VCNet: A Robust Approach to Blind Image Inpainting

Yi Wang[1], Ying-Cong Chen[2], Xin Tao[3], and Jiaya Jia[1,4]

[1]The Chinese University of Hong Kong [2]MIT CSAIL [3]Kuaishou Technology [4]SmartMore

Why blind inpainting

- How to repair these cases?

Why blind inpainting

- How to repair these cases?

Why blind inpainting

- How to repair these cases?

Why blind inpainting

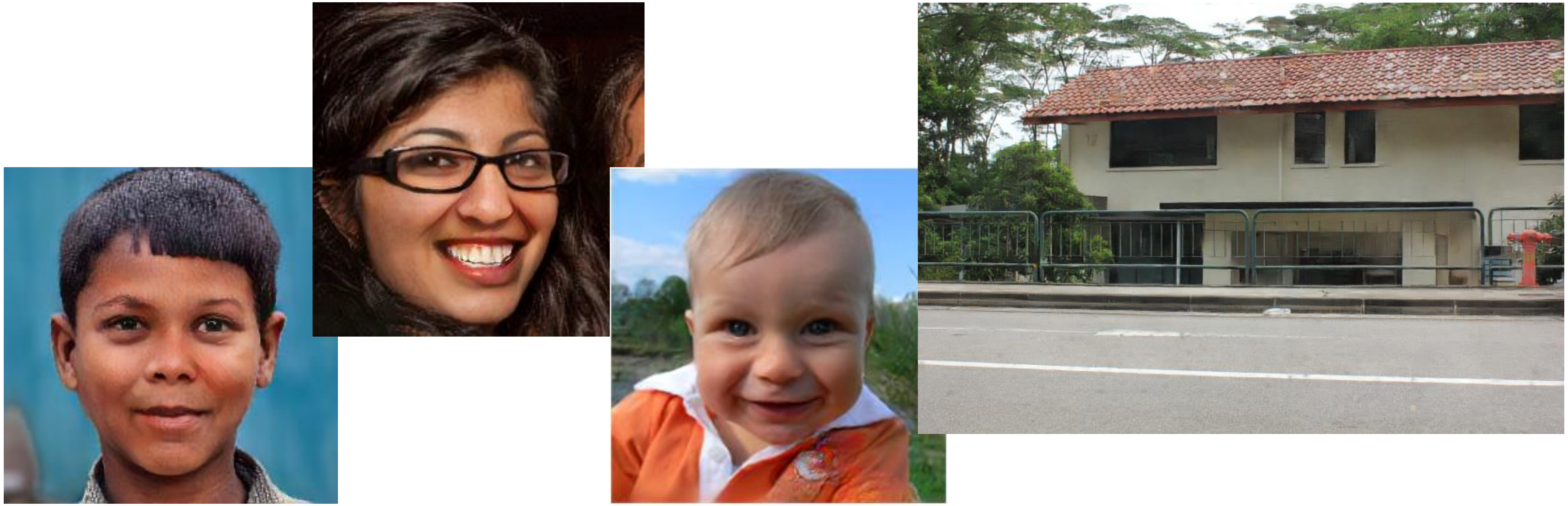
- How to repair these cases?

Why blind inpainting

- How to repair these cases?

