# VRetouchEr: Learning Cross-frame Feature Interdependence with Imperfection Flow for Face Retouching in Videos
# Supplementary Materials

Wen Xue[1], Le Jiang[1], Lianxin Xie[1], Si Wu[1,2,3*], Yong Xu[1,2,3] and Hau San Wong[4]
[1]School of Computer Science and Engineering, South China University of Technology
[2]Peng Cheng Laboratory [3]PAZHOU LAB
[4]Department of Computer Science, City University of Hong Kong
{csxuewen, csjiangle, cslianxin.xie}@mail.scut.edu.cn
{cswusi, yxu}@scut.edu.cn, cshswong@cityu.edu.hk

## 1. Results on FFHQR

We visually compare the proposed VRetouchEr with the competing methods, including MPRNet, RestoreFormer, GPEN, BIPNet, ProPainter, AutoRetouch and BPFRe. The comparison is performed on FFHQR, and the synthesized results of the methods are shown in Figure 1. One can observe that VRetouchEr outperforms the competing methods in synthesizing the clear face images, which are more consistent with the manual retouching results.

## 2. Results on MRFV

We further compare VRetouchEr with the competing methods on three in-the-wild videos of MRFV, and their results are shown in Figures 2-5, Figures 6-9 and Figures 10-13. Specifically, we present the retouching results of the methods on a number of representative frames in Figures 2, 6 and 10. We also plot the per-frame PSNR scores of the methods in Figure 3, 7 and 11. VRetouchEr is able to remove the imperfections, while the competing methods fail to achieve stable and satisfactory retouching performance. Further, VRetouchEr is compared with the second best method: BPFRe, in facial imperfection localization. We visualize the localization maps in Figures 4, 8 and 12, and plot the per-frame Soft-IoU scores in Figures 5, 9 and 13. We consider that more stable and accurate imperfection localization leads to the superior retouching performance of VRetouchEr.

## 3. Analysis on Imperfection Localization

We perform an experiment to investigate the complementarity between the Imperfection Localization module (IL) and the latent transformer $T$. First, We build a variant by disabling imperfection prediction and refinement networks, and the resulting model is referred to as ‘w/o IL’. Due to lack of explicit imperfection localization, ‘w/o IL’ learns the mapping from the domain of raw images to the one of clear face images. For the input frames, we visualize the activation maps of ‘w/o IL’ and VRetouchEr in Figure 14. We can make the following observations: ‘w/o IL’ cannot apply consistent attentions on the imperfections of the frames. VRetouchEr achieves better performance in localizing and removing imperfections, since the imperfection localization map and the activation map are complementary to a certain extent.

## 4. Ethical Statement

It is imperative to clarify that our technique is not designed to support any form of undesirable editing. Recognizing the importance of embracing diverse beauty standards, we are acutely aware of and concerned about the possible adverse effects our technology might have. We will encompass guidelines and ethical practices for face retouching to ensure responsible use. Furthermore, we acknowledge the critical issue of algorithmic bias. To address this, we are committed to ensuring fairness and inclusivity in our training dataset and in the development of our model.

