Poster Session
THU-AM-155

Self-supervised Non-uniform Kernel Estimation with Flow-based Motion Prior for Blind Image Deblurring

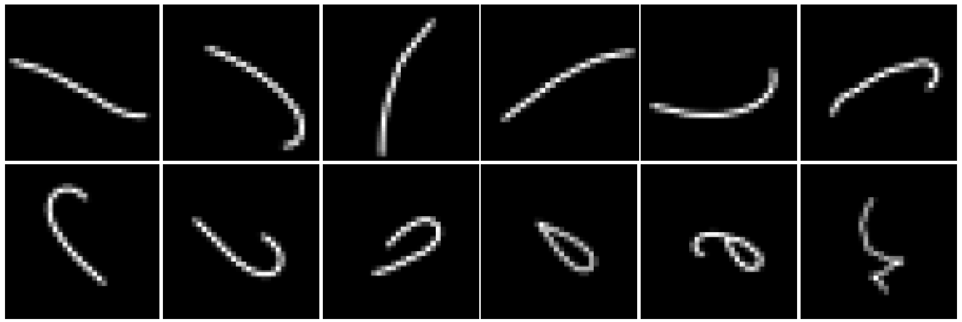
Zhenxuan Fang[1] Fangfang Wu[1]* Weisheng Dong[1] Xin Li[2] Jinjian Wu[1] Guangming Shi[1]
[1]Xidian University [2]West Virginia University
zxfang@stu.xidian.edu.cn wufangfang@xidian.edu.cn wsdong@mail.xidian.edu.cn
xin.li@mail.wvu.edu jinjian.wu@mail.xidian.edu.cn gmshi@xidian.edu.cn

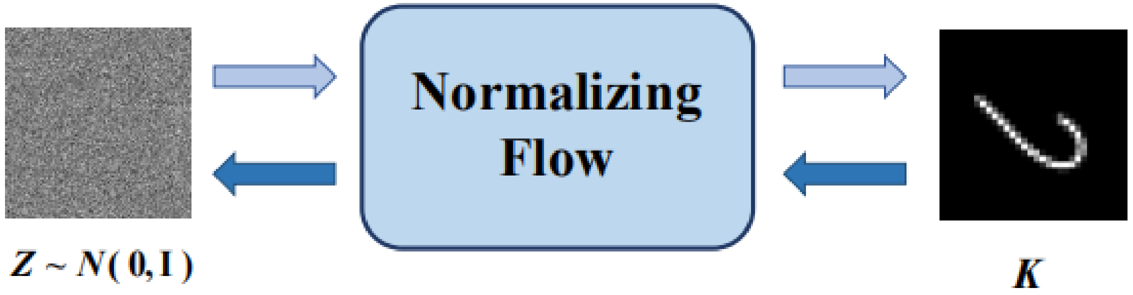
Overview

- Most blind image deblurring methods ignore the prior information about motion blur, and accurate estimation of spatially varying blur kernels is challenging.