Why blind inpainting

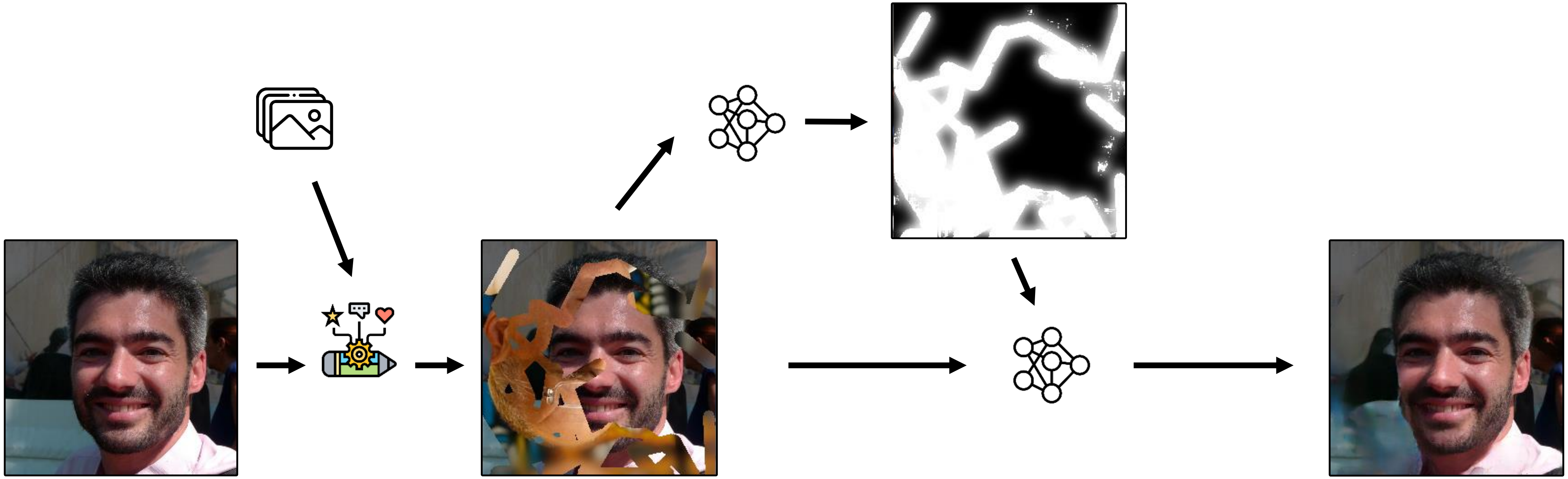
- How to repair these cases *automatically*?



What we propose in this paper

Blind inpainting method

- Robust against *unseen degradation* patterns & *mask errors.*



Our method

- Data synthesis

$$\mathbf{I} = \mathbf{O} \odot (\mathbf{1} - \mathbf{M}) + \mathbf{N} \odot \mathbf{M}$$

Where $\mathbf{I} \in \mathbb{R}^{h \times w \times c}$ is a degraded image (contaminated by unknown visual signals), $\mathbf{O} \in \mathbb{R}^{h \times w \times c}$ is the corresponding ground truth of $\mathbf{I}$. $\mathbf{M} \in \mathbb{R}^{h \times w \times \mathbf{1}}$ is a binary region mask (0 for known pixels and 1 otherwise) and $\mathbf{N} \in \mathbb{R}^{h \times w \times \boldsymbol{c}}$ is a noisy visual signal.

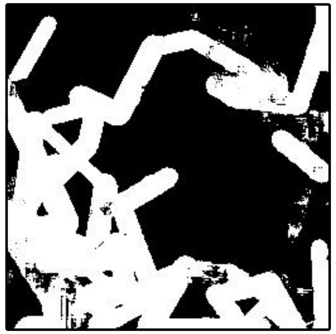
- About the possible image contamination,
  - $\mathbf{N} \rightarrow$ what,
  - $\mathbf{M} \rightarrow$ where.

Our method

- About the possible image contamination,
  - $\mathbf{N} \rightarrow$ what,
  - $\mathbf{M} \rightarrow$ where.
- Intuition: make $\mathbf{N}$ is indistinguishable as much as possible from $\mathbf{I}$ on image pattern.
  - Discriminative models cannot decide if a local region is corrupted without seeing its context.
  - A neural system trained with such data has the potential to work on unknown contamination.