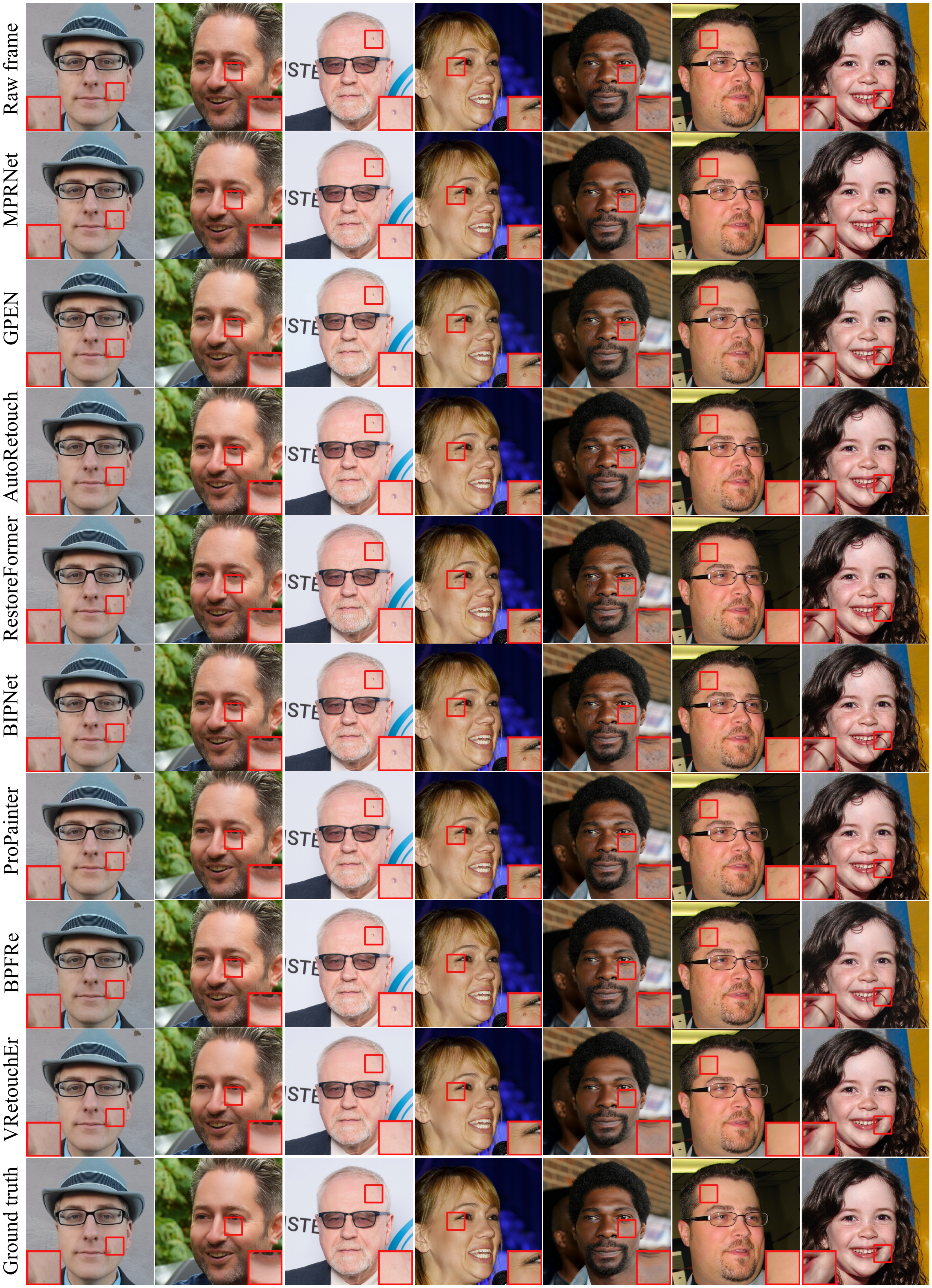

Figure 1. Visual comparison between VRetouchEr and the competing methods on FFHQR images.

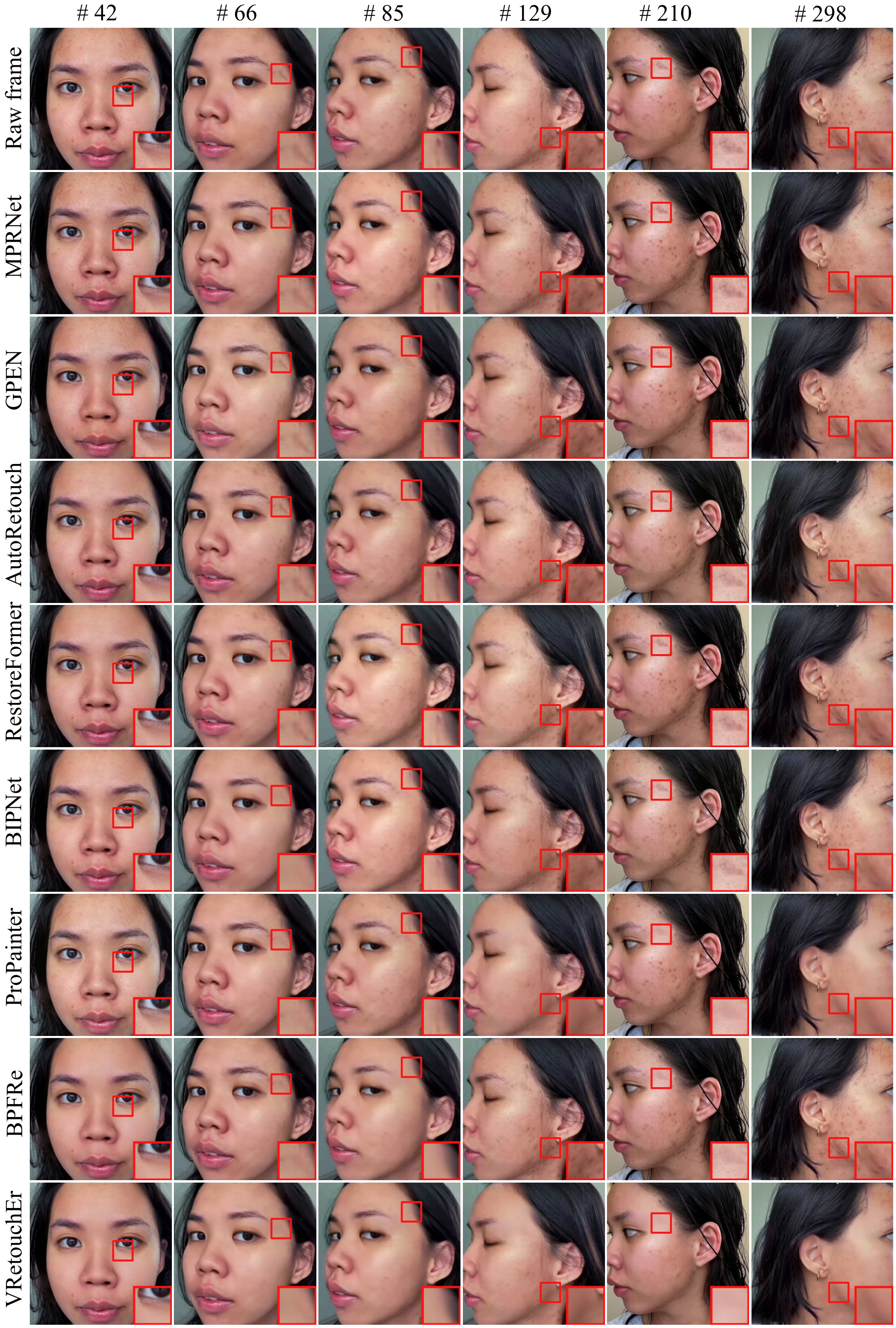

Figure 2. The retouching results of VRetouchEr and the competing methods on in-the-wild video frames.

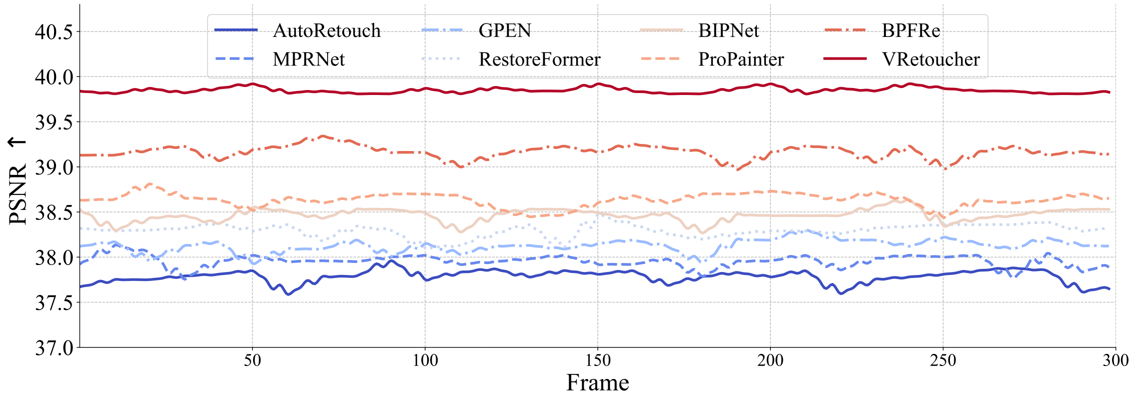

Figure 3. PSNR scores of VRetouchEr and the competing methods on in-the-wild video frames.

