Monocular Depth Estimation using Multi-Scale Continuous CRFs as Sequential Deep Networks

Dan Xu, *Student Member, IEEE,* Elisa Ricci, *Member, IEEE,* Wanli Ouyang, *Senior Member, IEEE,* Xiaogang Wang, *Senior Member, IEEE,* Nicu Sebe, *Senior Member, IEEE*

Abstract—Depth cues have been proved very useful in various computer vision and robotic tasks. This paper addresses the problem of monocular depth estimation from a single still image. Inspired by the effectiveness of recent works on multi-scale convolutional neural networks (CNN), we propose a deep model which fuses complementary information derived from multiple CNN side outputs. Different from previous methods using concatenation or weighted average schemes, the integration is obtained by means of continuous Conditional Random Fields (CRFs). In particular, we propose two different variations, one based on a cascade of multiple CRFs, the other on a unified graphical model. By designing a novel CNN implementation of mean-field updates for continuous CRFs, we show that both proposed models can be regarded as sequential deep networks and that training can be performed end-to-end. Through an extensive experimental evaluation, we demonstrate the effectiveness of the proposed approach and establish new state of the art results for the monocular depth estimation task on three publicly available datasets, *i.e.* NYUD-V2, Make3D and KITTI.

Index Terms—Monocular Depth Estimation, Convolutional Neural Networks (CNN), Deep Multi-Scale Fusion, Conditional Random Fields (CRFs).

✦

1 INTRODUCTION

WHILE estimating the depth of a scene from a single image is a natural ability for humans, devising computational models for accurately predicting depth information from RGB data is a challenging task. Many attempts have been made to address this problem in the past. In particular, recent works have achieved remarkable performance thanks to powerful deep learning models [11], [12], [30], [36]. Assuming the availability of a large training set of RGB-depth pairs, monocular depth prediction from single images can be regarded as a pixel-level continuous regression problem and Convolutional Neural Network (CNN) architectures are typically employed.

In the last few years significant efforts have been made in the research community to improve the performance of CNN models for pixel-level prediction tasks (*e.g.* semantic segmentation, contour detection). Previous works have shown that, for depth estimation as well as for other pixel-level classification or regression problems, more accurate estimates can be obtained by combining information from multiple scales [9], [11], [46], [48]. This can be achieved in different ways, *e.g.* fusing feature maps corresponding to different network layers or designing an architecture with multiple inputs corresponding to images at different resolutions. Other works have demonstrated that, by adding a Conditional Random Field (CRF) in cascade to

- *Dan Xu, Nicu Sebe are with the Department of Information Engineering and Computer Science, University of Trento, Trento, Italy. E-mail: {dan.xu, niculae.sebe}@unitn.it*
- *Elisa Ricci is with Fondazione Bruno Kessler. Email: eliricci@fbk.eu*
- *Wanli Ouyang is with the School of Electrical and Information Engineering, The University of Sydney. Email: wanli.ouyang@sydney.edu.au*
- *Xiaogang Wang is with the Department of Electronic Engineering, The Chinese University of Hong Kong. Email: xgwang@ee.cuhk.edu.hk*
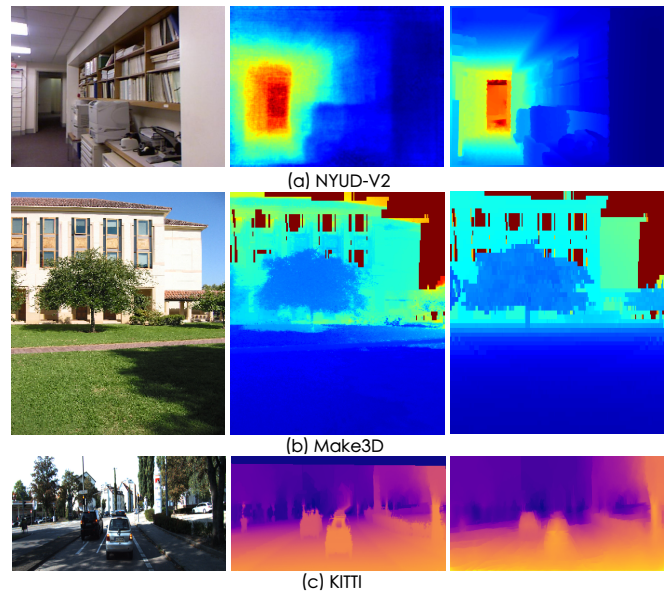
Fig. 1. Monocular depth estimation results on three different benchmark datasets, *i.e.* NYUD-V2 (the 1st row), Make3D (the 2nd row) and Kitti (the 3rd row), using the proposed multi-scale CRF model with a pre-trained CNN (*e.g.* VGG Convolution-Deconvolution [34]). From left to right, each column is original RGB images, the recovered depth maps and the groundtruth, respectively.

a convolutional neural architecture, the performance can be greatly enhanced and the CRF can be fully integrated within the deep model enabling end-to-end training with back-propagation [51]. However, these works mainly focus on pixel-level prediction problems in the discrete domain (*e.g.* semantic segmentation). While complementary, so far these strategies have been only considered in isolation and no previous works have exploited multi-scale information within a CRF inference framework.

In this paper we argue that, benefiting from the flexibility

and the representational power of graphical models, we can optimally fuse representations derived from multiple CNN side-output layers using structured constraints, improving performance over traditional multi-scale strategies. By exploiting this idea, we introduce a novel framework to estimate depth maps from single still images. Opposite to previous work fusing multi-scale features by weighted averaging or concatenation, we propose to integrate multi-layer side-output information by designing a novel approach based on continuous CRFs. Specifically, we present two different methods. The first approach is based on a single multi-scale unified CRF model, while the other considers a cascade of scale-specific CRFs. We also show that, by introducing a common CNN implementation for mean-fields updates in continuous CRFs, both models are equivalent to sequential deep networks and an end-to-end approach can be devised for training. Through extensive experimental evaluation we demonstrate that the proposed CRF-based approach produces more accurate depth maps than traditional multi-scale approaches for pixel-level prediction tasks [16], [46]. Moreover, by performing experiments on the publicly available NYU Depth V2 [43], Make3D [41] and KITTI [14] datasets, we show that our approach is able to robustly reconstruct depth with good visual quality (Fig.1) and outperforms state of the art methods for the monocular depth estimation task.

This paper extends our earlier work [50] through proposing and investigating different multi-scale connection structures for message passing, further enriching the related works, providing more approach details, and significantly expanding experimental results and analysis. To summarize, the contribution of this paper is threefold:

- Firstly, we propose a novel approach for predicting depth maps from RGB inputs which exploits multi-scale estimations derived from CNN inner semantic layers by structurally fusing them within a unified CNN-CRF framework.
- Secondly, as the task of pixel-level depth prediction implies inferring a set of continuous values, we show how mean field (MF) updates can be implemented as sequential deep models, enabling end-to-end training of the whole network. We believe that our MF implementation will be useful not only to researchers working on depth prediction, but also to those interested in other problems involving continuous variables. Therefore, our code is made publicly available at https://github.com/danxuhk/ContinuousCRF-CNN.git.
- Thirdly, our experiments demonstrate that the proposed multi-scale CRF framework is superior to previous methods integrating information from different semantic network layers by combining multiple losses [46] or by adopting feature concatenations [16]. We also show that our approach outperforms state of the state of the art monocular depth estimation methods on public benchmarks and that the proposed CRF-based models can be employed in combination with different pre-trained CNN architectures, consistently enhancing their performance.

The remainder of this paper is organised as follows. We first introduce related work in Section 2, and then the proposed multi-scale CRF models for monocular depth estimation is presented in Section 3. We further elaborate how the proposed models can be implemented as sequential neural network for end-to-end joint optimization in Section 4. The experimental results and analysis are elaborated in Section 5, and we conclude the paper in Section 6.

2 RELATED WORK

Our approach is built upon recent successes of deep CNN architectures for image classification [17], [23], [44] and fully convolutional networks for dense semantic image segmentation [33], [34]. We briefly introduce the most related works by organizing them into three main aspects, *i.e.* monocular depth estimation, multi-scale CNN and dense pixel-level prediction via combination of CNN and CRFs.

Monocular depth estimation. Previous approaches for depth estimation from single images can be grouped into three main categories: (i) methods operating on hand crafted features, (ii) methods based on graphical models and (iii) methods adopting deep convolutional neural networks.

Earlier works addressing the depth prediction task belong to the first category. Hoiem *et al.* [18], [19] proposed photo pop-up, a fully automatic method for creating a basic 3D model from a single photograph by introducing an assumption of 'ground-vertical' geometric structure. Karsch *et al.* [20] developed Depth Transfer, a non parametric approach based on SIFT Flow, where the depth of an input image is reconstructed by transferring the depth of multiple similar images and then applying some warping and optimizing procedures. Instead of directly recovering depth from appearance features, Liu *et al.* [29] explored using semantic scene segmentation results to guide the 3-D depth reconstruction. Similarly, Ladicky *et al.* [25] also demonstrated the benefit of combining semantic object labels with depth features. However, the hand-crafted representations are not robust enough for this challenging problem.

In the second category, some works exploited the flexibility of graphical models to reconstruct depth information. For instance, Delage *et al.* [10] proposed a dynamic Bayesian framework for recovering 3D information from indoor scenes. A discriminatively-trained multiscale Markov Random Fields (MRFs) were introduced in [39], [40], in order to optimally fuse local and global features. Depth estimation was treated as an inference problem in a discrete-continuous CRF model in [32]. However, these works did not employ deep networks.