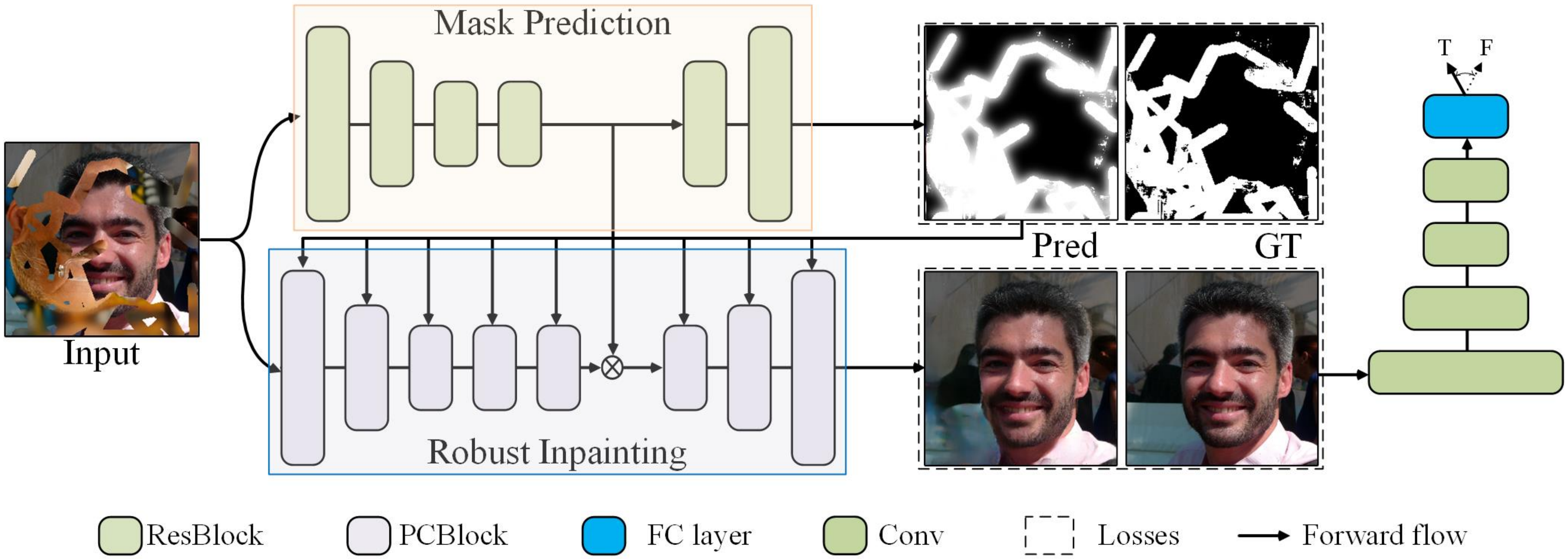
Our method

- Using real-world image patches to form $\mathbf{N}$ and free-form stokes as $\mathbf{M}$
  - $\mathbf{M}$ is dilated a bit by the iterative Gaussian smoothing
  - Employing alpha blending in the contact region between $\mathbf{N}$ and $\mathbf{O}$
- Training tuples $< \mathbf{I}_i, \mathbf{O}_i, \mathbf{M}_i, \mathbf{N}_i >_{|i=1,\ldots,m}$



