Blur2Blur

Blur Conversion for Unsupervised Image Deblurring on Unknown Domains

CVPR
SEATTLE, WA
JUNE 17-21, 2024


Bang-Dang Pham[1]


Cuong Pham[1,3]


Phong Tran[2]


Rang Nguyen[1]


Anh Tran[1]


Minh Hoai[1,4]

[1]VinAI Research, Vietnam [2]MBZUAI, UAE [3]Posts & Telecommunications Inst. of Tech., Vietnam [4]University of Adelaide, Australia

https://zero1778.github.io/blur2blur/

Problem & Motivation

How can we

deblur blurry images captured by any specific camera 📷?

Problem & Motivation

- Data-Driven Approaches
  - Supervised Deblurring

✅ High-quality restored results

✅ Consistent and Reliable

❌ Need paired data

❌ Overfit blur kernel space

❌ Underperform on unseen-blur

  - Camera-Specific Deblurring

✅ Learn well camera blur kernel

❌ Require "expensive" setups (beam splitter, geometrical alignment,...)

- Unsupervised Approaches

✅ No need paired data

❌ Cannot restore complex blur patterns

Problem & Motivation

- Data-Driven Approaches
    - Supervised Deblurring

✅ High-quality restored results

✅ Consistent and Reliable

❌ Need paired data

❌ Overfit blur kernel space

❌ Underperform on unseen-blur

    - Camera-Specific Deblurring

✅ Learn well camera blur kernel

❌ Require “expensive” setups (beam splitter, geometrical alignment,...)

- Unsupervised Approaches

✅ No need paired data

❌ Cannot restore complex blur patterns

Our Objective

- Data-Driven Approaches
  - Supervised Deblurring

    ✅ High-quality restored results — ❌ Need paired data

    ✅ Consistent and Reliable — ❌ Overfit blur kernel space

    ❌ Underperform on unseen-blur
  - Camera-Specific Deblurring

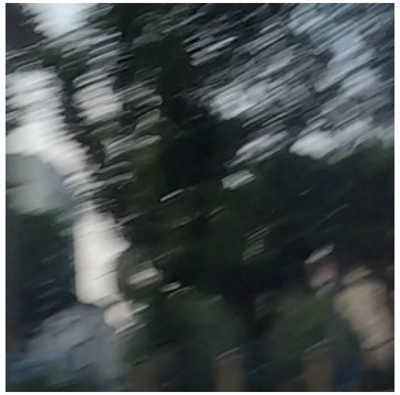
    ✅ Learn well camera blur kernel — ❌ Require “expensive” setups (beam splitter, geometrical alignment,...)
- Unsupervised Approaches

  ✅ No need paired data — ❌ Cannot restore complex blur patterns

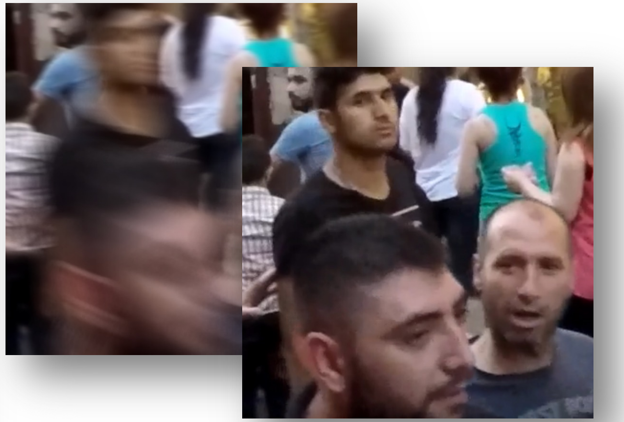
Proposed Method

We propose a novel framework called **Blur2Blur** –
converting images with Unknown Blur kernel into Known Blur kernel version, effectively deblurred by a supervised model while preserving original content