Overview

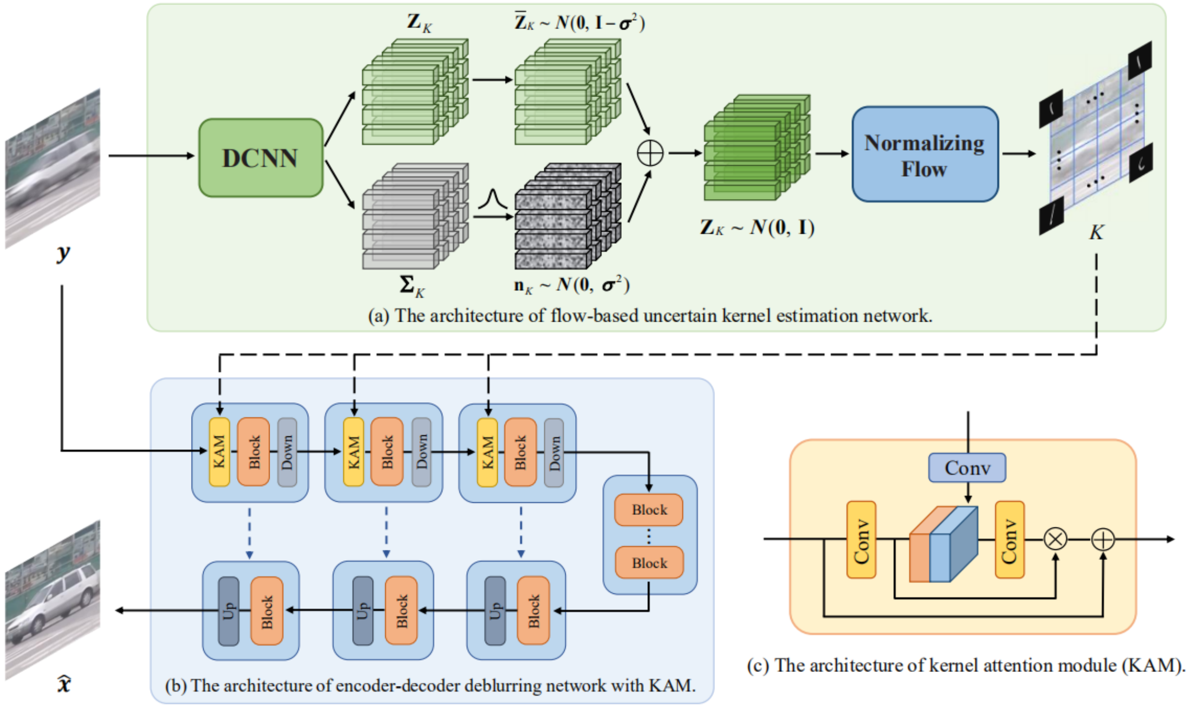
- We propose to represent the non-uniform motion blur kernels in a latent space by normalizing flow. Our latent space approach allows CNNs to predict spatially varying latent codes rather than kernels.





Overview

- We introduce uncertainty learning to the latent code estimation process to improve performance and robustness.
- We tackle the problem of lacking motion kernel ground truth in a self-supervised manner.



(a) The architecture of flow-based uncertain kernel estimation network.

(b) The architecture of encoder-decoder deblurring network with KAM.

(c) The architecture of kernel attention module (KAM).

$$\mathcal{L}_{KE} = \frac{1}{N}\sum_{n=1}^{N} \|\boldsymbol{x}_n \otimes f_{\boldsymbol{\theta}}[\mathbf{G}(\boldsymbol{y}_n)] - \boldsymbol{y}_n\|_1 ,$$

Introduction

- Blind single image deblurring can be mathematically formulated as

$$\boldsymbol{y} = \mathrm{B}(\boldsymbol{x}, \boldsymbol{k}) + \boldsymbol{n},$$

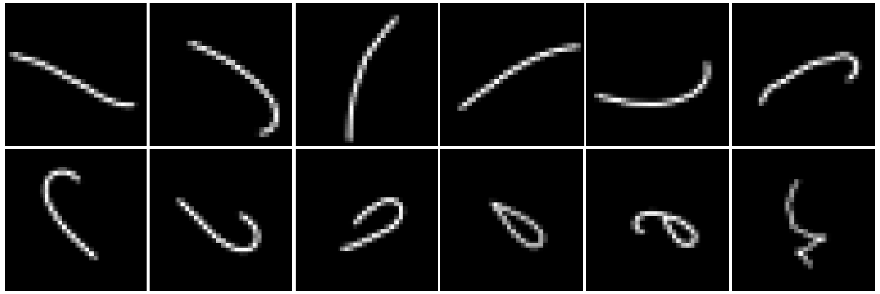
- Existing deep learning-based methods have been proposed for blind image deblurring, but they have limitations

  - The characteristics of blur in real scenarios are complex, accurate estimation of non-uniform blur kernel is challenging.

