Noname manuscript No.
(will be inserted by the editor)

Deblur and Deep Depth from Single Defocus Image

Saeed Anwar [1,2] · **Zeeshan Hayder** [1] · **Fatih Porikli** [1]

Received: date / Accepted: date

Abstract In this paper, we tackle depth estimation and blur removal from a single out-of-focus image. Previously, depth is being estimated, and blurred is removed using multiple images; for example, from multiview or stereo scenes, but doing so with a single image is challenging. Earlier works of monocular images for depth estimated and deblurring either exploited geometric characteristics or priors using hand-crafted features. Lately, there is enough evidence that deep convolutional neural networks (CNN) significantly improved numerous vision applications; hence, in this article, we present a depth estimation method that leverages rich representations learned from cascaded convolutional and fully connected neural networks operating on a patch-pooled set of feature maps. Furthermore, from this depth, we computationally reconstruct an all-focus image *i.e.* removing the blur and achieve synthetic re-focusing, all from a single image.

Our method is fast, and it substantially improves depth accuracy over the state-of-the-art alternatives. Our proposed depth estimation approach can be utilized for everyday scenes without any geometric priors or extra information. Furthermore, our experiments on two benchmark datasets consist images of indoor and outdoor scenes *i.e.* Make3D and NYU-v2 demonstrate superior performance in comparison to other available depth estimation state-of-the-art methods by reducing the root-mean-squared error by **57%** and **46%**, and state-of-the-art blur removal methods by **0.36 dB** and **0.72 dB** in PSNR, respectively. This improvement in-depth estimation and deblurring is further demonstrated by the superior performance using real defocus images against images captured with a prototype lens.

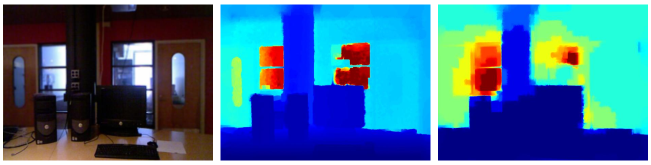
Keywords Depth Estimation · Depth Map · Blur Removal · Deblurring · Deconvolution · Convolutional Neural Network (CNN) · Defocus · Out-of-focus

1 Introduction

Recovering depth from a single image has many applications in computer vision and image processing. Many algorithms in computer vision or image processing take advantage of depth information, for example, pose estimation [1] and semantic labeling [2] *etc.* Recently, Microsoft introduced an affordable RGBD camera called Kinect for capturing the indoor depth along with RGB images; however, the computer vision community still uses available RGB datasets for evaluation for the different applications. Similarly, for outdoor applications, mostly LiDAR is used, but because of infrared interference, the acquire depth is noisy. This has commenced extensive research attention on the problem of estimating depths from single RGB images. Furthermore, it is an ill-posed problem, as an acquired image scene may signify many real-world scenarios.

Similarly, removing blur from an image has attained much focus due to easy access to hand-held camera equipment. Deblurring has applications in object recognition [3], classification [4], and image segmentation [5] *etc.* Therefore, recovering original image from its defocus version has attracted much attention, and recently numerous techniques have been put forward. Due to the inherent information loss, this reconstruction task requires strong prior knowledge or multiple observations

Saeed Anwar
E-mail: saeed.anwar@csiro.au
1 College of Engineering and Computer Science (CECS), The Australian National University (ANU).
2 Data61, The Commonwealth Scientific and Industrial Research Organisation (CSIRO), Australia.



Input image Original depth Our depth

Fig. 1 Example of depth prediction using our proposed model on NYU-v2 [15].

to produce effective results. For example, deconvolution with natural image priors [6–10], hybrid cameras [11–13] and blurred/noisy image pairs [14] are among the notable solutions.

In this article, our focus is on removing blur and estimating depth from a single image. Currently, the methods that rely on depth-from-defocus using coded-aperture have augmented the capacity to retrieve depth from a single image computationally. Subsequently, the recovered depth can be employed to generate a sharp image that further provides the possibilities to create synthetic views, change image focus, and enlarge the field's depth. The fundamental concept is to embed a coded pattern into a regular camera's aperture to achieve blur due to defocus. This causes the image to have different distinctive spatial structure.

The currently available depth from defocus methods relies on statistical models, encoding the sharp image structure as both the depth and radiance are not known, hence making the problem is ill-posed. Due to significant variance in the visual world, the depth recovery is very much ineffectual. User intervention is required to predict accurate and reliable depth maps though coded-aperture techniques have improved the depth discrimination. The coded-aperture techniques come with its inherent problem, such as the quality of the deblurred image is diminished due to less light transmitted. Similarly, the blur induced by coded-aperture is difficult to invert. Such problems have motivated researchers to look into more unconventional methods to recover the depth from defocus and deblur the out-of-focus image using the retrieved depth.

This exposition is an extension of our preliminary work [16] and introduces an alternative to the conventional aperture design and depth estimation methods for a single image. Reclaiming depth from the 2D images is analogous to estimating the third physical dimension lost during the imaging process. For this purpose, several existing approaches incorporate additional information to regularize this inherently ill-posed inverse problem. On the contrary, we propose a novel convolutional neural network-based depth estimation method without injecting any additional information. Only synthetically generated out-of-focus images are used as input to the network. This model learns the evidence using a quadratic energy minimization function using the difference between the synthetic images and the ground truth. Furthermore, the learned evidence *i.e.* the depth information is then incorporated in the defocusing problem. This encourages to have a kernel for each small neighborhood based on the intensity of the depth map.

The organization of our paper is as follows. We review related depth estimation and deblurring work in the next section. In section 3, we present our network architecture for depth estimation followed by our approach to incorporate the depth for deblurring in section 4. We evaluate and compare against the state-of-the-art depth estimation and deblurring techniques in section 5. We finally conclude our paper in section 6

2 Related Work

Here, we briefly discuss common techniques for depth estimation and deblurring algorithms with the capacity for generating a clean image when the input image is defocused.

2.1 Depth estimation

The techniques employed for extracting depth from an RGB image is enormous, and reporting it in this exposition is challenging; however, we provide a comprehensive review of the most common state-of-the-art competitive methods in this section of the paper.

2.1.1 Single-Shot coded aperture

Levin *et al.* [17] proposed the first technique to recover depth from a single image. An aperture mask was designed based on a prior derived from the probability distribution of the gradients of natural gray-scale images. Veeraraghavan *et al.* [18] proposed a coded aperture technique optimizing the aperture patterns based on the shape of power spectra. Similarly, Noguer *et al.* [19] projected a dotted pattern over the scene while the Zhou *et al.* [20] method placed an optical diffuser in front of the lens to obtain defocus images. These single-shot coded aperture approaches do not explicitly take into account of image structure and noise [21]. Some may require manual intervention to generate reliable depth maps [17]. Most importantly, spectral distortion introduced by the aperture mask hinders the ability to remove blur since spatial frequencies are systematically attenuated in the captured image [22,23].

2.1.2 Depth from focus

As an alternative to single-shot coded aperture techniques, some methods apply a focus measure for individual pixels across multiple images taken at different focal lengths [24]. The depth map is computed by assigning each pixel the position in the focal stack for which the focus measure of that pixel is maximal. This means the depth resolution is directly proportional to the number of images available. To augment the resolution, filters are applied to focus measure [25,26], or smooth surfaces are fitted to the estimated depths [27]. Depth-from-defocus with a single image is also targeted by numerous methods [28–34] that used focus measures and filters.