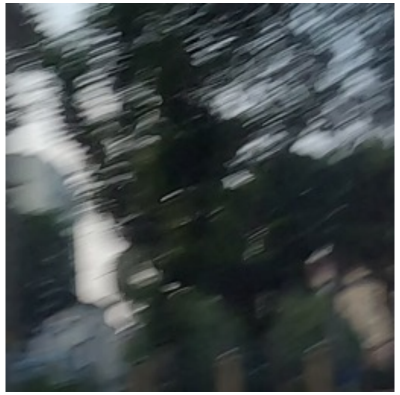
Unknown Blur



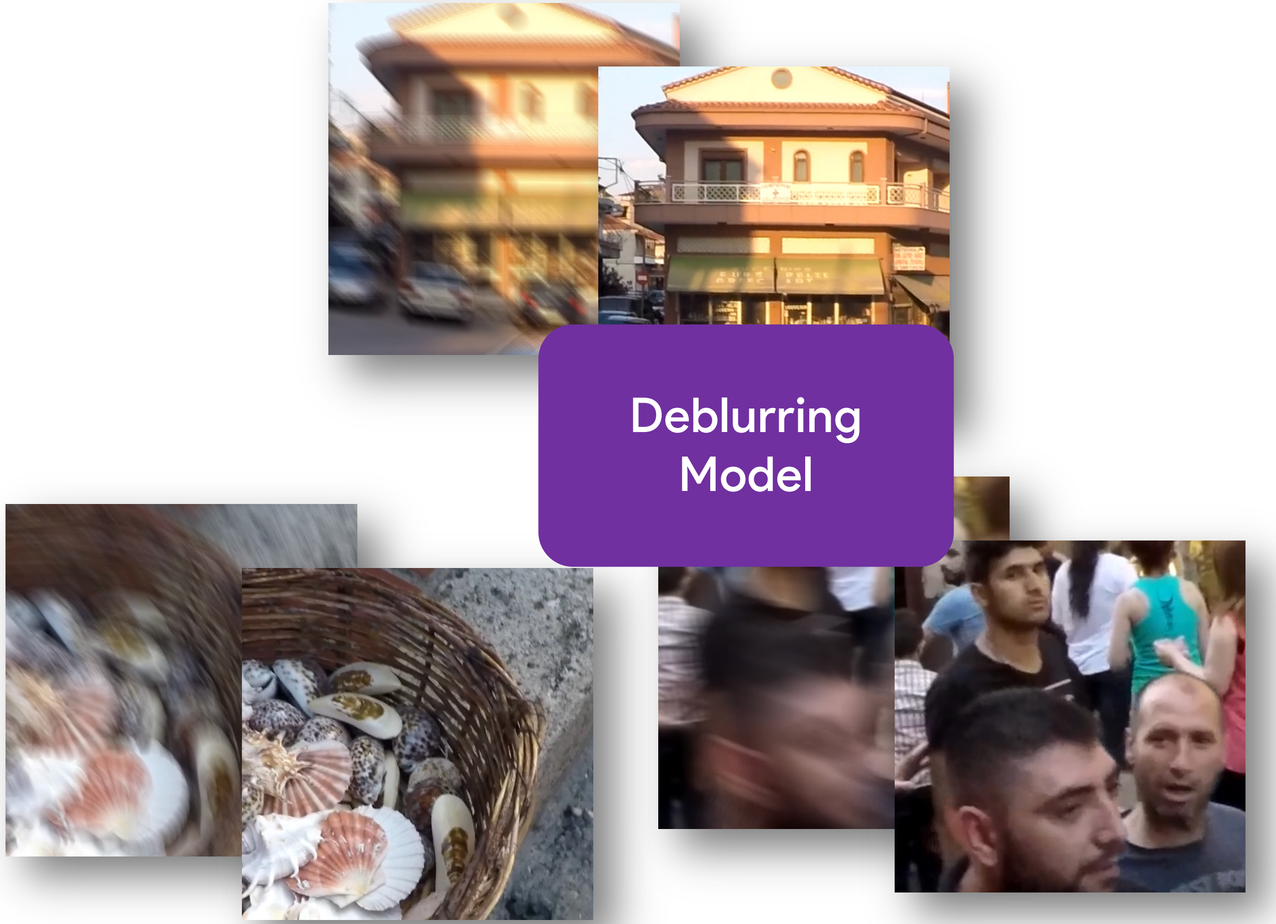
If we use another **known paired data**



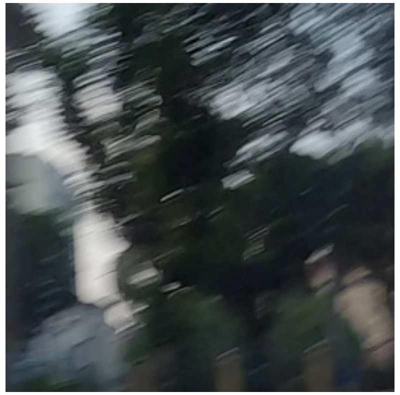
Unknown Blur



If we use another **known paired data** to train deblurring model



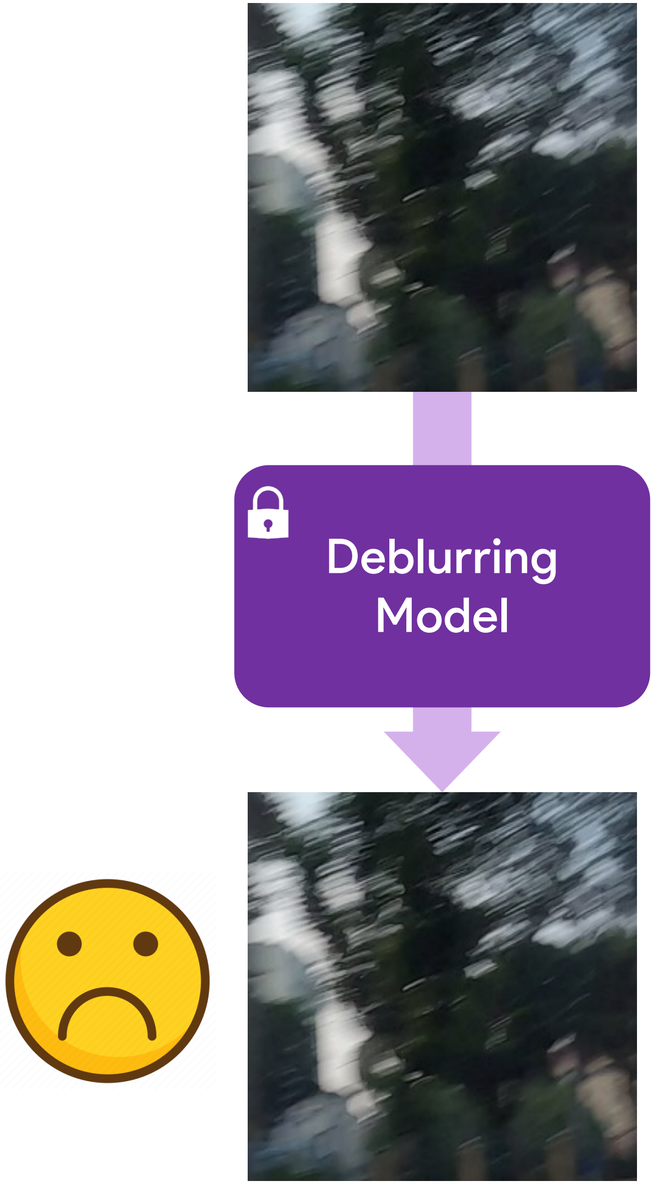
Unknown Blur



If we use another known paired data to train deblurring model

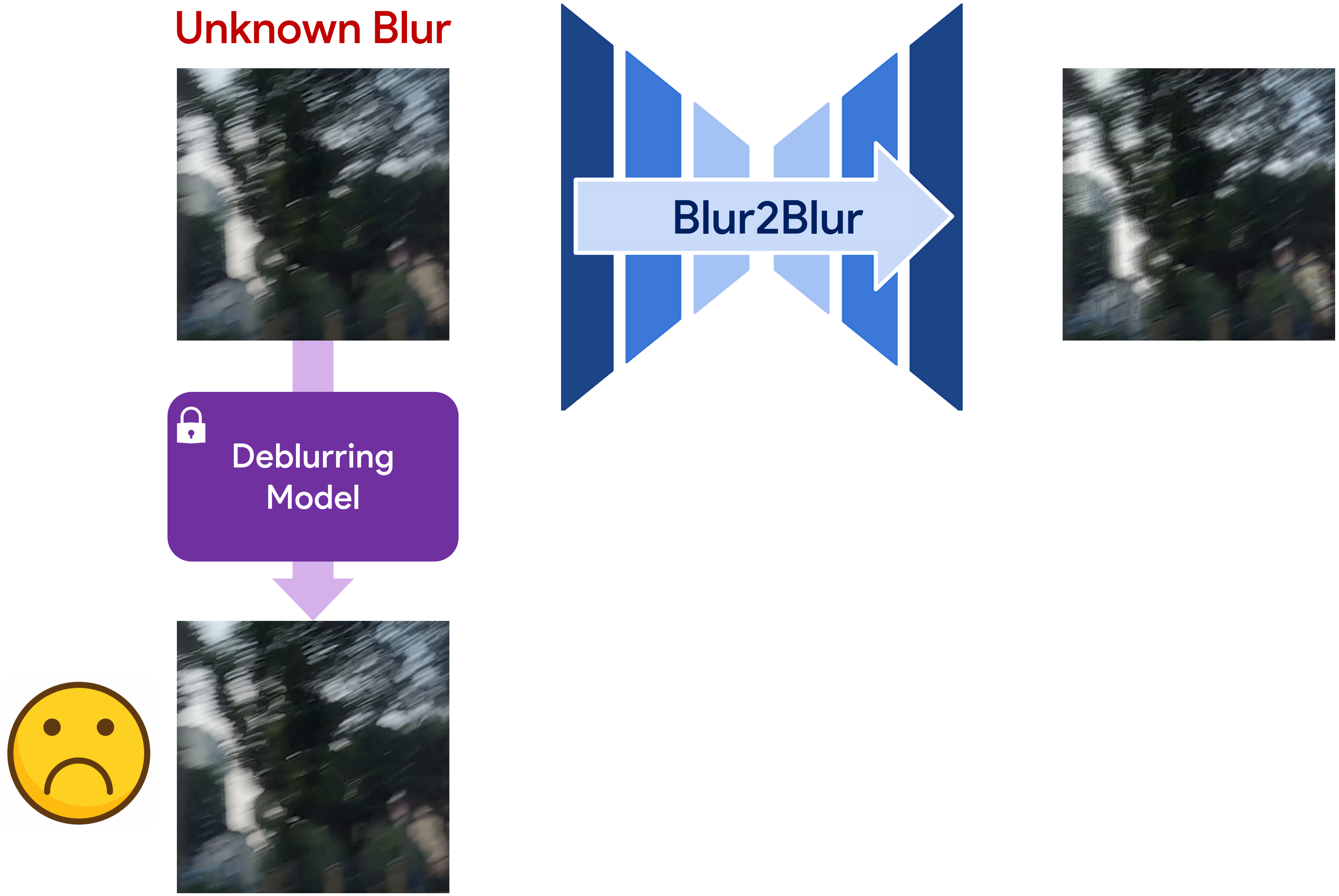
