraining, we employ the ranking loss which encourages a small difference between depths if the ground-truth ordinal relation is equality and encourages a large difference otherwise. Specifically, consider a training image  $I$  with  $K$  pairs of points with ground-truth ordinal relations  $R = \{(i_k, j_k, r_k)\}, k \in [1, \dots, K]$ , where  $i_k$  and  $j_k$  are the two points of  $k$ -th pair, and  $r_k$  is the ground-truth depth relation between  $i_k$  and  $j_k$ : closer (+1), farther (-1) and equal (0). Let  $z$  be the output depth map of our deep residual network and  $z_{i_k}, z_{j_k}$  be the predicted depth values of  $i_k$  and  $j_k$ . The ranking loss is defined as:

$$L_{rank}(I, R, z) = \sum_{k=1}^K E(I, i_k, j_k, r, z), \quad (5)$$

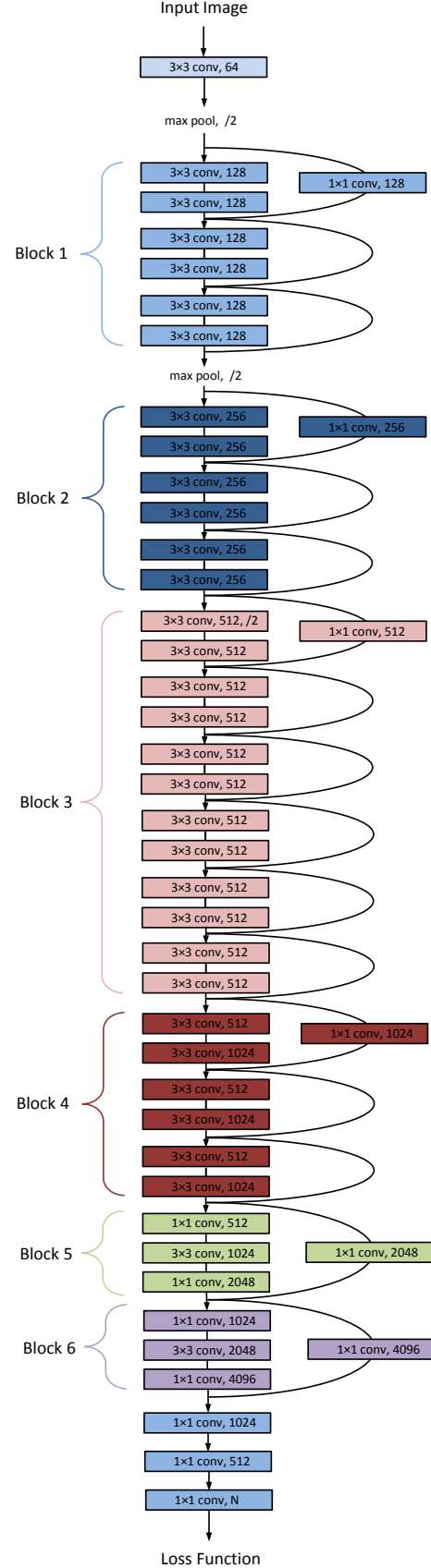


Fig. 3: Detailed structure of our deep residual network. It has 6 convolution blocks, each with different numbers of residual units.



### relative loss

where  $E(I, i_k, j_k, r, z)$  is the loss of the  $k$ -th pair:

$$E = \begin{cases} \log(1 + \exp(-z_{i_k} + z_{j_k})), & r_k = +1; \\ \log(1 + \exp(z_{i_k} - z_{j_k})), & r_k = -1; \\ (z_{j_k} - z_{i_k})^2, & r_k = 0. \end{cases} \quad (6)$$

After pretraining, we finetune our network with discretized metric depths. We use the pixel-wise multinomial logistic loss defined as:

$$L_{\log} = -\frac{1}{N} \sum_{i=1}^N \sum_{D=1}^B H(D_i^*, D) \log(P(D|z_i)), \quad (7)$$

where  $D_i^* \in [1, \dots, B]$  is the ground-truth depth label of pixel  $i$  and  $B$  is the total number of discretization bins.  $N$  is the number of pixels.  $P(D|z_i) = e^{z_{i,D}} / \sum_{d=1}^B e^{z_{i,d}}$  is the probability of pixel  $i$  labelled with  $D$ .  $z_{i,d}$  is the output of last convolutional layer in the network.