More recent approaches for depth estimation are based on CNNs [11], [27], [30], [38], [45]. For instance, Eigen *et al.* [12] proposed a multi-scale approach for depth prediction, considering two deep networks, one performing a coarse global prediction based on the entire image, and the other refining predictions locally. This approach was extended in [11] to handle multiple tasks (*e.g.* semantic segmentation, surface normal estimation). Wang *et al.* [45] introduced a CNN for joint depth estimation and semantic segmentation. The obtained estimates were further refined with Hierarchical CRFs. The most similar work to ours is [30], where the representational power of deep CNN and continuous CRFs is jointly exploited for depth prediction. However, the method proposed in [30] is based on superpixels and the information associated to multiple scales is not exploited in their graphical model.
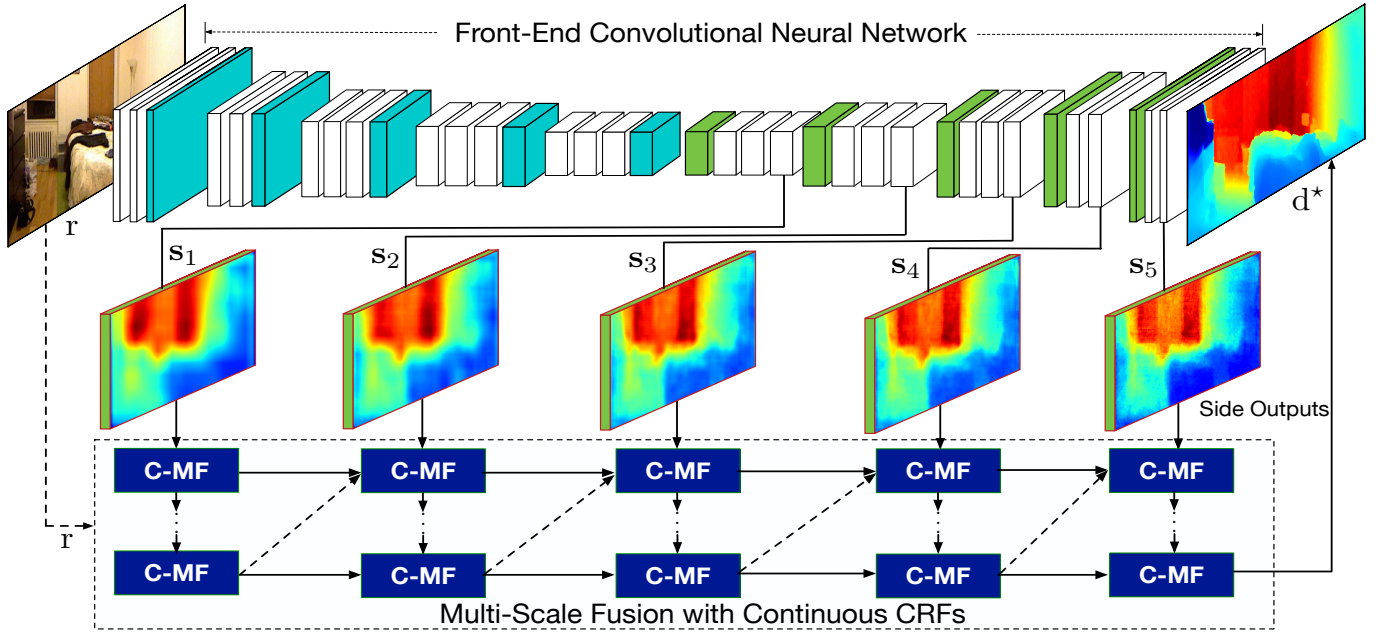
Fig. 2. Overview of the proposed deep architecture. Our model is composed of two main components: a front-end CNN and a fusion module. The fusion module uses continuous CRFs to integrate multiple side output maps of the front-end CNN. We consider two different CRFs-based multi-scale models and implement them as sequential deep networks by stacking several elementary blocks, the C-MF blocks.
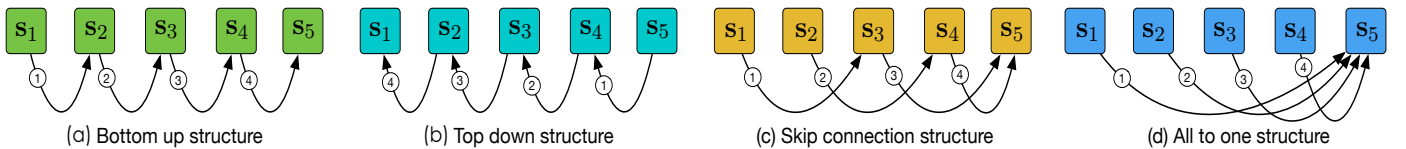

Fig. 3. Illustration of different multi-scale message passing structures for the integration of the multi-scale predictions s_1 to s_5 produced from the front-end convolutional network. The arrows represent the direction of the message passing, and the numbers in circles represent the order. The dashed line box in Fig. 2 shows a bottom-up passing structure.

Multi-Scale CNNs. The problem of combining information from multiple scales has recently received considerable interest in various computer vision tasks. In [46] a deeply supervised fully convolutional neural network was proposed for edge detection by weighted combination of multiple side outputs. Skip-layer networks, where the feature maps derived from different semantic layers of a primary front-end network are jointly considered in an output layer, have also become very popular [3], [6], [33]. Other works considered multi-stream architectures, where multiple parallel networks receiving inputs at different scale are fused [4]. Cai *et al.* [5] proposed a multi-scale method via combining the predictions obtained from feature maps with different resolution for object detection. Dilated convolutions (*e.g. dilation* or *à trous*) have been also employed in different deep network models in order to aggregate multi-scale contextual information [7]. However, in these works, the multi-scale representations or estimations are typically combined by using simple concatenation or weighted averaging operation. We are not aware of previous works exploring fusing deep multi-scale information within a CRF framework.

Dense pixel-level prediction via combination of CNN and CRFs. The combination of CNN and CRFs has shown great usefulness for dense pixel-level structured prediction [21], [42]. Some existing works utilize CRFs as a post processing module for further refining the predictions from the CNN [8], [35]. To benefit from end-to-end learning,

Zhang *et al.* [51] proposed a CRF-RNN model which jointly optimizes a front-end deep network with a discrete CRF for semantic image segmentation. Xu *et al.* [47] proposed an attention-gated deep CRF framework for pixel-level contour prediction. However, as far as we know, this work is a first attempt to combine multi-scale continuous CRFs with deep convolutional neural network for constructing a unified model for end-to-end monocular depth estimation.

3 MULTI-SCALE CRF MODELS FOR MONOCULAR DEPTH ESTIMATION

In this section we introduce our deep model with the designed multi-scale continuous CRFs for monocular depth estimation from RGB images. We first formalize the problem of depth prediction and give a brief overview of the proposed approach. Then, we describe two different variants of the proposed multi-scale model, one based on a cascade of CRFs and the other on a single multi-scale unified CRFs.

3.1 Problem Formulation and Overview

Following previous works we formulate the task of depth prediction from monocular RGB input as the problem of learning a non-linear mapping $F : \mathcal{I} \to \mathcal{D}$ from the image space $\mathcal{I}$ to the output depth space $\mathcal{D}$. More formally, let $\mathcal{Q} = \{(\mathbf{r}_i, \bar{\mathbf{d}}_i)\}_{i=1}^{Q}$ be a training set of Q pairs, where $\mathbf{r}_i \in \mathcal{I}$

denotes an input RGB image with N pixels and $\bar{\mathbf{d}}_i \in \mathcal{D}$ represents its corresponding real-valued depth map.

For learning F we consider a deep model made of two main building blocks (Fig. 2). The first component is a CNN architecture with a set of intermediate side outputs $\mathcal{S} = \{\mathbf{s}_l\}_{l=1}^{L}$, $\mathbf{s}_l \in R^N$, produced from L different layers with a mapping function $f_s(\mathbf{r}; \Theta, \boldsymbol{\theta}_l) \rightarrow \mathbf{s}_l$. For simplicity, we denote with Θ the set of front-end network layer parameters and with $\boldsymbol{\theta}_l$ the parameters of the network branch producing the side output associated to the l-th layer (see Section 5.1 for details of our implementation). In the following we denote this network as the front-end CNN.

The second component of our model is a fusion block. As shown in previous works [3], [33], [46], features generated from different CNN layers capture complementary information. The main idea behind the proposed fusion block is to use CRFs to effectively integrate the side output maps of our front-end CNN for robust depth prediction. Our approach develops from the intuition that these representations can be combined within a sequential framework, *i.e.* performing depth estimation at a certain scale and then refining the obtained estimates in the subsequent level. Specifically, we introduce and compare two different multi-scale models, both based on CRFs, and corresponding to two different versions of the fusion block. The first model is based on a **single multi-scale unified CRFs**, which integrates information available from different scales and simultaneously enforces smoothness constraints between the estimated depth values of neighboring pixels and neighboring scales. The second model implements a **cascade of scale-specific CRFs**: at each scale l a CRF is employed to recover the depth information from side output maps $\mathbf{s}_l$ and the outputs of each CRF model are used as additional observations for the subsequent model. In Section 3.2.1 we describe the two models in details, while in Section 4 we show how they can be implemented as sequential deep networks by stacking several elementary blocks. We call these blocks C-MF blocks, as they implement Mean Field updates for Continuous CRFs (Fig. 2).

3.2 Multi-scale Fusion with Continuous CRFs

We now elaborate the proposed CRF-based models for fusing multi-scale side-outputs derived from different semantic layers of the front-end deep convolutional neural networks.

3.2.1 Multi-Scale Unified CRF Model

Given a vector $\hat{\mathbf{s}}$ with a dimension of $L \times N$ obtained by concatenating the side output score maps $\{\mathbf{s}_1, \ldots, \mathbf{s}_L\}$ and a vector $\mathbf{d}$ with a dimension of $L \times N$ expressing real-valued output variables, we define a CRF modeling the following conditional distribution:

$$P(\mathbf{d}|\hat{\mathbf{s}}) = \frac{1}{Z(\hat{\mathbf{s}})} \exp\{-E(\mathbf{d}, \hat{\mathbf{s}})\}, \tag{1}$$

where $Z(\hat{\mathbf{s}}) = \int_{\mathbf{d}} \exp\{-E(\mathbf{d}, \hat{\mathbf{s}})\} d\mathbf{d}$ is the partition function [26] acting as a normalization factor for probabilities. The energy function is defined as:

$$E(\mathbf{d}, \hat{\mathbf{s}}) = \sum_{i=1}^{N} \sum_{l=1}^{L} \phi(d_i^l, \hat{\mathbf{s}}) + \sum_{i,j} \sum_{l,k} \psi(d_i^l, d_j^k), \tag{2}$$

and d_i^l indicates the hidden variable associated to scale l and pixel i. The first term is the sum of quadratic unary terms defined as:

$$\phi(d_i^l, \hat{\mathbf{s}}) = (d_i^l - s_i^l)^2, \tag{3}$$

where s_i^l is the regressed depth value at pixel i and scale l obtained with $f_s(\mathbf{r}; \Theta, \boldsymbol{\theta}_l)$. The second term is the sum of pairwise potentials describing the relationship between pairs of hidden variables d_i^l and d_j^k and is defined as follows:

$$\psi(d_i^l, d_j^k) = \sum_{m=1}^{M} \beta_m w_m(i, j, l, k, \mathbf{r})(d_i^l - d_j^k)^2, \tag{4}$$

where $w_m(i, j, l, k, \mathbf{r})$ is a weight which specifies the relationship between the estimated depth of the pixels i and j at scale l and k, respectively; M is the number of kernels.

To perform inference we rely on the mean-field theory to approximate $P(\mathbf{d}|\hat{\mathbf{s}})$ with another distribution $Q(\mathbf{d}|\hat{\mathbf{s}})$, where $Q(\mathbf{d}|\hat{\mathbf{s}}) = \prod_{i=1}^{N} \prod_{l=1}^{L} Q_{i,l}(d_i^l|\hat{\mathbf{s}})$, expressing a product of independent marginals. By minimizing the Kullback-Leibler divergence between the distribution of P and Q, we obtain the solution of Q. As the log distribution $\log Q_{i,l}(d_i^l|\hat{\mathbf{s}})$ has a quadratic form w.r.t. d_i^l and can be represented as Gaussian distribution, the following mean-field updates can be derived:

$$\gamma_{i,l} = 2\left(1 + 2\sum_{m=1}^{M} \beta_m \sum_k \sum_{j,i} w_m(i, j, l, k, \mathbf{r})\right), \tag{5}$$

$$\mu_{i,l} = \frac{2}{\gamma_{i,l}}\left(s_i^l + 2\sum_{m=1}^{M} \beta_m \sum_k \sum_{j,i} w_m(i, j, l, k, \mathbf{r})\mu_{j,k}\right). \tag{6}$$

Here $\gamma_{i,l}$ and $\mu_{i,l}$ are the variance and mean of the distribution $Q_{i,l}$, respectively.