Deblurring Model

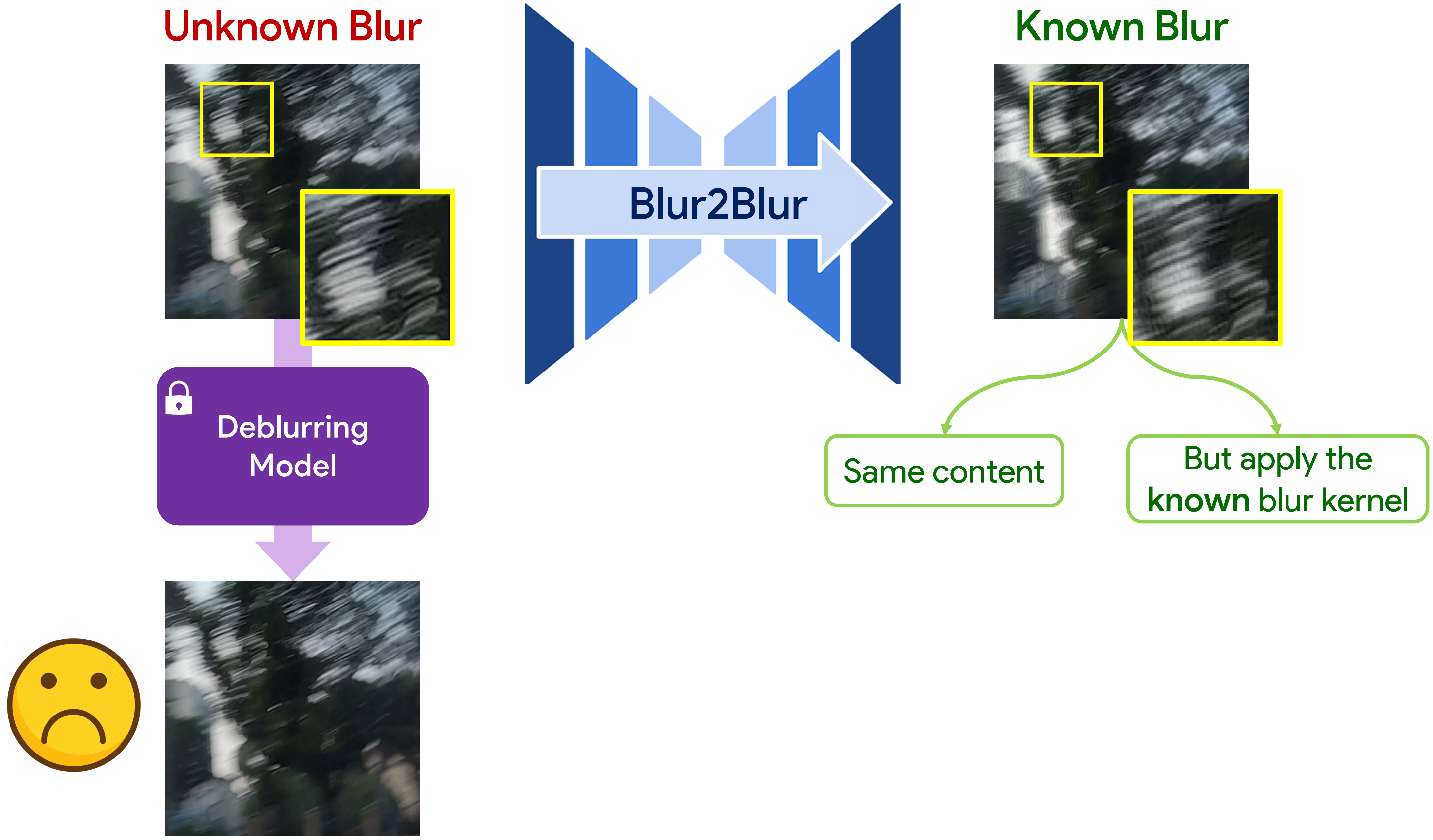
Unknown Blur

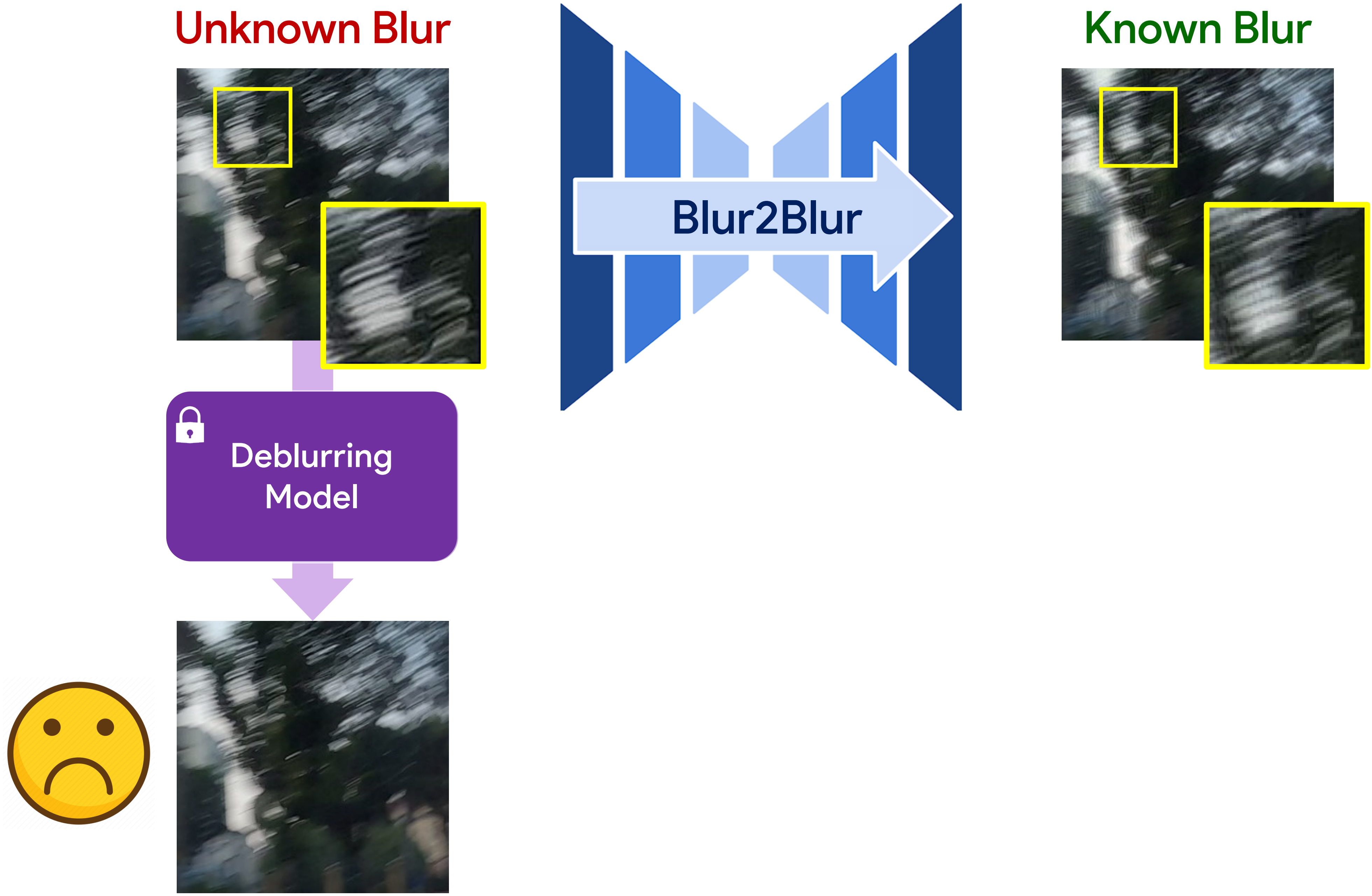


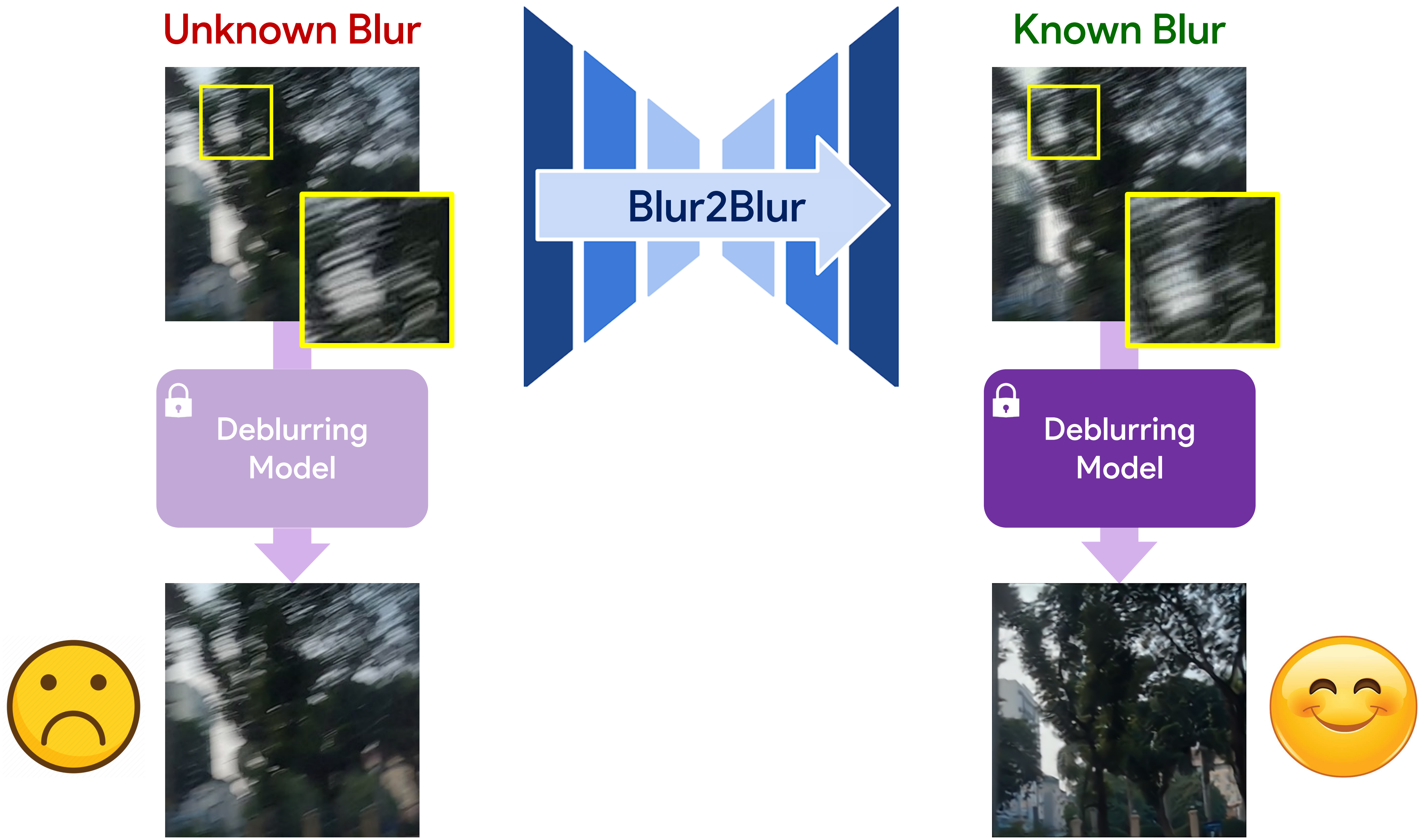
use another known paired data to train deblurring model

FAILED to deblur the unknown blur kernel

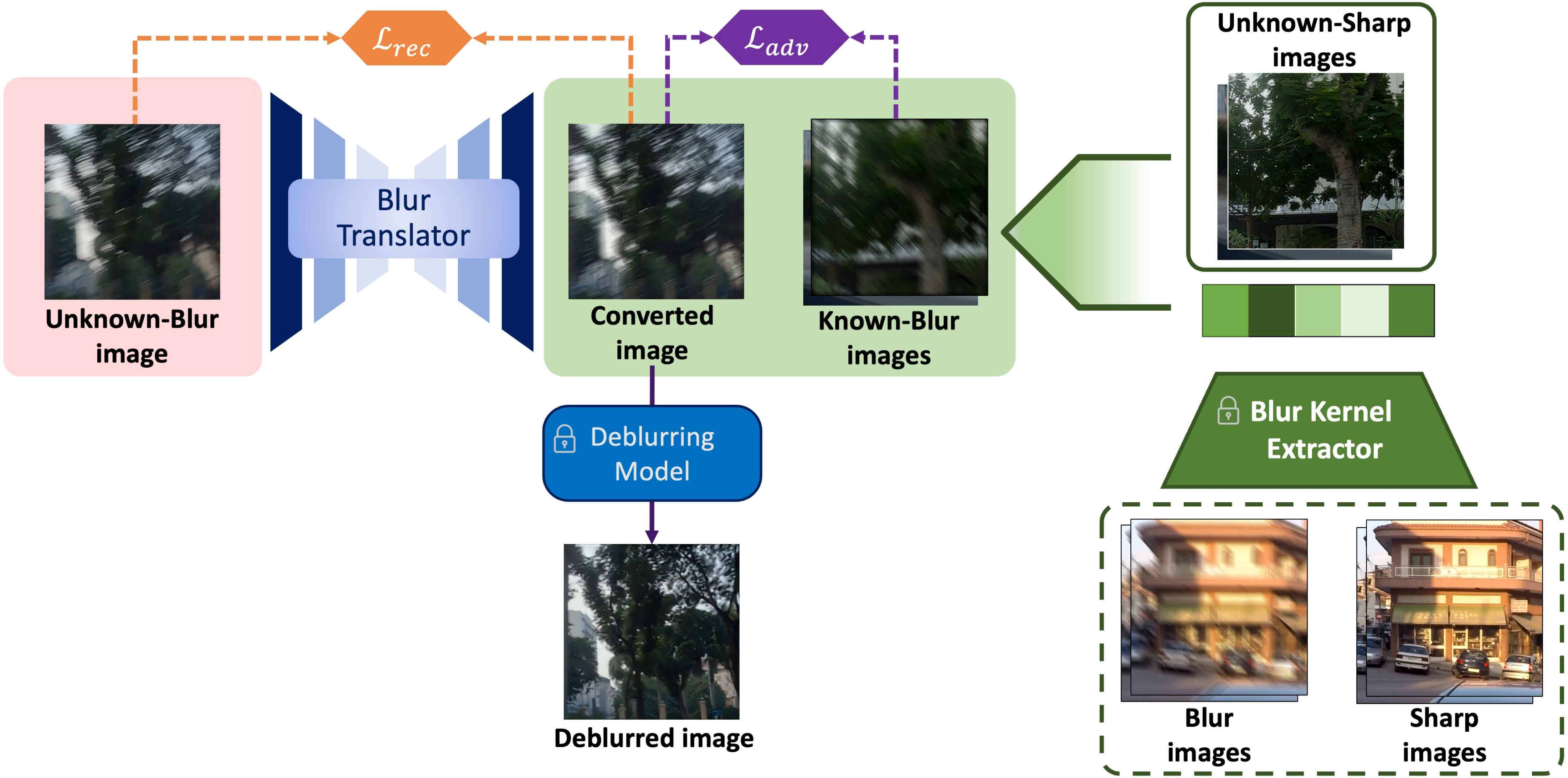

Unknown Blur
Blur2Blur
Deblurring
Model


Unknown Blur
Deblurring
Model
Blur2Blur
Known Blur
Same content
But apply the
known blur kernel