Although we formulate depth estimation as a classification task by discretizing continuous depth values into several bins, the depth labels are different with the labels of other classification tasks (e.g., semantic segmentation). Predicted depth labels that are closer to ground-truth should have more contribution in updating network weights. This is achieved through the information gain matrix  $H$  in Eq. (7). It is a  $B \times B$  symmetric matrix with elements  $H(p, q) = \exp[-\alpha(p - q)^2]$  and  $\alpha$  is a constant. During training, we equally discretize the continuous depths in the log space into several bins and during prediction, we set the depth value of each pixel to be the center of its corresponding bin.

## IV. EXPERIMENTS

We organize our experiments into the following 3 parts: 1) We demonstrate the benefit of the pretraining on our proposed Relative Depth in Stereo (RDIS) dataset by comparing with other pretraining schemes; 2) We evaluate the metric depth estimation on indoor and outdoor benchmark RGB-D datasets and analyze the contributions of some key components in our approach; 3) We evaluate both metric and relative depth estimation and compare with state-of-the-art results. During pretraining and finetuning, we apply online data augmentation including random scaling and flipping. We apply the following measures for metric depth evaluation:

- root mean squared error (rms):  $\sqrt{\frac{1}{T} \sum_p (d_{gt} - d_p)^2}$
- average relative error (rel):  $\frac{1}{T} \sum_p \frac{|d_{gt} - d_p|}{d_{gt}}$
- average  $\log_{10}$  error (log10):  $\frac{1}{T} \sum_p |\log_{10} d_{gt} - \log_{10} d_p|$
- root mean squared log error (rmslog):  $\sqrt{\frac{1}{T} \sum_p (\log d_{gt} - \log d_p)^2}$
- accuracy with threshold  $thr$ :  
percentage (%) of  $d_p$  s.t.  $\max(\frac{d_{gt}}{d_p}, \frac{d_p}{d_{gt}}) = \delta < thr$

where  $d_{gt}$  and  $d_p$  are the ground-truth and predicted depths respectively of pixels, and  $T$  is the total number of pixels in all the evaluated images. As for the relative depth evaluation, we report the Weighted Human Disagreement Rate (WHDR) [11], the average disagreement rate with human annotators, weighted by their confidence (here set to 1). We implement our network training based on the MXNet [38].

TABLE I: Comparison between different numbers of pairs during pretraining. The model is pretrained on our RDIS dataset and finetuned on NYUD2 and KITTI datasets. For each dataset, each row represents different numbers of ground-truth pairs in each input image during pretraining.

	Accuracy			Error		
	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$	rel	log10	rms
NYUD2						
100 pairs	63.8%	90.2%	97.5%	0.202	0.090	0.816
500 pairs	68.2%	92.5%	98.6%	0.178	0.081	0.750
1K pairs	<b>71.1%</b>	<b>93.3%</b>	<b>98.6%</b>	<b>0.173</b>	<b>0.077</b>	<b>0.721</b>
5K pairs	63.0%	89.1%	97.1%	0.208	0.092	0.828
KITTI						
100 pairs	70.6%	88.3%	94.6%	0.230	0.088	6.357
500 pairs	74.1%	90.0%	95.3%	0.205	0.079	5.900
1K pairs	<b>74.2%</b>	<b>90.0%</b>	<b>95.5%</b>	<b>0.205</b>	<b>0.079</b>	<b>5.828</b>
5K pairs	70.2%	87.3%	94.1%	0.223	0.088	6.629

### A. Benefit of pretraining

In this section, we show the benefit of the pretraining with our proposed RDIS dataset. Since our proposed RDIS dataset is densely labelled with relative depths, we need first to determine the number of ground-truth pairs in each image during pretraining. We randomly sample 100, 500, 1K and 5K ground-truth pairs in each input image during pretraining and finetune on both the NYUD2 [21] and KITTI [23] datasets.