2.1.3 Depth from multiple images

Many algorithms utilized multiple images [35–37] for recovering the depth of a scene. In [35], the authors used texture invariant rational operators to predict a precise dense depth map. However, accurate customization of those filters is an open question. Xu *et al.* [37] uses two blur observations of the same scene to estimate the depth and remove the blur. Li *et al.* [38] measured shading in a scene to refine depth from defocus, iteratively. Recently, Shahid *et al.* [39] employed the content-adaptive blurring (CAB), a multi-focus image focus algorithm to detect non-blurry regions in the image. The CAB algorithm induces blur depending on the image's content and analyzes the neighborhood of the blur for the image quality, whether blur should be applied or not keeping the quality. Hence, non-uniform blur is induced in the focused regions, while the already blurry areas receive limited or no blur.

2.1.4 Convolutional neural networks (CNN) methods

Success of deep convolutional neural networks in image classification [40,41], segmentation [5], object detection [42,43] and recognition [44], inspired single image depth estimation [34]. Recent works of Su *et al.* [45], Kar *et al.* [46], Eigen *et al.* [47] and Fayao *et al.* [34] are relevant to our method. [45] and [46] use a single sharp image to estimate depth map. However, both of these works focus on 3D reconstruction of already known segmented objects. More recently, [34] and [47] proposed CNN based approaches for depth estimation. Our algorithm differs from both of these works; [34] learns the unary and pairwise potentials from sharp images while [47] use CNN as a black box to estimate depth map using sharp images.

More recently, another multi-focused image fusion method known as Deep Regression Pair Learning (DRPL) [48] is proposed, which takes the whole image instead of patches employing two source images generating two binary masks. To improve performance, DRPL utilizes gradient loss and SSIM loss. On the contrary, we employ out-of-focus images and apply different blur measures to steer our CNN.

2.2 Deblurring

Recent deblurring works have imposed constraints on the sparsity of image gradients *e.g.* Levin *et al.* [10] used the hyper-Laplacian prior, Cho [9] applied the ℓ_2 prior, Krishnan *et al.* [49] employed the ℓ_1/ℓ_2 prior. Similarly, Xu *et al.* [50] introduced two-stage optimization with dominant edges in the image, whereas Whyte *et al.* [51] used auxiliary variables in the Richardson-Lucy deblurring algorithm. Our deblurring method incorporates a pixel-wise depth map to deblur the images.

2.2.1 Edge priors

Single image deblurring approaches use edges extracted implicitly or explicitly for kernel estimation. A few notable examples of deblurring approaches utilize image edge information for computing the blur kernel. Many deblurring methods (*e.g.* [9] and [50]) rely on the detection and selection of sharp edges through bilateral filtering, shock filtering, and magnitude thresholding. Similarly, for spatially varying kernels, step edges are predicted by [52] from the blurred ones. Furthermore, another method proposed by [53] used the detected edges to calculate the Radon transform of the point-spread functions (PSF). Recently, [54] proposed hyper-laplacian prior. Similarly, [55] estimated the blur kernel by using only non-saturated pixels. Here, the purpose is to reduce the ringing artifacts in the latent image by discarding the saturated pixels in the image during the blur kernel estimation. Edge priors are also employed in text image deblurring, Pan *et al.* [56] used ℓ_0 prior to extract edges. This approach is beneficial when the background is smooth; however, it fails to perform satisfactorily in textured image regions. A problem with edge priors is selecting wrong edges for the blur kernel estimation and can happen most often as there are multiple copies of the same edge because of the blur kernel. Extracting the definite edge, in this case, would be quite challenging. Finally, images with limited texture (*e.g.* faces and text) often do not augment from methods using edge priors.

2.2.2 Probabilistic priors

Another line of research adopted a probabilistic perspective modeling the posterior probability for the latent image and the blur kernel. Fergus *et al.* [6] illustrated that modeling the distribution of gradient image as a mixture of Gaussians while that of the blur kernel as exponential distributions. Improving upon this, Shan *et al.* [8] introduced a Maximum a Posteriori (MAP) model assuming a Gaussian noise model, which leads to having constraints on the image *i.e.* gradient sparsity and the blur kernel *i.e.* ℓ_1-norm regularization. On the other hand, [10] proposed an iterative EM strategy by maximizing the posterior distribution over all the potential latent images with the best kernel in hand.

2.2.3 GMM priors

Another approach is to employ certain information from the latent image patches instead of using the entire image for deblurring *e.g.* [57–59]. The GMM model is used by Zoran *et al.* [57] to learn an ample variety of patch appearances prompting imprecise convergence of the solution pair. Similarly, Sun *et al.* [58] exploited atomic structures such as edges, corners, T-junctions *etc.* in the prior learned from the natural and artificial images. Subsequently, *et al.* [59] employed the recurrence property of natural images in multi-scale fashion for estimating the blur kernel.

2.2.4 Learning methods

Recently, due to the availability of a significant amount of training data, many researchers opted to pursue the path of learning from the data (*e.g.* see [60] and [61]). To determine the blur kernels, Schuler *et al.* [60], introduced a convolution neural network composed of the stack of the many CNN modules. This model is applicable for specific kernel size, mainly below 17×17. Similarly, Chakrabarti *et al.* [61] presented a CNN model that learns the Fourier coefficients of the blur kernel, which is then used as input to the non-blind deblurring for recovering the original image. Although it improves the qualitative results; however, it suffers when dense textures exist in the image.

2.2.5 Class-Specific priors

Lately, many methods have incorporated the class information into the image deblurring [62,63]. An approach for photo enhancement by Joshi *et al.* [64] is recently proposed, which uses personal photo albums. One limitation of this method is the manual annotation of the external faces to separate it from the background for segmentation and matting. Subsequently, Hacohen *et al.* [65] investigated class-specific prior, which needs a dense similarity between the blurred image and the sharp reference image. Although this method's result is appealing; however, its applicability is restricted due to the firm requirement for the similar content reference image. In related work, Sun *et al.* [66] tackled the non-blind deblurring problem by incorporating the same content external example images into a prior to transfer mid and high-frequencies to the blurred image. Pan *et al.* [67] presented a deblurring method for face images, where a similar external face image is selected among the training set, then prominent features such mouth, eyes, and lower contour are annotated manually, which guides the deblurring process.

Our approach is different from all the mentioned image deblurring algorithms. Our method does not require any manual annotations or similarity between the training and blurred images. Furthermore, our approach does not rely on external training images, rather the depth map is estimated by our CNN network. We describe our strategy for deblurring in detail in the next section.

2.3 Contributions

This paper aims for depth estimation and blur removal by leveraging rich representations learned from cascaded convolutional and fully connected neural networks operating on patch pooled feature maps. Current techniques estimate depth from sharp images by relying on manually tuned statistical models. Their depth accuracy is limited due to the visual world's variance, and usually, human intervention is required. In contrast, our method benefits from the correspondence between the blurred image and the depth map. Learning the filters to capture these inherent associations through a deep network acts as a prior for estimating better depth.We also exploit the depth of field to sharpen the out-of-focus image. To the best of our knowledge, predicting depth from a single out-of-focus image using deep neural networks has not been investigated before.

We claim the following contributions in this paper.

- Predicting depth from a single out-of-focus image using deep neural networks by exploiting dense overlapping patches.
- Aligning depth discontinuities between the patches of interest using bilateral filtering.
- Incorporating depth map to estimate per pixel blur kernels for non-uniform image deblurring.

