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

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

Abstract

Face Video Retouching is a complex task that often requires labor-intensive manual editing. Conventional image retouching methods perform less satisfactorily in terms of generalization performance and stability when applied to videos without exploiting the correlation among frames. To address this issue, we propose a Video Retouching transformEr to remove facial imperfections in videos, which is referred to as VRetouchEr. Specifically, we estimate the apparent motion of imperfections between two consecutive frames, and the resulting displacement vectors are used to refine the imperfection map, which is synthesized from the current frame together with the corresponding encoder features. The flow-based imperfection refinement is critical for precise and stable retouching across frames. To leverage the temporal contextual information, we inject the refined imperfection map into each transformer block for multi-frame masked attention computation, such that we can capture the interdependence between the current frame and multiple reference frames. As a result, the imperfection regions can be replaced with normal skin with high fidelity, while at the same time keeping the other regions unchanged. Extensive experiments are performed to verify the superiority of VRetouchEr over state-of-the-art image retouching methods in terms of fidelity and stability.

1. Introduction

With the development of digital media enhancement, face retouching in videos plays an important role in improving the facial appearance of persons in dynamic sequences. There are a wide range of applications, from professional video production to the burgeoning field of live-streaming and virtual conferencing. These applications require the meticulous removal of visual imperfections. On the other hand, the enhancements should be imperceptible to preserve the subject's natural appearance.

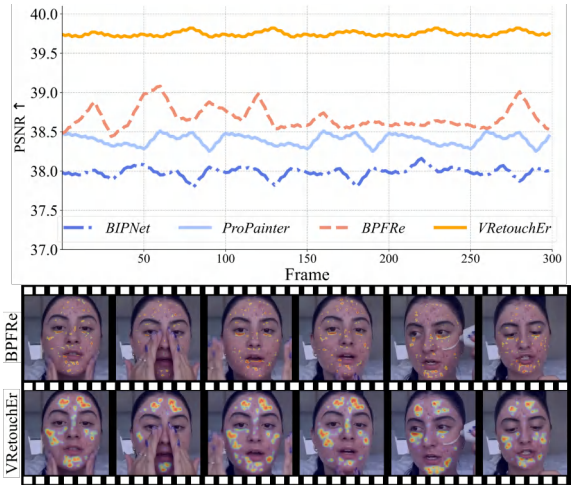

Figure 1. An example to demonstrate the superiority of VRetouchEr over the main competing methods. (*Upper row*) The quantitative results of the methods in terms of PSNR. (*Middle row*) Imperfection detection results of BPFRe. (*Bottom row*) Imperfection detection results of VRetouchEr.

Recent video enhancement methods [23–25] fall short in the context of face video retouching due to their inability to accurately model and track the movement of facial imperfections over time. For instance, the optical flow technique utilized in ProPainter [51] lacks the necessary precision for facial flaw localization. The image-to-image translation methods [21, 33, 48] and GAN-based methods [17, 45, 47] are designed for static image retouching, and fall short in maintaining the performance and stability when addressing image sequences. The deficiency of paired training data further impedes the progress in this domain. Hence, there is a significant gap in current methodologies to perform face video retouching, underscoring the need for handling the

temporal and spatial dynamics of facial imperfections in video streams.

In this work, we address the limitations of existing retouching methods by proposing a face Video Retouching transformEr with facial imperfection flow-based multi-frame attention, which is referred to as VRetouchEr. The core idea of our framework is to incorporate an imperfection flow module to estimate the displacement vectors of imperfections between consecutive frames. Conditioned on the estimation, we can correct the imperfection detection results in each frame via adaptive modulation at specific pixel locations. It is worth noting that the resulting motion information plays an important role in localizing facial imperfection across frames in a stable and precise way. Further, we design a multi-frame masked attention mechanism to synthesize high-fidelity content in the imperfection regions. To achieve this, we allow long-distance interactions between the facial regions from different frames, since multiple frames provide richer information of normal skin characteristics. On the other hand, the estimated imperfection maps are used to weight the intermediate features, such that the imperfections are suppressed. As a result, the attention mechanism enables stable and precise retouching across frames. In summary, the main contribution of this work are as follows: (a) We propose VRetouchEr, a novel face video retouching approach that addresses the challenging of imperfection localization and removal over image sequences. (b) By estimating imperfection flow to correct the imperfection localization in each frame, the obtained spatial information becomes more stable and reliable, which is beneficial to stabilizing the retouching performance. (c) By performing multi-frame masked attention computation, VRetouchEr is able to leverage the contextual information from different frames and synthesize more precise retouching results than existing methods operating on single frame.