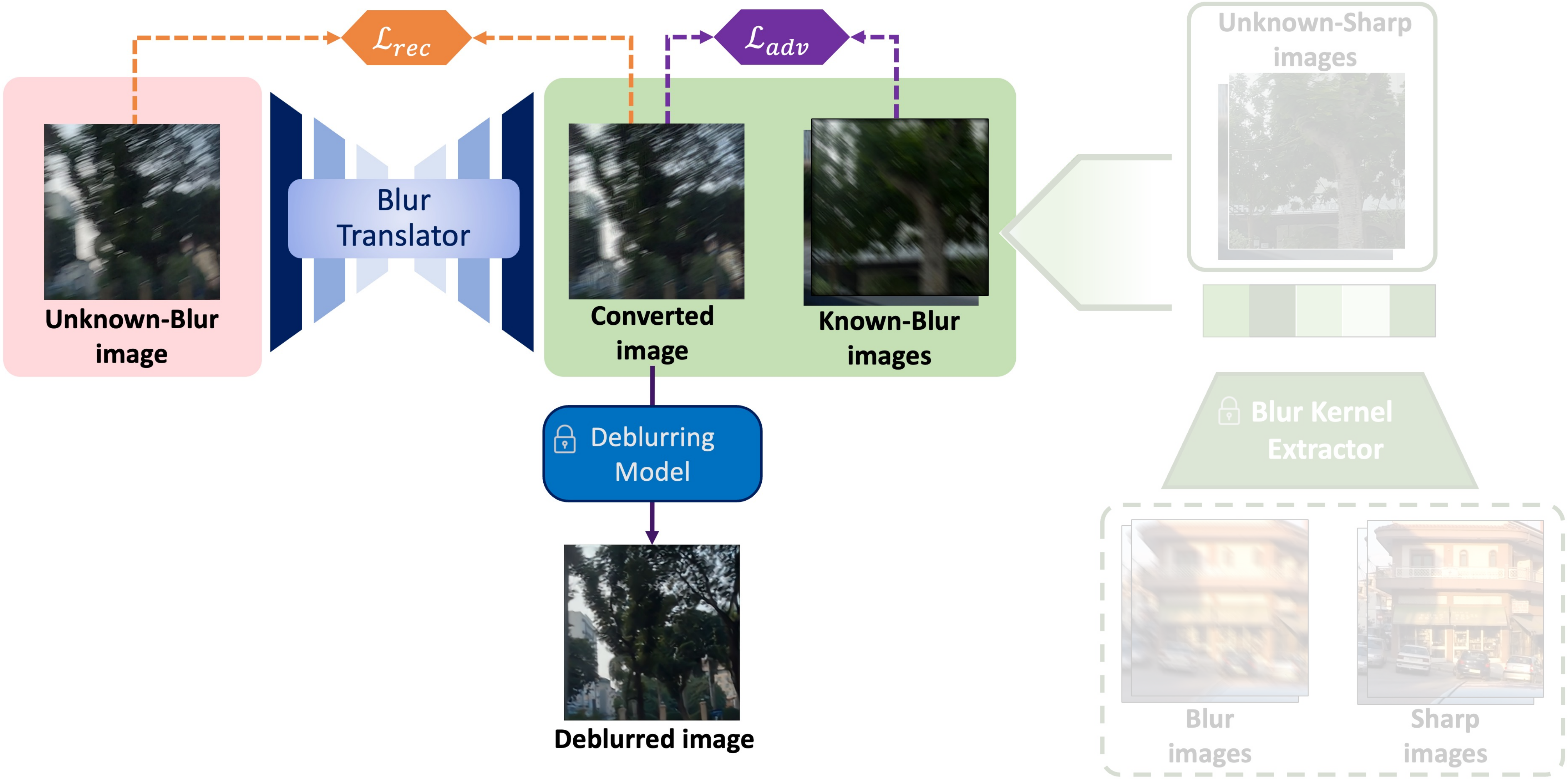

Unknown Blur
Blur2Blur
Known Blur
Deblurring
Model


Unknown Blur
Blur2Blur
Known Blur
Deblurring Model
Deblurring Model

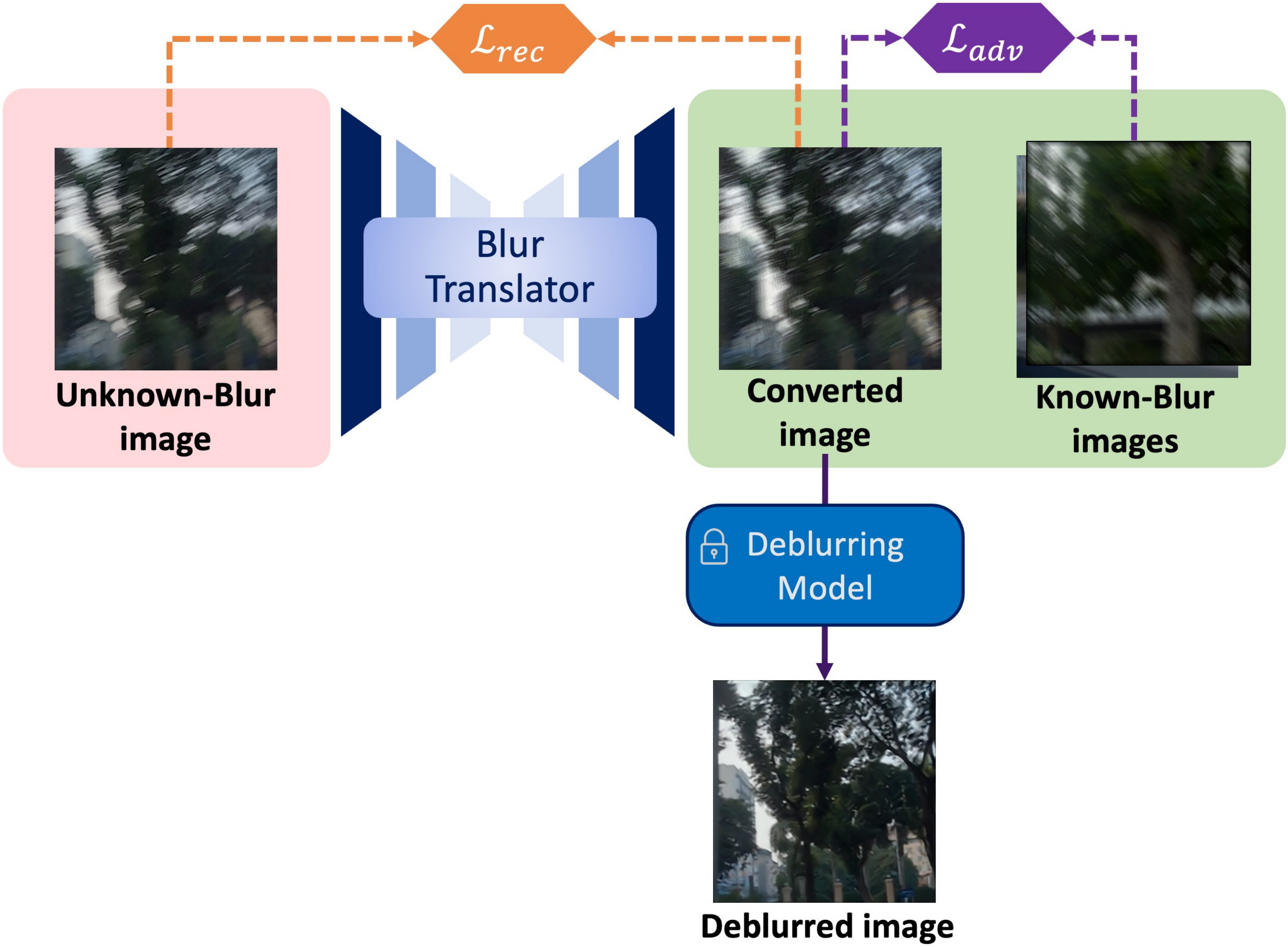
How it works


$\mathcal{L}_{rec}$
$\mathcal{L}_{adv}$
Unknown-Sharp images
Blur Translator
Unknown-Blur image
Converted image
Known-Blur images
Blur Kernel Extractor
Deblurring Model
Deblurred image
Blur images
Sharp images

How it works


$\mathcal{L}_{rec}$
$\mathcal{L}_{adv}$
Unknown-Sharp images
Unknown-Blur image
Blur Translator
Converted image
Known-Blur images
Deblurring Model
Blur Kernel Extractor
Deblurred image
Blur images
Sharp images

How it works



To generate Converted Image

- Preserved input content $\mathcal{L}_{rec}$

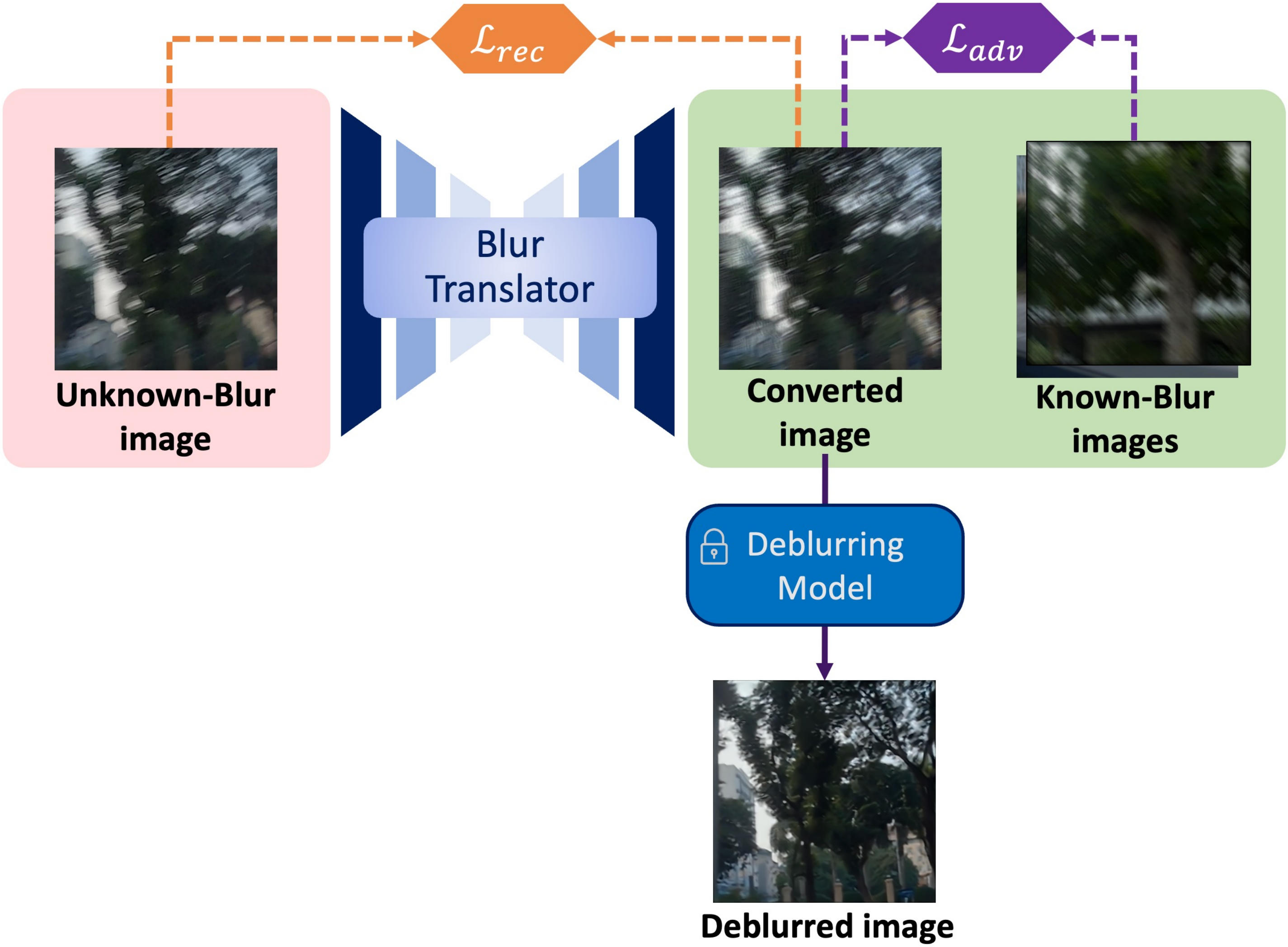
$$\mathcal{L}^{G}_{rec}(G) = \frac{1}{M}\sum_{i=1}^{M}\frac{1}{t_i}\mathbb{E}_{y_i\sim\mathcal{B}}[||\phi(y_i) - \phi(G(y_i))||_1]$$

where M is the number of levels
y_i is the input image at scale level i
$\phi(.)$ is a pretrained feature extractor