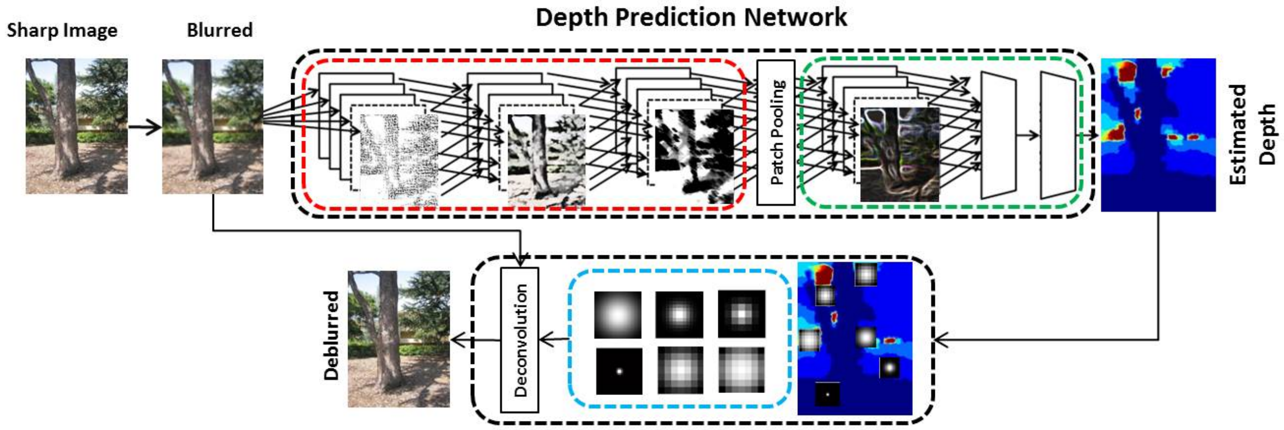


Fig. 2 A representation of the overall method. The sharp image is defocused using circular kernels which simulates capture with a regular aperture. This out-of-focus image is passed to the network ***A*** (shown in red) to compute fully convolutional feature map. A patch pooling extract respective feature map at keypoint locations in the image, which are then propagated through network ***B*** (shown in green) to estimate the depth. Lastly, kernels are computed from the depth map, which is applied to the blurred image, that results in an all-focus image.

3 Depth Estimation

To estimate depth, we introduce a learning procedure using a modified deep neural network [41] by incorporating image-level global context and local evidence. The global context is captured using a fully convolutional network, and the local evidence is absorbed using a fully connected network through patch pooling. In the following section, we discuss the individual components of our system in more detail.

3.1 Network Architecture

The architecture of our network is inspired by the VGG 16-layer very deep network [41]. The input to our proposed network at both the training and testing stage is a fixed-size RGB image. The only preprocessing we apply is mean-normalization, *i.e.* subtracting the mean RGB value from each pixel (computed separately for Make3D [68] and the NYU-v2 [15] training dataset).

Next, the image is passed through a stack of convolutional layers, each consist of traditional 3×3 receptive field filters. In contrast to the [41], we didn't utilize 1×1 convolution filters, which can be seen as a linear transformation of the input channels (followed by non-linearity). A fixed convolution stride of one pixel and the convolutional layer's spatial padding is such that the spatial resolution is preserved after convolution, *i.e.* padding one pixel for 3×3 convolutional layers. Spatial pooling is carried out by five max-pooling layers followed by some of the convolutional layers. Max-pooling is employed over a 2×2 pixel window, with a stride of two. Convolutional layers are followed by the patch pooling layer, which takes the RGB image as input and generates a dense grid of patches over the entire image. It also pools the feature map for each corresponding patch and returns a fixed-size output. Three Fully-Connected (FC) layers follow the patch pooling layer; the first two have 4096 channels, while the third performs 21-way[1] depth estimation and thus contains 21 channels (one for each sampled depth class). The final loss layer is the softmax layer.

This objective function inherently minimizes the multinomial logistic loss, and it maps the output scores of the last fully-connected layer to a probability distribution over classes using the softmax function.

$$\hat{p}_i = \exp(y_i) / \left[\sum_{c=1}^{C} \exp(y_{i^c}) \right] . \tag{1}$$

The computed multinomial logistic loss is then computed for the softmax output class probabilities as

$$E = \frac{-1}{N} \sum_{i=1}^{N} \log(\hat{p}_{(i,L_i)}), \tag{2}$$

where L_i is the quantized depth label for each pixel in the image.

3.2 Depth Prediction

Our network is a cascade of two smaller networks, as shown in Fig. 2. The convolutional deep network ***A*** (shown in red) is designed specifically to enforce the global

[1] Taken from literature for fair comparison

image-level information in depth estimation. It is followed by a shallow, fully connected network $\boldsymbol{B}$ (shown in green) that processes small local evidence for further refinement.

Here, unlike typical networks, images are neither cropped nor warped to prevent them from unintended blur artifacts. The network $\boldsymbol{A}$ operates on the full-scale out-of-focus images and comprises 13 convolutional and four max-pooling layers. The output of the network $\boldsymbol{A}$ is a pixel-level feature map, and we argue, this is essential in modeling depth dynamic range. Furthermore, each layer of data is a four-dimensional array of size $N \times C \times H \times W$. Where N is the number of images in a batch, H is the height of the image, W is the width of the image, and C is the feature (or channel) dimension. The first layer receives the N number of out-of-focus images Y, and in the subsequent layers, input location corresponds to the receptive field regions in the image. The convolution, pooling, and activation functions are the basic components, and since these operators are translation invariant, they apply to local input regions and depend only on relative spatial positions. In any n-th layer, the feature value f_{ij} for the data vector y_{ij} at location $(i,\ j)$ is computed by

$$f_{ij}^{(n)} = \Psi_{ks}(f_{(i+\delta i, j+\delta j)}^{(n-1)}, 0 < \delta_i, \delta j < k), \tag{3}$$

where k denotes the kernel size of the layer, s is subsampling (by a factor of four in both spatial axes) and Ψ_{ks} is the layer type.

3.3 Patch-pooling

Pixel depth prediction requires multiple deconvolutional layers to access an original size image feature map and a pixel-level regression to obtain a full-scale depth map. In practice, pixel-level regression with deep and large network architectures require a comparably large number of iterations for convergence in back-propagation, making the training memory intensive and slow. To overcome this issue, we introduce a small set of keypoint locations on a regular grid to perform patch pooling. This novel patch pooling layer uses max-pooling to convert the computed network response inside a region of interest into a feature map with a fixed spatial extent of $H \times W$ (e.g., 64×64), where H and W are layer hyper-parameters that are independent of any particular patch. A patch Φ is a rectangular window into the convolutional feature map. A tuple (r, c, h, w) defines each patch and specifies its top-left corner (r, c) and its height and width (h, w). The spatial pyramid-pooling [69] layer is carried out on the output of the network $\boldsymbol{A}$ feature map. For a pyramid level with $n \times m$ keypoints, the patch Φ_{ij} corresponding to (i, j)-th keypoint is denoted by

$$\Phi_{ij} = [\lfloor \frac{i-1}{n} w \rfloor, \lceil \frac{i}{n} w \rceil] x [\lfloor \frac{j-1}{m} h \rfloor, \lceil \frac{j}{m} h \rceil]. \tag{4}$$

Intuitively, the floor operation is performed on the left and top boundary while on the right and bottom boundary, the ceiling. These patches are densely extracted from the entire image and hence, overlap. We extract the respective feature map region for each image patch corresponding to a keypoint. In the backward direction [43], the function computes the gradient of the loss function (i.e. softmax loss) with respect to each of its input data vector y_{ij} at location $(i,\ j)$ in n-th layer by following the argmax switches as

$$\frac{\partial L}{\partial y_{ij}^{n}} = \sum_{\Phi} \sum_{k} [ij = ij^{*}(\Phi_{ij}, k)] \frac{\partial L}{\partial y_{ij}^{n+1}}. \tag{5}$$

Each mini-batch contains a number of patches *i.e.* $\Phi = [\Phi_{00}, \ldots, \Phi_{nm}]$, with the corresponding patch-pooling output y_{ij}^{n+1}. The input data pixel y_{ij}^{n} is a part of several patches, thus (possibly) assigned many different labels k. The partial derivative $\partial L / \partial y_{ij}^{n+1}$ is accumulated if ij is the argmax switch selected for y_{ij}^{n+1} by max pooling. In back-propagation, the partial derivatives $\partial L / \partial y_{ij}^{n+1}$ are already computed by the backward functions of the next layer (*i.e.* the network $\boldsymbol{B}$) on top of our patch-pooling layer.