To define the weights $w_m(i, j, l, k, \mathbf{r})$ we introduce the following assumptions. First, we assume that the estimated depth at scale l only depends on the depth estimated at previous scale. Second, for relating pixels at the same and at previous scale, we set weights depending on m kernel functions K_m^{ij}, which consists of Gaussian kernels with form of $\exp\left(-\frac{\|\mathbf{h}_i^m - \mathbf{h}_j^m\|_2^2}{2\theta_m^2}\right)$. Here, $\mathbf{h}_i^m$ and $\mathbf{h}_j^m$ indicate some features derived from the input image $\mathbf{r}$ for pixels i and j. θ_m are user-defined bandwidth parameters [22]. Following previous works [22], [51], we use pixel positions and color values as features, leading to two kernel functions, *i.e.* a bilateral appearance kernel using both the pixel positions and the color value features and a spatial smoothness kernel using only the pixel positions features, for modeling dependencies of pixels at scale l and other two for relating pixels at neighboring scales. Under these assumptions, the mean-field updates (5) and (6) can be rewritten as:

$$\gamma_{i,l} = 2\left(1 + 2\sum_{m=1}^{2} \beta_m \sum_{j \neq i} K_m^{ij} + 2\sum_{m=3}^{4} \beta_m \sum_{j,i} K_m^{ij}\right), \tag{7}$$

$$\mu_{i,l} = \frac{2}{\gamma_{i,l}}\left(s_i^l + 2\sum_{m=1}^{2} \beta_m \sum_{j \neq i} K_m^{ij}\mu_{j,l}, \right.$$
$$\left. + 2\sum_{m=3}^{4} \beta_m \sum_{j,i} K_m^{ij}\mu_{j,l-1}\right). \tag{8}$$

The parameters β_m need to be learned during training. We will present the details of the parameter optimization in
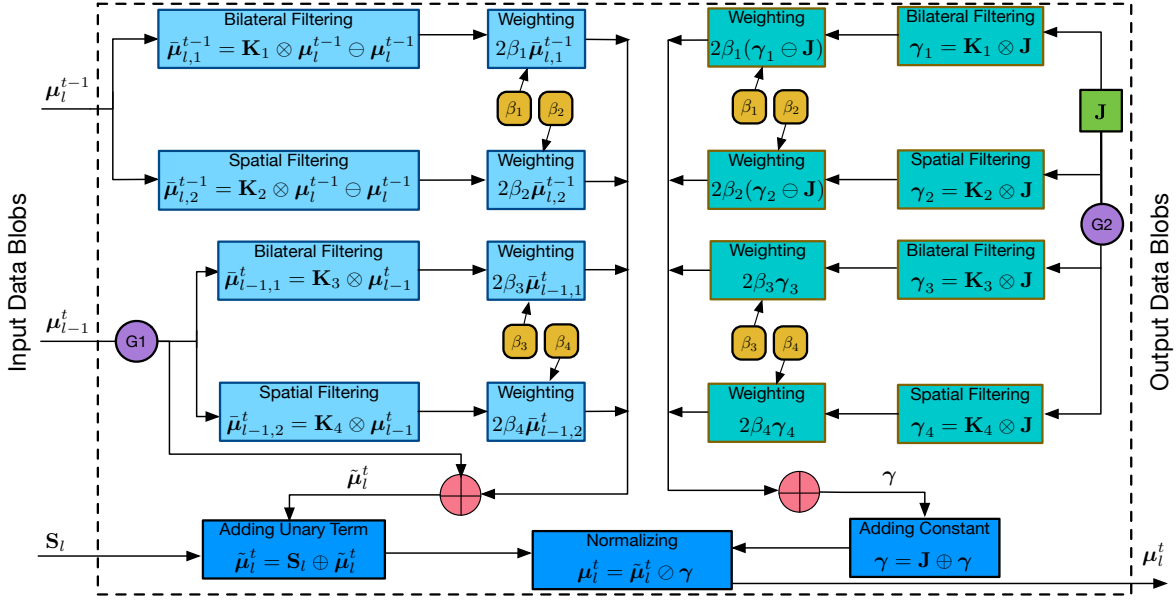
Fig. 4. Detailed computing flow graph of the proposed C-MF block. $\mathbf{J}$ represents a $W \times H$ matrix with all elements equal to one. The symbols $\oplus$, $\ominus$, $\oslash$ and $\otimes$ indicate element-wise addition, subtraction, division and Gaussian convolution operation, respectively. G1 and G2 represent two gate functions for controlling the computing flow.

Section 4. Given a new test image, the optimal $\tilde{\mathbf{d}}$ can be computed via maximizing the log conditional probability [37], *i.e.* $\tilde{\mathbf{d}} = \arg\max_{\mathbf{d}} \log(Q(\mathbf{d}|\mathbf{S}))$, where $\tilde{\mathbf{d}} = [\mu_{1,1}, ..., \mu_{N,L}]$ is a vector of the $L \times N$ mean values associated to $Q(\mathbf{d}|\hat{\mathbf{s}})$. We take the estimated variables at the finest scale L (*i.e.* $\mu_{1,L}, ..., \mu_{N,L}$) as our predicted depth map $\mathbf{d}^\star$.

3.2.2 Multi-Scale Cascade CRF Model

The cascade model is based on a set of L CRF models, each one associated to a specific scale l, which are progressively stacked such that the estimated depth at previous scale can be used as observations of the CRF model in the following scale level. Each CRF is used to compute the output vector $\mathbf{d}^l$ and it is constructed considering the side output representations $\mathbf{s}^l$ and the estimated depth at the previous step $\tilde{\mathbf{d}}^{l-1}$ as observed variables, *i.e.* $\mathbf{o}^l = [\mathbf{s}^l, \tilde{\mathbf{d}}^{l-1}]$. The associated energy function of the CRF model is defined as:

$$E(\mathbf{d}^l, \mathbf{o}^l) = \sum_{i=1}^{N} \phi(d_i^l, \mathbf{o}^l) + \sum_{i \neq j} \psi(d_i^l, d_j^l). \quad (9)$$

The unary and pairwise terms can be defined analogously to the above-introduced unified multi-scale model. In particular the unary term, reflecting the similarity between the observation o_l^i and the hidden depth value d_i^l, is:

$$\phi(y_i^l, \mathbf{o}^l) = (d_i^l - o_i^l)^2, \quad (10)$$

where o_i^l is obtained via combining the regressed depth from the side output $\mathbf{s}^l$ and the map $\mathbf{d}^{l-1}$ estimated by the CRF at previous scale. In our implementation we simply consider $o_i^l = s_i^l + \tilde{d}_i^{l-1}$, but other alternative strategies can be also considered. The pairwise potentials, used to force neighboring pixels with similar appearance to have close depth values, are:

$$\psi(d_i^l, d_j^l) = \sum_{m=1}^{M} \beta_m K_m^{ij}(d_i^l - d_j^l)^2, \quad (11)$$

where we consider $M = 2$ Gaussian kernels, one for appearance features, and the other accounting for pixel positions.

Similar to the multi-scale CRF model, under mean-field approximation, the following updates can be derived:

$$\gamma_{i,l} = 2\left(1 + 2\sum_{m=1}^{M} \beta_m \sum_{j \neq i} K_m^{ij}\right), \quad (12)$$

$$\mu_{i,l} = \frac{2}{\gamma_{i,l}}\left(o_i^l + 2\sum_{m=1}^{M} \beta_m \sum_{j \neq i} K_m^{ij} \mu_{j,l}\right). \quad (13)$$

At the test time, we use the estimated depth variables corresponding to the cascade CRF model of the finest scale L as our final predicted depth map $\mathbf{d}^\star$.

4 MULTI-SCALE MODELS AS SEQUENTIAL DEEP NETWORKS

In this section, we describe how the two proposed CRFs-based models can be implemented as sequential deep networks, enabling end-to-end training of our whole deep network model (the front-end CNN and the fusion module). We first show how the mean-field iterations derived for the multi-scale and the cascade models can be implemented by designing a common structure, the continuous mean-field updating (C-MF) block, consisting into stack of a series of CNN operations. Then, we present the resulting sequential network structures and details of the training phase for optimizing the whole deep network.

4.1 C-MF: a Common CNN Implementation of Continuous Mean-Field Updating

By analyzing the two proposed CRF models, we can observe that the mean-field updates derived for the cascade and for the multi-scale models share common terms. As stated above, the main difference between the two is the way the estimated depth at previous scale is handled at the current scale. In the multi-scale CRFs, the relationship among neighboring scales is modeled in the hidden variable space, while

(a) The proposed multi-scale cascade CRF model as sequential neural network using the C-MF block.

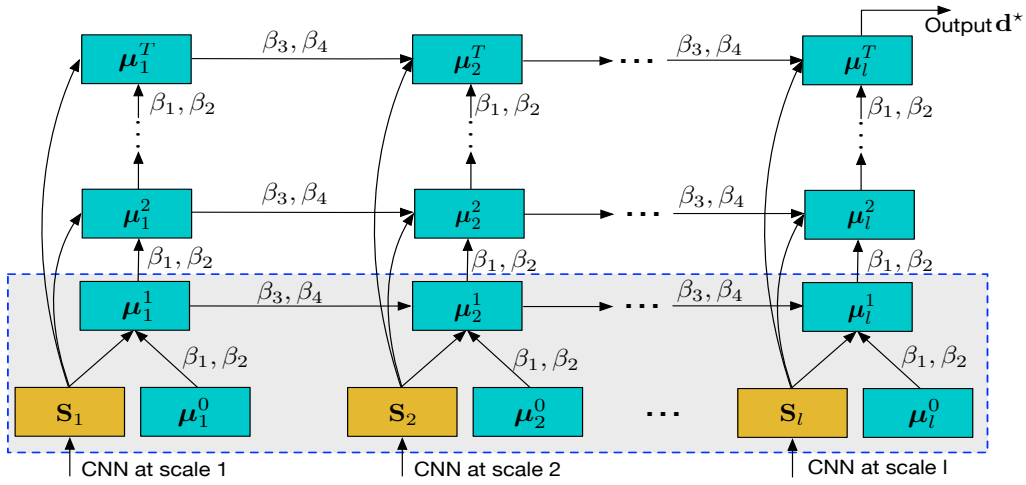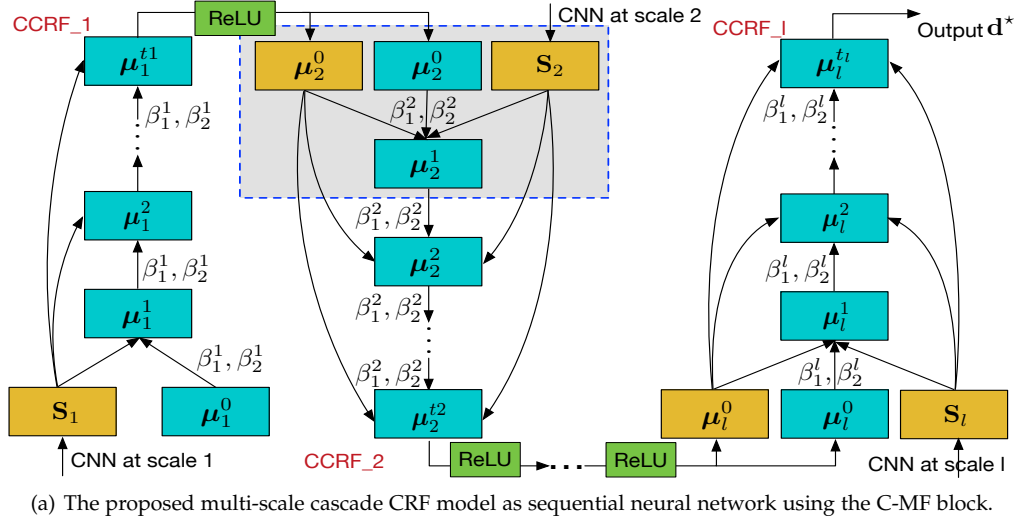(b) The proposed multi-scale unified CRF model as sequential neural network using the C-MF block.