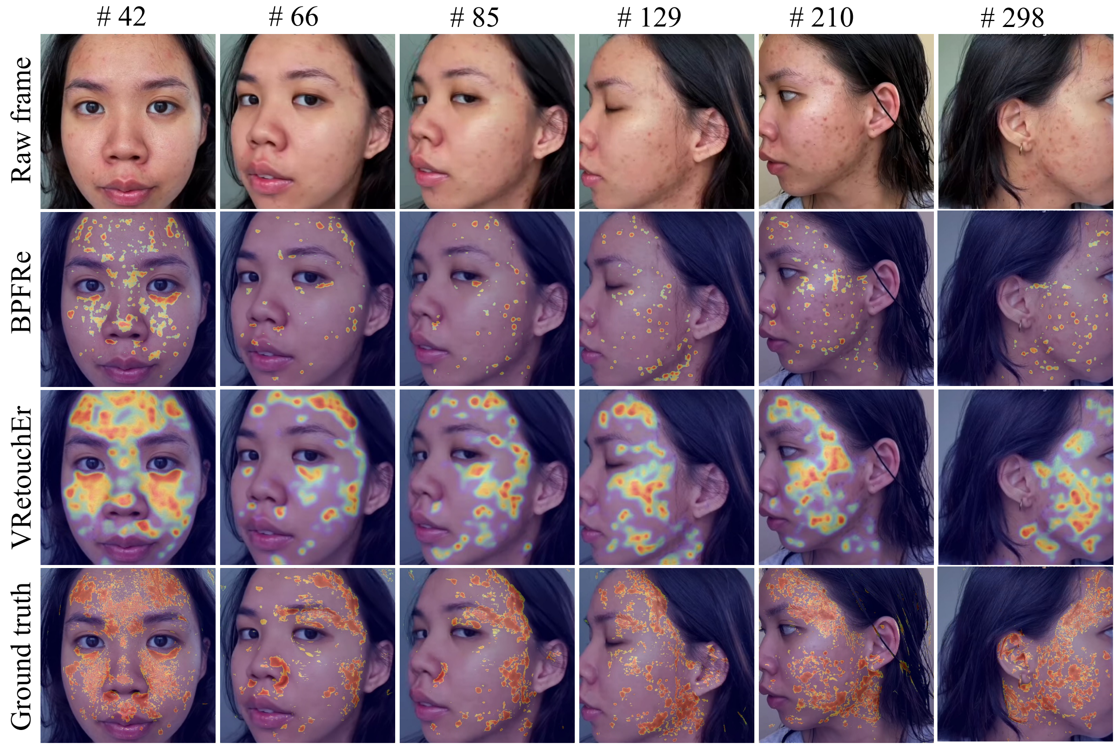

Figure 4. Visual comparison between VRetouchEr and BPFRe in imperfection localization on in-the-wild video frames.

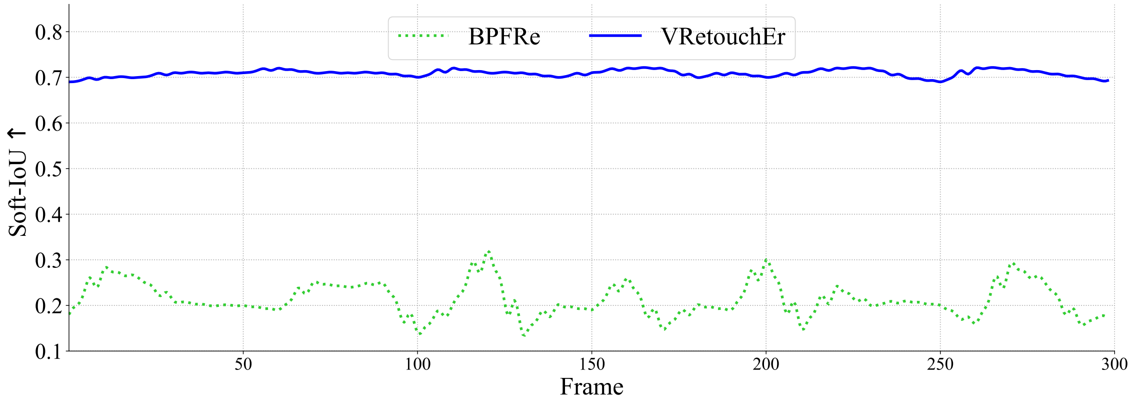

Figure 5. Soft-IoU scores of VRetouchEr and BPFRe in imperfection localization on in-the-wild video frames.