  - End-to-end methods ignore the information of motion prior.

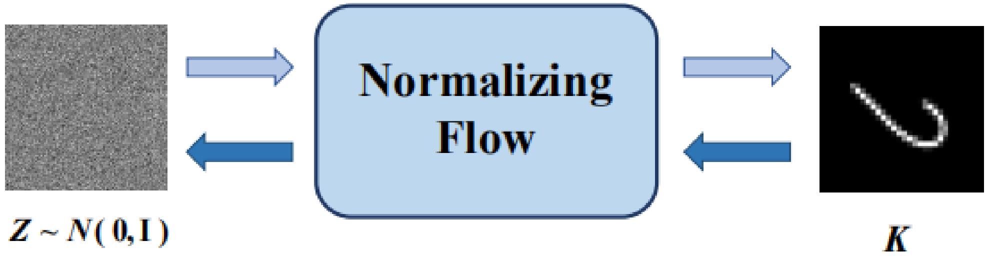
Proposed Method

- Represent the complex motion blur kernel into a simple Gaussian distribution by a normalizing flow.
  - ➢ The simulated motion blur kernels.



  - ➢ The flow-based motion prior model.

$$\mathcal{L}(\mathbf{k};\boldsymbol{\theta}) = -\log p_Z\left(f_{\boldsymbol{\theta}}(\mathbf{k})\right) - \log\left|\det\left(\frac{\partial f_{\boldsymbol{\theta}}(\mathbf{k})}{\partial \mathbf{k}}\right)\right|,$$

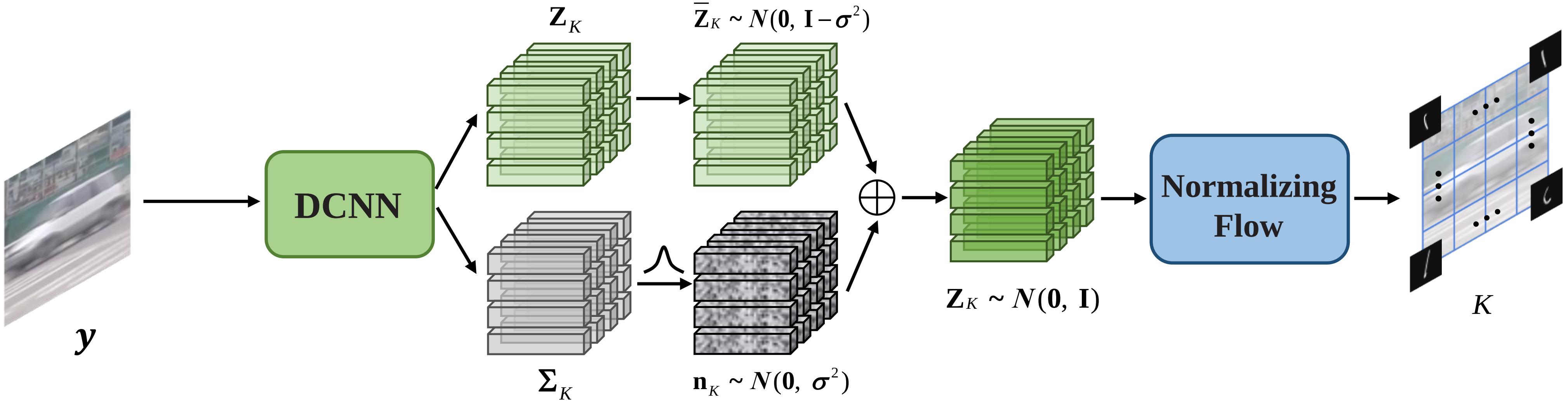


Proposed Method

- Self-supervised Kernel Estimation in Latent Space
  - To overcome the problem of lacking ground truth of blur kernel, we propose to estimate the blur kernel in a self-supervised manner

