Pyramid Architecture Search for Real-Time Image Deblurring
Supplementary Material

Xiaobin Hu[1,2,7], Wenqi Ren[2]*, Kaicheng Yu[3,6], Kaihao Zhang[4], Xiaochun Cao[2], Wei Liu[5], Bjoern Menze [1,7]
[1]Informatics, TU München [2]SKLOIS, IIE, CAS [3]CVLab, EPFL [4]Australian National University
[5]Tencent AI Lab [6]abacus.Ai [7]Quantitative Biomedicine, University of Zurich

1. Overview

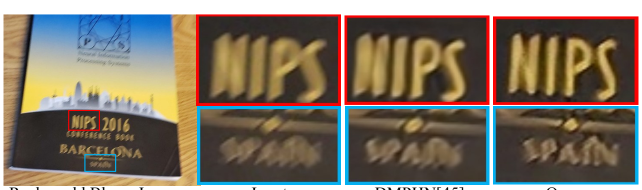
In this supplementary document, we present additional results to complement the paper. Firstly, we provide the detailed configurations and parameters of our proposed architectures (PyNAS$_d$ and PyNAS$_s$) found by our PyNAS (pyramid neural architecture search network). Secondly, more qualitative comparisons with the state-of-the-art algorithms are added in the supplementary.

2. Comparison with real-world results:

As suggested, we compare our approach against two recent deblurring algorithms (CVPR19 [45] and CVPR20 [48]) on the real-world RWBI dataset [48] which captured by various hand-held devices. The dataset consists of 3112 blurry images and no ground truths are provided. We report the quantitative results in Table 1 by the no-reference metric of PIQE on the RWBI dataset. An example is shown in Fig.1. These results demonstrate that the proposed NAS-based method can generalize on real-world conditions and restore higher-realistic deblurred results than the other SOTA methods.

Table 1. Quantitative results with different methods. ↓ means that the lower the better. All models are trained on the GoPro and then generalized on the real-world RWBI dataset.

Criteria	CVPR2019 [45]	CVPR2020 [48]	Ours
PIQE ↓	32.22	30.36	**12.41**



Real-world Blurry Image Inputs DMPHN[45] Ours

Figure 1. Qualitative comparison on real-world blurry images (RWBI).

*Corresponding author

3. Discussion

Searching strategy on top structure: We adopt a weighted random sampling method (a discrete empirical probability model) via assigning the higher probability to the better scale depth (i.e., scale=3,4). We tried to embed the scale depth into gradient back-propagation but the searching was not stable and also affected the performance of the bottom variable searching. We will clarify this in our revised manuscript.

Structure and performance difference with DMPHN: Compared with DMPHN, our PyNAS is an automatic and flexible framework by introducing a large search space including top and bottom variables. In DMPHN, each scale inherits the same structure without considering different requirements of the receptive field at each scale. But PyNAS studies the discrepancy of receptive field at each scale, and then stack different structures for further joint optimization via cutting off unnecessary links or optimizing basic structures. Thus, even for the same patch scheme (PyNAS$_s$), our method decreases 1/3 parameter size (10Mb) of DMPHN. In addition, our network architecture obtained by PyNAS only occupies 1/2 inference time of DMPHN.

Theoretical analysis and concrete proofs on NAS: We outline the evidence in a high level why NAS discovered architectures are surpassing human-designed ones. *i*). The human-designed encoder-decoder structure like DMPHN assumes that each scale uses the same structure without considering the discrepancy of the receptive field at each scale. *ii*). The scale depth, patch scheme, and basic structure are not well-optimized but only empirically designed in a manual way. Specifically, some links from blur features obtained by shallow layers, which probably deteriorate the deblurring reconstruction, can be removed by our NAS. *iii*). As seen in Table 3 of paper, our multi-scale search strategy (PyNAS) is much better than the random policy which is usually adopted in a human-designed process.

More Analyses with similar structures: Different from similar structures [5, 21, 35], our PyNAS adopts the multi-patch mechanism to better represent blur kernel knowledge.