Fig. 5. Description of the proposed two CRF models as sequential deep networks. The blue and yellow boxes indicate the estimated variables and observations, respectively. The parameters β_m are used for mean-field updates. As in the cascade model parameters are not shared among different CRFs, we use the notation β_1^l, β_2^l to denote parameters associated to the l-th scale.

in the cascade CRFs the depth estimated at previous scale acts as an observed variable.

Starting from this observation, in this section we show how the computation of Eq. (8) and Eq. (13) can be implemented with a common structure. Figure 4 describes in details these computations. In the following, for the sake of clarity, we introduce matrix representation. Let $\mathbf{S}_l \in \mathbb{R}^{W \times H}$ be the matrix obtained by rearranging the $N = W \times H$ pixels corresponding to the side output vector $\mathbf{s}_l$ and $\boldsymbol{\mu}_l^t \in \mathbb{R}^{W \times H}$ the matrix of the estimated output depth variables associated to scale l and mean-field iteration t. To implement the multi-scale model at each iteration t, $\boldsymbol{\mu}_l^{t-1}$ and $\boldsymbol{\mu}_{l-1}^t$ are convolved by two Gaussian kernels. Following [22], we use a spatial and a bilateral kernel. As Gaussian convolutions represent the computational bottleneck (requiring a complexity of $\mathcal{O}(N^2)$) in the mean-field iterations, we adopt the permutohedral lattice implementation [1] to approximate the filter response calculation reducing the computational cost from quadratic to linear [37]. The weighing of the parameters β_m is performed as a convolution with a 1×1 kernel. Then, the outputs are combined and are added to the side-output maps $\mathbf{S}_l$. Finally, a normalization step follows,

corresponding to the calculation of Eq. (7). The normalization matrix $\boldsymbol{\gamma} \in \mathbb{R}^{W \times H}$ is also computed by considering convolutions with Gaussian kernels and weighting with parameters β_m. It is worth noting that the normalization step in our mean-field updates for continuous CRFs is substantially different from that of discrete CRFs in CRF-RNN [51] based on a softmax function.

In the cascade CRF model, differently from the multi-scale unified CRF model, $\boldsymbol{\mu}_{l-1}^t$ acts as an observed variable. To design a common C-MF block among the two models, we introduce two gate functions G1 and G2 (Fig. 4) controlling the computing flow and allowing to easily switch between the two approaches. Both gate functions accept a user-defined boolean parameter. In our setting, the value 1 corresponds to the multi-scale CRF and the value 0 corresponds to the cascade model. Specifically, if G1 is equal to 1, the gate function G1 passes $\boldsymbol{\mu}_{l-1}^t$ to the Gaussian filtering block, otherwise passes it to the element-wise addition block with the computed message. Similarly, G2 controls the computation of the normalization terms and switches between the computation of Eq. (7) and Eq. (12). In other words, if G2 equals to 0, then the Gaussian filtering and weighting

operations for γ_3 and γ_4 are disabled. Importantly, for each step in the C-MF block we implement the calculation of error differentials for the back-propogation as in [51].

There are two different types of CRF parameters to be learned, *i.e.* the bandwidth parameters θ_m and the Gaussian-kernel weights β_m. For optimizing these CRF parameters, similar to [22], the bandwidth values θ_m are pre-defined for simplifying the calculation, and we implement the backward differential computation for the weights of Gaussian kernels β_m. In this way β_m are learned automatically with back-propagation.

4.2　From Mean-Field Updates to Sequential Deep Networks

Fig. 4 illustrates the implementation of the proposed two CRF-based models using the designed C-MF block described above. In the figure, each blue-dashed box is associated to a mean-field iteration. The cascade model as shown in Fig. 5(b) consists of L single-scale CRFs. At the l-th scale, t_l mean-field iterations are performed and then the estimated depth outputs are passed to another CRF model of the subsequent scale after a Rectified Linear Unit (ReLU) operation. The ReLU used here has two aspects of consideration: first the depth predictions should be always positive, and second we want to increase the nonlinearity of the sequential network for better mapping. To implement a single-scale CRF, we stack t_l C-MF blocks and make them share the parameters, while we learn different parameters for different CRFs. For the multi-scale model, one full mean-field update involves L scales simultaneously, obtained by combining L C-MF blocks. We further stack T iterations for learning and inference. The parameters corresponding to different scales and different mean-field iterations are shared. In this way, by using the common C-MF layer, we implement the two proposed multi-scale continuous CRFs models as deep sequential networks enabling end-to-end training with the front-end network.

4.3　Multi-Scale Message Passing Structures

The proposed work aims at multi-scale structured fusion and prediction, the connection structure between the different multi-scale predictions for message passing plays an important role in the performance. In this section, we thus propose and investigate different message passing structures. Fig. 3 illustrates several structures include top down structure, skip-connection structure and all to one structure. The top down structure is similar to the bottom up structure depicted in Fig. 2, which gradually refines the score maps from coarse to fine. The skip connection structure aims at utilizing more complementary information via skipping scales. The all to one structure uses all the other scales to refine the finest scale. Since all the message passing structures involve two scales at each time, we are able to build all these proposed connection structures by using the proposed aforementioned neural-network implemented C-MF block. The experimental investigation of these structures is illustrated in the experimental part.

TABLE 1
The parameter details of the sub-network for generating the side output from the last-scale convolutional block of ResNet-50.

Name	conv_s5_1	deconv_s5_1	deconv_s5_2
Type	conv	deconv	deconv
Kernel	$3 \times 3 \times 1024$	$4 \times 4 \times 512$	$4 \times 4 \times 256$
Stride, Padding	1, 1	2, 1	2, 1
Activation	ReLU	ReLU	ReLU
Name	deconv_s5_3	deconv_s5_4	pred
Type	deconv	deconv	deconv & crop
Kernel	$4 \times 4 \times 128$	$4 \times 4 \times 64$	$4 \times 4 \times 1$
Stride, Padding	2, 1	2, 1	2, 1
Activation	ReLU	ReLU	-

4.4　Optimization of The Whole Network

We train the whole network using a two phase scheme. In the first phase (pretraining), the parameters of the base front-end network $\boldsymbol{\Theta}$ and the parameters of the side-output generation sub-branch networks $\boldsymbol{\vartheta} = \{\boldsymbol{\theta}_l\}_{l=1}^{L}$ are learned by minimizing the sum of L distinct side losses as in [46], corresponding to L side outputs. We define the optimization objective using a square loss over Q training samples as follows:

$$\{\boldsymbol{\Theta}^*, \boldsymbol{\vartheta}^*\} = \arg\min_{\boldsymbol{\Theta}, \boldsymbol{\theta}_l} \sum_{l=1}^{L} \sum_{i=1}^{Q} \|f_s(\mathbf{r}_i; \boldsymbol{\Theta}, \boldsymbol{\theta}_l) - \tilde{\mathbf{d}}_i\|_2^2, \quad (14)$$

where $\tilde{\mathbf{d}}_i$ denotes the i-th ground-truth sample. In the second phase (fine tuning), we initialize the front-end network with the learned parameters $\{\boldsymbol{\Theta}^*, \boldsymbol{\vartheta}^*\}$ in the first phase, and jointly fine-tune with the proposed multi-scale CRF models to compute the optimal value of the parameters $\boldsymbol{\Theta}$, $\boldsymbol{\vartheta}$ and $\boldsymbol{\beta}$, with $\boldsymbol{\beta} = \{\beta_m\}_{m=1}^{M}$. The entire network is learned with Stochastic Gradient Descent (SGD) by minimizing a square loss

$$\{\boldsymbol{\Theta}^*, \boldsymbol{\vartheta}^*, \boldsymbol{\beta}^*\} = \arg\min_{\boldsymbol{\Theta}, \boldsymbol{\vartheta}, \boldsymbol{\beta}} \sum_{i=1}^{Q} \|F(\mathbf{r}_i; \boldsymbol{\Theta}, \boldsymbol{\vartheta}, \boldsymbol{\beta}) - \tilde{\mathbf{d}}_i\|_2^2. \quad (15)$$

When the whole network optimization is finished, the test can be performed end-to-end, *i.e.* given a test RGB image as input the network directly outputs an estimated depth map.

5　EXPERIMENTS

To demonstrate the effectiveness of the proposed multi-scale CRF models for monocular depth prediction, we performed experiments on three publicly available datasets: the NYU Depth V2 [43], the Make3D [39] and the KITTI [14] datasets. In the following we first describe the experimental setup and the implementation details, and then present the experimental results and analysis.