The network $\boldsymbol{B}$ operates on the sampled feature map, which is defined as 64×64 spatial neighborhood for each sampled keypoint in the image. This network is shallow and consists of only fully connected layers. It is designed specifically to predict *one* depth value for each keypoint in the image. Network $\boldsymbol{A}$, patch-pooling and network $\boldsymbol{B}$ are trained jointly as outlined in Sections 3.5 and 3.6.

3.4 Depth Estimation with Fast Bilateral Filtering

The depth map $\tilde{Z}$ predicted by the network $\boldsymbol{B}$ is not continuous, however, the spatial dimensions of $\tilde{Z}$ and out-of-focus image Y are the same, but $\tilde{Z}$ has regions with missing values. To estimate the missing pixels in $\tilde{Z}$, we interpolate using nearby keypoints.

Furthermore, our intuition is that the color intensity discontinuities must be aligned with the depth discontinuities between the patches of interest. The predicted depth values at the nearby keypoint locations are used to interpolate each pixel's depth of out-of-focus image. Using fast bilateral filtering with Y, the smoothness constraint on the boundary pixels and the edge alignment constraint on the image pixels can be simultaneously satisfied. Inpainting of depth map is an ill-posed problem; therefore, an additional prior on the structure is required. The filtered depth map Z is a combination of $\tilde{Z}$ (data term) and Y bilateral features (smoothness term), which is inspired by [70].

3.5 Training

We adopt a pragmatic two-step training scheme to learn shared features via alternating optimization. We first train network $\boldsymbol{A}$ based on back-propagation to learn weights. Then, fixing the weights for the network $\boldsymbol{A}$, we train network $\boldsymbol{B}$. Besides, we jointly fine-tune both networks, once network $\boldsymbol{A}$ and $\boldsymbol{B}$ are fully trained individually. The training is carried out by mini-batch gradient descent to optimize the softmax objective.

In all the experimental settings, the batch size is a single image and its keypoint locations. The number of keypoints is set to 15K patches for NYU-v2 and 7K for the Make3D dataset. The learning rate is initially set to 10^{-2} and then decreased by a factor of ten after 15K iterations. In total, we train our system only for 25K iterations (five epochs), hence reducing the learning rate only once.

As with any gradient-descent framework, the initialization of the network weights is crucial. Improper initialization can stall the convergence due to the numerical instability of gradients in deep networks. To address this issue, we use the pre-trained object recognition network weights to initialize our model. We train the networks for depth prediction using a 21-bin strategy, as described in Section 3.4.

3.6 Testing

After jointly fine-tuning both networks $\boldsymbol{A}$ and $\boldsymbol{B}$, we follow the standard test procedure. Given a color input image of size $H \times W \times C$, we extract patches corresponding to all keypoint locations in the image, forward-propagate them through the network $\boldsymbol{A}$ and compute the full image feature map. Subsequently, we perform patch-pooling to extract the features for each corresponding region and forward-propagate them to the network $\boldsymbol{B}$. The output of the network $\boldsymbol{B}$ along with the input image is post-processed using the fast bilateral filtering to estimate the full resolution continuous-valued responses for each pixel in the input image.

4 Deblurring/Refocusing

After computing the depth map for the out-of-focus image, we construct a sharp image in-focus at all pixels. For this purpose, each pixel of the image is deconvolved using the kernels for every pixel in the depth map. These kernels are directly set from the estimated depth values. Since the deconvolution is done for each pixel, there are no visible artifacts generated near depth discontinuities.

This pixel-based deconvolution approach is more effective in comparison to [37, 72] where regions near depth discontinuities exhibit ringing artifacts. We use a modified version of non-blind deblurring by [17]

$$E(x_{ij}) = \|x_{ij} * k^d_{ij} - y_{ij}\|^2 + \tau \|\nabla x_{ij}\|^{0.8}, \tag{6}$$

where x_{ij} is the in-focus image pixel, y_{ij} is the out-of-focus image pixel and k^d_{ij} is the kernel at location (i, j). The first term of Eq 6 is called the data fidelity term and minimizes the difference between the ground-truth image and the blurry image. This aim is to keep the deblur image faithful to the original image. The second term of Eq 6 is called regularization or a prior due to the equation's ill-posedness. An effective prior such as the one based on gradient sparsity helps avoid the trivial solutions and guides the process to the meaningful outcome.

For each pixel in image Y, we first compute the kernel k^d_{ij} from the depth map Z at ij-th pixel position. Next, each pixel of the sharp image X is obtained by deconvolving a patch of 25×25 centered around the same pixel of Y with k^d_{ij} using eq 6. This technique ensures that the deconvolved pixel will not be affected by ringing artifacts. The sharp image X is generated by aggregating all deconvolved pixels x_{ij} into their original positions. Although this process of deblurring patches is more accurate but computationally expensive.

5 Experimental Analysis and Discussion

In this section, we present both qualitative and quantitative evaluations and comparisons against state-of-the-art methods such as Make3D [68], DepthTransfer [71], DFD [72], and DCNF-FCSP [34]. Similarly for deblurring, we compare with [51], [9], [49]. [73], and [50].

We use average relative error (rel), root-mean-square error (rms), and average log_{10} error for depth estimation and Peak-Signal-to-Noise Ratio (PSNR) for blur removal. Depth estimation experiments were performed using the Caffe framework for efficient inference at the test time. This platform also allows sharing features during training. In step-wise training, stochastic gradient descent mini-batches are sampled randomly from N images. Nevertheless, we use all patches for the sampled images in the current mini-batch—overlapping patches from the same image share computation and memory in the forward and backward passes. Training is done on

Table 1 Network configuration for our depth estimation, using dense patch pooling, is based on VGG16 [41]. For brevity, we use *conv* for convolutional layer, *relu* for activation function, *ip* for inner product and *interp* for bilinear upsampling.

Layer	**1-2**	**3-4**	**5-7**	**8-10**	**11-13**	**14-15**	**16**
Type	conv+relu	conv+relu	conv+relu	conv+relu	conv+relu	ip+relu	ip+interp
Filter Size	3×3	3×3	3×3	3×3	3×3	-	-
No. of Filter	64	128	256	512	512	4k	20
Pooling	max	max	max	max	**patch**	-	-

Table 2 Comparison for depth estimation on Make3D [68] dataset. Our method achieves the best in all error evaluation metrics using the training/test partition provided by [68].