Compared with the information loss of blur kernel due to the image degradation of the multi-scale framework, we utilize the multi-patch scheme to exploit blur kernel priors and better represent global non-uniform knowledge. Furthermore, in contrast to [33, 43, 45], our PyNAS can automatically design different structures based on the different requirements of the receptive field at each scale.

More comparisons with recent work in Table 2: To make our results more convincing, we compare our PyNAS with **Five** most recent deblurring methods in Table 2. For [A1], to ensure a fair comparison, we choose the best models under the constraint of real-time inference. From Table 2, Our PyNAS architecture still performs much better than the previous most recent manual architecture.

Table 2. The most recent quantitative results on the VideoDeblurring dataset. The models are trained on the GoPro dataset and then generalized on the VideoDeblurring dataset [31].

Models	PSNR	Models	PSNR
Kupyn (ICCV2019) [15]	28.53	Zhang (CVPR2019) [45]	30.25
Zhang (CVPR2020) [48]	30.11	Ours (PyNAS$_d$)	**31.01**

4. Network Architectures

Two network architectures are found by our PyNAS algorthim: one is the lightweight (PyNAS$_s$) which searches the identical cells for each scale and can share the weight among scales. The other is the (PyNAS$_d$) which searches individual cell structure of each scale and then stack them together. The learned PyNAS$_s$ is a pyramid shape (1-2-4-8) network and mainly consists of dilation blocks and standard convolution blocks, while the PyNAS$_d$ is a 1-3-9 patch scheme network and contains massive dilation blocks and convolution blocks with large kernel size. Detailed configurations (e.g., kernel size and operators) of PyNAS$_d$ and PyNAS$_s$ can be found in Figure 1 and Figure 2. The selected standard operators are mainly consisted of 3×3 convolution block (3×3_Con), 5×5 convolution block (5×5_Con) and 3×3 dilation block (3×3_Dilation).

5. Visualization Quality of Real-time Image Deblurring

We show the visualization of different models for images containing large motion blur and zoom in the main object. Compared with recent deep learning based methods, the rehabilitated images of our method are clearer and sharper at the edges. The content of our deblurred images are well recovered, e.g., the numbers of advertisement and license plate are deblurred perfectly, while others fail to show clear numbers.