The standard NYUD2 training set contains 795 images. We split the 795 images into a training set with 400 images and a validation set with 395 images. We discretize the continuous metric depth values into 100 bins in the log space. As for the KITTI dataset, we apply the same split in [3] which contains 700 training images and 697 test images. We further evenly split the 700 training images into a training set and a validation set. We only use left images and discretize the continuous metric depth values into 50 bins in the log space. We cap the maximum depth to be 80 meters. For both the NYUD2 and KITTI datasets, we finetune on our split training sets and evaluate on our validation sets. During finetuning, we ignore the missing values in ground-truth depth maps and only evaluate on valid points. We do not apply the information gain matrix in this experiment. The results are illustrated in Table I. As we can see from the table that for both indoor and outdoor datasets, the performance increases with the number of pairs and achieves the best when using 1K pairs per input image. Further increasing the number of pairs does not improve the performance. For pretraining with 5K pairs of points, we further add dropouts and evaluate the accuracy with  $\delta < 1.25$  on the NYUD2 dataset. The accuracies are 63.7%, 63.1%, 62.6% and 62.3% with 32K, 35K, 40K and 43K iterations. It demonstrates that the performance decrease is caused by overfitting. In the following experiments, we all sample 1K ground-truth pairs in each input image during pretraining.

In order to demonstrate the quality of our proposed RDIS dataset, we conduct experimental comparisons against several pretraining schemes: 1) Directly finetune our ResNet model on RGB-D datasets without pretraining (Direct); 2) Pretrain our ResNet model on the DIW [12] dataset and finetune on RGB-D datasets (DIW); 3) Pretrain our ResNet model using our

TABLE II: Test results on the NYUD2 and KITTI datasets with different pretraining. For each dataset, the first row is the result without pretraining; the second row is the result with pretraining on the DIW dataset; the third row is the result with pretraining using our RDIS images but the ground-truth relative depths are generated by the Deep3D [6] model; the last row is the result with pretraining on our RDIS dataset.

	Accuracy			Error		
	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$	rel	log10	rms
NYUD2						
Direct	73.3%	93.5%	98.1%	0.186	0.075	0.666
DIW	77.3%	95.4%	98.9%	0.160	0.066	<b>0.600</b>
Deep3D	72.5%	92.8%	97.8%	0.191	0.077	0.683
Ours	<b>78.1%</b>	<b>95.4%</b>	<b>98.9%</b>	<b>0.157</b>	<b>0.065</b>	0.604
KITTI						
Direct	77.3%	92.1%	96.9%	0.173	0.070	5.890
DIW	79.7%	93.7%	97.8%	0.154	0.064	5.251
Deep3D	76.1%	91.9%	97.1%	0.178	0.072	5.765
Ours	<b>82.9%</b>	<b>94.3%</b>	<b>98.2%</b>	<b>0.142</b>	<b>0.058</b>	<b>5.066</b>

RDIS images and finetune on RGB-D datasets, the ground-truth relative depths for pretraining are generated using the Deep3D model [6] (Deep3D).

We finetune on the standard training set of the NYUD2 which contains 795 images and evaluate on the standard test set which contains 654 images. The continuous metric depth values are discretized into 100 bins in the log space. The parameter  $\alpha$  of the information gain matrix defined in Eq. (7) is set to 2.0. As for the KITTI dataset, we finetune on the same 700 training images and evaluate on the same 697 test images as in [3]. The continuous metric depth values are discretized into 50 bins in the log space and the maximum depth value is capped to be 80 meters. The parameter  $\alpha$  is set to 0.2. We ignore the missing ground-truth values during both finetuning and evaluation.

We show the results in Table II. We can see from the table that the pretraining on our proposed RDIS dataset improves the depth estimation of both indoor and outdoor datasets significantly, and even outperforms the pretraining on the DIW dataset. Notably, compared to the DIW dataset which contains 421K training images with manually labelled relative depths, our RDIS dataset contains only 97652 images, and the relative ground-truth depths are generated by existing stereo matching algorithm.

### B. Component analysis

In this section, we evaluate metric depth estimation on the indoor NYUD2 and outdoor KITTI datasets and analyze the contributions of some key components of our approach. We use the same dataset settings with the second experiment in Sec. IV-A.