Make3D	**Error (C1)** (lower is better)			**Error (C2)** (lower is better)		
Depth	**rel**	**log10**	**rms**	**rel**	**log10**	**rms**
Saxena[68]	-	-	-	0.370	-	-
DT[71]	1.744	0.407	7.089	1.820	0.415	7.787
DCNF [34]	1.644	0.397	6.725	1.698	0.403	7.310
DFD [72]	0.733	-	4.446	1.000	-	5.149
Ours	**0.213**	**0.075**	**2.560**	**0.202**	**0.312**	**0.079**

Table 3 Quantitative comparison of our depth algorithm on NYU-v2 [15] dataset with the current state-of-the-art alternatives. Our method achieves the best in all error evaluation metrics. Note that the results of [68] and Depth-Transfer [71] are reproduced from [47].

NYU-v2	[72]	[68]	[71]	[47]	[34]	**Ours**
rel	0.609	0.349	0.350	0.215	0.213	**0.094**
log10	-	-	0.131	-	0.087	**0.039**
rms	2.758	1.214	1.200	0.907	0.759	**0.347**

a standard desktop with an NVIDIA Tesla K40c GPU with 12GB memory.

5.1 Network Architecture

In this section of the paper, we present the proposed network parameters for the depth estimation. VGG16 [41] inspires our network. The configuration of our network is shown in the Table 1. Our network is composed of 16 convolution layers and three fully-convolutional layers. Like VGG16, our network also starts from 64 channels and the increased by a product of two after each pooling layer, eventually reaching 512 channels. The total number of parameters in our CNN network is similar to VGG16 *i.e.* 138M.

5.2 Preparing Synthetic Images

The Synthetic out-of-focus images with spatially varying blur kernels were generated using the corresponding ground truth depth maps. For this purpose, we selected two image datasets having ground truth depth maps for as described in section 5.3. The depth variation is dependent on the collection methods and type of sensors used, *e.g.* Make3D [68] dataset has a depth variation of 1-80 meters with more than two depth layers. Subsequently, the Gaussian blur kernel is generated from each depth layer and applied to the corresponding sharp image.

5.3 Datasets

We performed experimental validation on two datasets: NYU-v2 [15] and Make3D [68]. For depth estimation, we use the standard test images provided with these datasets, while for blur removal we use randomly selected subset of images from each dataset.

NYU-v2: This dataset consists of 1449 color and depth images of indoor scenes. The dataset is split into 795 images for training and 654 images for the test. All images are resized to 420×640, and white borders are removed.

Make3D: This dataset consists of 534 color and depth images of outdoor scenes. It is split into 400 images for training and 134 images for the test. All images are resized to 460×345.

5.4 Depth Estimation

Table 2 shows the results for Make3D dataset. Our proposed method outperforms for all metrics as well as both C1 and C2 errors. In terms of root mean square (rms) our method is leading by a margin of **1.87** (C1 error) and **5.07** (C2 error) from the second best performer.

In [34], the superpixel pooling method extracts useful regions from the convolutional feature maps instead of image crops. However, their superpixel strategy does not take into account the overlapping regions. Besides this, the number of superpixels per image are small and vary in size. In contrast, the patches we select are very dense and have overlapping areas, which helps to predict pixel-wise depth across different patches more accurately. We observe that the keypoint locations and dense patches on a regular grid are more beneficial than non-overlapping superpixel segments. The method in [72] is trained explicitly for estimating depth from out-of-focus images; therefore, it outperforms other alternatives that use sharp images only.

The results for the NYU-v2 dataset are shown in Table 3. Concerning *rms* error our method is **0.412** higher than the best performing method among the alternatives with similar observations for log_{10} and *rel*. Since the current state-of-the-art methods fail to exploit out-of-focus images (except [72]), we reproduced their original results for the NYU-v2 dataset. In contrast, our method takes out-of-focus images to estimate depths and still able to outperform all competing methods by a significant margin. Some qualitative results are shown in Fig. 3. The proposed method has captured the depth accurately for the near and distant objects in the scene.

5.5 Removing Non-uniform Blur

In this section, we evaluate our blur removal method on test images from NYU-v2 and Make3D. The proposed deblurring method outperforms all competing algorithms on all test images for non-uniform blur. Figure 5 shows the results generated by our and competing schemes for different images. Our algorithm delivers higher visual quality than its counterparts. Furthermore, our algorithm can restore high-frequency texture details with a closer resemblance to the ground truth than existing methods due to estimating blur kernels from depth layers. In Fig. 5, the highly textured patterns on walls are adeptly reproduced by our algorithm, while these details are missing in the results of the other methods. In this example, most of the other methods tend to smoothen out the variation of the background texture along with one of its principal directions. Besides, some methods introduce additional artifacts and artificial textures.

We present a sample result on the NYU-v2 dataset here. Figure 4 shows the results generated by our and competing methods for different images. Our algorithm attains higher visual quality than its counterparts. It is observed that our algorithm can restore high-frequency texture details with a closer resemblance to the ground-truth than existing methods due to estimating blur kernels from depth layers. In Fig. 4, the highly-textured patterns on walls are adeptly reproduced by our algorithm, while these details are not clearly visible in the results of the other methods. In this example, most of the other methods tend to smoothen out the variation of the background texture along with one of its principal directions. Furthermore, some methods introduce additional artifacts and artificial textures.

In Table 4, we report the blur removal accuracy of our algorithm, measured by PSNR across all the test images, with the highest PSNR in each comparison is highlighted in bold. The average improvement (in PSNR) by our non-uniform deblurring algorithm over the state-of-the-art methods for NYU-v2 is at least **0.72 dB**, and for Make3D is at least **0.36 dB** on test images as shown in Table 4. This significant improvement demonstrates the advantage of incorporating a deep neural network-based depth map for kernel estimation in blur removal. The numerical results of our deblurring approach are faithful to the visual observation presented earlier in Figures 4 and 5.

5.6 Real Out-of-focus Images

In this experiment, we evaluate the proposed method on real-world blurred images. Comparisons with state-of-the-art methods [34,68,71] are shown in figure 6. In the bird example, the objects close to the camera are in focus while the background is out-of-focus, which is reflected in our results, while other baseline methods fail to capture the relationship between depth and blur and hence, do not perform well in this scenario.

In our last experiment, we present the visual results on the real-world out-of-focus face image. Comparison with state-of-the-art methods are shown in figure 7. In the face example, there are approximately two layers of blur corresponding to different levels, as shown in figure 7. Our method put the face and background in different depth levels by exploiting the blur while [34] generates sharp boundaries for the face but puts a different level of depth for the same face layer. Thus, making our method more useful in a practical situation in the presence of non-uniform blur. Although our approach for the depth estimation outperformed; however, we acknowledge that our proposed algorithm does not estimate the depth on real images as accurately as on the synthetic images.