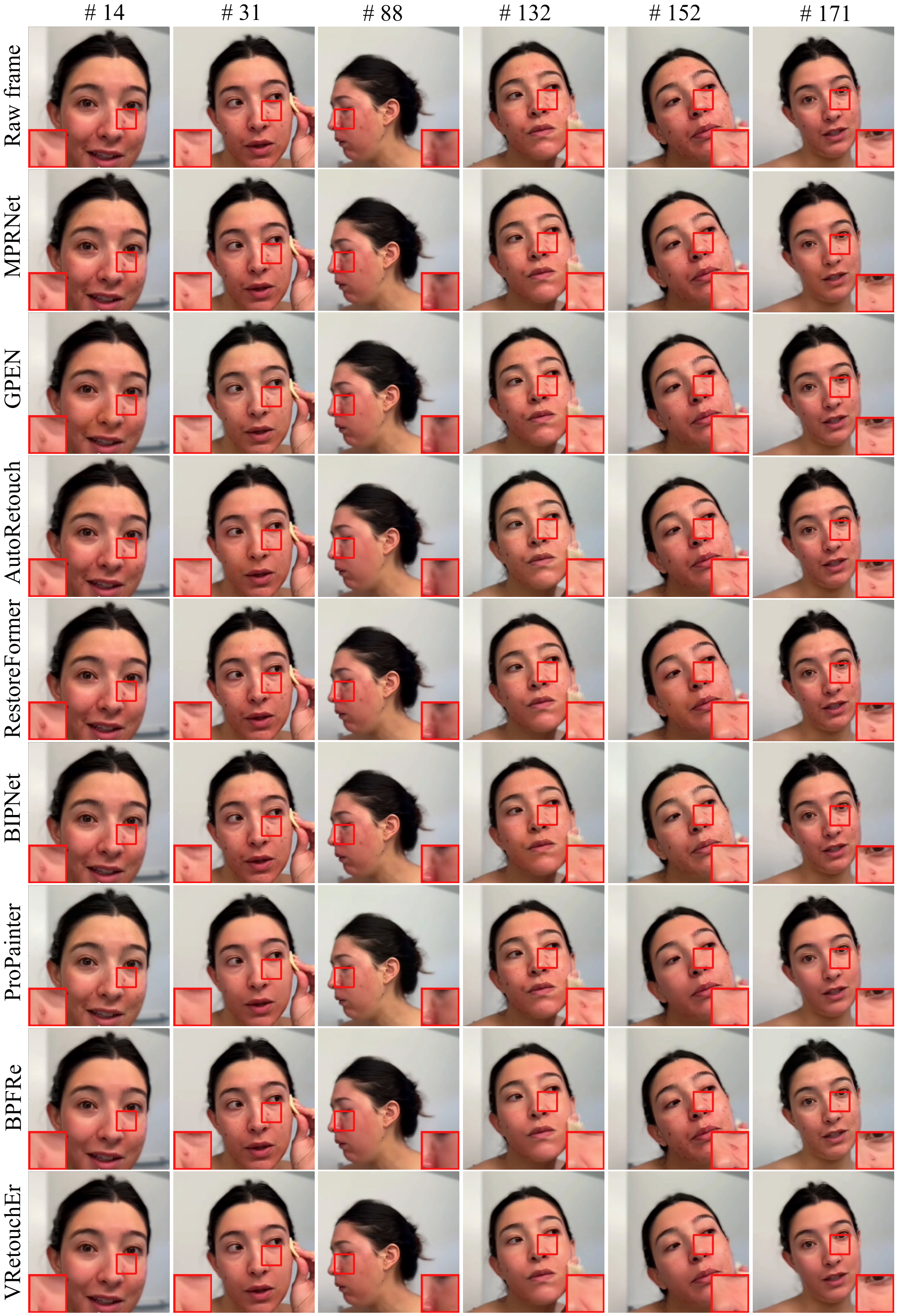

Figure 6. The retouching results of VRetouchEr and the competing methods on in-the-wild video frames.

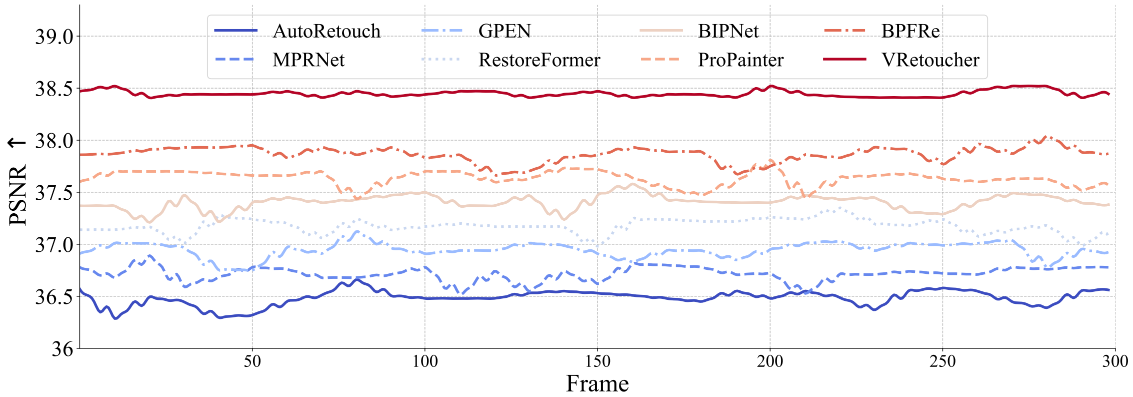

Figure 7. PSNR scores of VRetouchEr and the competing methods on in-the-wild video frames.