- ➢ Perceptual Loss
- ➢ Adopt multi-scale architecture backbone (MIMO-Unet[1])

[1] *Sung-Jin Cho et al. Rethinking Coarse-to-Fine Approach in Single Image Deblurring. ICCV, 2021*

How it works



To generate Converted Image

- Preserved input content $\mathcal{L}_{rec}$

$$\mathcal{L}^{G}_{rec}(G) = \frac{1}{M}\sum_{i=1}^{M}\frac{1}{t_i}\mathbb{E}_{y_i\sim\mathcal{B}}[||\phi(y_i) - \phi(G(y_i))||_1]$$

- Applied Known-blur kernel $\mathcal{L}_{adv}$

$$\mathcal{L}_{adv}(G, D) = \mathbb{E}_{y\sim\mathcal{K}}[\log D(y)] + \mathbb{E}_{y\sim\mathcal{B}}[\log(1 - D(G(y)))].$$

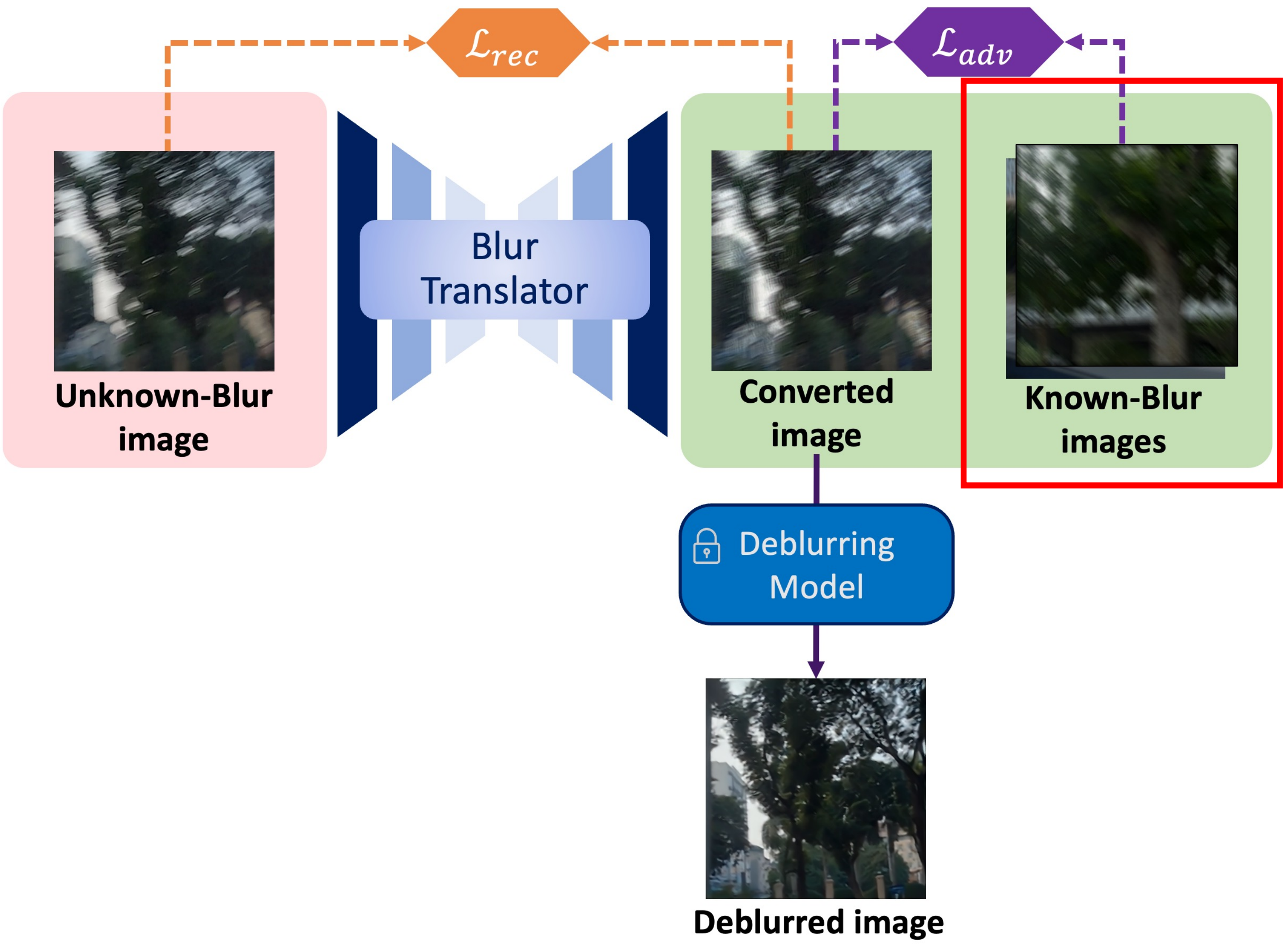
- Penalty Gradient Regularization

$$\mathcal{L}^{D}_{grad}(D) = \mathbb{E}_{\hat{y}\sim\hat{\mathcal{B}}}[(\|\nabla_{\hat{y}} D(\hat{y})\|_2 - 1)^2]$$

- Total Loss

$$\mathcal{L}^{G}_{total}(G, D) = \mathcal{L}_{adv}(G, D) + \lambda_{rec}\mathcal{L}_{rec}(G)$$
$$\mathcal{L}^{D}_{total}(G, D) = -\mathcal{L}_{adv}(G, D) + \lambda_{grad}\mathcal{L}_{grad}(D)$$

How it works

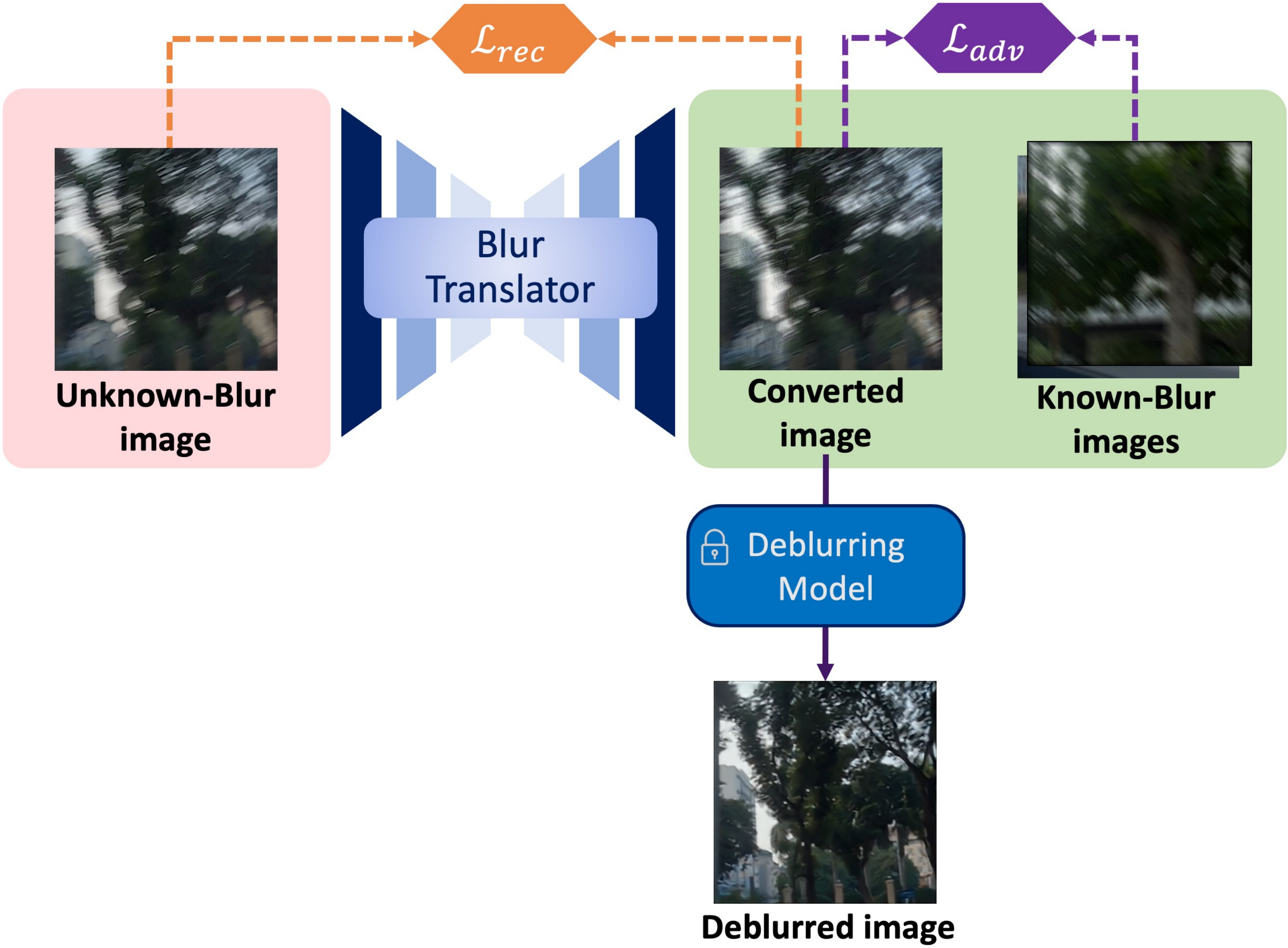


However...