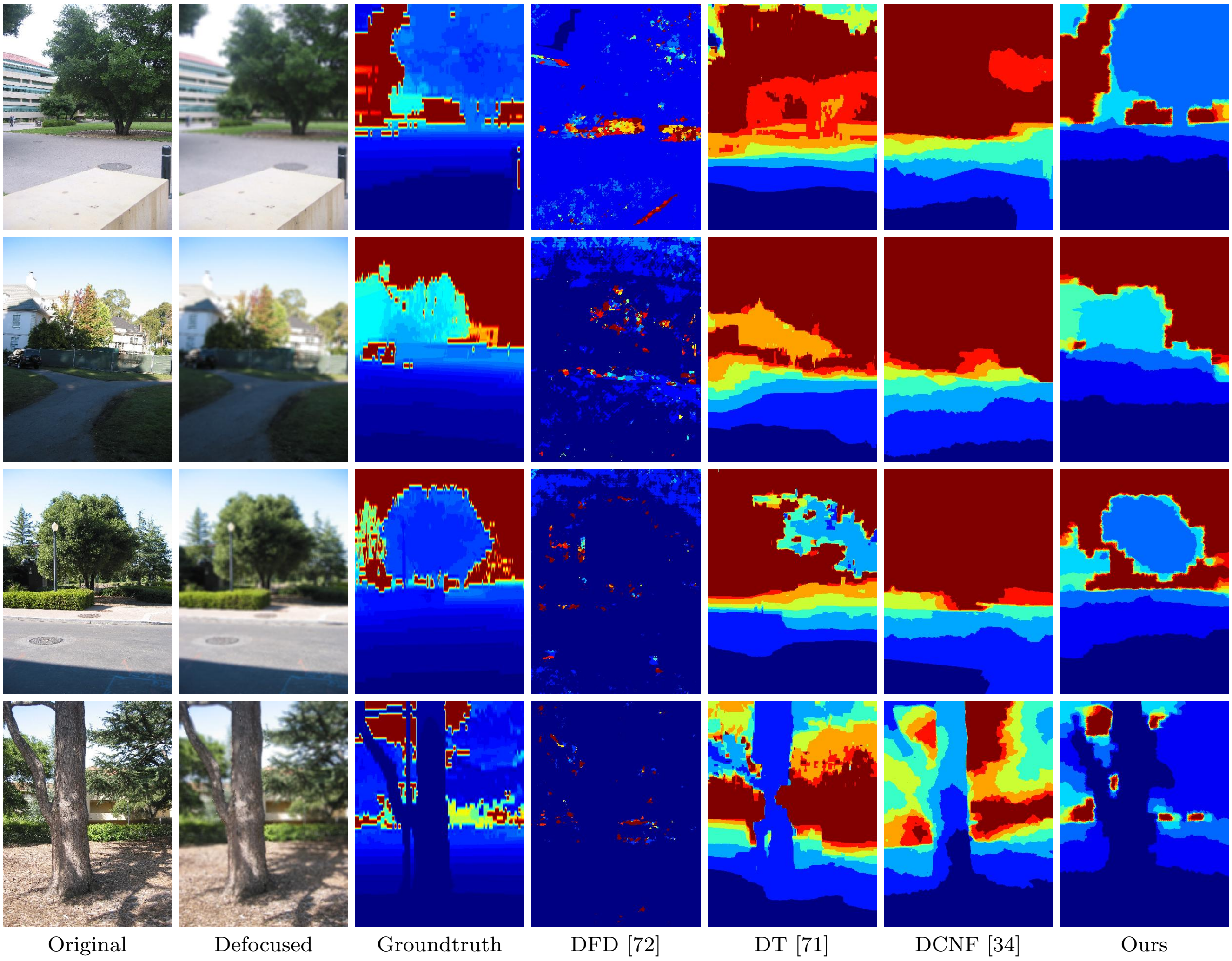


Fig. 3 Qualitative comparison of depth estimation on [68] dataset. Our method predicted the depth levels more accurately as compared to the competitive methods. Red color represents far while blue represents near.

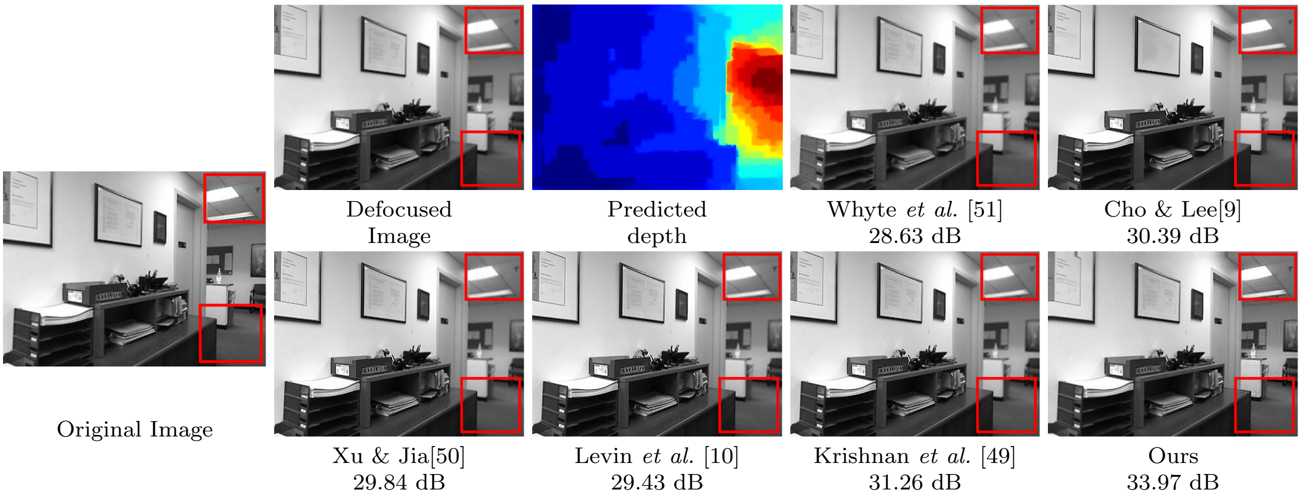
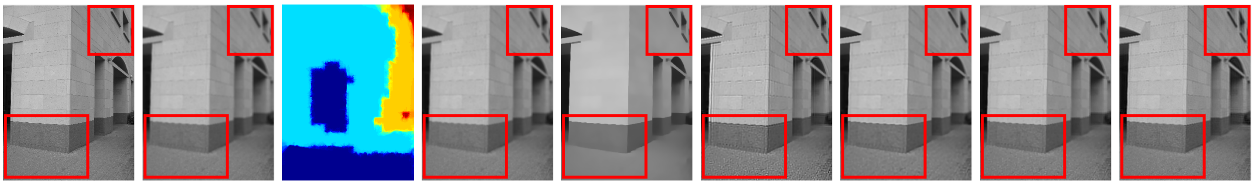


Fig. 4 Qualitative comparison of our deblurring results on NYU-v2 [68] dataset with state-of-the-art deblurring methods. The difference can be seen in the red box and best viewed at higher magnification.

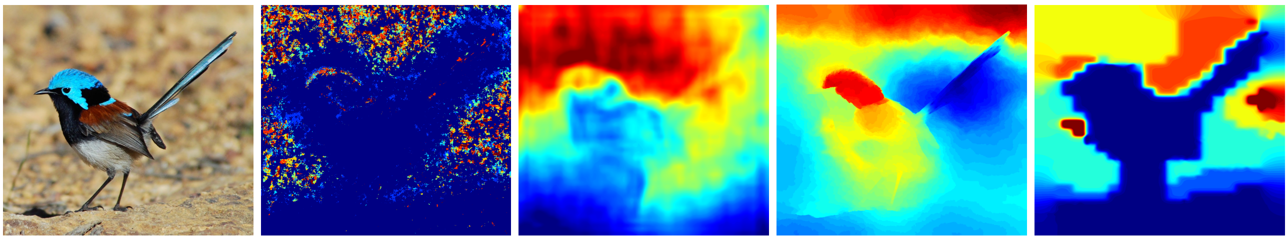


Original | Defocus-ed Image | Predicted depth | [51] 28.85 dB | [74] 28.42 dB | [50] 27.43 dB | [10] 29.66 dB | [49] 29.51 dB | Ours 30.27 dB

Fig. 5 An example from Make3D [68] dataset. Our deblurring method has recovered more details without producing ringing artifacts. Best viewed at higher magnification.