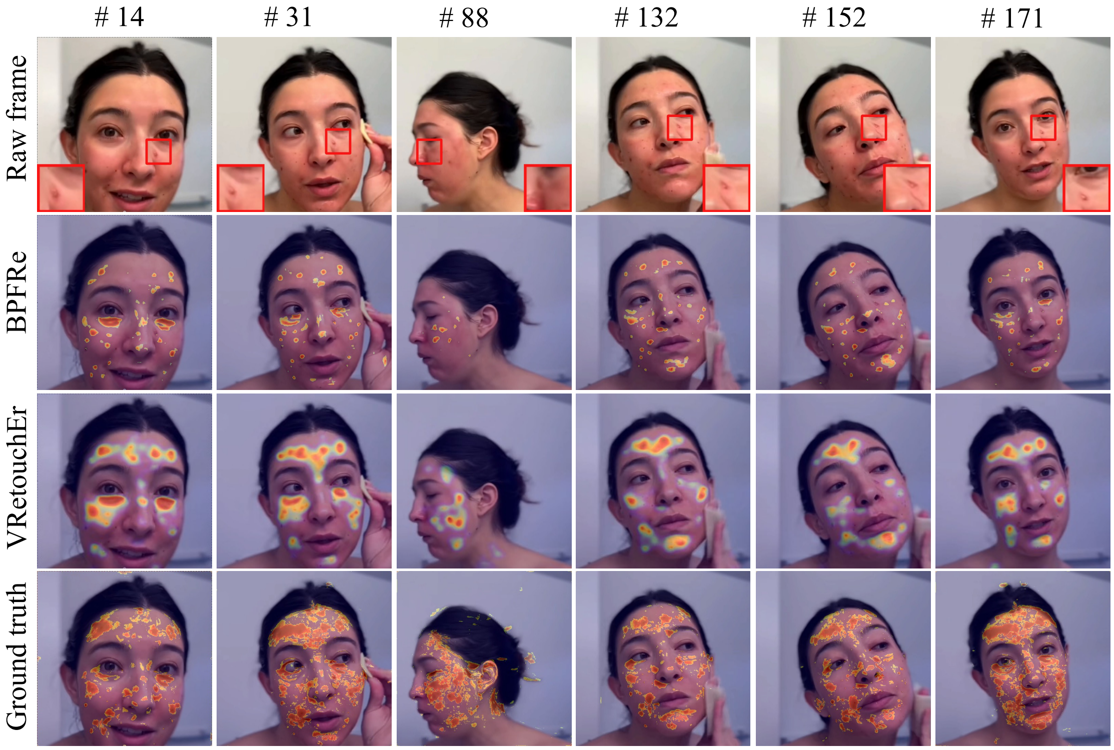

Figure 8. Visual comparison between VRetouchEr and BPFRe in imperfection localization on in-the-wild video frames.

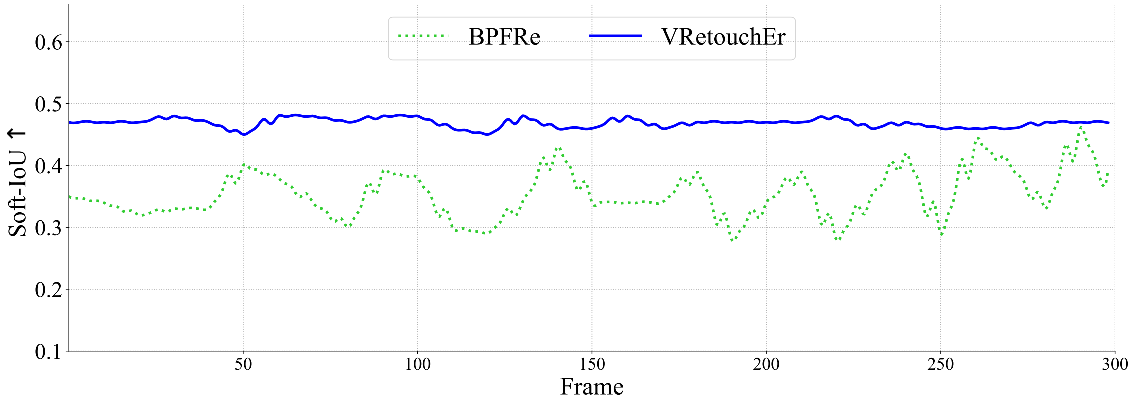

Figure 9. Soft-IoU scores of VRetouchEr and BPFRe in imperfection localization on in-the-wild video frames.