References

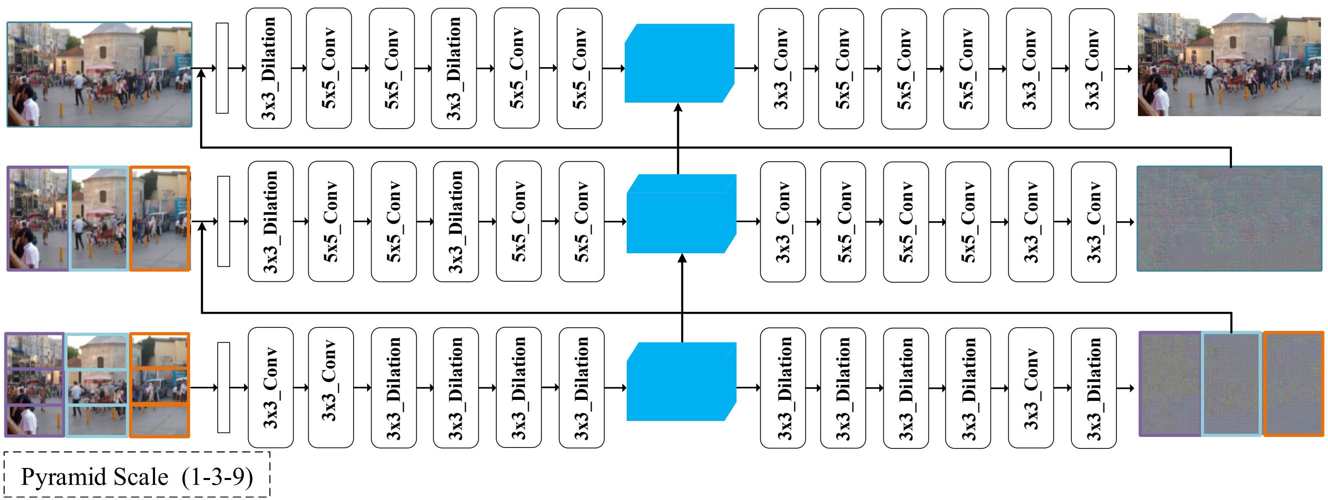


Figure 2. PyNAS$_d$: Our proposed pyramid architecture search (PyNAS) using the pyramid patch scheme (1-3-9) and scale depth (3). The non-overlap multi-patch hierarchy is used as the input of the network. PyNAS searches the whole encoder and decoder structure of each scale of the network from the operator candidates and the path binarization is exploited to search for the best operator. It is noteworthy that our PyNAS finds a better pyramid network architecture (1-3-9) using less inference time and shallower scale depth.

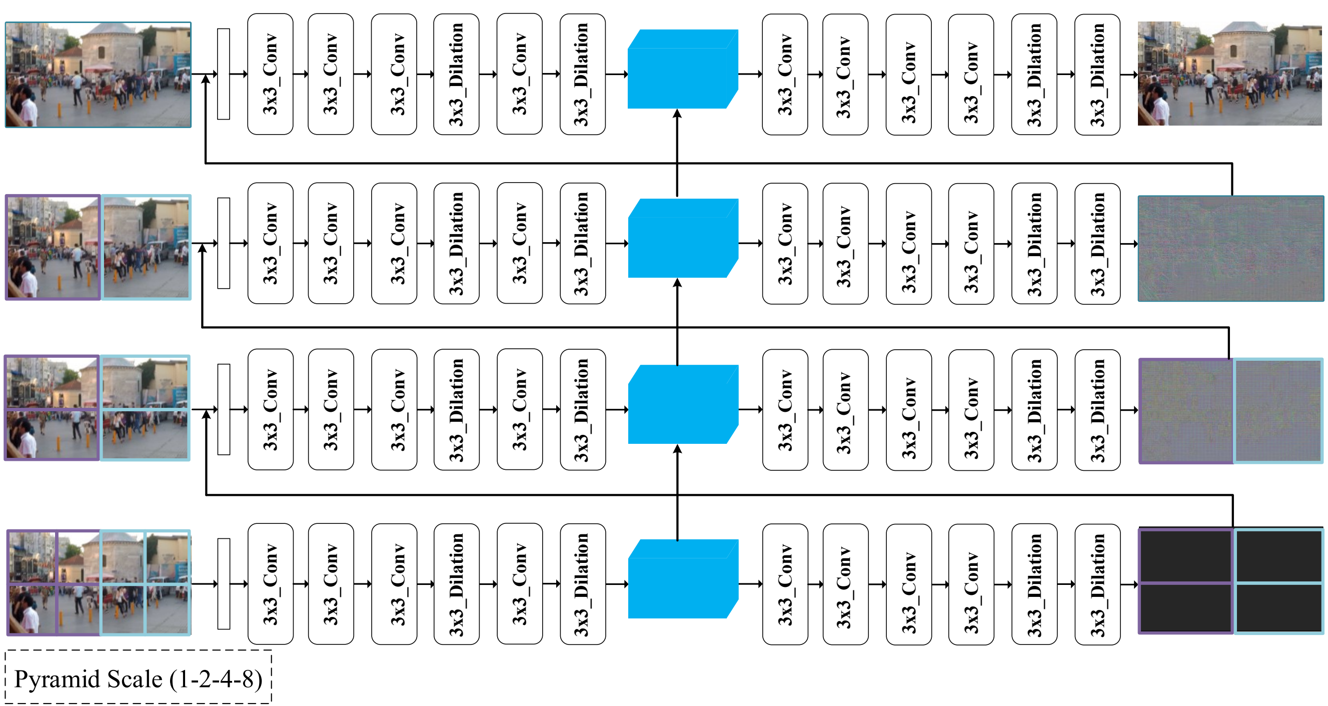


Figure 3. PyNAS$_s$: Our proposed pyramid architecture search (PyNAS) using the pyramid patch scheme (1-2-4-8) and scale depth (4). The non-overlap multi-patch hierarchy is used as the input of the network. PyNAS searches the whole encoder and decoder structure of each scale of the network from the operator candidates and the path binarization is exploited to search for the best operator.