5.1　Experimental Setup

5.1.1　Datasets

The **NYU Depth V2** dataset [43] contains 120K unique pairs of RGB and depth images captured with a Microsoft Kinect. The datasets consists of 249 scenes for training and 215 scenes for testing. The images have a resolution of 640×480. To speed up the training phase, following previous works [30], [53] we consider only a small subset of images. This subset has 1449 aligned RGB-depth pairs: 795 pairs are used for training, 654 for testing. Following [12], we perform data augmentation for the training
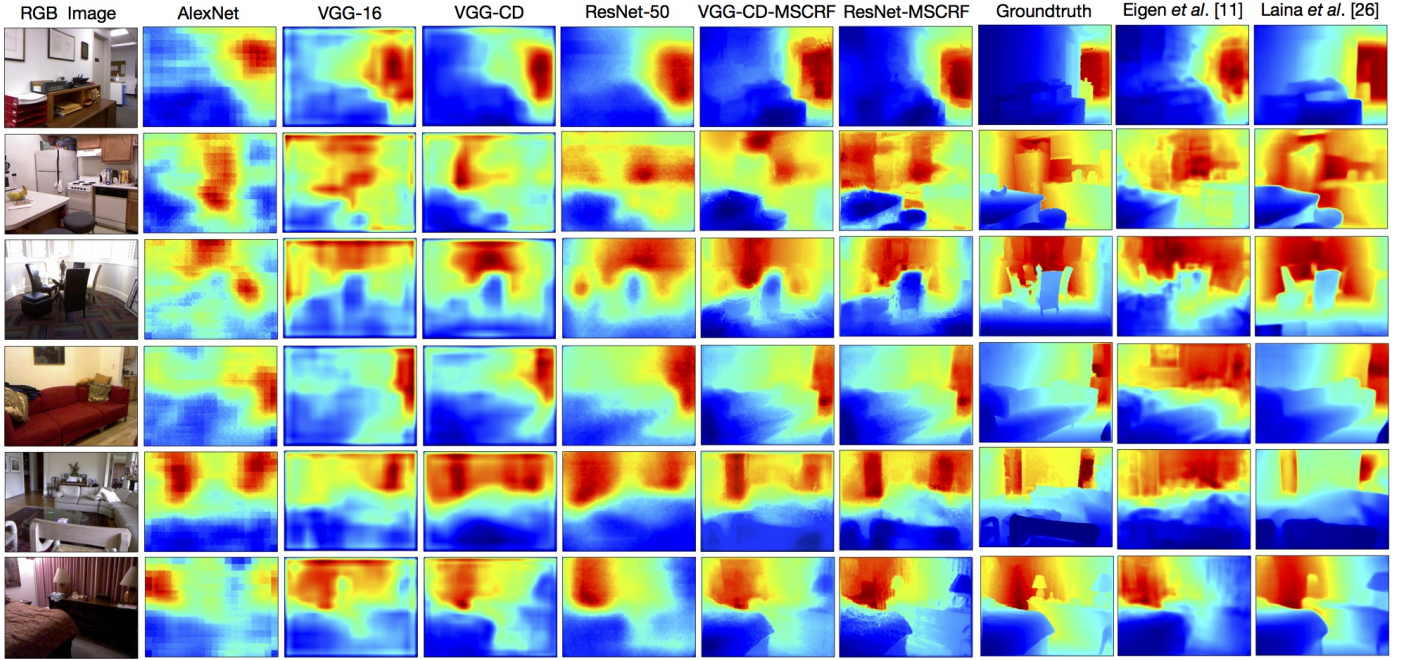
Fig. 6. Examples of qualitative depth prediction results of different methods on the NYU v2 test dataset. Different front-end deep network architectures are investigated. VGG-CD-MSCRF and ResNet-MSCRF represent our approach with the proposed multi-scale continuous CRF model plugged on VGG-CD and ResNet-50 network respectively.

samples. The RGB and depth images are scaled with a ratio $\rho \in \{1, 1.2, 1.5\}$ and the depths are divided by ρ. Additionally, we horizontally flip all the samples and randomly crop them to 320×240 pixels. The data augmentation phase produces 4770 training pairs in total.

The **Make3D** dataset [39] contains 534 RGB-depth pairs, split into 400 pairs for training and 134 for testing. We resize all the images to a resolution of 460×345 as done in [32] to preserve the aspect ratio of the original images. We adopted the same data augmentation scheme used for NYU Depth V2 dataset but, for $\rho = \{1.2, 1.5\}$ we randomly generate two samples each via cropping, obtaining 4K training samples.

The **KITTI** dataset [14] is built for various computer vision tasks within the context of autonomous driving, which contains depth videos captured through a LiDAR sensor deployed on a driving vehicle. For the training and testing split, we follow the protocol made by Eigen *et al.* [12] for a better comparison with existing works. Specifically, 61 scenes are selected from the raw data. Total 22,600 images from 32 scenes are used for training, and 697 images from the other 29 scenes are used for testing. Following [13], the ground-truth depth maps are generated by reprojecting the 3D points collected from velodyne laser into the left monocular camera. The resolution of RGB images are reduced half from original 1224×368 for training and testing.

5.1.2 Evaluation Metrics

Following previous works [11], [12], [45], we adopt the following evaluation metrics to quantitatively assess the performance of our depth prediction model. Specifically, we consider:

- mean relative error (rel): $\frac{1}{P} \sum_{i=1}^{P} \frac{|\tilde{d}_i - d_i^\star|}{d_i^\star}$;
- root mean squared error (rms): $\sqrt{\frac{1}{P} \sum_{i=1}^{P} (\tilde{d}_i - d_i^\star)^2}$;

- mean log10 error (log10):
  $\frac{1}{P} \sum_{i=1}^{P} \| \log_{10}(\tilde{d}_i) - \log_{10}(d_i^\star) \|$;
- scale invariant rms log error as used in [12], rms(sc-inv.);
- accuracy with threshold t: percentage (%) of $d_i^\star$, subject to $\max(\frac{d_i^\star}{\tilde{d}_i}, \frac{\tilde{d}_i}{d_i^\star}) = \delta < t$ ($t \in [1.25, 1.25^2, 1.25^3]$).

Where $\tilde{d}_i$ and $d_i^\star$ is the ground-truth depth and the estimated depth at pixel i respectively; P is the total number of pixels of the test images.

5.2 Implementation Details

We implemented the proposed deep model using the popular Caffe framework [15] on a single Nvidia Tesla K80 GPU with 12 GB memory. More details on the front-end CNN architectures, the generation of multi-scale side outputs and the parameter settings are elaborated as follows.

5.2.1 Front-end CNN Architectures

To study the influence of the frond-end CNN, we consider several network architectures including: (i) AlexNet [23], (ii) VGG-16 [44], (iii) a fully convolutional encoder-decoder network derived from VGG-16, referred as VGG-ED [2], (iv) a Convolution-Deconvolution network based on VGG-16, referred as VGG-CD [34], and (v) ResNet-50 [17]. For AlexNet, VGG-16 and ResNet-50, we obtain the side outputs from the last semantic convolutional layer of different convolutional blocks, in which each the layer produces feature maps with the same shape. The scheme utilized for the generation will be introduced in the next section. The number of side outputs considered in our experiments is 5, 5 and 4 for AlexNet, VGG-16 and ResNet-50, respectively. As VGG-ED and VGG-CD have been widely used for dense pixel-level prediction tasks, we also investigate them in the experimental analysis. Both VGG-ED and VGG-CD have a

TABLE 2
Quantitative performance comparison of different front-end deep network architectures and the proposed two multi-scale CRF models associated with the pretrained front-end networks on the NYU Depth V2 dataset.

Network Architecture	Error (lower is better)			Accuracy (higher is better)		
	rel	log10	rms	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
AlexNet (pretrain)	0.265	0.120	0.945	0.544	0.835	0.948
VGG-16 (pretrain)	0.228	0.104	0.836	0.596	0.863	0.954
VGG-ED (pretrain)	0.208	0.089	0.788	0.645	0.906	0.978
VGG-CD (pretrain)	0.203	0.087	0.774	0.652	0.909	0.979
ResNet-50 (pretrain)	0.168	0.072	0.701	0.741	0.932	0.981
AlexNet + cascade-CRFs	0.231	0.105	0.868	0.591	0.859	0.952
VGG-16 + cascade-CRFs	0.193	0.092	0.792	0.636	0.896	0.972
VGG-ED + cascade-CRFs	0.173	0.073	0.685	0.693	0.921	0.981
VGG-CD + cascade-CRFs	0.169	0.071	0.673	0.698	0.923	0.981
ResNet-50 + cascade-CRFs	**0.143**	**0.065**	**0.613**	**0.789**	**0.946**	**0.984**

symmetric network structure, and five side outputs are then generated from the different blocks of the decoder or the deconvolutional network part.

5.2.2 Generation of multi-scale CNN side-outputs

Our approach can be applied with any multi-scale front-end CNN models including those with skip-connections. We here briefly describe the scheme we adopt to build CNN side outputs from the front-end CNN for the multi-scale fusion with CRFs. In [46] a convolutional layer is first used to generate a score map from the feature map and then a deconvolutional (*deconv*) layer is adopted as a bilateral upsampling operator to enlarge the score map such as to obtain the same size of the input image. However, we noticed that by adopting the approach in [46] the generated side outputs associated to the feature maps with smaller size are very coarse, causing a lot scene details missing. To address this problem, after the convolutional layer, we stack several *deconv* layers, each of them enlarging the output map by two times. A Rectified Linear Unit (ReLU) is applied after each *deconv* layer. After the last deconv layer we use a crop layer to cut the extra margin and obtain a side output with the same resolution of the ground-truth image. We employ this scheme to obtain side outputs for AlexNet, VGG-16 and ResNet-50, while for VGG-CD and VGG-ED, we use the same setting as in [46], as their decoder or deconvolutional part is able to obtain more fine-grained side outputs. Table 1 shows detailed network parameters used to obtain the side output from the last convolutional block of ResNet-50 (*i.e.* from the layer *res5c*).

5.2.3 Parameters settings

As described in Section 4.4, training consists of a pretraining and a fine tuning phase. In the first phase, we train the front-end CNN with parameters initialized with the corresponding ImageNet pretrained models. For AlexNet, VGG-16, VGG-ED and VGG-CD, the batch size is set to 12 and for ResNet-50 to 8. The learning rate is initialized at 10^{-11} and decreases by 10 times around every 50 epochs. 80 epochs are performed for pretraining in total. The momentum and the weight decay are set to 0.9 and 0.0005, respectively. When the pretraining is finished, we connect all the side

outputs of the front-end CNN to our CRFs-based multi-scale deep models for end-to-end training of the whole network. In this phase, the batch size is reduced to 6 and a fixed learning rate of 10^{-12} is used. The same parameters of the pre-training phase are used for momentum and weight decay. The bandwidth weights for the Gaussian kernels are obtained through cross validation. The number of mean-field iterations is set to 5 for efficient training for both the cascade CRFs and multi-scale CRFs. We do not observe significant improvement using more than 5 iterations. Training the whole network takes around ~ 25 hours on the Make3D dataset, ~ 28 hours on the KITTI dataset and ~ 31 hours on the NYU v2 dataset.

5.3 Experimental Results

To present the experimental results, we start from an ablation study for investigating the performance impact of different front-end network architectures, the effectiveness of the proposed CRF-based multi-scale fusion models and the influence of the stacking orders for making the sequential neural network. Then we compare the overall performance with the state of the art methods, and finally the qualitative results and running time are analyzed.

5.3.1 Evaluation of different front-end CNN architectures

As discussed above, the proposed multi-scale CRF-based fusion models are general and different deep architectures can be used for the front-end network. In this section we evaluate the impact of this choice on the depth estimation performance. We consider both the case of the pretrained front-end models (*i.e.* only side losses are employed but the multi-scale CRF models are not plugged), indicated with 'pretrain', and the case of the fine-tuned models, including the front-end network with the multi-scale cascade CRFs (cascade-CRFs). The results of the experiments are shown in Table 2. As expected, in both cases deeper CNN architectures produced more accurate predictions, and ResNet-50 achieves the best performance among all the front-end networks. Moreover, VGG-CD is slightly better than VGG-ED, and both these models outperforms VGG-16, showing that the symmetric network structure is beneficial for the

TABLE 3
Quantitative baseline comparison with different multi-scale fusion schemes, and with the continuous CRF as a post-processing module on the NYU Depth V2 dataset. The number of scales is investigated for both multi-scale models with a bottom up message passing structure.