2. Related Work

2.1. Image-to-Image Translation

Image-to-image translation aims to learn a mapping across domains of visually distinguishable images, while preserving the content representation to a certain extent [19, 27, 29, 34]. Convolutional Neural Networks (CNN) and Generative Adversarial Networks (GAN) [14] have led to significant progress in this domain. As a representative GAN-based method, Pix2Pix [17] aimed at learning a mapping to minimize the pixel-wise discrepancy. Imposing cycle consistency regularization on the original and the reconstructed images, as in CycleGAN [52] and DiscoGAN [19], is also observed to be effective. A multi-stage method MPRNet [48] leveraged the high-level contextual information and spatial details to improve synthesis results. On the other hand, GPEN [47] combined a U-shaped CNN with a GAN, in which the generative prior was useful for high-quality image generation. To perform effective translation among multiple domains, StarGAN [6,7] learnt a unified translation framework, which was conditional on the domain label. Additionally, the latent space of a pre-trained GAN was observed to possess semantic organization, which allowed semantic editing on images [8, 13, 18]. By leveraging GAN inversion methods [15, 32, 36] to project images back to the latent space, attribute manipulation was performed by imposing the attribute-associated transformations on the resulting latent vectors. To address the multi-frame setting, burst image restoration methods [3, 43] utilized cross-frame cues to merge burst shot frames, which ultimately led to a significant improvement in restoration performance. In addition, BIPNet [11] distilled and aligned frame-wise features through an edge-focused alignment module, and then refined the resolution progressively.

2.2. Vision Transformer

The Vision Transformer (ViT) architecture [10, 44, 46] has emerged as a powerful framework, drawing inspiration from transformer research in natural language processing (NLP) [9, 37]. By incorporating the multi-head attention mechanism, positional encoding and position-wise feed-forward networks, ViT was able to capture the relationships among input features. ViT achieved significant improvements across multiple vision tasks, such as object detection [4, 53], semantic segmentation [39, 50] and image-to-image translation [5, 16]. Building on ViT, the Swin-Transformer [30] improved the original design by arranging the image data in a more organized manner, leading to better results in computer vision tasks. For image synthesis, VQGAN [12] combined the transformer's ability to capture complex relationships with the image generation capability of GANs, resulting in increased realism of images generated from text descriptions. RestoreFormer [42] learnt a dictionary in terms of key-value pairs from high-quality data to restore degraded images. Transformers have also shown to be effective for video inpainting [24, 51] and restoration [25, 26]. ProPainter [51] incorporated optical flow into an inpainting module, and a dual-domain feature propagation approach was adopted to synthesize the content in the masked regions.

2.3. Face Retouching

Face retouching is a specialized task in the domain of image translation, which focuses on the removal of imperfections from human faces in images or videos, while preserving facial attributes as much as possible. Traditional retouching approaches [1,2,22,28,38] typically adopt a global smoothing strategy for the entire face region, which may result in the loss of fine-grained details. AutoRetouch [33] adapted a GAN-based framework to face retouching, and