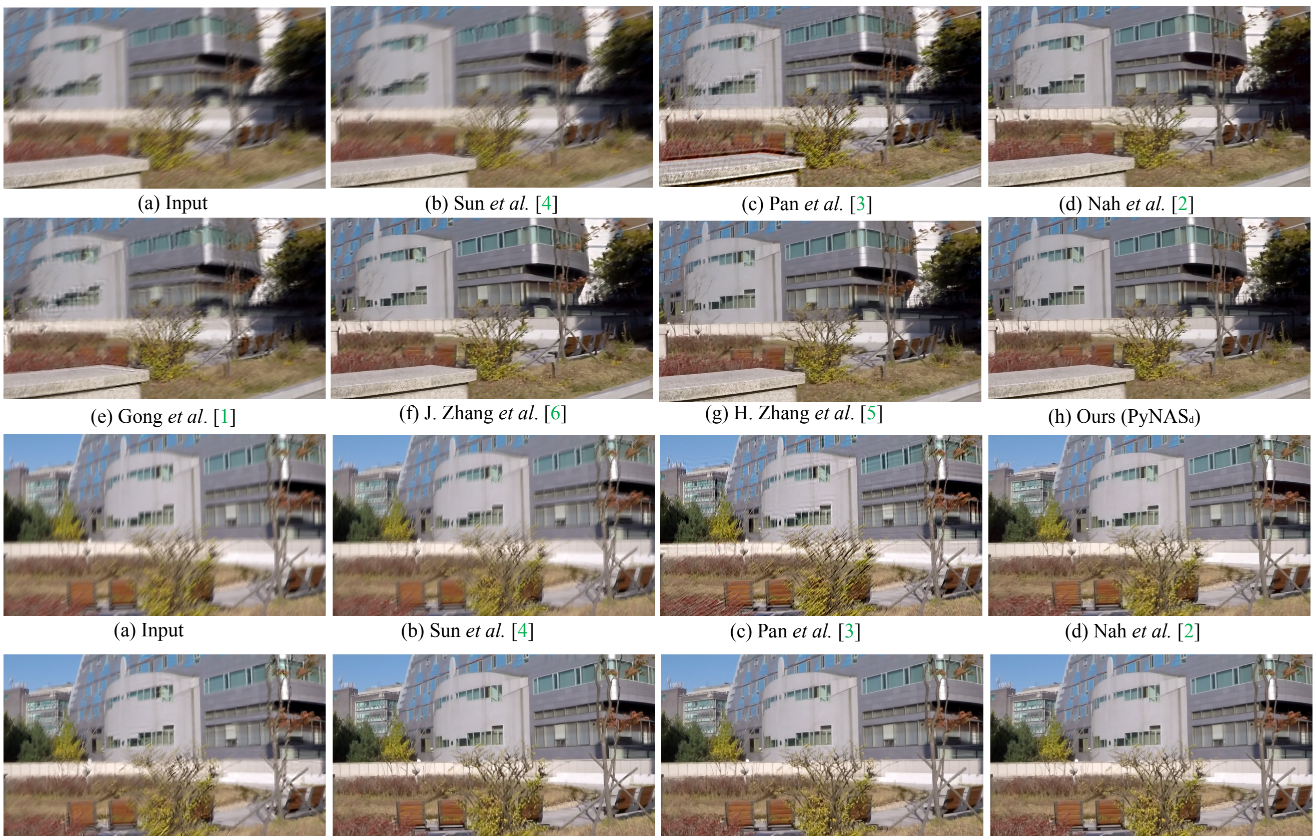
(a) Input (b) Sun *et al.* [4] (c) Pan *et al.* [3] (d) Nah *et al.* [2]

(e) Gong *et al*. [1] (f) J. Zhang *et al*. [6] (g) H. Zhang *et al*. [5] (h) Ours (PyNAS$_d$)

(a) Input (b) Sun *et al.* [4] (c) Pan *et al.* [3] (d) Nah *et al.* [2]

(e) Gong *et al*. [1] (f) J. Zhang *et al*. [6] (g) H. Zhang *et al*. [5] (h) Ours (PyNAS$_d$)

Figure 4. Visual comparison with state-of-the-art Image Deblurring methods on GOPR0868-11-00.