1) *Network comparisons*: In this part, we compare our deep residual network architecture against two baseline networks: deep residual network with 101 and 152 layers in [28]. We pretrain the 3 models on our RDIS dataset and finetune on the NYUD2 dataset. Similar to our network architecture, we replace the last 1000-way classification layers of ResNet101 and ResNet152 with one channel convolutional layers during

pretraining and 100-way classification layers during finetuning. We also add two convolutional layers with 1024 and 512 channels respectively before the last layer. We do not apply the information gain matrix in this experiment. The results are illustrated in Table III. From the table we can see that our network architecture outperforms the deeper ResNet101 and ResNet152.

TABLE III: Test results on the NYUD2 dataset with different network architectures. The first row is the result of the ResNet101, the second row is the result of the ResNet152, the last row is the result of our network.

	Accuracy			Error		
	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$	rel	log10	rms
Res101	76.1%	94.7%	98.5%	0.170	0.071	0.632
Res152	76.2%	94.9%	98.5%	0.169	0.070	0.626
Ours	<b>77.8%</b>	<b>95.3%</b>	<b>98.8%</b>	<b>0.159</b>	<b>0.066</b>	<b>0.606</b>

2) *Benefit of information gain matrix*: In this part, we evaluate the contribution of the information gain matrix during finetuning. We pretrain the network on our RDIS dataset and finetune on both the NYUD2 and KITTI datasets with and without the information gain matrix. The parameter  $\alpha$  defined in Eq. (7) is set to 2.0 and 0.2 respectively for the NYUD2 and KITTI datasets. The results are illustrated in Table IV. As we can see from the table that the information gain matrix improves the performance of both indoor and outdoor depth estimation.

3) *Depth classification vs. depth regression*: In this part, we compare our depth estimation by classification with the conventional regression. We directly train the ResNet101 model without pretraining on our RDIS dataset. For depth regression, we use the  $L2$  loss. For our depth estimation as classification, we discretize the continuous depth values into 100 and 50 bins in the log space respectively for the NYUD2 and KITTI datasets. And we set the parameter  $\alpha$  to 2.0 and 0.2 respectively. We show the results in Table VII, from which we can see that our depth estimation by classification outperforms the conventional regression. This is because the regression tends to converge to the mean depth values, which may cause larger errors in areas that are either very far from or very close to the camera. The classification naturally produces the confidence of a depth estimation in the form of probability distribution. Based on the probability distribution, we can apply the information gain loss to alleviate the problem.

TABLE IV: Test results on the NYUD2 and KITTI datasets with and without information gain matrix. For each dataset, the first row is the result without information gain matrix, the following row is the result with information gain matrix.

	Accuracy			Error		
	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$	rel	log10	rms
NYUD2						
Plain	77.8%	95.3%	98.8%	0.159	0.066	0.606
Infogain	<b>78.1%</b>	<b>95.4%</b>	<b>98.9%</b>	<b>0.157</b>	<b>0.065</b>	<b>0.604</b>
KITTI						
Plain	80.5%	93.9%	97.7%	0.158	0.064	5.415
Infogain	<b>82.9%</b>	<b>94.3%</b>	<b>98.2%</b>	<b>0.142</b>	<b>0.058</b>	<b>5.066</b>

TABLE V: Comparison with state-of-the-art results on the NYUD2 dataset. The first 5 rows are the results by recent depth estimation methods, the last row is the result by our approach.

	Accuracy			Error		
	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$	rel	log10	rms
Wang et al. [27]	60.5%	89.0%	97.0%	0.210	0.094	0.745
Liu et al. [24]	65.0%	90.6%	97.6%	0.213	0.087	0.759
Eigen et al. [25]	76.9%	95.0%	98.8%	0.158	-	0.641
Laina et al. [13]	81.1%	95.3%	98.8%	<b>0.127</b>	<b>0.055</b>	0.573
Ours	<b>83.1%</b>	<b>96.2%</b>	<b>98.8%</b>	0.132	0.057	<b>0.538</b>

TABLE VI: Comparison with state-of-the-art results on the KITTI dataset. We cap the maximum depth to 50 and 80 meters to compare with recent works. For the work in [5], we also report their results with additional training images in the CityScapes dataset [39] and denote as Godard et al. CS.