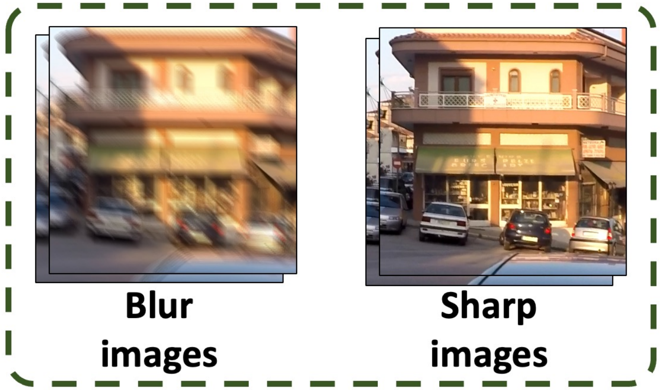
- Known-Blur images from another dataset could have:
  - ❖ Color distribution gap
  - ❖ Image resolution difference
  - ❖ Device-dependent noise pattern

➡ Negatively affect to Discriminator and Blur Translator

How it works

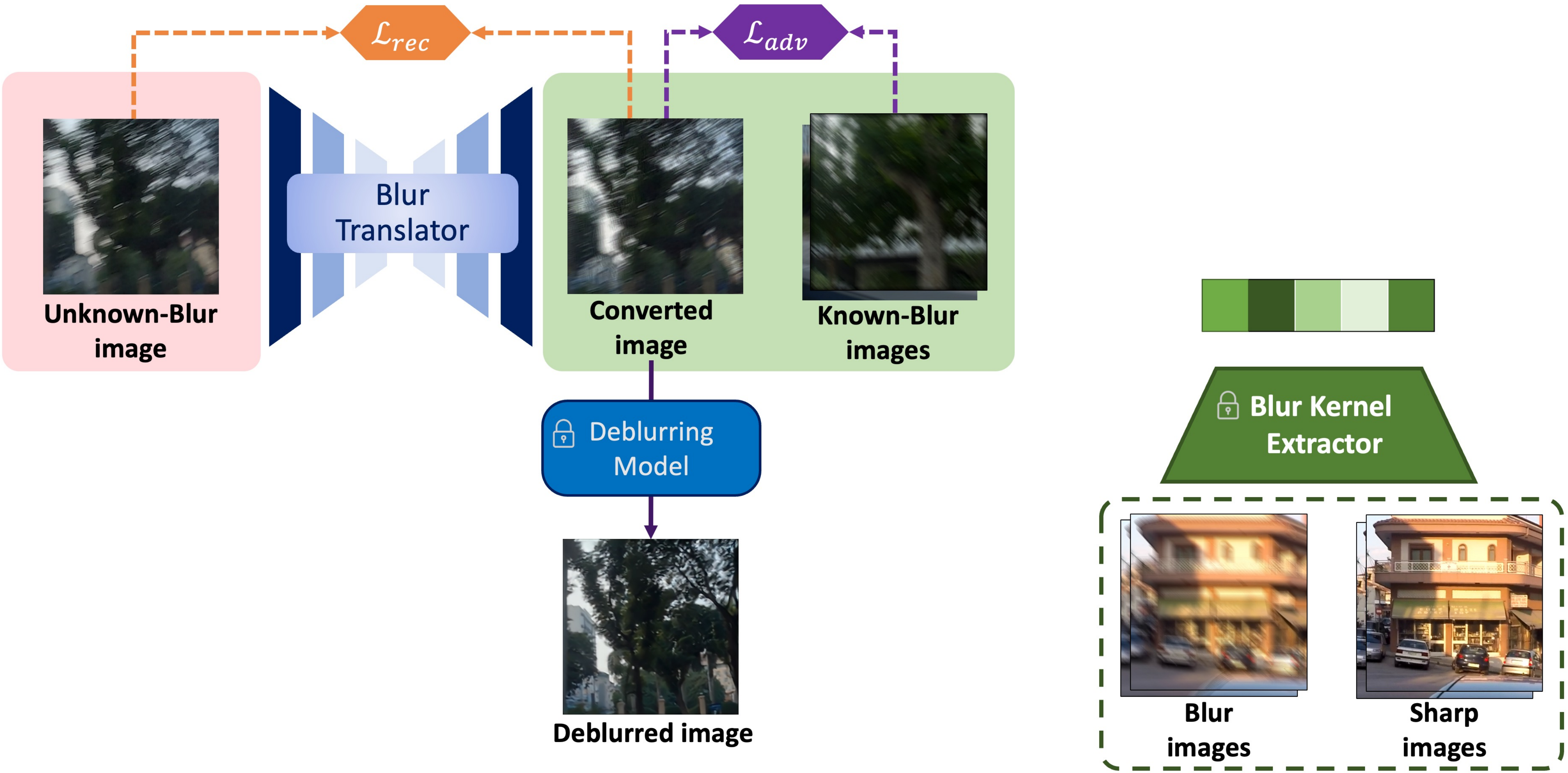


$\mathcal{L}_{rec}$
$\mathcal{L}_{adv}$
Blur Translator
Unknown-Blur image
Converted image
Known-Blur images
Deblurring Model
Deblurred image



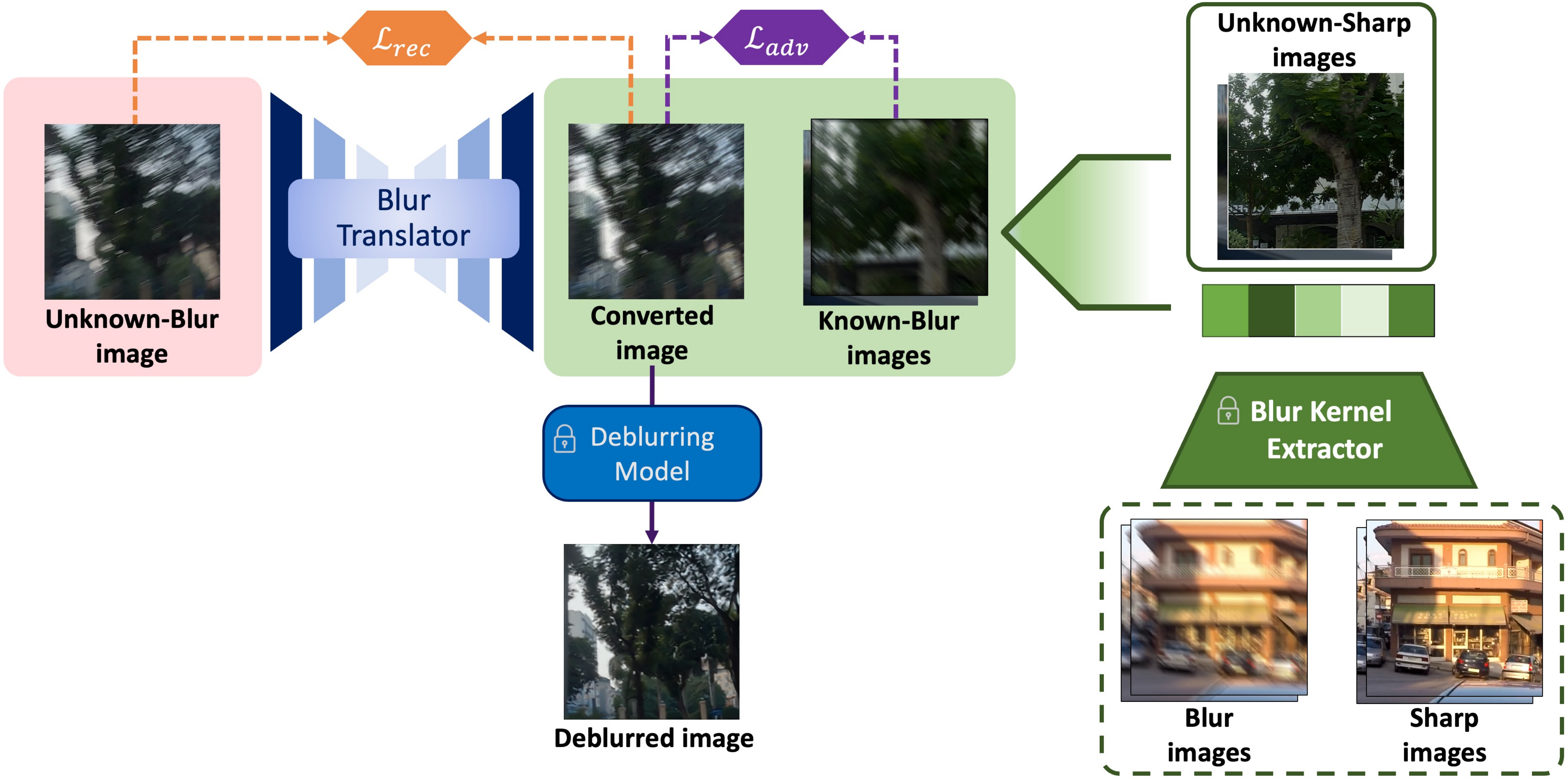
Blur images
Sharp images

How it works



$\mathcal{L}_{rec}$
$\mathcal{L}_{adv}$
Blur Translator
Unknown-Blur image
Converted image
Known-Blur images
Deblurring Model
Deblurred image
Blur Kernel Extractor
Blur images
Sharp images

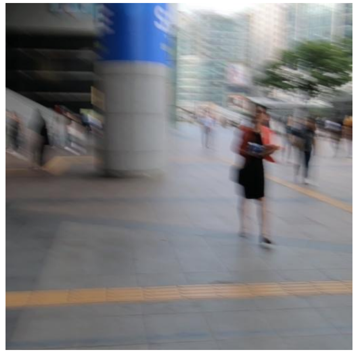
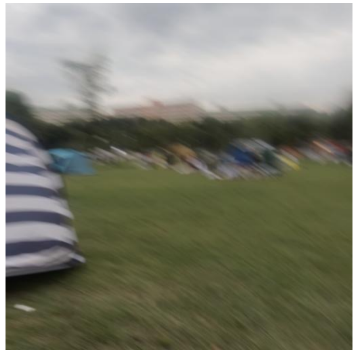
How it works


$\mathcal{L}_{rec}$
$\mathcal{L}_{adv}$
Unknown-Blur image
Blur Translator
Converted image
Known-Blur images
Unknown-Sharp images
Deblurring Model
Deblurred image
Blur Kernel Extractor
Blur images
Sharp images