Table 4 Quantitative comparison of our deblurring method on Make3D [68] and NYU-v2 [15] datasets with state of the art deblurring methods.

Deblurring	**Peak Signal to Noise Ratio (PSNR)** (Higher is better)					
	[51]	[74]	[50]	[49]	[10]	**Ours**
Make3D	19.95	20.46	20.71	20.29	20.67	**21.07**
NYU-v2	28.23	31.72	31.82	33.49	33.02	**34.21**



Real Image | DFD [72] | Eigen *et al.* [47] | DCFN [34] | Ours

Fig. 6 Real defocused image with unknown blur. Our method benefits from the amount of blur in the real images whereas other methods rely on the color and shape of the object which fails to recover the depth.

6 Conclusion

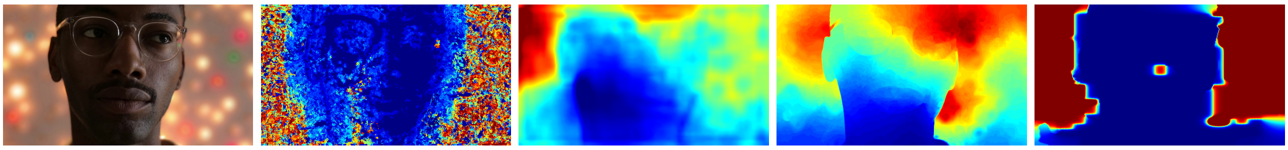
We have presented a novel deep convolutional neural framework that estimates the depth map from an out-of-focus image. This depth map is later utilized to deblur the same out-of-focus image. Furthermore, the patch-pooling strategy aims to extract feature maps at densely selected keypoint locations that are effective and efficient for depth estimation. The fundamental difference from existing methods is the formulation of the convolutional neural network-based depth estimation from defocus and incorporating the resulting depth map in deblurring. It should be noted here that competitive depth from defocus (DFD) methods require extra hardware constraints, such as using patterns on the aperture to predict the depth map. We have extensively validated our approach on indoor and outdoor benchmark datasets and observed that our method outperformed state-of-the-art depth estimation as well as the uniform and the non-uniform deblurring methods.

One of the limitations of our work is not incorporating any geometric cues for the depth estimation. This direction is worth investigating in the future. Our CNN model may also be applicable for other image processing tasks such as image denoising, image inpainting, and image super-resolution. Furthermore, different computer vision algorithms can also benefit from our model where depth is required and readily unavailable, for example, object detection, segmentation, and classification.

Our method benefits from out-of-focus blur to estimate the depth map; however, it will not be able to determine the depth in the presence of camera shake or motion blur. Similarly, our method is limited in handling outliers and noise present in the defocus image. Our future work will investigate fixed budget depth estimation from motion blur/camera-shake and joint computation of the depth map and the deblurred image. We will also employ techniques that can handle outliers and noise during deblurring.

References

1. X. Zhu and D. Ramanan, “Face detection, pose estimation, and landmark localization in the wild,” in *CVPR*, 2012.



Real Image DFD [72] Eigen *et al.* [47] DCFN [34] Ours

Fig. 7 Images are real defocused photos with unknown blur, and qualitative comparison shows significant improvement over state-of-the-art depth prediction methods. Our method benefits from the amount of blur in the real images, whereas other methods rely on the object's color and shape, which fails to recover the depth.

2. D. Eigen and R. Fergus, "Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture," in *ICCV*, 2015.
3. P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *CVPR*, 2001.
4. K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016.
5. J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *CVPR*, 2015.
6. R. Fergus, B. Singh, A. Hertzmann, S. T. Roweis, and W. T. Freeman, "Removing camera shake from a single photograph," 2006.
7. A. Levin, "Blind motion deblurring using image statistics," in *NIPS*, 2006.
8. Q. Shan, J. Jia, and A. Agarwala, "High-quality motion deblurring from a single image," *ACM Trans. Graph.*, 2008.
9. S. Cho and S. Lee, "Fast motion deblurring," in *ACM Transactions on Graphics (TOG)*, 2009.
10. A. Levin, Y. Weiss, F. Durand, and W. T. Freeman, "Understanding blind deconvolution algorithms," *TPAMI*, 2011.
11. S. K. Nayar and M. Ben-Ezra, "Motion-based motion deblurring," *TPAMI*, 2004.
12. F. Li, J. Yu, and J. Chai, "A hybrid camera for motion deblurring and depth map super-resolution," in *CVPR*, 2008.
13. Y.-W. Tai, H. Du, M. S. Brown, and S. Lin, "Image/video deblurring using a hybrid camera," in *CVPR*, 2008.
14. L. Yuan, J. Sun, L. Quan, and H.-Y. Shum, "Image deblurring with blurred/noisy image pairs," ser. SIGGRAPH, 2007.
15. P. K. Nathan Silberman, Derek Hoiem and R. Fergus, "Indoor segmentation and support inference from rgbd images," in *ECCV*, 2012.
16. S. Anwar, Z. Hayder, and F. Porikli, "Depth estimation and blur removal from a single out-of-focus image." in *BMVC*, vol. 1, 2017, p. 2.
17. A. Levin, R. Fergus, F. Durand, and W. T. Freeman, "Image and depth from a conventional camera with a coded aperture," *ACM Trans. Graph.*, 2007.
18. A. Veeraraghavan, R. Raskar, A. Agrawal, A. Mohan, and J. Tumblin, "Dappled photography: Mask enhanced cameras for heterodyned light fields and coded aperture refocusing," *ACM Trans. Graph.*, 2007.
19. F. Moreno-Noguer, P. N. Belhumeur, and S. K. Nayar, "Active refocusing of images and videos," *ACM Trans. Graph.*, 2007.
20. C. Zhou, O. Cossairt, and S. Nayar, "Depth from diffusion," in *CVPR*, 2010.
21. C. Zhou and S. Nayar, "What are good apertures for defocus deblurring?" in *ICCP*, 2009.
22. C. Zhou, S. Lin, and S. K. Nayar, "Coded aperture pairs for depth from defocus and defocus deblurring," *IJCV*, 2011.
23. A. Levin, "Analyzing depth from coded aperture sets," in *ECCV*, 2010.
24. S. Pertuz, D. Puig, and M. A. Garcia, "Analysis of focus measure operators for shape-from-focus," *PR*, 2013.
25. M. Mahmood and T. S. Choi, "Nonlinear approach for enhancement of image focus volume in shape from focus," *TIP*, 2012.
26. S. O. Shim and T. S. Choi, "A fast and robust depth estimation method for 3d cameras," in *ICCE*, 2012.
27. M. Subbarao and T. Choi, "Accurate recovery of three-dimensional shape from image focus," *TPAMI*, 1995.
28. S. Bae and F. Durand, "Defocus magnification," *CG Forum*, 2007.
29. F. Calderero and V. Caselles, "Recovering relative depth from low-level features without explicit t-junction detection and interpretation," *IJCV*, 2013.
30. Y. Cao, S. Fang, and F. Wang, "Single image multi-focusing based on local blur estimation," in *ICIG*, 2011.
31. S. Zhuo and T. Sim, "Defocus map estimation from a single image," *PR*, 2011.
32. V. P. Namboodiri and S. Chaudhuri, "Recovery of relative depth from a single observation using an uncalibrated (real-aperture) camera," in *CVPR*, 2008.
33. M. Liu, M. Salzmann, and X. He, "Discrete-continuous depth estimation from a single image," in *CVPR*, 2014.
34. F. Liu, C. Shen, and G. Lin, "Deep convolutional neural fields for depth estimation from a single image," in *CVPR*, 2015.
35. M. Watanabe and S. K. Nayar, "Rational filters for passive depth from defocus," *IJCV*, 1998.
36. C. Paramanand and A. N. Rajagopalan, "Non-uniform motion deblurring for bilayer scenes," in *CVPR*, 2013.
37. L. Xu and J. Jia, "Depth-aware motion deblurring," in *ICCP*, 2012.
38. C. Li, S. Su, Y. Matsushita, K. Zhou, and S. Lin, "Bayesian depth-from-defocus with shading constraints," in *CVPR*, 2013.
39. M. S. Farid, A. Mahmood, and S. A. Al-Maadeed, "Multi-focus image fusion using content adaptive blurring," *Information fusion*, vol. 45, pp. 96–112, 2019.
40. A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *NIPS*, 2012.
41. K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.
42. R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *CVPR*, 2014.
43. R. Girshick, "Fast r-cnn," in *ICCV*, 2015.