	Accuracy			Error		
	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$	rel	rmslog	rms
Cap 80 meters						
Liu et al. [24]	65.6%	88.1%	95.8%	0.217	-	7.046
Eigen et al. [3]	69.2%	89.9%	96.7%	0.190	0.270	7.156
Godard et al. [5]	81.8%	92.9%	96.6%	0.141	0.242	5.849
Godard et al. CS [5]	83.6%	93.5%	96.8%	0.136	0.236	5.763
Ours	<b>89.0%</b>	<b>96.7%</b>	<b>98.4%</b>	<b>0.120</b>	<b>0.192</b>	<b>4.533</b>
Cap 50 meters						
Garg et al. [4]	74.0%	90.4%	96.2%	0.169	0.273	5.104
Godard et al. [5]	84.3%	94.2%	97.2%	0.123	0.221	5.061
Godard et al. CS [5]	85.8%	94.7%	97.4%	0.118	0.215	4.941
Ours	<b>89.7%</b>	<b>96.8%</b>	<b>98.4%</b>	<b>0.117</b>	<b>0.189</b>	<b>3.753</b>

TABLE VII: Test results of depth estimation by classification and regression on the NYUD2 and KITTI datasets. For each dataset, the first row is the result of regression, the following row is the result of classification.

	Accuracy			Error		
	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$	rel	log10	rms
NYUD2						
regression	66.9%	92.1%	98.0%	0.215	0.084	0.730
classification	<b>72.3%</b>	<b>92.7%</b>	<b>98.3%</b>	<b>0.195</b>	<b>0.077</b>	<b>0.691</b>
KITTI						
regression	68.9%	89.4%	91.1%	0.256	0.092	7.160
classification	<b>79.9%</b>	<b>93.7%</b>	<b>97.6%</b>	<b>0.166</b>	<b>0.067</b>	<b>5.443</b>

TABLE VIII: Comparison with state-of-the-art results on the DIW dataset. The evaluation metric is Weighted Human Disagreement Rate (WHDR).

Method	WHDR
Baseline [12]	31.37%
Eigen [12]	25.70%
Chen-NYU [12]	31.31%
Chen-DIW [12]	22.14%
Chen-NYU-DIW [12]	14.39%
Ours-RDIS	18.05%
Ours-NYU-RDIS	<b>11.55%</b>

### C. State-of-the-art comparisons

In this section, we evaluate metric depth estimation on the NYUD2 and KITTI datasets and compare with recent depth estimation methods. During pretraining, we use 1K pairs of points in each input image. During finetuning, we discretize the continuous metric depths into 100 and 50 bins in log

space for the NYUD2 and KITTI datasets respectively. We also evaluate relative depth estimation on the Depth in the Wild (DIW) [12] dataset.

1) *NYUD2*: We finetune our model on the raw NYUD2 training set and test on the standard 654 images. We set the parameter  $\alpha$  of the information gain matrix to be 2.0. We compare our approach against several prior works and report the results in Table V, from which we can see that we achieve state-of-the-art results of 4 evaluation metrics without using any multi-scale network architecture, up-sampling or CRF post-processing. Fig. 4 illustrates some qualitative evaluations of our method compared against Liu et al. [24] and Eigen et al. [25].

2) *KITTI*: We finetune our model on the same training set in [5] which contains 33131 images and test on the same 697 images in [3]. But different with the depth estimation method proposed in [5] which applies both the left and right images in stereo pairs, we only use the left images. The missing values in the ground-truth depth maps are ignored during finetuning and evaluation. We set the parameter  $\alpha$  of the information gain matrix to be 0.2. In order to compare with the recent state-of-the-art results, we cap the maximum depth to both 80 meters and 50 meters and present the results in Table VI. We outperform state-of-the-art results of all evaluation metrics significantly. Some qualitative results are illustrated in Fig. 5. Our method yields outstanding visual predictions.