Method	Error (lower is better)			Accuracy (higher is better)		
	rel	log10	rms	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
HED [46]	0.185	0.077	0.723	0.678	0.918	0.980
Hypercolumn [16]	0.189	0.080	0.730	0.667	0.911	0.978
C-CRF	0.193	0.082	0.742	0.662	0.909	0.976
Ours (single-scale)	0.187	0.079	0.727	0.674	0.916	0.980
Ours - cascade (3-scale)	0.176	0.074	0.695	0.689	0.920	0.980
Ours - cascade (5-scale)	0.169	0.071	0.673	0.698	0.923	**0.981**
Ours - unified (3-scale)	0.172	0.072	0.683	0.691	0.922	**0.981**
Ours - unified (5-scale)	**0.163**	**0.069**	**0.655**	**0.706**	**0.925**	**0.981**

TABLE 4
Quantitative performance evaluation of different message passing structures for the cascade CRF model via building the sequential deep network with the proposed C-MF block on the NYU Depth V2 dataset.

Method	Error (lower is better)			Accuracy (higher is better)		
	rel	log10	rms	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Top down structure	0.175	0.072	0.688	0.689	0.919	0.979
Bottom up structure	0.169	0.071	0.673	0.698	0.923	0.981
Skip connection structure	0.161	0.070	0.664	0.709	0.923	0.981
All to one structure	**0.154**	**0.068**	**0.648**	**0.725**	**0.927**	**0.981**

TABLE 5
Overall performance comparison with state of the art methods on the NYU Depth V2 dataset. Our approach achieves the best on most of the metrics, while the runners-up Eigen and Fergus [11] and Laina *et al.* [27] employ more training data than ours. ResNet-50-unified means using ResNet-50 front-end network with the proposed multi-scale unified CRF model.

Method	Error (lower is better)				Accuracy (higher is better)		
	rel	log10	rms	rms (sc-inv.)	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Karsch *et al.* [41]	0.349	-	1.214	0.325	0.447	0.745	0.897
Ladicky *et al.* [20]	0.35	0.131	1.20	-	-	-	-
Liu *et al.* [32]	0.335	0.127	1.06		-	-	-
Ladicky *et al.* [25]	-	-	-	-	0.542	0.829	0.941
Zhuo *et al.* [53]	0.305	0.122	1.04	-	0.525	0.838	0.962
Liu *et al.* [30]	0.230	0.095	0.824	-	0.614	0.883	0.975
Wang *et al.* [45]	0.220	0.094	0.745	-	0.605	0.890	0.970
Eigen *et al.* [12]	0.215	-	0.907	0.219	0.611	0.887	0.971
Roi and Todorovic [38]	0.187	0.078	0.744	-	-	-	-
Eigen and Fergus [11]	0.158	-	0.641	0.171	0.769	0.950	0.988
Laina *et al.* [27]	0.129	0.056	0.583	-	0.801	0.950	0.986
Ours (ResNet-50-unified-4.7K-bottom up)	0.139	0.063	0.609	0.163	0.793	0.948	0.984
Ours (ResNet-50-unified-95K-bottom up)	0.121	0.052	0.586	0.149	0.811	0.954	0.987
Ours (ResNet-50-unified-95K-all to one)	**0.108**	**0.045**	**0.579**	**0.142**	**0.823**	**0.957**	**0.987**

TABLE 6
Overall performance comparison with state of the art methods on the Make3D dataset. Our approach outperforms all the competitors w.r.t. the C2 Error, and performs only slightly worse on the *rel* metric of the C1 Error than Laina *et al.* [27] using Huber loss and significantly larger training data.

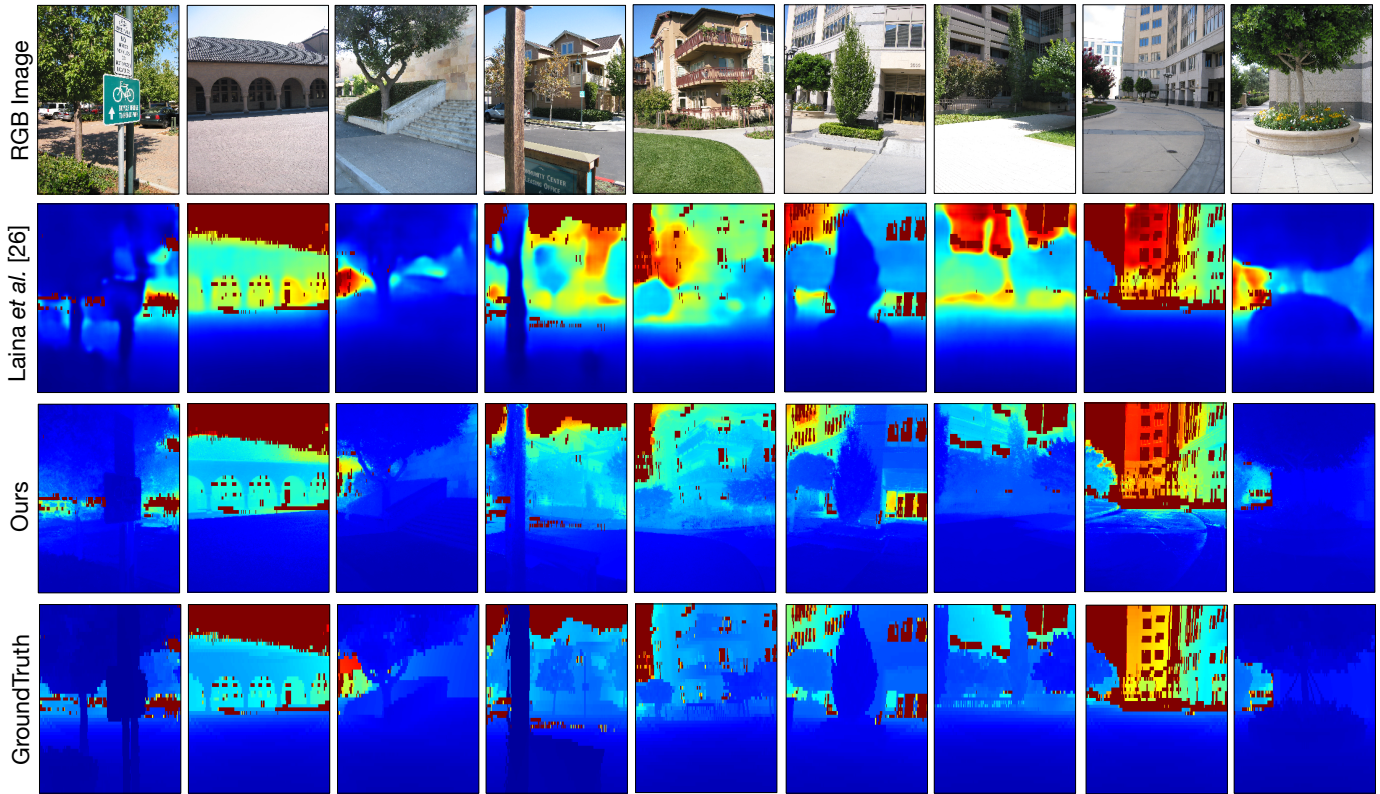| Method	C1 Error				C2 Error		
	rel	log10	rms	rms (sc-inv.)	rel	log10	rms
Karsch *et al.* [20]	0.355	0.127	9.20	-	0.361	0.148	15.10
Liu et al. [32]	0.335	0.137	9.49	-	0.338	0.134	12.60
Liu et al. [30]	0.314	0.119	8.60	-	0.307	0.125	12.89
Li et al. [28]	0.278	0.092	7.19	-	0.279	0.102	10.27
Laina *et al.* [27] (ℓ_2 loss)	0.223	0.089	4.89	-	-	-	-
Laina *et al.* [27] (Huber loss)	0.176	0.072	4.46	-	-	-	-
Ours (ResNet-50-cascade-bottom up)	0.213	0.082	4.67	0.245	0.221	4.79	8.81
Ours (ResNet-50-unified-bottom up)	0.206	0.076	4.51	0.237	0.212	4.71	8.73
Ours (ResNet-50-unified-10K-bottom up)	0.184	0.065	4.38	0.219	0.198	4.53	8.56
Ours (ResNet-50-unified-10K-all to one)	**0.174**	**0.059**	**4.27**	**0.211**	**0.185**	**4.41**	**8.43**

Fig. 7. Examples of depth prediction results on the Make3D dataset. The four rows from up to bottom are the input test RGB images, the results produced from Laina *et al.* [27], the results of our ResNet50-MSCRF model and the groundtruth depth maps, respectively.

dense pixel-level prediction problems. Importantly, for all considered front-end networks there is a significant increase in performance when applying the proposed CRF-based models.

Figure 6 depicts some examples of predicted depth maps using different front-end networks on the NYU Depth V2 test dataset. As we can see from the figure, the qualitative results confirm that the deeper architecture leads to better depth recovery. By comparing the reconstructed depth maps obtained with pretrained models (*e.g.* using only the front-end networks VGG-CD and ResNet-50) with those generated with our multi-scale models, it is clear that our approach remarkably improves prediction accuracy and visual quality.

5.3.2 Evaluation of different multi-scale CRF fusion models

To evaluate the effectiveness of the proposed CRF-based multi-scale fusion models, we conduct experiments on the NYU Depth V2 dataset and consider the following baselines:

(i) the 'HED' method in [46], where multiple side outputs are fused with a weighted averaging scheme and the sum of multiple side output losses is jointly minimized as deep supervision with a cross-entropy loss, while we use the square loss as our problem involves continuous variables;

(ii) the 'Hypercolumn' method [16], where multi-scale feature maps generated from different semantic network layers are concatenated and fused;

(iii) a continuous CRF ('C-CRF') applied on the prediction of the front-end network, *i.e.* plugging after the last output layer as a post-processing module without end-to-end training.

For the first two baselines, we want to compare our models with other popular methods for fusing multi-scale CNN information, while the third one aims at demonstrating the effectiveness of the continous CRF itself. In these experiments we consider VGG-CD as the front-end CNN architecture. The results of the comparison are shown in Table 3. It is evident that with our CRF-based fusion models (both the cascade CRFs and the unified CRFs) more accurate depth maps can be obtained, demonstrating that our idea of integrating complementary information derived from CNN side output maps within a graphical model framework is more effective than traditional fusion schemes. Table 3 also compares the proposed cascade and unified models. As expected, the unified model produces more accurate depth maps, at the price of an increased computational cost. This can also be observed from Table 2. The C-CRF (in Table 3) improves the depth estimation at all metrics over the VGG-CD (pretrain) (in Table 2) with a clear gap, showing the CRF model is very useful for refining the deeply predicted map. By jointly learning with the front-end (*i.e.* end-to-end training), ours (single-scale) further boosts the performance. Finally, we analyze the impact of adopting multiple scales and compare our complete models (5 scales) with their version when only a single and three side output layers are used. It is evident that the performance can be improved by increasing the number of scales.