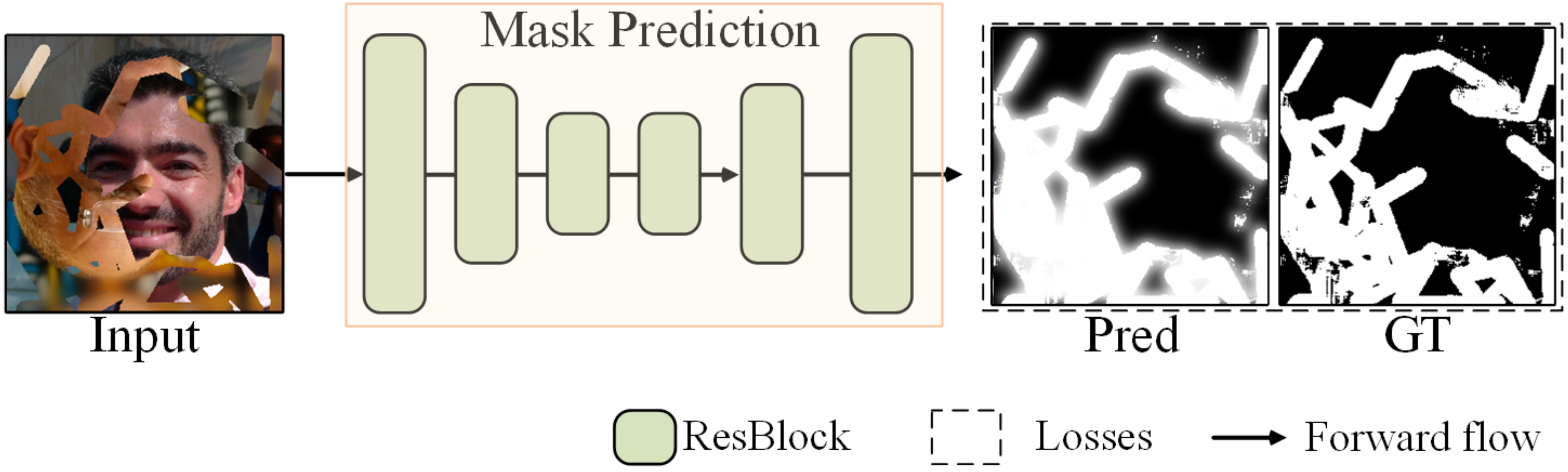
Our method

- Framework: mask prediction + image inpainting



Our method

- Mask prediction
  - To predict potential visually inconsistent area of a given image
  - Formulate it as a binary pixel-level classification.
    - A self-adaptive loss to balance positive- and negative-sample classification



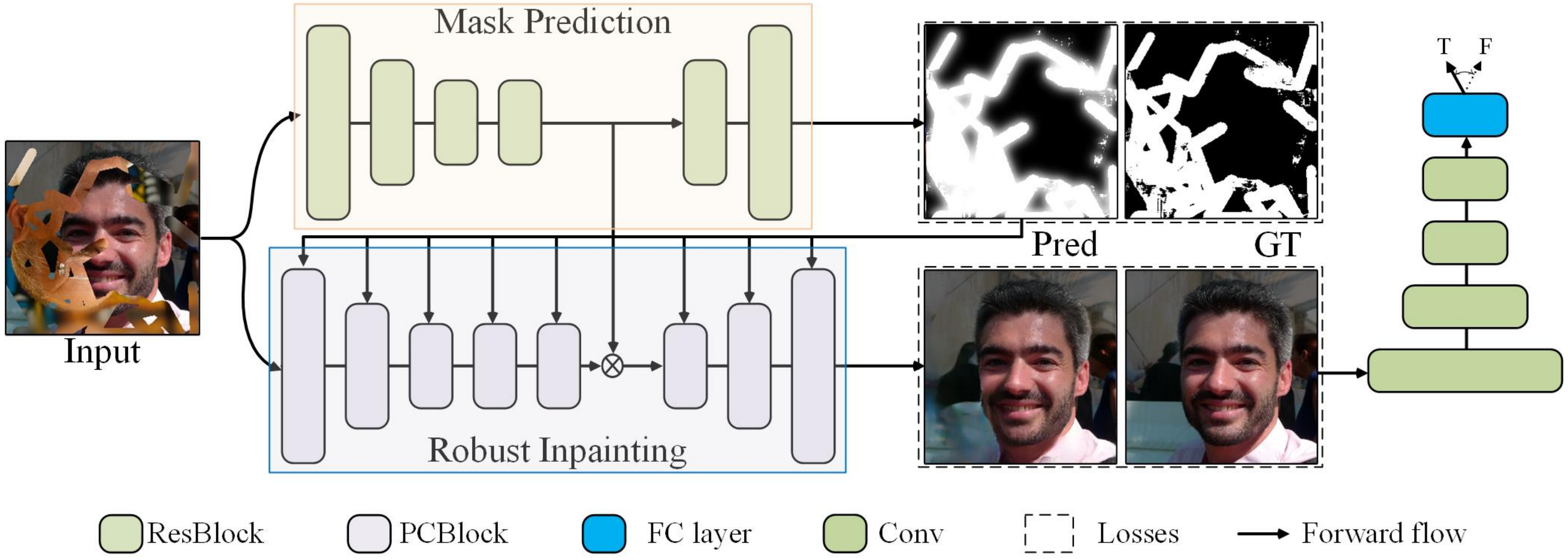
Our method

- Mask prediction
  - The optimization target of MPN is to detect all corrupted regions.
  - We propose to detect the inconsistency region of the image.
    - If these regions are detected correctly, other corrupted regions can be naturally blended to the image