3) *DIW*: We evaluate relative depth estimation on the DIW test set and report the WHDR of 7 methods in Table VIII: 1) a baseline method that uses only the location of the query points: classify the lower point to be closer or guess randomly if the two points are at the same height (Baseline); 2) the

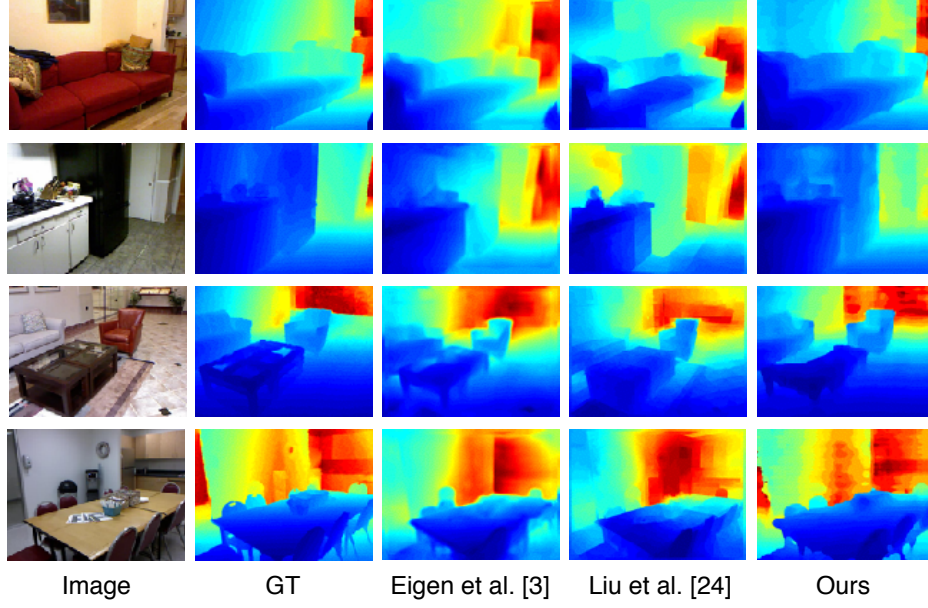


Fig. 4: Qualitative comparisons with state-of-the-art results on the NYUD2 dataset. The first two columns are RGB images and ground-truth depths respectively. The following 4 columns are predictions. Depths are shown in color (red is far, blue is close).

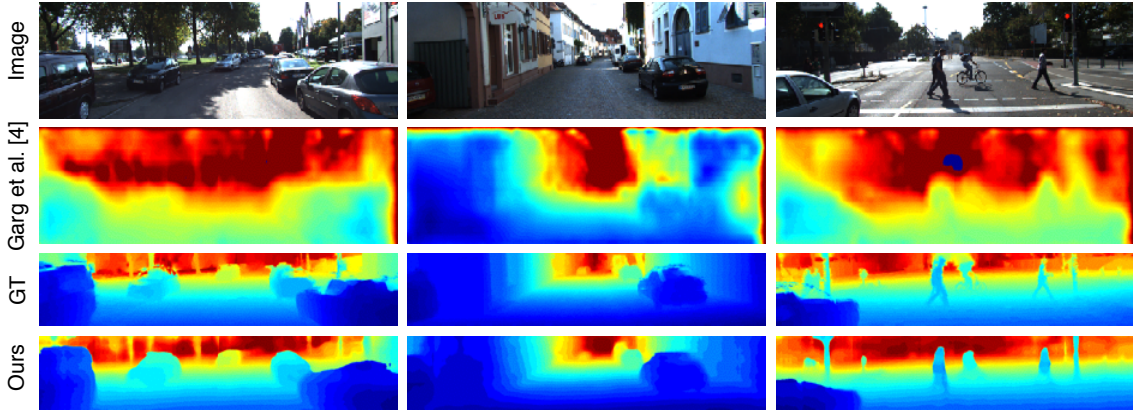


Fig. 5: Qualitative comparisons with state-of-the-art results on the KITTI dataset. Depths are shown in color (red is far, blue is close). Since the ground-truth captured by the velodyne is very sparse, we inpaint the ground-truth for visualization purposes. We also crop the ground-truth and our predictions to mask out the vast sky regions.