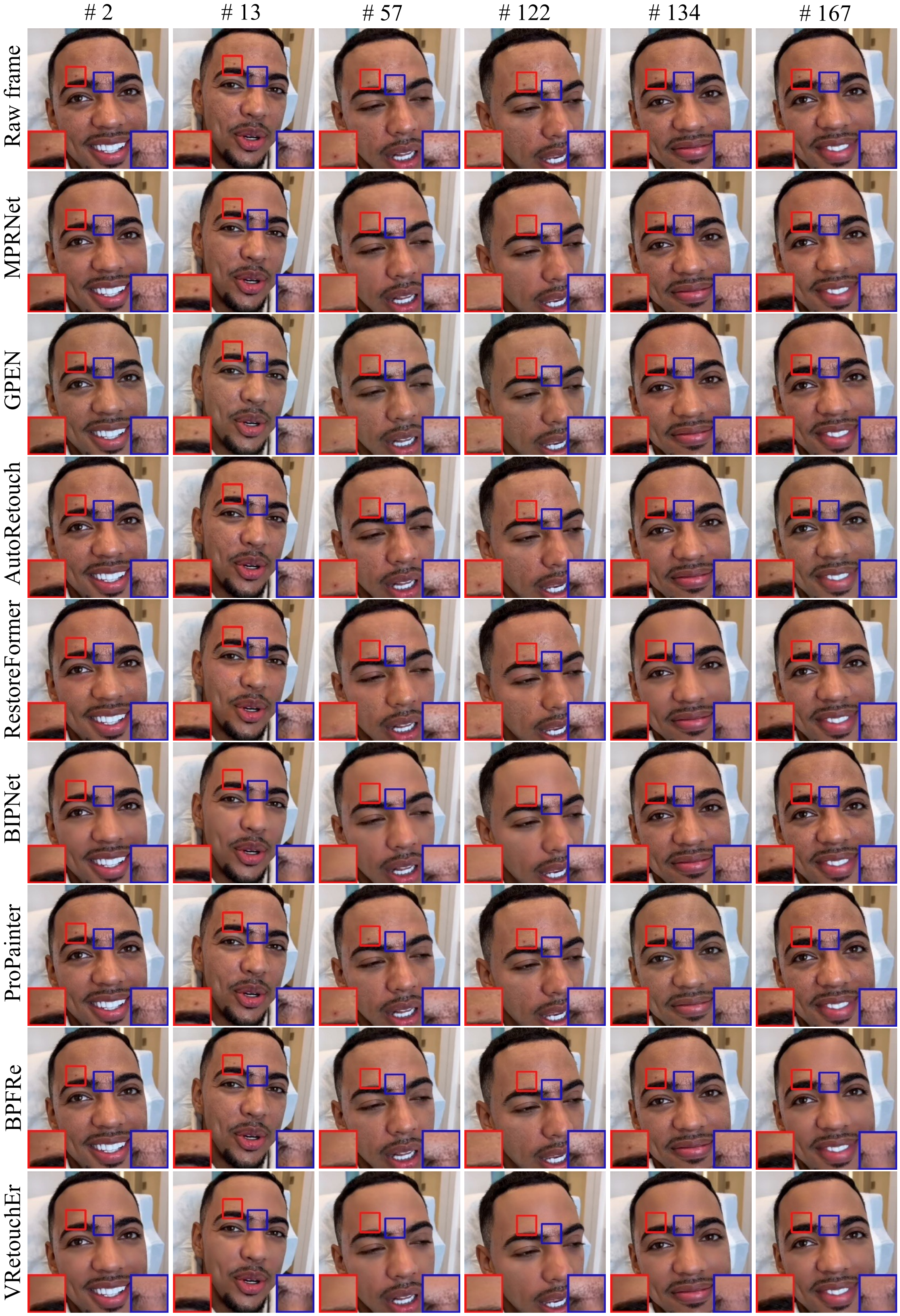

Figure 10. The retouching results of VRetouchEr and the competing methods on in-the-wild video frames.

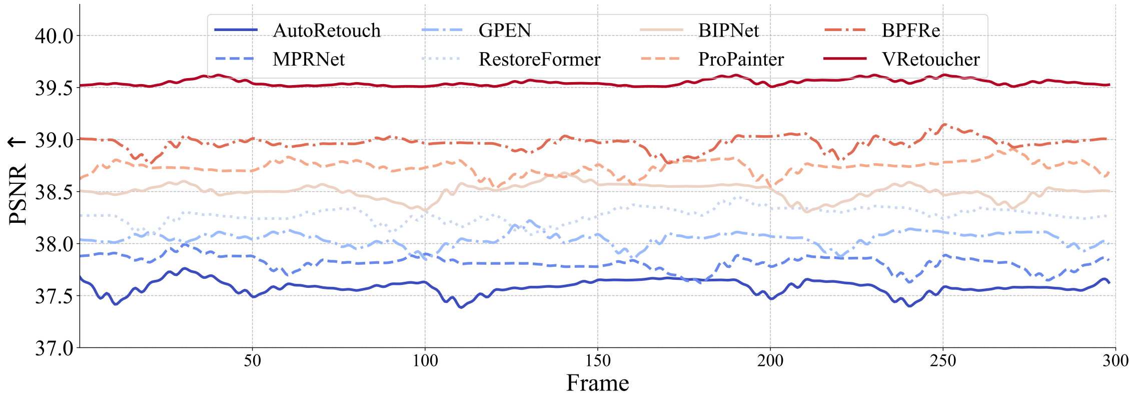

Figure 11. PSNR scores of VRetouchEr and the competing methods on in-the-wild video frames.

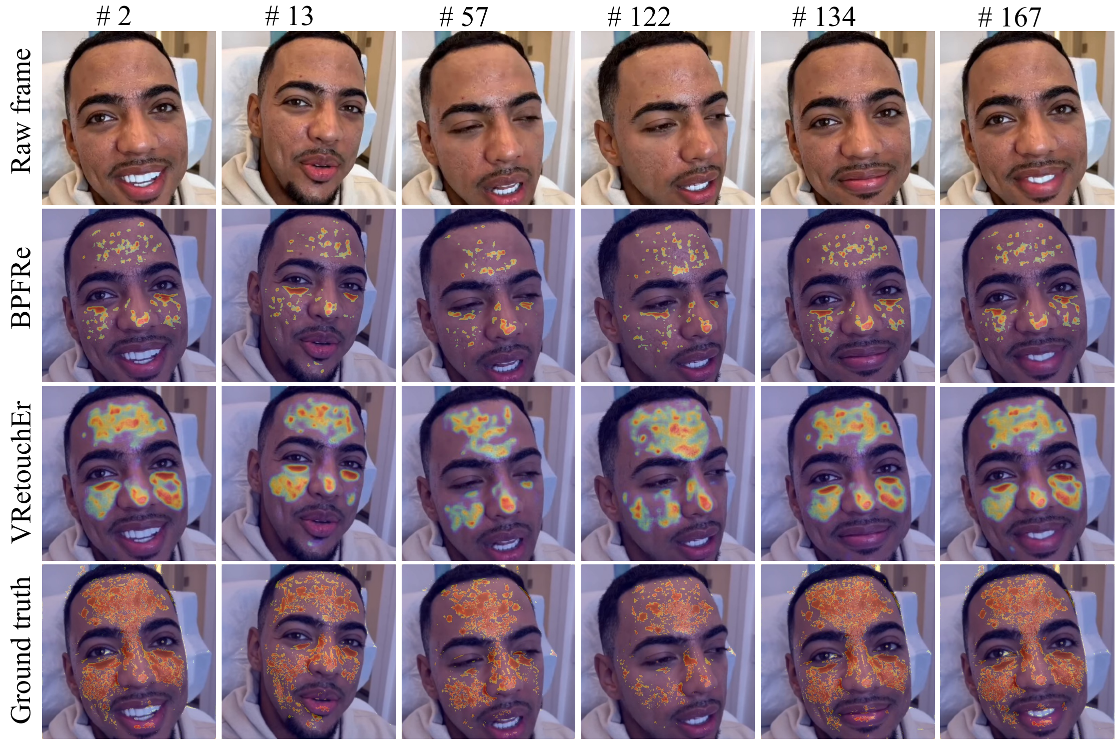

Figure 12. Visual comparison between VRetouchEr and BPFRe in imperfection localization on in-the-wild video frames.