GT　Input　Training MPN alone　Joint training　Face-swap

Our method

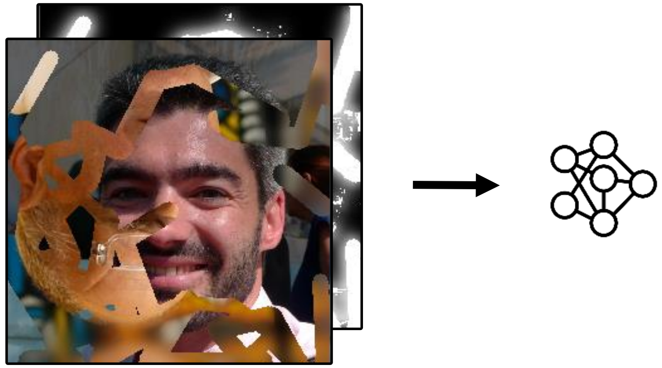
- Image inpainting
  - To inpaint inconsistent parts based on the predicted mask and context
  - Repairing corrupted regions requires knowledge from contextual information



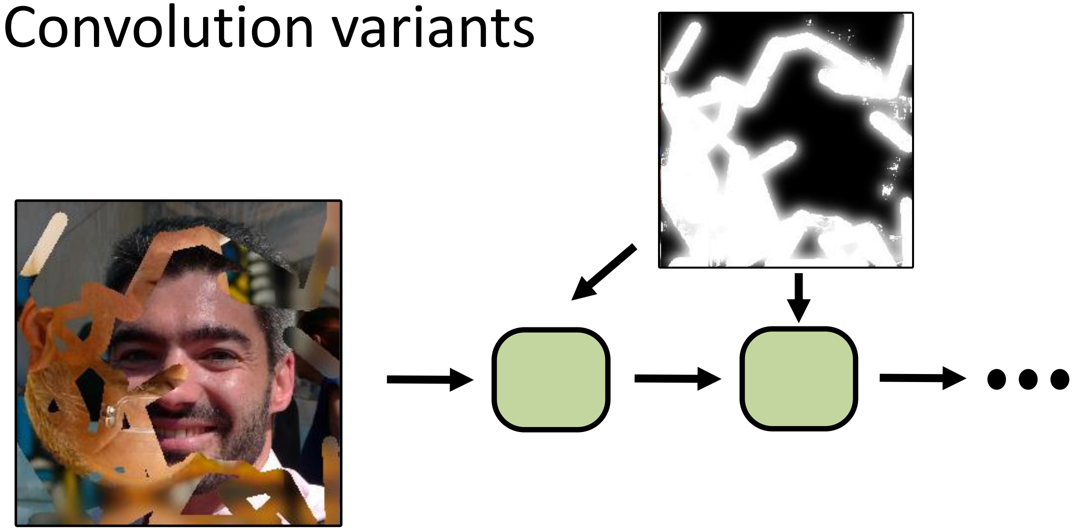
Our method

- How to exploit the predicted mask

Naïve concatenation



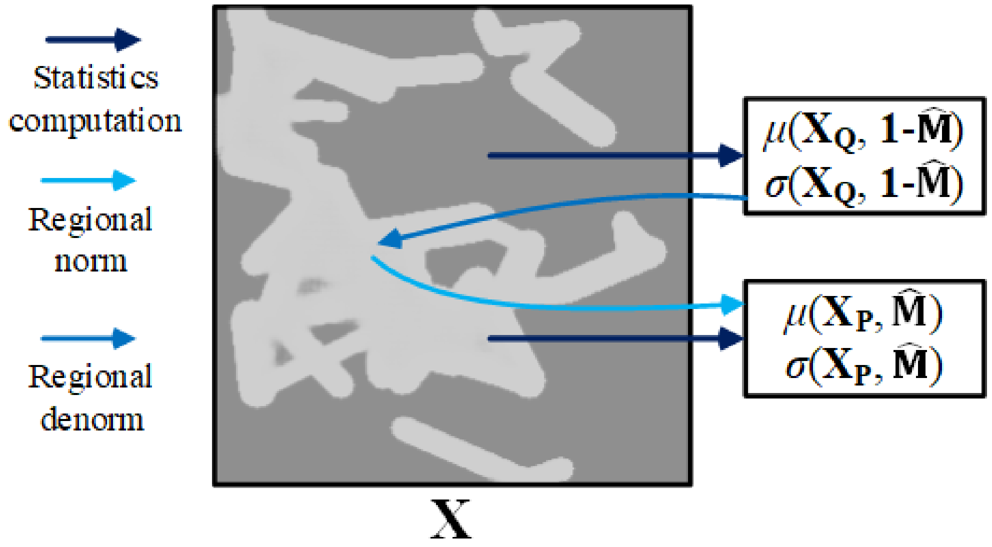
Convolution variants



Mask guided conv (e.g., partial conv, gated conv)

Our method