(a) Input (b) Sun *et al.* [4] (c) Pan *et al.* [3] (d) Nah *et al.* [2]

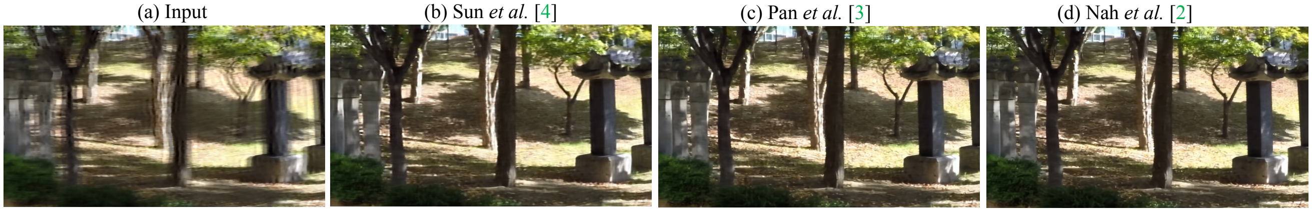
(e) Gong *et al*. [1] (f) J. Zhang *et al*. [6] (g) H. Zhang *et al*. [5] (h) Ours ($PyNAS_d$)

(a) Input (b) Sun *et al.* [4] (c) Pan *et al.* [3] (d) Nah *et al.* [2]

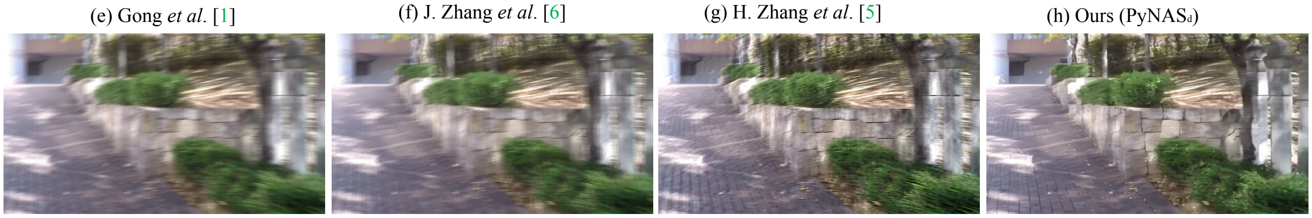
(e) Gong *et al*. [1] (f) J. Zhang *et al*. [6] (g) H. Zhang *et al*. [5] (h) Ours ($PyNAS_d$)

Figure 5. Visual comparison with state-of-the-art Image Deblurring methods on GOPR0869-11-00.

(a) Input (b) Sun *et al.* [4] (c) Pan *et al.* [3] (d) Nah *et al.* [2]