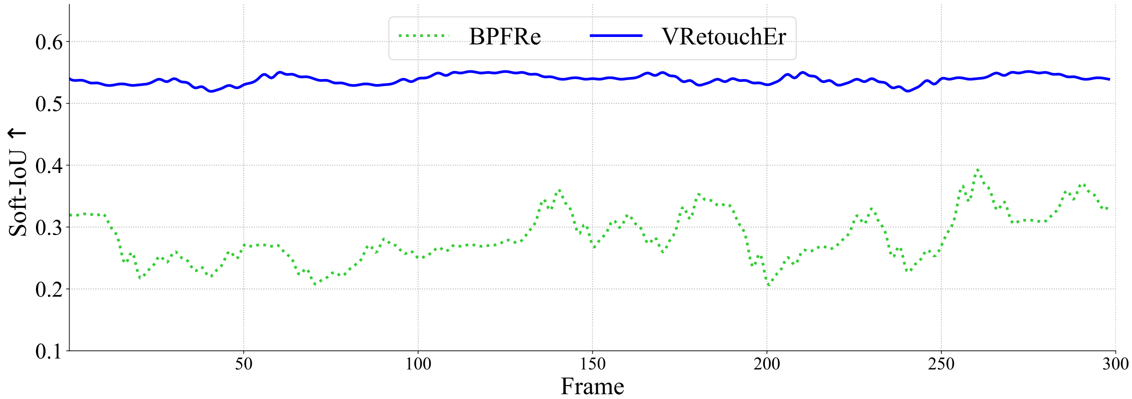

Figure 13. Soft-IoU scores of VRetouchEr and BPFRe in imperfection localization on in-the-wild video frames.

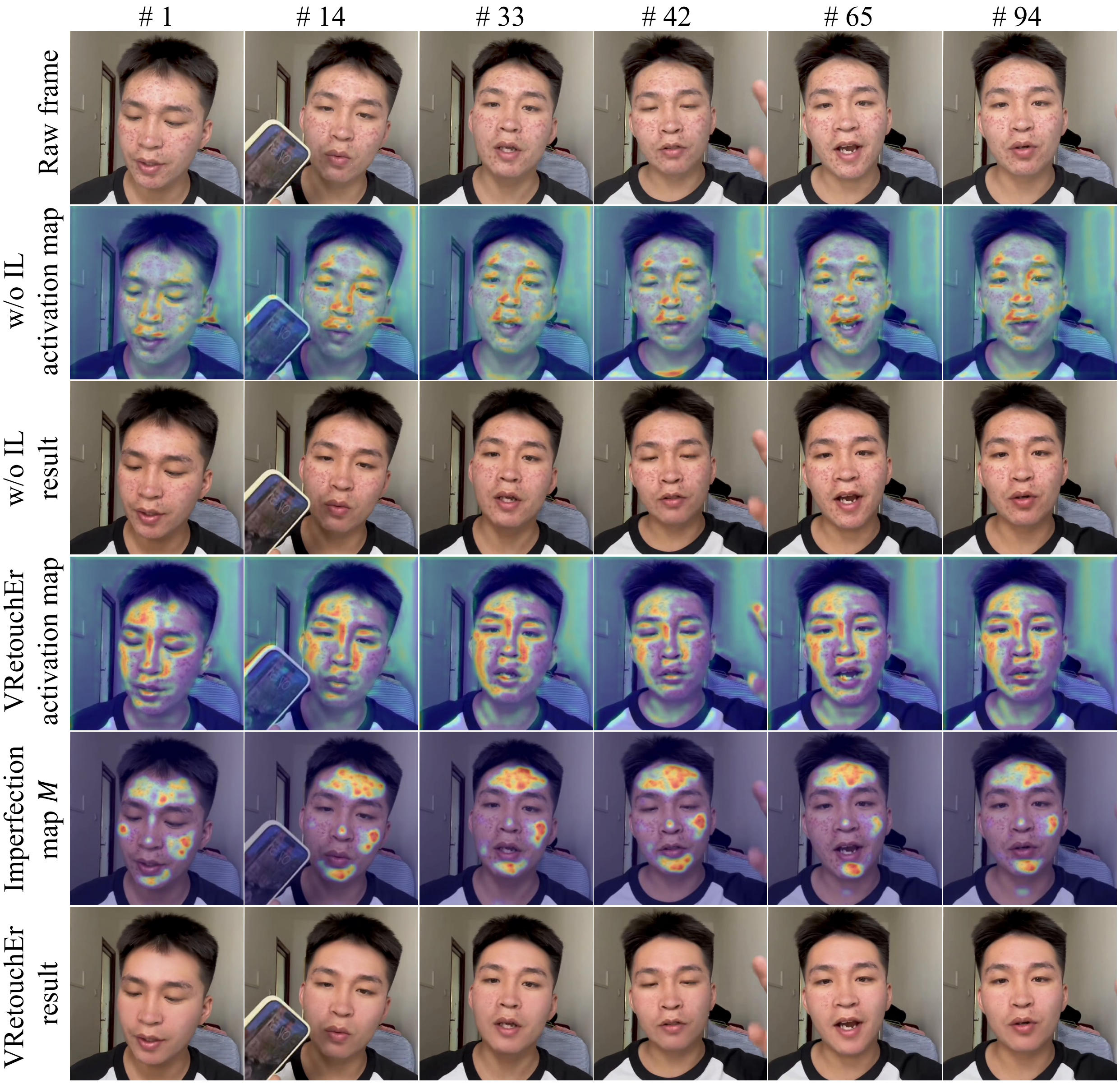

Figure 14. Visualization of the activation maps of VRetouchEr and the ablative model on in-the-wild video frames.