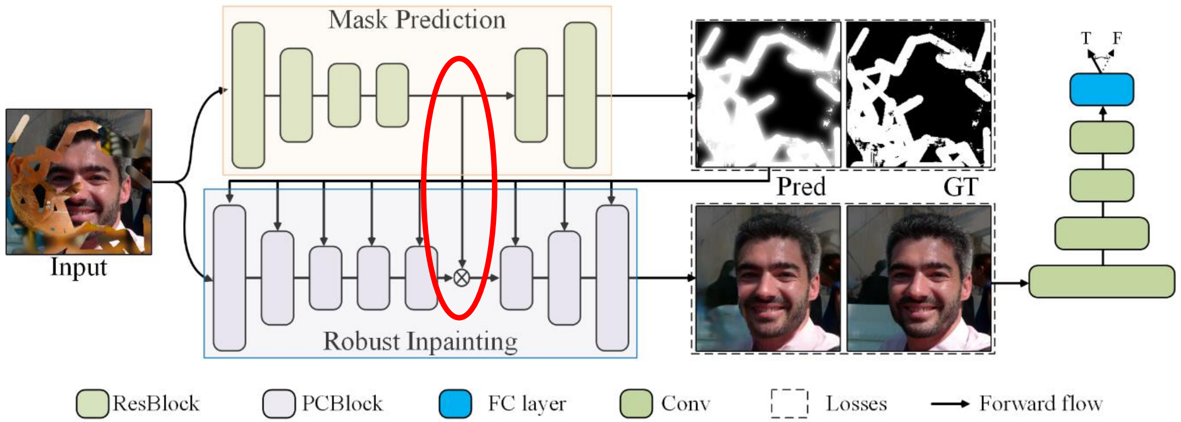
- How to exploit the predicted mask
  - Probabilistic context normalization (PCN): transferring contextual information in different layers
  - PCN = context feature transfer + feature preserving



$$\mathrm{PCN}(\mathbf{X}, \mathbf{H}) = [\beta \boldsymbol{\tau}(\mathbf{X}, \mathbf{H}) \odot \mathbf{H} + (1 - \beta)\mathbf{X} \odot \mathbf{H}] + \mathbf{X} \odot \overline{\mathbf{H}}$$

Our method

- Other designs in image inpainting
  - Feature fusion
    - Exploit discriminative features in inpainting