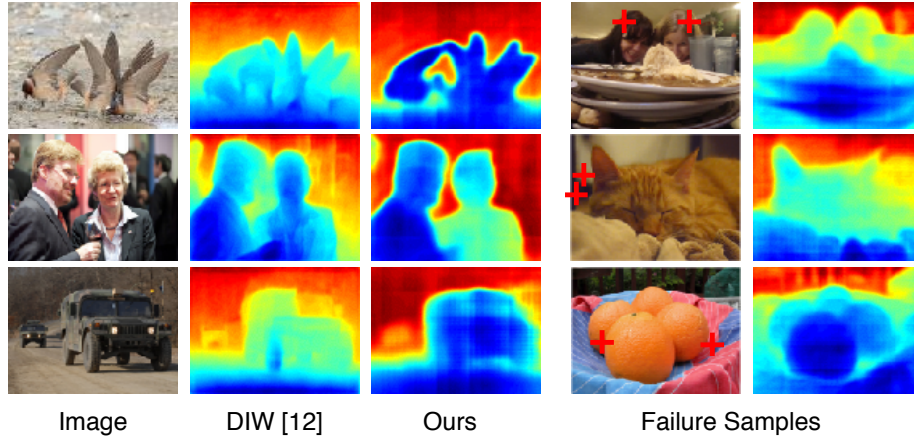


Fig. 6: Some examples of relative depth estimation of the DIW dataset. The first column are the RGB images, the second column are the predictions of Chen et al. [12], the third column are our predictions. The last two columns are some failure samples of our approach. The pairs of points labelled with ground-truth ordinal relations are marked as red crosses.



model trained by Eigen et al. [25] on the raw NYUD2 dataset (Eigen); 3) the model trained by Chen et al. [12] on the raw NYUD2 dataset (Chen-NYU); 4) the model trained by Chen et al. [12] on the DIW dataset (Chen-DIW). 5) the model by Chen et al. [12] pretrained on the raw NYUD2 dataset and finetuned on the DIW dataset (Chen-NYU-DIW). 6) our model trained on our RDIS dataset (Ours-RDIS). 7) our model pretrained on the raw NYUD2 dataset and finetuned on our RDIS dataset (Ours-NYU-RDIS). From the table we can see that even though we do not train our model on the DIW training set, we achieve state-of-the-art result on the DIW test set. We show some of our predicted relative depth maps as well as some failure samples in Fig. 6, from which we can see that our predicted relative depth maps are visually better. As for the failure samples, we can also predict satisfactory relative depth maps. Notably, the ground-truth pairs in failure samples are those points with almost equal distance. Given the fact that the equal relation is absent in DIW, we can conclude that we reach the nearly perfect performance on the DIW test set.

## V. CONCLUSION

We have proposed a new dataset Relative Depths in Stereo (RDIS) containing images labelled with dense relative depths. The ground-truth relative depths are obtained through existing stereo algorithm as well as manual post-processing. We have shown that augmenting benchmark RGB-D datasets with our proposed RDIS dataset, the performance of single-image depth estimation can be improved significantly.

Note that the goal of this work is to predict depths from single monocular images. However the application of our proposed RDIS dataset is not limited to this. With the learning scheme based on relative depths, we can perform 2D-to-3D conversion like Deep3D [6]. We leave this as our future work.