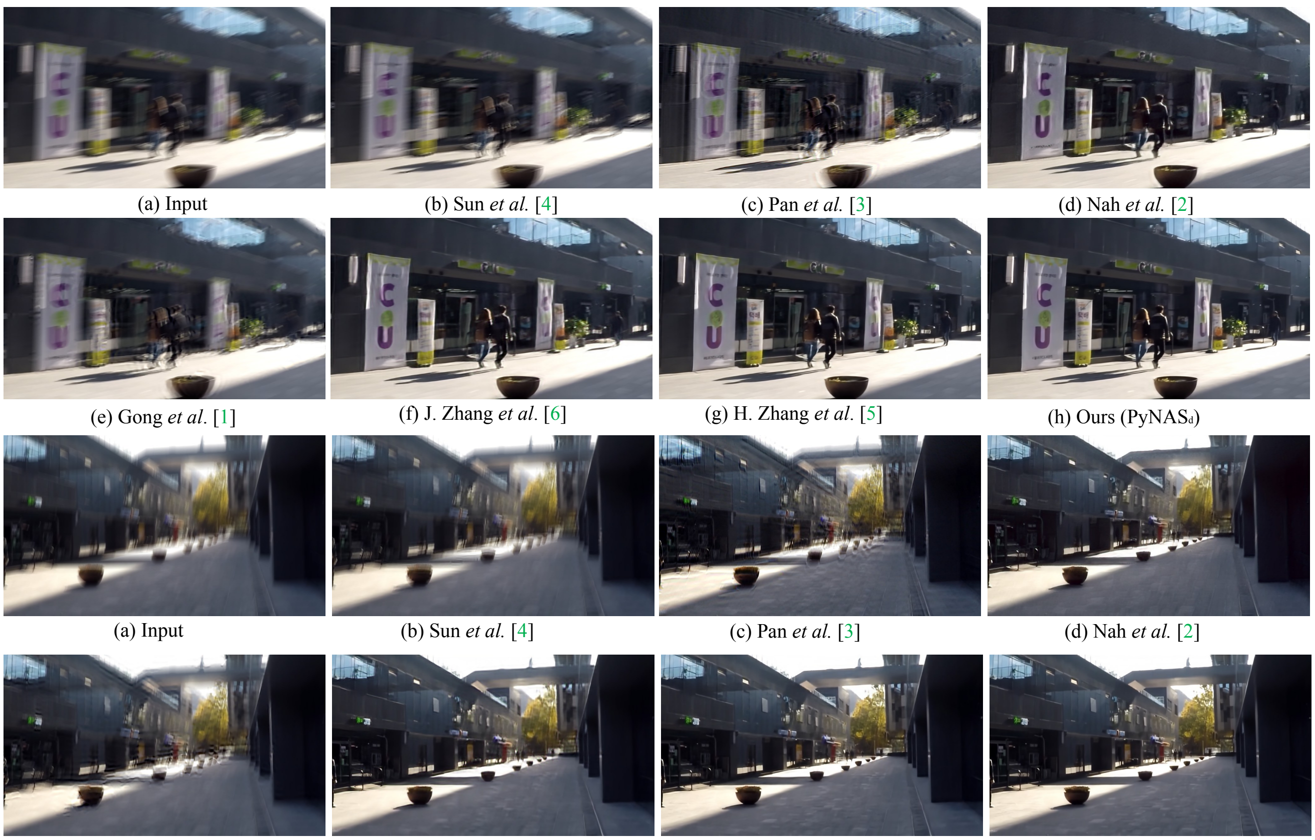
(e) Gong *et al.* [1] (f) J. Zhang *et al.* [6] (g) H. Zhang *et al.* [5] (h) Ours ($PyNAS_d$)



(a) Input (b) Sun *et al.* [4] (c) Pan *et al.* [3] (d) Nah *et al.* [2]

(e) Gong *et al.* [1] (f) J. Zhang *et al.* [6] (g) H. Zhang *et al.* [5] (h) Ours ($PyNAS_d$)

Figure 6. Visual comparison with state-of-the-art Image Deblurring methods on GOPR0871-11-00.



(a) Input (b) Sun *et al.* [4] (c) Pan *et al.* [3] (d) Nah *et al.* [2]

(e) Gong *et al*. [1] (f) J. Zhang *et al*. [6] (g) H. Zhang *et al*. [5] (h) Ours ($PyNAS_d$)

(a) Input (b) Sun *et al.* [4] (c) Pan *et al.* [3] (d) Nah *et al.* [2]

(e) Gong *et al*. [1] (f) J. Zhang *et al*. [6] (g) H. Zhang *et al*. [5] (h) Ours ($PyNAS_d$)

Figure 7. Visual comparison with state-of-the-art Image Deblurring methods on GOPR0881-11-01.