  - A comprehensive optimization target

$$\mathcal{L}_g(\hat{\mathrm{O}}, \mathrm{O}) = \underbrace{\lambda_r ||\hat{\mathrm{O}} - \mathrm{O}||_1}_{\text{reconstruction term}} + \underbrace{\lambda_s ||V^l_{\hat{\mathrm{O}}} - V^l_{\mathrm{O}}||_1}_{\text{semantic consistency term}} + \underbrace{\lambda_f \mathcal{L}_{mrf}(\hat{\mathrm{O}}, \mathrm{O})}_{\text{texture consistency term}} + \underbrace{\lambda_a \mathcal{L}_{adv}(\hat{\mathrm{O}}, \mathrm{O})}_{\text{adversarial term}}$$

Experiments

- Quantitative evaluation

Table 1: Quantitative results on the testing sets from different methods

Method	FFHQ-2K			Places2-4K			ImageNet-4K		
	BCE↓	PSNR↑	SSIM↑	BCE↓	PSNR↑	SSIM↑	BCE↓	PSNR↑	SSIM↑
CA [43]	1.297	16.56	0.5509	0.574	18.12	0.6018	0.450	17.68	0.5285
GMC [37]	0.766	20.06	0.6675	0.312	20.38	0.6956	0.312	19.56	**0.6467**
PC [22]	**0.400**	20.19	0.6795	0.273	19.73	0.6682	0.229	19.53	0.6277
GC [44]	0.660	17.16	0.5915	0.504	18.42	0.6423	0.410	18.35	0.6416
Our VCN	**0.400**	**20.94**	**0.6999**	**0.253**	**20.54**	**0.6988**	**0.226**	**19.58**	0.6339

Experiments

- Qualitative evaluation: synthetic ones

Input CA GMC PC GC Ours

Experiments

- Qualitative evaluation: dealing with other shaped masks

Input

Prediction

Input

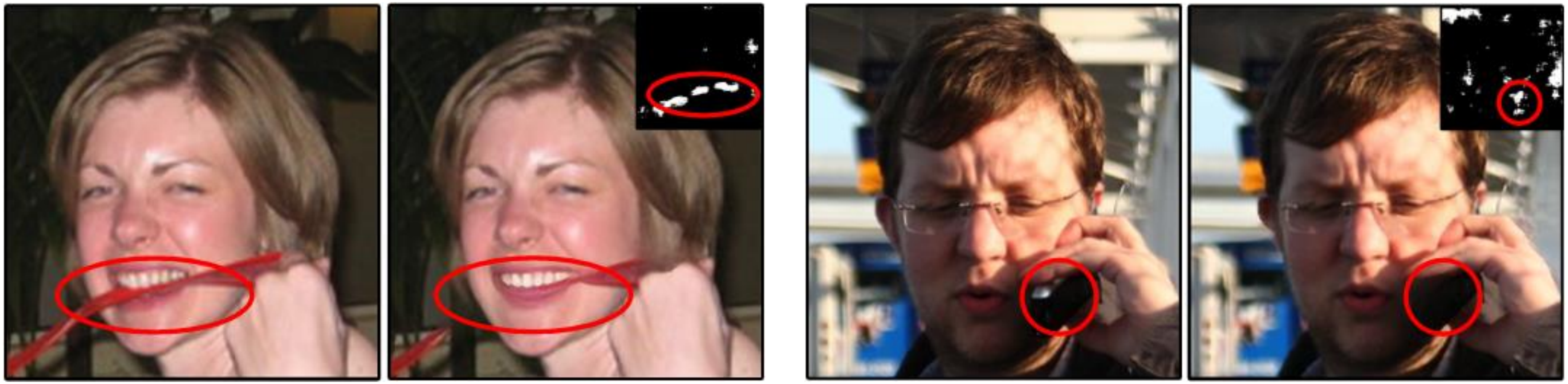
Prediction

Experiments

- Qualitative evaluation: dealing with different types of noisy patterns

Experiments

- Qualitative evaluation: dealing with real occluded faces



Input Prediction Input Prediction

Applications: graffiti removal


Input


Prediction


Input


Prediction


Input


Prediction

Applications: raindrop removal

Input

Prediction

Applications: face-swap

References

Input

Face-swap results

Thanks for watching!

Project website: https://github.com/shepnerd/blindinpainting_vcnet