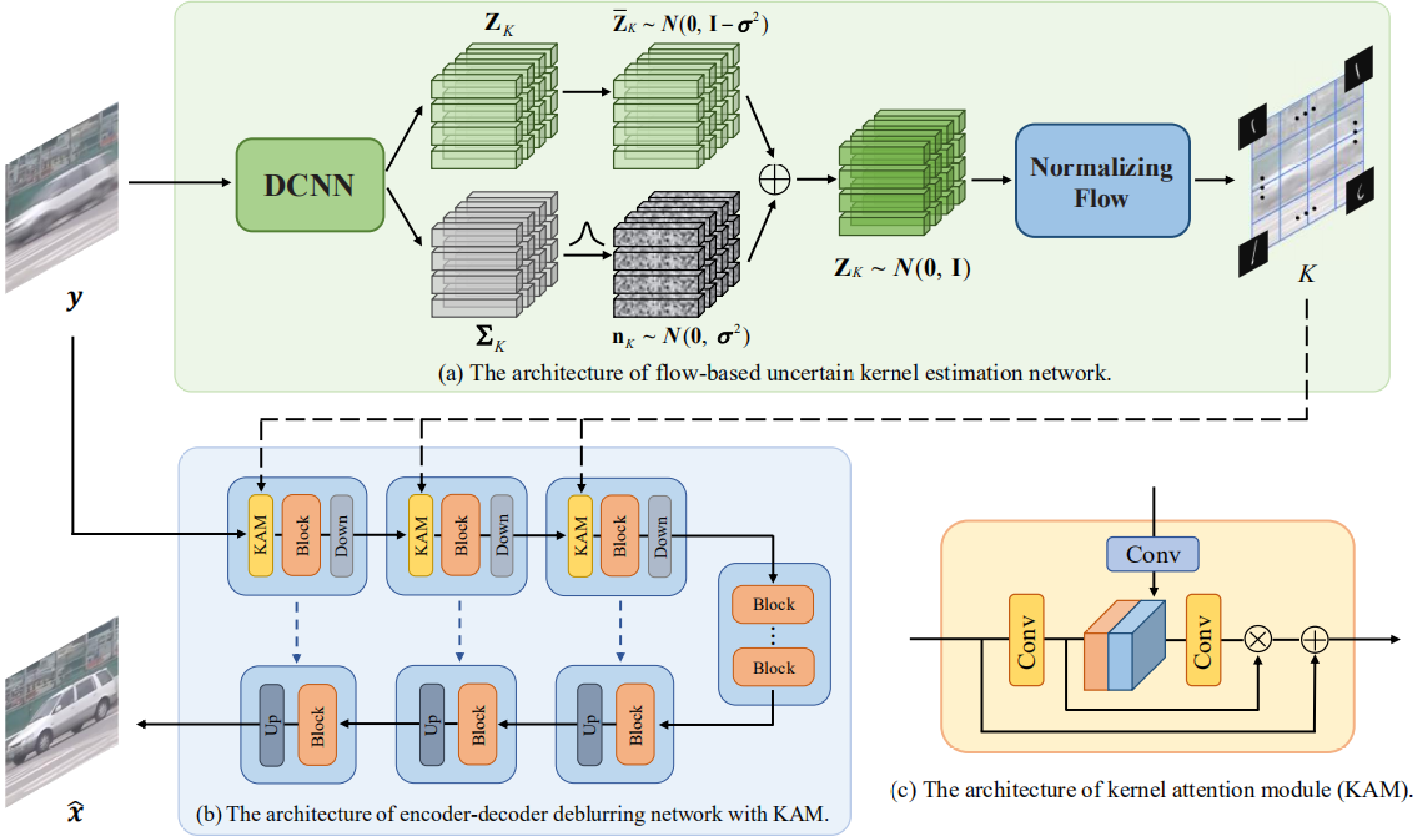
$$\mathcal{L}_{KE} = \frac{1}{N}\sum_{n=1}^{N} \| \boldsymbol{x}_n \otimes f_{\boldsymbol{\theta}}[\mathbf{G}(\boldsymbol{y}_n)] - \boldsymbol{y}_n \|_1$$

  - The architecture of flow-based uncertain kernel estimation network with uncertainty learning



Proposed Method

- Uncertain Flow-based Prior Network (UFPNet)



(a) The architecture of flow-based uncertain kernel estimation network.