## 5. Network Architecture

The framework of VRetouchEr consists of a multi convolutional layer encoder $E$, a latent retouching transformer $T$, a multi convolutional decoder $G$, an imperfection flow estimation network $S$, a imperfection localization network $E$, and a convolutional patch discriminator $D$. We adopt SpyNet architecture [34] for $S$. The architecture details of $E, T, G$ and $N$ are presented in Tables 1-4.

Table 1. The network architecture of the encoder $E$ used in the experiments.

| Encoder $E$<br>Input: Raw image $X_t, \{X_r^i\} \in \mathbf{R}^{512\times512\times3}$ | | |
|---|---|---|
| Layer | Activation | Output size |
| ConvBlk | Leaky ReLU | $512 \times 512 \times 64$ |
| ConvBlk | Leaky ReLU | $256 \times 256 \times 128$ |
| ConvBlk | Leaky ReLU | $128 \times 128 \times 256$ |
| ConvBlk | Leaky ReLU | $64 \times 64 \times 512$ |
| ConvBlk | Leaky ReLU | $64 \times 64 \times 512$ |
| ConvBlk | Leaky ReLU | $64 \times 64 \times 512$ |

Table 2. The network architecture of the transformer $T$ used in the experiments,'MMA' denote the multi-frame masked attention block

| Transformer $T$.<br>Input: feature $f_t, M_t, \{M_r^i\}, \{f_r^i\} \in \mathbf{R}^{64\times64\times512}$ | | | | |
|---|---|---|---|---|
| Layer | window size | heads | Norm | MLP ratio |
| MMA | $6 \times 6 \times 16$ | 8 | Layer Norm | 2.0 |
| MMA | $6 \times 6 \times 16$ | 8 | Layer Norm | 2.0 |
| MMA | $6 \times 6 \times 16$ | 8 | Layer Norm | 2.0 |
| MMA | $6 \times 6 \times 16$ | 8 | Layer Norm | 2.0 |
| MMA | $6 \times 6 \times 16$ | 8 | Layer Norm | 2.0 |

Table 3. The network architecture of the decoder $G$ used in the experiments. 'AdaIN' denotes adaptive instance normalization.

| Decoder $G$<br>Input: retouched feature $\hat{f}_t \in \mathbf{R}^{64\times64\times512}$ | | | |
|---|---|---|---|
| Layer | Normalization | Activation | Output size |
| ConvBlk | AdaIN | Leaky ReLU | $64 \times 64 \times 512$ |
| ConvBlk | AdaIN | Leaky ReLU | $64 \times 64 \times 512$ |
| ConvBlk | AdaIN | Leaky ReLU | $64 \times 64 \times 512$ |
| ConvBlk | AdaIN | Leaky ReLU | $128 \times 128 \times 256$ |
| ConvBlk | AdaIN | Leaky ReLU | $256 \times 256 \times 128$ |
| ConvBlk | AdaIN | Leaky ReLU | $512 \times 512 \times 64$ |
| ConvBlk | AdaIN | Leaky ReLU | $512 \times 512 \times 3$ |

Table 4. The network architecture of the imperfection localization network $N$ used in the experiments.

| Imperfection localization network $N$<br>Input: raw image $\{X_i\} \in \mathbf{R}^{512\times512\times3}$, features $\{f_r^i\} \in \mathbf{R}^{64\times64\times512}$ | | |
|---|---|---|
| Layer | Activation | Output size |
| ConvBlk | Leaky ReLU | $512 \times 512 \times 64$ |
| ConvBlk | Leaky ReLU | $256 \times 256 \times 128$ |
| ConvBlk | Leaky ReLU | $128 \times 128 \times 256$ |
| ConvBlk | Leaky ReLU | $64 \times 64 \times 512$ |

## 6. Summary of VRetouchEr

The constituent networks of VRetouchEr are jointly optimized from scratch, and the training procedure is summarized in Algorithm 1.

**Algorithm 1** Pseudo-code of training the constituent networks in VRetouchEr.

```
**Input**: Paired training data $\{\mathbb{X}, \mathbb{Y}\}$.
**Initialize**: Encoder $E$ parameterized by $\theta_E$, imperfection flow estimator $S$ parameterized by $\theta_S$, imperfection localization network $N$ parameterized by $\theta_N$, Transformer $T$ parameterized by $\theta_T$, decoder $G$ parameterized by $\theta_G$, discriminator $D$ parameterized by $\theta_D$, align factors $\{\alpha, \beta\}$, learning rates $\varepsilon$ and # epochs $\Gamma$.
**for** $t = 1$ to $\Gamma$ **do**
    **for** each mini-batch **do**
        Sample the paired sequences $\{X, Y\}$.
        Feed each frame of $X$ into $E$ to produce the feature $f_{\{r/t\}}$.
        Feed $X$ into $S$ to obtain the imperfection flow $O$.
        Feed $O$ and $f_r$ into the FIR module to obtain the refined imperfection map $M$ by Eqs.(1-3).
        Feed $M$, $X$ and $X_t$ into $T$ to obtain the transformed feature $\hat{f}_t$ by Eqs.(4-6).
        Feed $\hat{f}_t$ into $G$ to synthesize image $\hat{y} \in \hat{Y}$.
        Optimize $N$ and $S$ by using Adam:
        $\{\theta_N, \theta_S\} \leftarrow \texttt{Adam}\big(\nabla(L_{flow} + L_{imp}), \{\theta_N, \theta_S\}, \varepsilon\big)$,
        Optimize $E$, $T$ and $G$ by using Adam:
        $\{\theta_E, \theta_T, \theta_G\} \leftarrow \texttt{Adam}\big(\nabla(L^{rec}_{con} + L^{syn}_{adv}), \{\theta_E, \theta_T, \theta_G\}, \varepsilon\big)$,
        Optimize $D$ by using Adam:
        $\theta_D \leftarrow \texttt{Adam}\big(\nabla(-L^{real}_{adv}), \theta_D, \varepsilon\big)$.
    **end for**
**end for**
**Return**: $\theta_E$, $\theta_S$, $\theta_N$, $\theta_T$ and $\theta_G$.
```