5.3.3 Evaluation of multi-scale message passing structures

We evaluate the influence of different multi-scale message passing structures using the cascade CRF model. Four connection structures as depicted in Fig. 3 are compared. Table 4

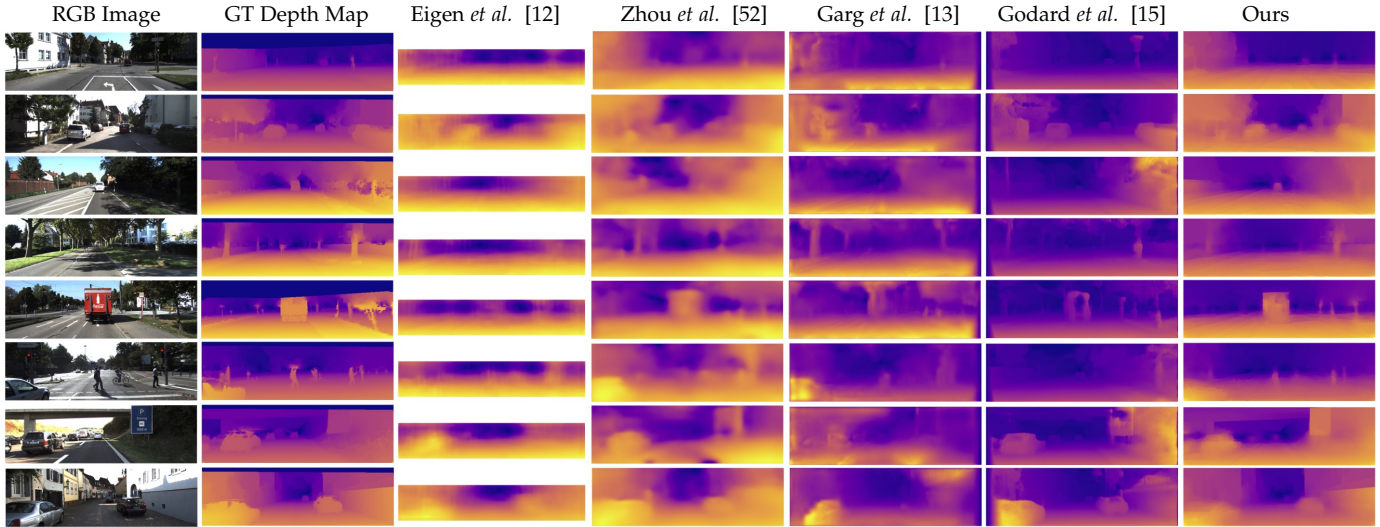| RGB Image | GT Depth Map | Eigen *et al.* [12] | Zhou *et al.* [52] | Garg *et al.* [13] | Godard *et al.* [15] | Ours |

Fig. 8. Examples of depth prediction results on the KITTI raw dataset. Qualitative comparison with other depth estimation methods on this dataset is presented. The sparse ground-truth depth maps are interpolated for better visualization.

TABLE 7

Overall performance comparison with state of the art methods on the KITTI raw dataset. Our approach obtains very competitive performance over all the competitors w.r.t. all the evaluation metrics on the testing set given by Eigen *et al.* [12]. For the setting, caps means different gt/predicted depth range and stereo means using left and right images captured from two monocular cameras in the training phase. Ours uses a unified model considering both the bottom up and the all to one network structure.

Method	Setting		Error (lower is better)				Accuracy (higher is better)		
	range	stereo	rel	sq rel	rms	rms (sc-inv.)	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Saxena *et al.* [41]	0-80m	No	0.280	-	8.734	0.327	0.601	0.820	0.926
Eigen *et al.* [12]	0-80m	No	0.190	-	7.156	0.246	0.692	0.899	0.967
Liu *et al.* [30]	0-80m	No	0.217	0.092	7.046	-	0.656	0.881	0.958
Zhou *et al.* [52]	0-80m	No	0.208	1.768	6.858	-	0.678	0.885	0.957
Kuznietsov *et al.* [24] (only supervised)	0-80m	No	-	-	4.815	-	0.845	0.957	0.987
Garg *et al.* [13]	0-80m	Yes	0.177	1.169	5.285	-	0.727	0.896	0.962
Garg *et al.* [13] L12 + Aug 8x	1-50m	Yes	0.169	1.080	5.104	-	0.740	0.904	0.958
Godard *et al.* [15]	0-80m	Yes	0.148	1.344	5.927	-	0.803	0.922	0.964
Kuznietsov *et al.* [24]	0-80m	Yes	-	-	**4.621**	-	**0.852**	**0.960**	**0.986**
Ours (ResNet-50 Pretrain)	0-80m	No	0.152	0.973	4.902	0.176	0.782	0.931	0.975
Ours (ResNet-50 Fine-tune-bottom up)	0-80m	No	0.132	0.911	4.791	0.162	0.804	0.945	0.981
Ours (ResNet-50 Fine-tune-all to one)	0-80m	No	**0.125**	**0.899**	4.685	**0.154**	0.816	0.951	0.983

shows the monocular depth estimation results on NYUD-v2 dataset. The comparison results confirm that the message passing structure indeed has an impact on the final performance. The bottom up and top down structures have similar performance, while the skip-connection structure slightly outperform these two. The all to one structure performs the best, producing around 2.0% gain in terms of the *rel* metric than the top down structure, which means that directly passing message to the finest prediction scale from the rest scales can absorb more complementary information than the gradual passing fashions used in the first three structures.

5.3.4 Comparison with state of the art

We also compare our approach with state of the art methods on all the datasets. For previous works we directly report results taken from the original papers. Table 5 shows the results of the comparison on the NYU Depth V2 dataset. For our approach we consider the cascade model and use two different training sets for pretraining: the small set of 4.7K pairs employed in all our experiments and a larger set of 95K images as in [27]. Note that for fine tuning we only use the small set. As shown in the table, our approach outperforms all competing methods and it is the second best model when we use only 4.7K images. This is remarkable

considering that, for instance, in [11] 120K image pairs are used for training. Our model achieves the best results on all the metrics via using 95K pretraining samples and using the proposed all to one message passing structure.

We also perform a comparison with several state of the art methods on the Make3D dataset (Table 6). Following [32], the error metrics are computed in two different settings, *i.e.* considering (C1) only the regions with ground-truth depth less than 70 and (C2) the entire image. It is clear that the proposed approach is significantly better than previous methods. In particular, comparing with Laina *et al.* [27], the best performing method in the literature, it is evident that our approach, both in case of the cascade and the multi-scale models, outperforms [27] by a significant margin when Laina *et al.* also adopt a square loss. It is worth noting that in [27] a training set of 15K image pairs is considered, while we employ much less training samples. By increasing our training data (*i.e.* ∼ 10K in the pretraining phase), our multi-scale CRF model also outperforms [27] with Huber loss (log10 and rms metrics). The final performance is further boosted by considering the all to one structure similar to NYUD v2 dataset. Finally, it is very interesting to compare the proposed method with the approach in Liu *et al.* [30], since
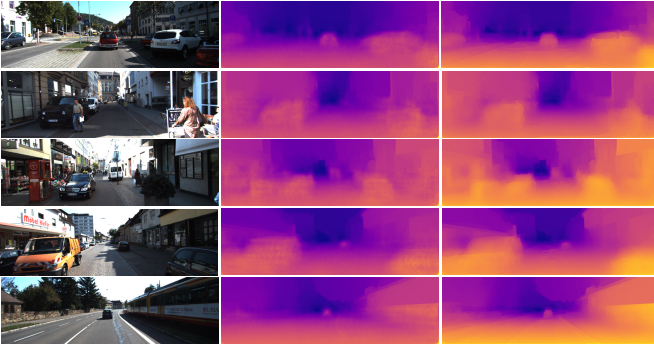
Fig. 9. Examples of depth prediction results on the KITTI raw dataset. The middle column and the right column show the pretrained and the fine-tuned estimation results respectively.

in [30] a CRF model is also employed within a deep network trained end-to-end. Our method significantly outperforms [30] in terms of accuracy. Moreover, in [30] a time of 1.1sec is reported for performing inference on a test image but the time required by superpixels calculations is not taken into account. Oppositely, with our method computing the depth map for a single image takes about 1 sec in total.

The state of the art comparison on KITTI dataset is shown in Table 7. The competitors include Saxena *et al.* [39], Eigen *et al.* [12], Liu *et al.* [31], Zhou *et al.* [52], Garg *et al.* [13], Godard *et al.* [15] and Kuznietsov *et al.* [24]. As the same setting of ours, the first four methods use single monocular images in the training phase, while the last two considered two monocular images with a stereo setting for training. Among the first four competitors, Eigen *et al.* [12] significantly outperforms the others in terms of the metric of the mean relative error (*rel*), due to the usage of large-scale training data (more than 1 million samples). While our model achieves much better performance than Eigen *et al.* [12] in all metrics with much less data (22.6K samples). Although the training of the last two methods (requiring two monocular images) is not equal to our setting, the proposed approach with both the bottom-up and the all to one structures still produces better results than them with clear performance gap in all metrics. Kuznietsov *et al.* [24] reports results for both the stereo training and the monocular supervised training. It is not directly comparable with the stereo training setting, which is significantly different as it requires both left and right images from a binocular camera. Ours focuses on monocular depth estimation and achieves lower error performance comparing with theirs using the same monocular setting. Fig. 8 also shows some qualitative comparison results with these methods, further demonstrating the advantageous performance of our approach.

5.3.5   Qualitative depth estimation results

Fig. 6, 7 and 9 show some examples of the qualitative depth estimation results and the comparison with the competing methods on the NYUD-V2, Make3D and KITTI dataset respectively. It is clear that the proposed approach is able to produce sharper depth estimation with better visual quality compared with the classic CNN structures, which demonstrates the importance of the prediction aided by the CRFs with appearance and smoothness constraints. Fig. 9

also shows a qualitative comparison between the pretrained front-end CNN and the fine-tuned whole model. It can be observed that our approach can recover more scene structures and details. We believe that this is probably because the effective structured fusion of the coarse-to-fine multi-scale predictions of the deep network with the proposed CRF models. For the influence of the variance in the CRF model on the prediction errors, as the variance term is actually acted as a normalization factor after the message passing. It may have influence but the main influence is dominated by the predictions of deep front-end CNN based on our observation from the experimental results.

5.3.6   Empirical run-time analysis

Computational run-time complexity is an important aspect for deep structured prediction models. In this paragraph we provide a short discussion about the computational cost of the proposed CRFs-based models. As shown in the paper, the multi-scale CRF model achieves better accuracy and lower error than the cascade model for both the NYU Depth V2 and the Make3D experiments. However, as expected, the cascade model is more advantageous in terms of the running time. For instance, considering ResNet-50 as the front-end CNN, the time required at test phase for one image is 1.02 seconds w.r.t. the cascade model and 1.45 seconds w.r.t. the multi-scale model, and the image resolution is 320×240 pixels. Higher resolution of the network input usually brings more computational overhead. We also test the running time given the input resolution of 640×480 and it costs around 2.25 seconds for processing one image. We believe that if we reduce the receptive field of the CRF model from fully connected to partially connected, the computing time could be significantly reduced.