(b) The architecture of encoder-decoder deblurring network with KAM.

(c) The architecture of kernel attention module (KAM).

Proposed Method

- Training process

I. Pretrain the normalizing flow model to represent the motion blur kernel into a Gaussian distribution

II. The self-supervise loss is use to pretrain the kernel estimation network

$$\mathcal{L}_{KE} = \frac{1}{N}\sum_{n=1}^{N} \|\boldsymbol{x}_n \otimes f_{\boldsymbol{\theta}}[\mathbf{G}(\boldsymbol{y}_n)] - \boldsymbol{y}_n\|_1$$

III. The PSNR loss is used to train the deblurring network, meanwhile, we use the reblur loss which can be expressed as

$$\mathcal{L}_{reblur} = \frac{1}{N}\sum_{n=1}^{N} \|\mathcal{F}(\boldsymbol{y}_n) \otimes \mathcal{K}(\boldsymbol{y}_n) - \boldsymbol{y}_n\|_1 \,,$$

Experimental Results

JUNE 18-22, 2023 CVPR VANCOUVER, CANADA

- The comparison results on the benchmark datasets

Method	GoPro		HIDE		RealBlur-R		RealBlur-J		Params
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	(M)
DeepDeblur [27]	29.23	0.916	N/A	N/A	32.51	0.841	27.87	0.827	11.7
SRN [38]	30.26	0.934	28.36	0.915	35.66	0.947	28.56	0.867	6.8
DeblurGAN [19]	28.70	0.858	24.51	0.871	33.79	0.903	27.97	0.834	N/A
DeblurGAN-v2 [20]	29.55	0.934	26.61	0.875	35.26	0.944	28.70	0.866	60.9
DBGAN [49]	31.10	0.942	28.94	0.915	N/A	N/A	N/A	N/A	11.6
DMPHN [48]	31.20	0.945	29.09	0.924	35.70	0.948	28.42	0.860	21.7
MT-RNN [31]	31.15	0.945	29.15	0.918	N/A	N/A	N/A	N/A	2.6
SAPHN [36]	31.85	0.948	29.98	0.930	N/A	N/A	N/A	N/A	23.0
MIMO-UNet [7]	32.45	0.957	29.99	0.930	35.54	0.947	27.63	0.837	16.1
MPRNet [47]	32.66	0.959	30.96	0.939	35.99	0.952	28.70	0.873	20.1
HINet [5]	32.71	0.959	30.32	0.932	35.75	0.949	28.17	0.849	88.7
DeepRFT [26]	33.23	0.963	31.42	0.944	35.86	0.950	28.97	0.884	23.0
Stripformer [39]	33.08	0.962	31.03	0.940	36.07	0.952	28.82	0.876	20.0
MSDI-Net [22]	33.28	0.964	31.02	0.940	35.88	0.952	28.59	0.869	135.4
NAFNet [4]	33.69	0.967	31.32	0.943	35.50	0.953	28.32	0.857	67.8
UFPNet (ours)	**34.06**	**0.968**	**31.74**	**0.947**	**36.25**	**0.953**	**29.87**	**0.884**	80.3