## REFERENCES

- [1] A. Saxena, A. Ng, and S. Chung, "Learning depth from single monocular images," in *Proc. Adv. Neural Inf. Process. Syst.*, 2005.
- [2] F. Liu, C. Shen, and G. Lin, "Deep convolutional neural fields for depth estimation from a single image," in *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2015.
- [3] D. Eigen, C. Puhrsch, and R. Fergus, "Depth map prediction from a single image using a multi-scale deep network," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014.
- [4] R. Garg and I. Reid, "Unsupervised cnn for single view depth estimation: Geometry to the rescue," in *Proc. Eur. Conf. Comp. Vis.*, 2016.
- [5] C. Godard, O. Mac Aodha, and G. J. Brostow, "Unsupervised monocular depth estimation with left-right consistency," in *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2017.
- [6] J. Xie, R. Girshick, and A. Farhadi, "Deep3d: Fully automatic 2d-to-3d video conversion with deep convolutional neural networks," in *Proc. Eur. Conf. Comp. Vis.*, 2016.
- [7] J. Zbontar and Y. LeCun, "Stereo matching by training a convolutional neural network to compare image patches," *J. Mach. Learn. Res.*, vol. 17, 2016.
- [8] A. Spyropoulos, N. Komodakis, and P. Mordohai, "Learning to detect ground control points for improving the accuracy of stereo matching," in *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2014, pp. 1621–1628.
- [9] W. Luo, A. G. Schwing, and R. Urtasun, "Efficient deep learning for stereo matching," in *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2016.
- [10] K. Zhang, J. Lu, and G. Lafruit, "Cross-based local stereo matching using orthogonal integral images," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 7, pp. 1073–1079, 2009.
- [11] D. Zoran, P. Isola, D. Krishnan, and W. T. Freeman, "Learning ordinal relationships for mid-level vision," in *Proc. IEEE Int. Conf. Comp. Vis.*, 2015.
- [12] W. Chen, Z. Fu, D. Yang, and J. Deng, "Single-image depth perception in the wild," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016.
- [13] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab, "Deeper depth prediction with fully convolutional residual networks," in *Proc. IEEE Int. Conf. 3D Vision*, October 2016.
- [14] V. Hedau, D. Hoiem, and D. Forsyth, "Thinking inside the box: Using appearance models and context based on room geometry," in *Proc. Eur. Conf. Comp. Vis.*, 2010, pp. 224–237.
- [15] A. Gupta, M. Hebert, T. Kanade, and D. M. Blei, "Estimating spatial layout of rooms using volumetric reasoning about objects and surfaces," in *Proc. Adv. Neural Inf. Process. Syst.*, 2010.
- [16] A. G. Schwing and R. Urtasun, "Efficient exact inference for 3d indoor scene understanding," in *Proc. Eur. Conf. Comp. Vis.*, 2012.
- [17] K. Karsch, C. Liu, and S. B. Kang, "Depthtransfer: Depth extraction from video using non-parametric sampling," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2014.
- [18] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.
- [19] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learn. Rep.*, 2015.
- [20] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2015.
- [21] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor segmentation and support inference from rgbd images," in *Proc. Eur. Conf. Comp. Vis.*, 2012.
- [22] A. Saxena, M. Sun, and A. Y. Ng, "Make3d: Learning 3d scene structure from a single still image," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2009.
- [23] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *Int. J. Robt. Res.*, 2013.
- [24] F. Liu, C. Shen, G. Lin, and I. D. Reid, "Learning depth from single monocular images using deep convolutional neural fields," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2016.
- [25] D. Eigen and R. Fergus, "Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture," in *Proc. IEEE Int. Conf. Comp. Vis.*, 2015.
- [26] B. Li, C. Shen, Y. Dai, A. van den Hengel, and M. He, "Depth and normal estimation from monocular images using regression on deep features and hierarchical crfs," in *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, June 2015.
- [27] P. Wang, X. Shen, Z. Lin, S. Cohen, B. Price, and A. L. Yuille, "Towards unified depth and semantic prediction from a single image," in *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, June 2015.
- [28] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2016.
- [29] J. Flynn, I. Neulander, J. Philbin, and N. Snavely, "Deepstereo: Learning to predict new views from the world's imagery," in *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2016.
- [30] Z. Cao, T. Qin, T.-Y. Liu, M.-F. Tsai, and H. Li, "Learning to rank: from pairwise approach to listwise approach," in *Proc. Int. Conf. Mach. Learn.*, 2007.
- [31] H. Hirschmuller and D. Scharstein, "Evaluation of stereo matching costs on images with radiometric differences," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 9, pp. 1582–1599, 2009.
- [32] H. Hirschmuller, "Stereo processing by semiglobal matching and mutual information," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2008.
- [33] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *Proc. Eur. Conf. Comp. Vis.*, 2016.
- [34] Z. Wu, C. Shen, and A. van den Hengel, "Wider or deeper: Revisiting the resnet model for visual recognition," 2016. [Online]. Available: <https://arxiv.org/pdf/1611.10080.pdf>
- [35] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn.*, 2015.
- [36] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "Imagenet large scale visual recognition challenge," *Int. J. Comp. Vis.*, vol. 115, no. 3, pp. 211–252, 2015.
- [37] B. Zhou, A. Khosla, Lapedriza, A. Torralba, and A. Oliva, "Places: An image database for deep scene understanding," 2016. [Online]. Available: <https://arxiv.org/abs/1610.02055>

- [38] T. Chen, M. Li, Y. Li, M. Lin, N. Wang, M. Wang, T. Xiao, B. Xu, C. Zhang, and Z. Zhang, "Mxnet: A flexible and efficient machine learning library for heterogeneous distributed systems," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015.
- [39] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2016.