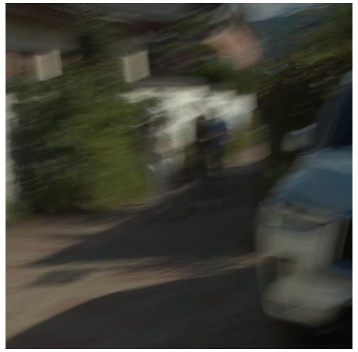
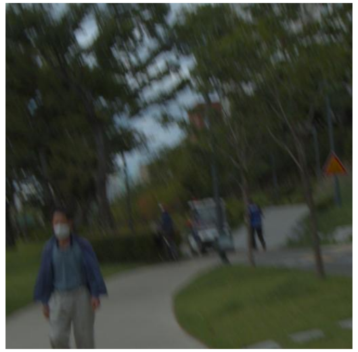
Experiments ~ Many-to-One

Unknown Blur

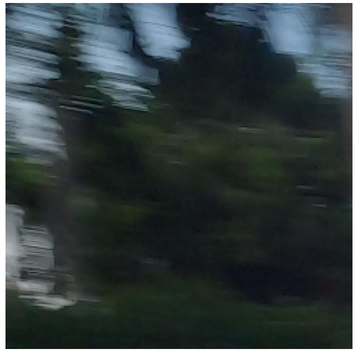
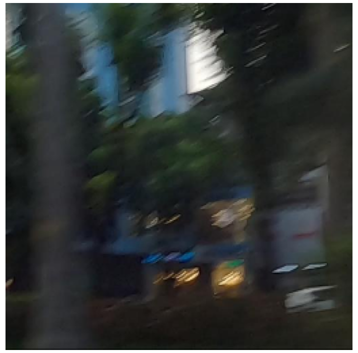
REDS
(Synthetic)

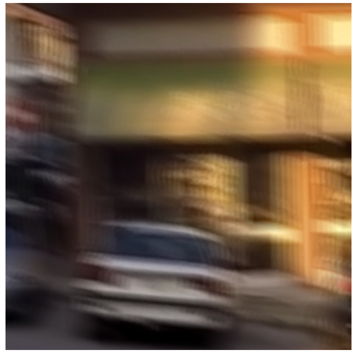
RSBlur
(Real)

RB2V
(Real)

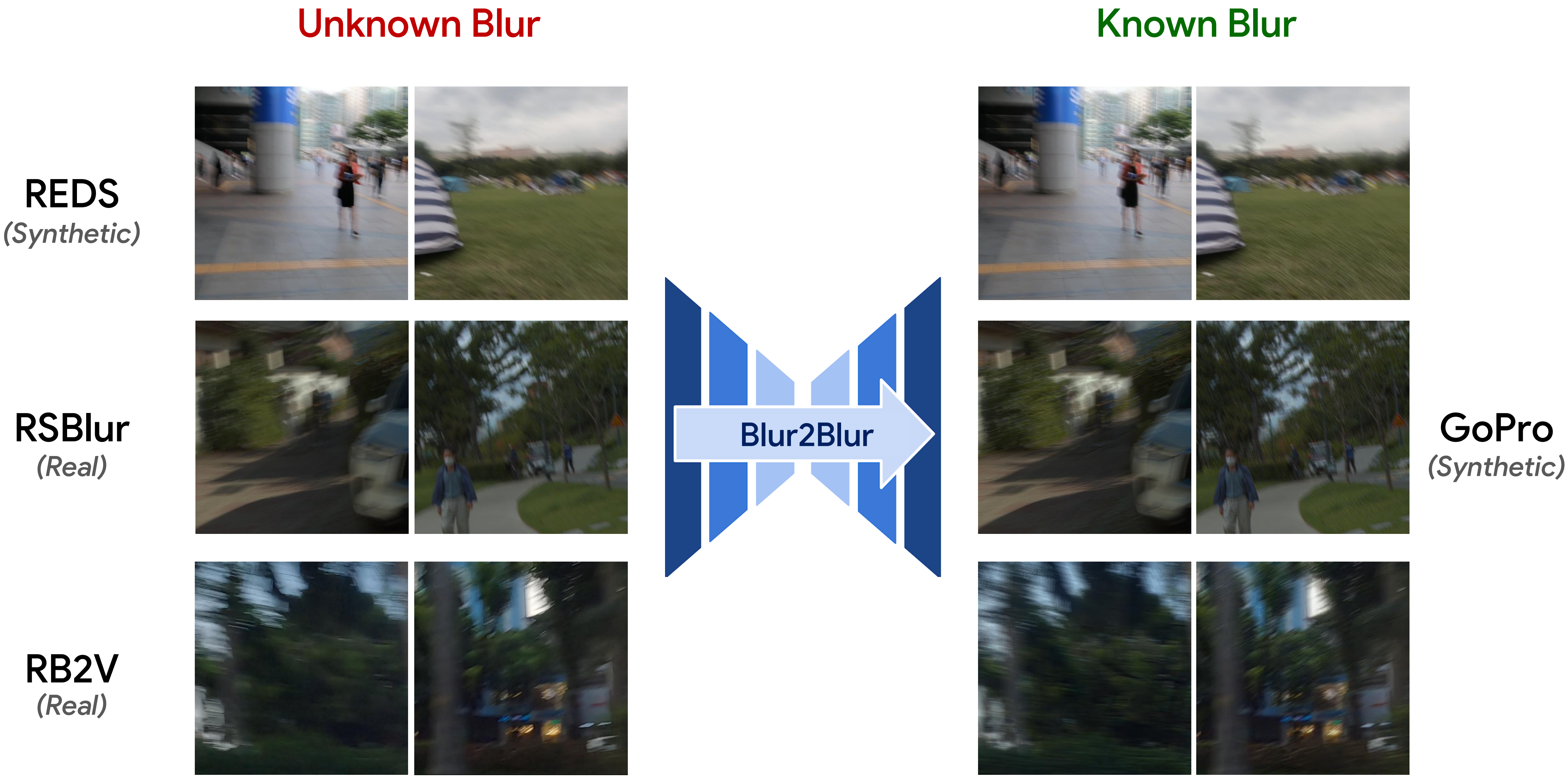
Known Blur



GoPro
(Synthetic)

(Pause and Zoom for best view)

Experiments ~ Many-to-One



(Pause and Zoom for best view)

Experiments ~ Many-to-One

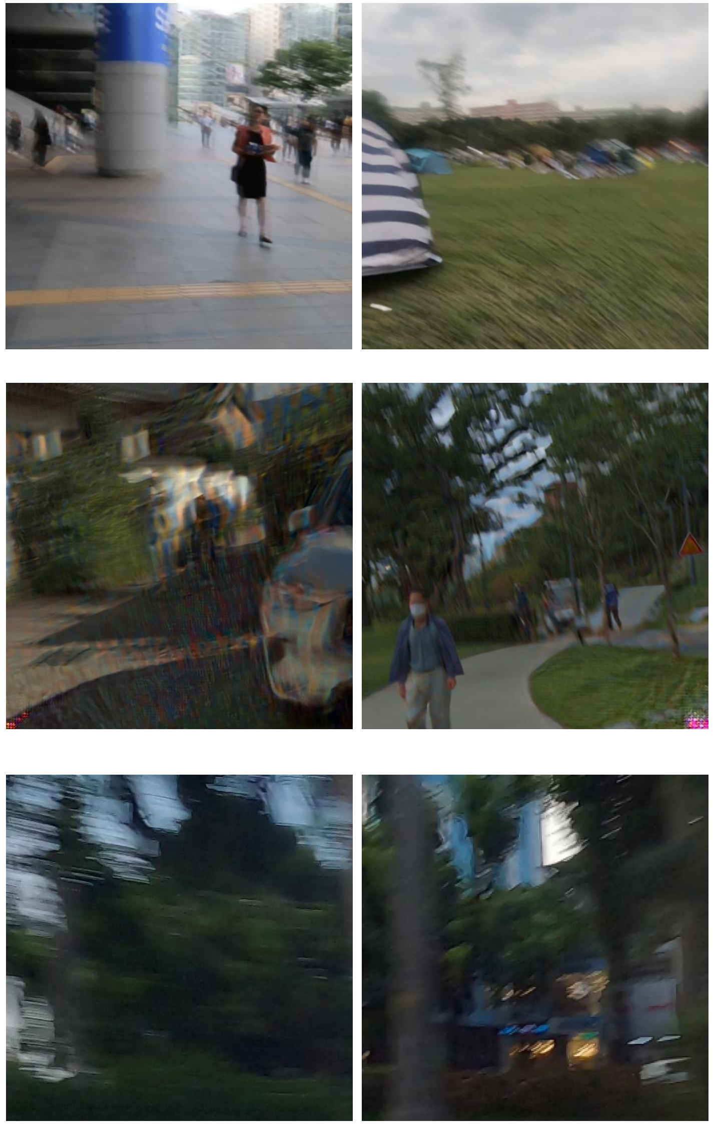
Unknown DeBlur
+ NAFNet(GoPro)

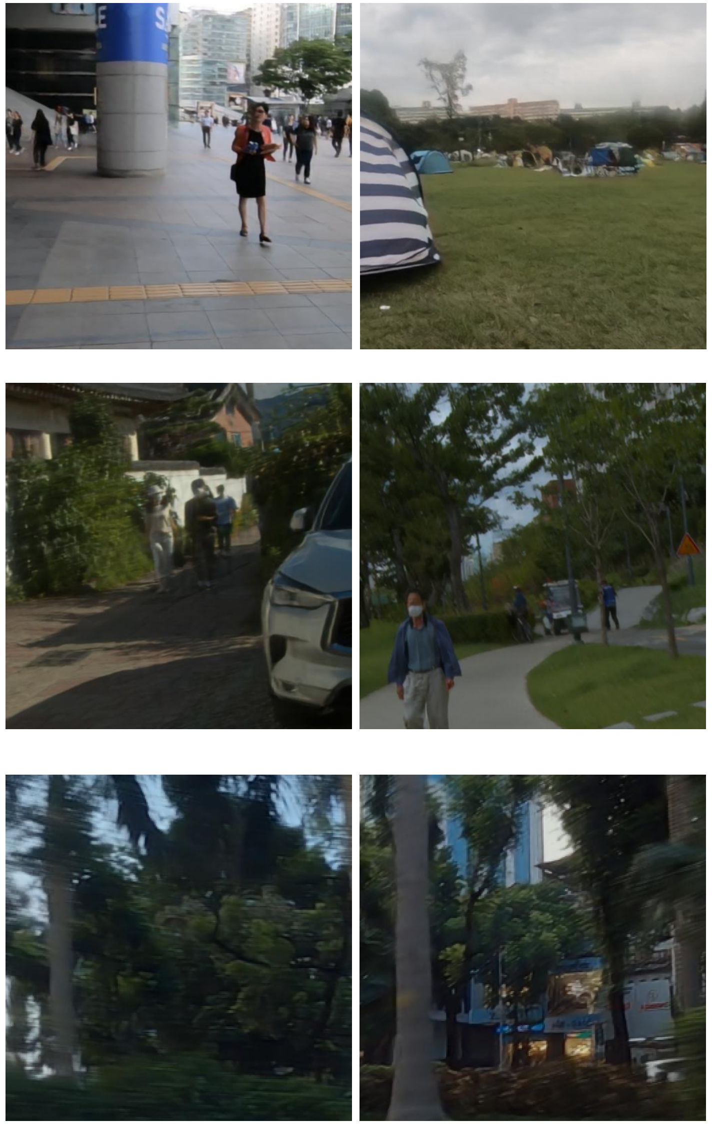
Known DeBlur
+ NAFNet(GoPro)