44. A. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "Cnn features off-the-shelf: an astounding baseline for recognition," in *CVPR Workshops*, 2014.
45. H. Su, Q. Huang, N. J. Mitra, Y. Li, and L. Guibas, "Estimating image depth using shape collections," *TG*, 2014.
46. A. Kar, S. Tulsiani, J. Carreira, and J. Malik, "Category-specific object reconstruction from a single image," in *CVPR*, 2015.
47. D. Eigen, C. Puhrsch, and R. Fergus, "Depth map prediction from a single image using a multi-scale deep network," in *NIPS*, 2014.
48. J. Li, X. Guo, G. Lu, B. Zhang, Y. Xu, F. Wu, and D. Zhang, "Drpl: Deep regression pair learning for multi-focus image fusion," *IEEE Transactions on Image Processing*, vol. 29, pp. 4816–4831, 2020.
49. D. Krishnan, T. Tay, and R. Fergus, "Blind deconvolution using a normalized sparsity measure," in *CVPR*, 2011.
50. L. Xu and J. Jia, "Two-phase kernel estimation for robust motion deblurring," in *ECCV*, 2010.
51. O. Whyte, J. Sivic, A. Zisserman, and J. Ponce, "Non-uniform deblurring for shaken images," *IJCV*, Jun. 2012.
52. N. Joshi, R. Szeliski, and D. Kriegman, "Psf estimation using sharp edge prediction," in *CVPR*, 2008.
53. T. S. Cho, S. Paris, B. K. Horn, and W. T. Freeman, "Blur kernel estimation using the radon transform," in *CVPR*, 2011.
54. D. Krishnan and R. Fergus, "Fast image deconvolution using hyper-laplacian priors," in *NIPS*, 2009.
55. O. Whyte, J. Sivic, and A. Zisserman, "Deblurring shaken and partially saturated images," *IJCV*, 2014.
56. J. Pan, Z. Hu, Z. Su, and M. H. Yang, "Deblurring text images via L0 regularized intensity and gradient prior," in *CVPR*, 2014.
57. D. Zoran and Y. Weiss, "From learning models of natural image patches to whole image restoration," in *ICCV*, 2011.
58. L. Sun, S. Cho, J. Wang, and J. Hays, "Edge-based blur kernel estimation using patch priors," in *ICCP*, 2013.
59. T. Michaeli and M. Irani, "Blind deblurring using internal patch recurrence," in *ECCV*, 2014.
60. C. J. Schuler, M. Hirsch, S. Harmeling, and B. Schölkopf, "Learning to deblur," *TPAMI*, 2016.
61. A. Chakrabarti, "A neural approach to blind motion deblurring," in *ECCV*, 2016.
62. S. Anwar, C. Phuoc Huynh, and F. Porikli, "Class-specific image deblurring," in *ICCV*, 2015.
63. S. Anwar, C. P. Huynh, and F. Porikli, "Image deblurring with a class-specific prior," *TPAMI*, 2017.
64. N. Joshi, W. Matusik, E. H. Adelson, and D. J. Kriegman, "Personal photo enhancement using example images," *ACM Trans. Graph*, 2010.
65. Y. Hacohen, E. Shechtman, and D. Lischinski, "Deblurring by example using dense correspondence," in *ICCV*, 2013.
66. L. Sun, S. Cho, J. Wang, and J. Hays, "Good Image Priors for Non-blind Deconvolution - Generic vs. Specific," in *ECCV*, 2014.
67. J. Pan, Z. Hu, Z. Su, and M. Yang, "Deblurring face images with exemplars," in *ECCV*, 2014.
68. A. Saxena, M. Sun, and A. Y. Ng, "Make3d: Learning 3d scene structure from a single still image," *TPAMI*, 2009.
69. K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," in *ECCV*, 2014.
70. A. Levin, A. Zomet, and Y. Weiss, "Learning how to inpaint from global image statistics," in *ICCV*, 2003.
71. K. Karsch, C. Liu, and S. B. Kang, "Depth transfer: Depth extraction from video using non-parametric sampling," *TPAMI*, 2014.
72. A. Chakrabarti and T. Zickler, "Depth and deblurring from a spectrally-varying depth-of-field," in *ECCV*, 2012.
73. A. Levin, Y. Weiss, F. Durand, and W. T. Freeman, "Efficient marginal likelihood optimization in blind deconvolution," in *CVPR*, 2011.
74. S. Cho and S. Lee, "Fast motion deblurring," ser. SIGGRAPH Asia, 2009.

Saeed Anwar is a Research Scientist in the CSIRO-Data61, Australia, and Adjunct Lecturer at Australian National University. He has received his Ph.D. degree from the Australian National University in 2019, the master's degree in Erasmus Mundus Vision and Robotics (Vibot), jointly offered by the Heriot-Watt University, United Kingdom, the University of Girona, Spain, and the University of Burgundy, France with distinction. His current research interests include computer vision, pattern recognition, deep learning, machine learning, image analysis and restoration, optimization, multimedia processing, medical systems, automotive perception, car navigation, intelligent transportation. Recently, he got Best Paper Award Nominee at IEEE CVPR 2020.

Zeeshan Hayder has a Doctor of Philosophy (Ph.D.) degree focused in Computer Engineering from The Australian National University and NICTA. He has also finished a research internship at Intel Visual Computing Lab, California, USA. His research interests include computer vision, machine learning, and artificial intelligence, particularly on deep structured models for large scene analysis, including object recognition, detection, classification, segmentation, and graphical modeling. He is also interested in computer vision and machine learning applications in robotics, cyber-physical systems, and ubiquitous computing. He served as a reviewer for over ten international journals and conferences.

Fatih Porikli is an IEEE Fellow and a Professor in the Research School of Engineering, Australian National University (ANU). He is also managing the Computer Vision Research Group at Data61/CSIRO. He has received his Ph.D. from New York University in 2002. Previously he served Distinguished Research Scientist at Mitsubishi Electric Research Laboratories. Prof. Porikli is the recipient of the R&D 100 Scientist of the Year Award in 2006. He won 4 best paper awards at premier IEEE conferences and received 5 other professional prizes. Prof. Porikli authored more than 150 publications and invented 66 patents. He is the co-editor of 2 books. He is serving as the Associate Editor of 5 journals for the past 8 years. He was the General Chair of

AVSS 2010 and WACV 2014, and the Program Chair of WACV 2015 and AVSS 2012. His research interests include computer vision, deep learning, manifold learning, online learning, and image enhancement with commercial applications in video surveillance, car navigation, intelligent transportation, satellite, and medical systems.