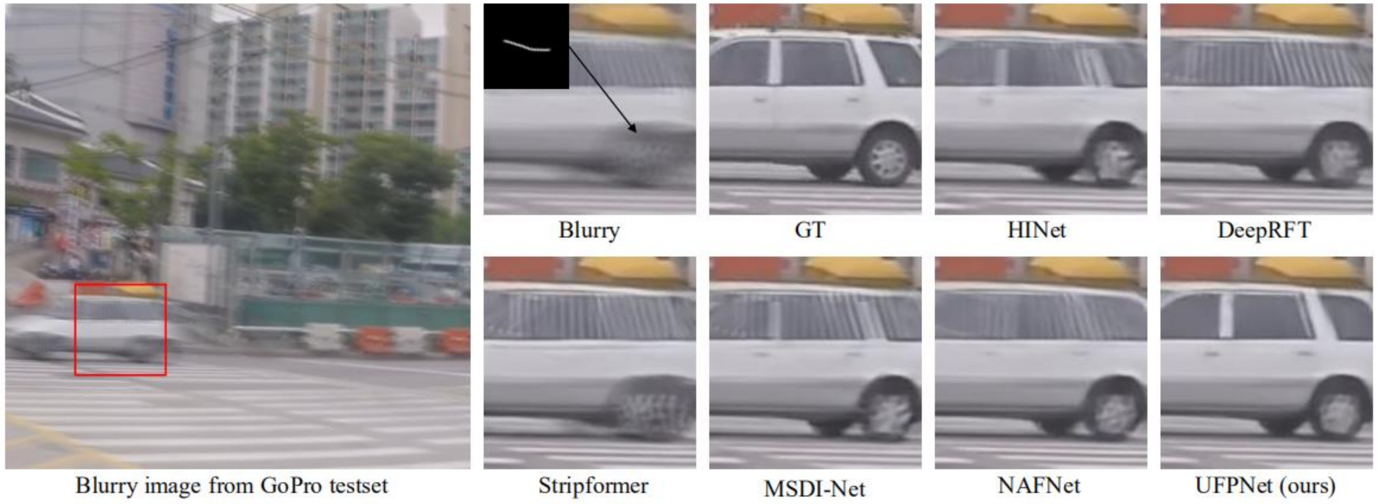
(1) The models are trained on the GoPro dataset

Method	RealBlur-R		RealBlur-J	
	PSNR	SSIM	PSNR	SSIM
DeblurGAN-v2 [20]	36.44	0.935	29.69	0.870
SRN [38]	38.65	0.965	31.38	0.909
MIMO-UNet [7]	N/A	N/A	31.92	0.919
MPRNet [47]	39.31	0.972	31.76	0.922
DeepRFT [26]	39.84	0.972	32.19	0.931
Stripformer [39]	39.84	0.974	32.48	0.929
UFPNet (ours)	**40.61**	**0.974**	**33.35**	**0.934**