6   CONCLUSION

In this paper, we introduced a novel approach for predicting depth maps from a single RGB image. The core of the method is a novel framework based on continuous CRFs for fusing multi-scale score-level side-outputs derived from different semantic CNN layers. We demonstrated that this framework can be used in combination with several common CNN architectures and can be implemented for end-to-end training. The extensive experiments confirmed the validity of the proposed multi-scale fusion approach. While this paper specifically addresses the problem of depth prediction, we believe that other tasks in computer vision involving pixel-level predictions of continuous variables, can also benefit from our implementation of the mean-field updating within the CNN framework.

Currently, the multi-scale fusion is performed on the score level. Further research direction will investigate the integration of both the feature- and the score-level multi-scale information within a unified graphical model. Moreover, the study of strategies for further improving the training and testing efficiency of the CNN-CRF models will also be an interesting aspect in the future work. The monocular depth estimation is particularly useful for various cross-modal recognition and detection tasks. A straightforward follow-up of this work would be designing a joint multi-task deep model to transfer the learned depth model for

aiding other similar dense prediction problems [49] such as contour detection and semantic segmentation.

REFERENCES

[1] A. Adams, J. Baek, and M. A. Davis. Fast high-dimensional filtering using the permutohedral lattice. In *Computer Graphics Forum*, volume 29, pages 753–762, 2010.

[2] V. Badrinarayanan, A. Handa, and R. Cipolla. Segnet: A deep convolutional encoder-decoder architecture for robust semantic pixel-wise labelling. *arXiv preprint arXiv:1505.07293*, 2015.

[3] G. Bertasius, J. Shi, and L. Torresani. Deepedge: A multi-scale bifurcated deep network for top-down contour detection. In *CVPR*, 2015.

[4] P. Buyssens, A. Elmoataz, and O. Lézoray. Multiscale convolutional neural networks for vision–based classification of cells. In *ACCV*, 2012.

[5] Z. Cai, Q. Fan, R. S. Feris, and N. Vasconcelos. A unified multi-scale deep convolutional neural network for fast object detection. In *ECCV*, 2016.

[6] D. Chen, D. Xu, H. Li, N. Sebe, and X. Wang. Group consistent similarity learning via deep crfs for person re-identification. In *CVPR*, 2018.

[7] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. *ICLR*, 2015.

[8] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *arXiv preprint arXiv:1606.00915*, 2016.

[9] L.-C. Chen, Y. Yang, J. Wang, W. Xu, and A. L. Yuille. Attention to scale: Scale-aware semantic image segmentation. *CVPR*, 2016.

[10] E. Delage, H. Lee, and A. Y. Ng. A dynamic bayesian network model for autonomous 3d reconstruction from a single indoor image. In *CVPR*, 2006.

[11] D. Eigen and R. Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *ICCV*, 2015.

[12] D. Eigen, C. Puhrsch, and R. Fergus. Depth map prediction from a single image using a multi-scale deep network. In *NIPS*, 2014.

[13] R. Garg, G. Carneiro, and I. Reid. Unsupervised cnn for single view depth estimation: Geometry to the rescue. In *ECCV*, 2016.

[14] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. Vision meets robotics: The kitti dataset. *IJRR*, 2013.

[15] C. Godard, O. Mac Aodha, and G. J. Brostow. Unsupervised monocular depth estimation with left-right consistency. In *CVPR*, 2017.

[16] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik. Hypercolumns for object segmentation and fine-grained localization. In *CVPR*, 2015.

[17] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015.

[18] D. Hoiem, A. A. Efros, and M. Hebert. Automatic photo pop-up. *ACM TOG*, 24(3):577–584, 2005.

[19] D. Hoiem, A. A. Efros, and M. Hebert. Geometric context from a single image. In *ICCV*, 2005.

[20] K. Karsch, C. Liu, and S. B. Kang. Depth transfer: Depth extraction from video using non-parametric sampling. *IEEE TPAMI*, 36(11):2144–2158, 2014.

[21] P. Knöbelreiter, C. Reinbacher, A. Shekhovtsov, and T. Pock. End-to-end training of hybrid cnn-crf models for stereo. *arXiv preprint arXiv:1611.10229*, 2016.

[22] V. Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. *NIPS*, 2011.

[23] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012.

[24] Y. Kuznietsov, J. Stückler, and B. Leibe. Semi-supervised deep learning for monocular depth map prediction. In *CVPR*, 2017.

[25] L. Ladicky, J. Shi, and M. Pollefeys. Pulling things out of perspective. In *CVPR*, 2014.

[26] J. Lafferty, A. McCallum, F. Pereira, et al. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML*, 2001.

[27] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab. Deeper depth prediction with fully convolutional residual networks. *arXiv preprint arXiv:1606.00373*, 2016.

[28] B. Li, C. Shen, Y. Dai, A. van den Hengel, and M. He. Depth and surface normal estimation from monocular images using regression on deep features and hierarchical crfs. In *CVPR*, 2015.

[29] B. Liu, S. Gould, and D. Koller. Single image depth estimation from predicted semantic labels. In *CVPR*, 2010.

[30] F. Liu, C. Shen, and G. Lin. Deep convolutional neural fields for depth estimation from a single image. In *CVPR*, 2015.

[31] F. Liu, C. Shen, G. Lin, and I. Reid. Learning depth from single monocular images using deep convolutional neural fields. *IEEE TPAMI*, 38(10):2024–2039, 2016.

[32] M. Liu, M. Salzmann, and X. He. Discrete-continuous depth estimation from a single image. In *CVPR*, 2014.

[33] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015.

[34] H. Noh, S. Hong, and B. Han. Learning deconvolution network for semantic segmentation. In *ICCV*, 2015.

[35] G. Papandreou, L.-C. Chen, K. Murphy, and A. L. Yuille. Weakly- and semi-supervised learning of a dcnn for semantic image segmentation. *arXiv preprint arXiv:1502.02734*, 2015.

[36] L. Porzi, S. R. Buló, A. Penate-Sanchez, E. Ricci, and F. Moreno-Noguer. Learning depth-aware deep representations for robotic perception. *IEEE Robotics and Automation Letters*, 2(2):468–475, 2017.

[37] K. Ristovski, V. Radosavljevic, S. Vucetic, and Z. Obradovic. Continuous conditional random fields for efficient regression in large fully connected graphs. In *AAAI*, 2013.

[38] A. Roy and S. Todorovic. Monocular depth estimation using neural regression forest. In *CVPR*, 2016.

[39] A. Saxena, S. H. Chung, and A. Y. Ng. Learning depth from single monocular images. In *NIPS*, 2005.

[40] A. Saxena, S. H. Chung, and A. Y. Ng. 3-d depth reconstruction from a single still image. *IJCV*, 76(1):53–69, 2008.

[41] A. Saxena, M. Sun, and A. Y. Ng. Make3d: Learning 3d scene structure from a single still image. *IEEE TPAMI*, 31(5):824–840, 2009.

[42] A. G. Schwing and R. Urtasun. Fully connected deep structured networks. *arXiv preprint arXiv:1503.02351*, 2015.

[43] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012.

[44] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[45] P. Wang, X. Shen, Z. Lin, S. Cohen, B. Price, and A. Yuille. Towards unified depth and semantic prediction from a single image. In *CVPR*, 2015.

[46] S. Xie and Z. Tu. Holistically-nested edge detection. In *ICCV*, 2015.

[47] D. Xu, W. Ouyang, X. Alameda-Pineda, E. Ricci, X. Wang, and N. Sebe. Learning deep structured multi-scale features using attention-gated crfs for contour prediction. In *NIPS*, 2017.

[48] D. Xu, W. Ouyang, E. Ricci, X. Wang, and N. Sebe. Learning cross-modal deep representations for robust pedestrian detection. In *CVPR*, 2017.

[49] D. Xu, W. Ouyang, X. Wang, and N. Sebe. Pad-net: Multi-tasks guided prediciton-and-distillation network for simultaneous depth estimation and scene parsing. In *CVPR*, 2018.

[50] D. Xu, E. Ricci, W. Ouyang, X. Wang, and N. Sebe. Multi-scale continuous crfs as sequential deep networks for monocular depth estimation. In *CVPR*, 2017.

[51] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. H. Torr. Conditional random fields as recurrent neural networks. In *ICCV*, 2015.

[52] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe. Unsupervised learning of depth and ego-motion from video. In *CVPR*, 2017.

[53] W. Zhuo, M. Salzmann, X. He, and M. Liu. Indoor scene structure analysis for single image depth estimation. In *CVPR*, 2015.

PLACE PHOTO HERE

Dan Xu is a Ph.D. candidate in the Department of Information Engineering and Computer Science, and a member of Multimedia and Human Understanding Group (MHUG) led by Prof. Nicu Sebe at the University of Trento. He was a research assistant in the Multimedia Laboratory in the Department of Electronic Engineering at the Chinese University of Hong Kong. His research focuses on computer vision, multimedia and machine learning. Specifically, he is interested in deep learning, structured prediction and cross-modal representation learning and the applications to scene understanding tasks. He received the Intel best scientific paper award at ICPR 2016.

Elisa Ricci received the PhD degree from the University of Perugia in 2008. She is an assistant professor at the University of Perugia and a researcher at Fondazione Bruno Kessler. She has since been a post-doctoral researcher at Idiap, Martigny, and Fondazione Bruno Kessler, Trento. She was also a visiting researcher at the University of Bristol. Her research interests are mainly in the areas of computer vision and machine learning. She is a member of the IEEE.

Wanli Ouyang received the PhD degree in the Department of Electronic Engineering, The Chinese University of Hong Kong. He is now a senior lecturer in the School of Electrical and Information Engineering at the University of Sydney, Australia. His research interests include image processing, computer vision and pattern recognition. He is a senior member of IEEE.

Xiaogang Wang received the PhD degree in Computer Science from Massachusetts Institute of Technology. He is an associate professor in the Department of Electronic Engineering at the Chinese University of Hong Kong since August 2009. He was the Area Chairs of ICCV 2011 and 2015, ECCV 2014 and 2016, ACCV 2014 and 2016. He received the Outstanding Young Researcher in Automatic Human Behaviour Analysis Award in 2011, Hong Kong RGC Early Career Award in 2012, and CUHK Young Researcher Award 2012.

Nicu Sebe is Professor with the University of Trento, Italy, leading the research in the areas of multimedia information retrieval and human behavior understanding. He was the General Co- Chair of the IEEE FG Conference 2008 and ACM Multimedia 2013, and the Program Chair of the International Conference on Image and Video Retrieval in 2007 and 2010, ACM Multimedia 2007 and 2011. He was the Program Chair of ICCV 2017 and ECCV 2016, and a General Chair of ACM ICMR 2017. He is a fellow of the International Association for Pattern Recognition.