REDS
(Synthetic)

RSBlur
(Real)

RB2V
(Real)





GoPro
(Synthetic)

(Pause and Zoom for best view)

Experiments ~ Many-to-One

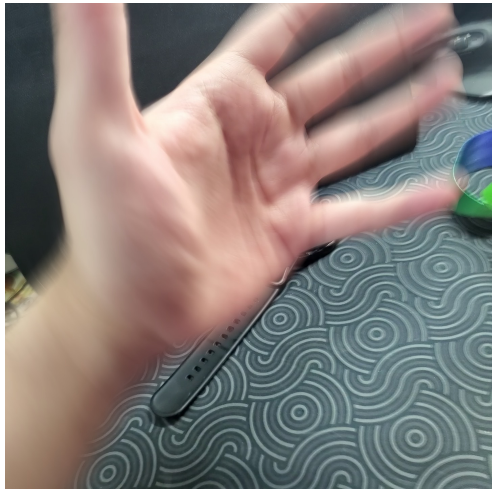
	RB2V_Street	REDS	RSBlur
NAFNet [3]			
w/ GoPro	24.78 / 0.714	25.80 / 0.880	26.33 / 0.790
w/ Synthetic Data	22.10 / 0.644	25.07 / 0.853	23.53 / 0.659
w/ Blur2Blur (GoPro)	**26.98** / **0.812**	**28.11** / **0.893**	**29.00** / **0.857**
w/ *the source domain**	28.72 / 0.883	29.09 / 0.927	33.06 / 0.888
Restormer [37]			
w/ GoPro	23.34 / 0.698	25.43 / 0.775	25.98 / 0.788
w/ Synthetic Data	23.78 / 0.655	24.76 / 0.753	23.34 / 0.651
w/ Blur2Blur (GoPro)	<u>25.97</u> / <u>0.750</u>	<u>27.55</u> / <u>0.885</u>	<u>28.89</u> / <u>0.850</u>
w/ *the source domain**	27.43 / 0.849	28.23 / 0.916	32.87 / 0.874
Generalized Deblurring			
BSRGAN [38]	23.31 / 0.645	26.39 / 0.803	27.11 / 0.810
RSBlur [25]	23.42 / 0.603	26.32 / 0.812	26.98 / 0.798
Unpaired Training			
CycleGAN [41]	21.21 / 0.582	23.92 / 0.775	23.34 / 0.782
DualGAN [35]	21.02 / 0.556	23.50 / 0.700	22.78 / 0.704

The pPSNR↑/pSSIM↑ scores. The best scores are in **bold** and the second best score are in <u>underline</u>. For a supervised method, NAFNet or Restormer, we assess its upper-bound of deblurring performance by training it on the *training set of the source dataset**.

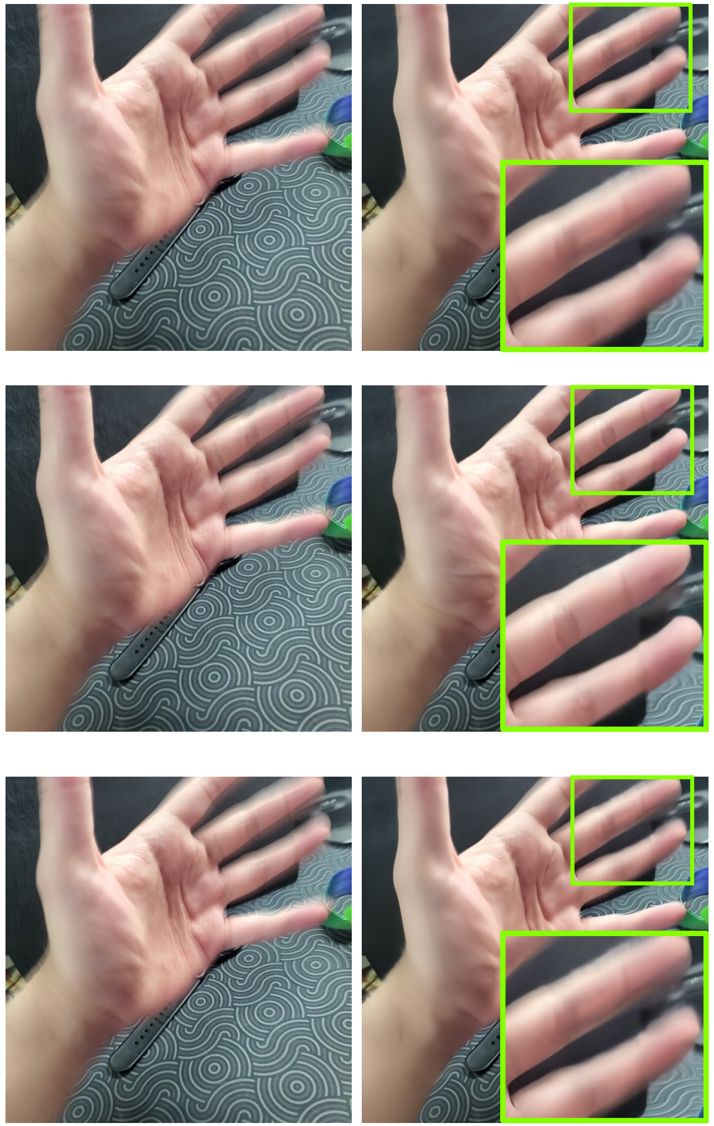
Experiments ~ One-to-Many

Unknown Blur

Samsung Galaxy
Note 10 Plus
(PhoneCraft dataset)



Known Blur Deblurred

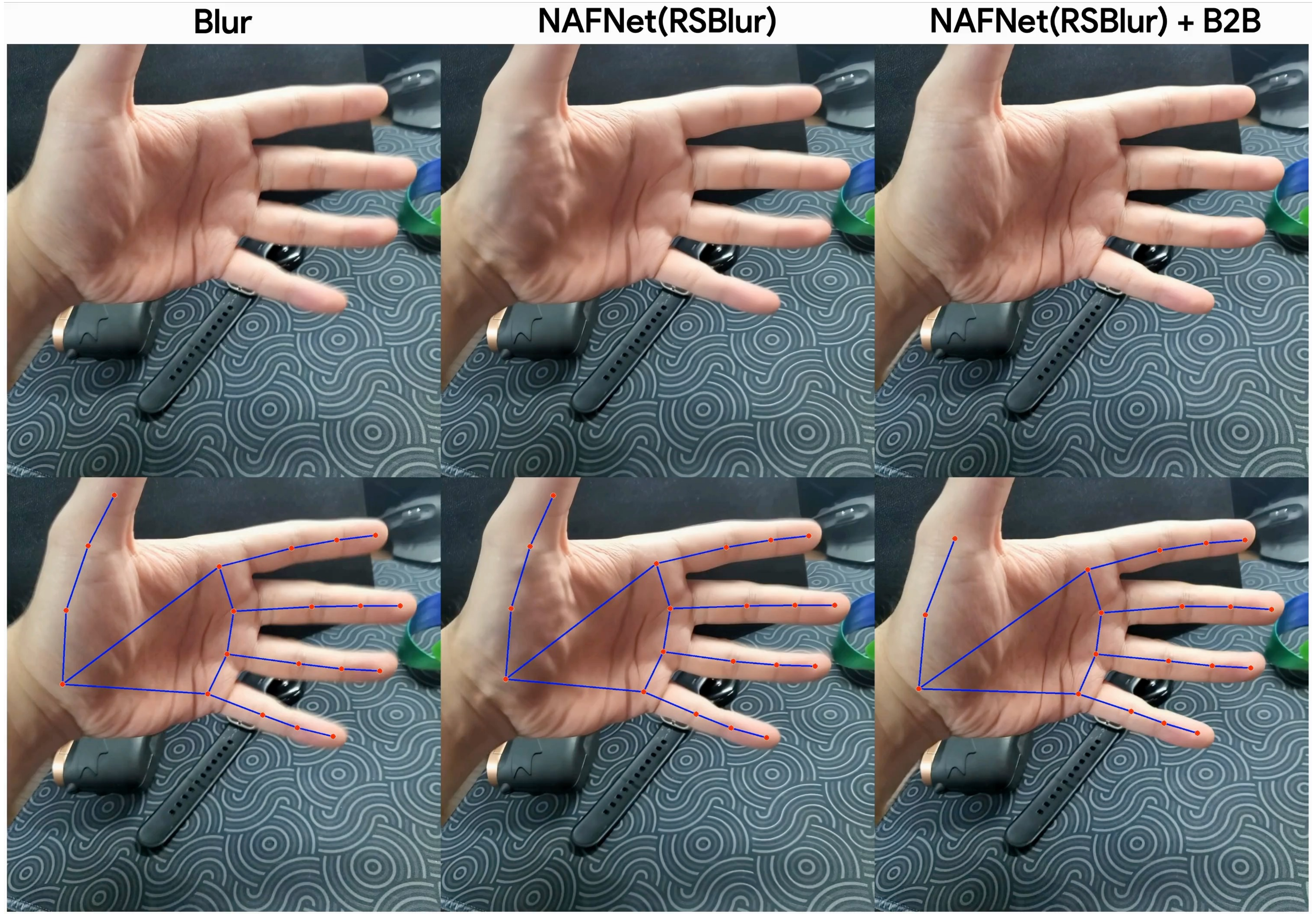


REDS
(Synthetic)

RSBlur
(Real)

GoPro
(Synthetic)

Experiments ~ Hand Pose Application



Experiments ~ Extreme Real Blur

Blurry Input

NAFNet(RSBlur)

NAFNet(RSBlur) + B2B

Conclusion

- We propose **Blur2Blur**, an effective approach to address the practical challenge of adapting image deblurring techniques to handle unseen blur

- "Plug-and-play module" for better utilizing pretrained of state-of-the-arts models.

- By conducting evaluations with real-world blurry datasets, affirming its role as a versatile deblurring model for general applications, opening up new unsupervised approaches.

- Contact for more: bangdang2000@gmail.com

Project Page

Code