(2) The models are trained on the RealBlur dataset

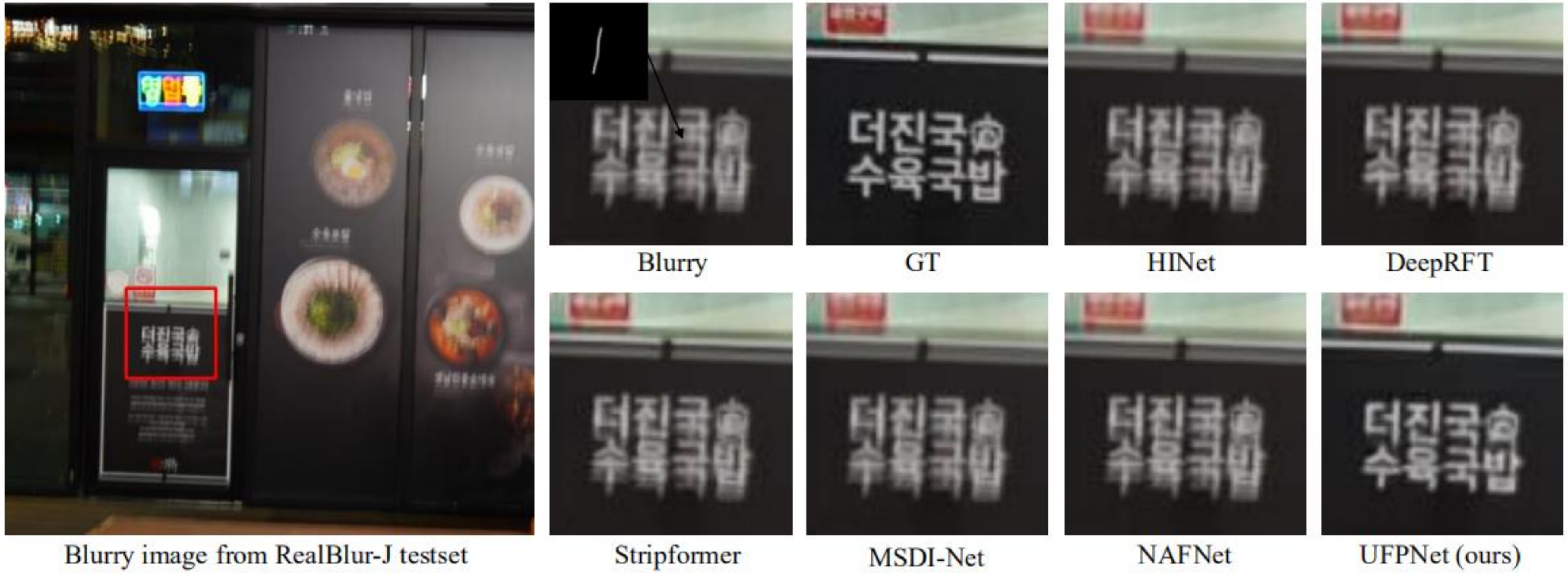
Experimental Results

- Visual comparison to other methods



Experimental Results

- Visual comparison to other methods



Blurry image from RealBlur-J testset

Experimental Results

- Ablation Studies

Whyte et al. [40]	Proposed KE-Net Baseline	Proposed KE-Net Flow prior	Proposed KE-Net UL	GoPro PSNR	GoPro SSIM	HIDE PSNR	HIDE SSIM	RealBlur-R PSNR	RealBlur-R SSIM	RealBlur-J PSNR	RealBlur-J SSIM
				33.69	0.967	31.32	0.943	35.50	0.951	28.32	0.857
✓				33.74	0.967	31.38	0.944	35.61	0.951	28.79	0.863
	✓			33.78	0.967	31.45	0.945	35.78	0.952	29.13	0.869
	✓	✓		33.83	0.967	31.53	0.946	35.91	0.952	29.32	0.872
	✓	✓	✓	**34.06**	**0.968**	**31.74**	**0.947**	**36.25**	**0.953**	**29.87**	**0.884**

Whyte et al. [40]	Proposed KE-Net Baseline	Proposed KE-Net Flow prior	Proposed KE-Net UL	PSNR	SSIM
✓				41.63	0.989
	✓			43.90	0.993
	✓	✓		44.56	0.994
	✓	✓	✓	**45.92**	**0.996**

Method	KE	GoPro PSNR	GoPro SSIM	HIDE PSNR	HIDE SSIM
MIMO-UNet [7]	×	32.45	0.957	29.99	0.930
	✓	**32.83**	**0.959**	**30.16**	**0.931**
MPRNet [47]	×	32.66	0.959	30.96	0.939
	✓	**33.04**	**0.967**	**31.13**	**0.941**
NAFNet [4]	×	33.69	0.964	31.32	0.943
	✓	**34.06**	**0.968**	**31.74**	**0.947**

Conclusions

- In this paper, we propose to represent the motion blur kernels in a latent space by a normalizing flow and designing CNNs to predict spatially varying latent codes instead of motion kernels.

- To further improve the accuracy and robustness of kernel estimation, we introduce uncertainty learning into the process of estimating latent codes.

- To address the issue of the lack of ground truth about the non-uniform motion kernel in real-world images, we tackle the training set generation in a self-supervised manner.

- Extensive experimental results on benchmark datasets show that the proposed method significantly outperforms existing state-of-the-art methods and demonstrated excellent generalization performance from GoPro to other real-world blur datasets.

THANK YOU FOR WATCHING！