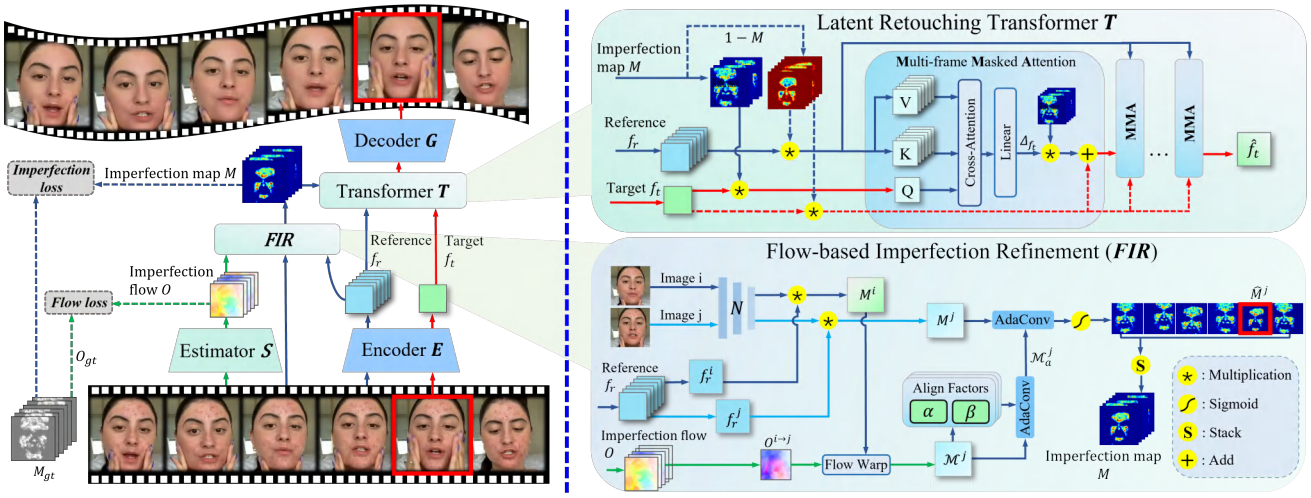

Figure 2. An overview of the proposed VRetouchEr's structure. An Encoder-Transformer-Decoder architecture is specifically designed for Face Video Retouching. VRetouchEr leverages two key modules: the Flow-based Imperfection Refinement (FIR) module and the Multi-frame Masked Attention (MMA) mechanism. FIR performs imperfection flow estimation, and further utilizes the displacement information to refines the imperfection localization on each frame. By injecting the imperfection information into the latent transformation, MMA leads to more accurate and consistent retouching outcomes throughout the frame sequence.

the encoder-decoder-based generator is induced to synthesize clean face images in an adversarial training process. To apply more attention on the local regions, ABPN [21] incorporated context-aware local retouching layers together with an adaptive blend pyramid layer for coarse-to-fine retouching. Similarly, BPFRe [45] performed imperfection detections followed by a two-stage retouching process, in which the intermediate features of a U-net were injected as side information into a StyleGAN generator to perform conditional clean face image generation.

There are fundamental differences between the proposed VRetouchEr and the above existing methods. First, this work focuses on face retouching in videos, while the video enhancement methods, like ProPainter, fall shot in this task due to lack of imperfection localization. Second, the state-of-the-art model, BPFRe, is designed for single-image retouching, thus its performance is unstable when applied to image sequences. To address the limitations, VRetouchEr performs imperfection flow estimation to improve the localization in terms of both stability and precision. Further, we adopt a task-specific transformer architecture by designing multi-frame masked attention mechanism, such that the long-distance relationship between facial features can be effectively captured across frames.

3. Methodology

3.1. Overview

In this section, we elaborate on the design of the proposed approach, VRetouchEr, for high-quality face retouching in videos. Given a raw image sequence X and the corresponding retouched one Y, our goal is to precisely locate the imperfections in X and replace with the content consistent with the surrounding normal skin. The key idea behind VRetouchEr is to leverage the temporal context information to stabilize face retouching over video frames. To achieve this, we adopt an encoder-transformer-decoder based architecture for VRetouchEr as illustrated in Figure 2. An encoder E extracts features of the target frame X_t and multiple reference frames X_r in sequence X, and a flow estimation network S is incorporated to model the motions of imperfections between consecutive frames. The estimated imperfection flow O is fed into a Flow-based Imperfection Localization (FIR) module to improve the estimation stability and precision over the sequence. The resulting imperfection map M, together with extracted target feature f_t and reference feature f_r, are fed into the latent transformer denoted by T to restore the clean face features. In each intermediate block, we can collect richer information of normal skin from the reference frames, and our Multi-frame Masked Attention (MMA) mechanism enables the model to concentrate on the most relevant parts. Finally, the decoder G rebuilds each target frame from the transformed features.

3.2. Flow-based Imperfection Refinement

The FIR module is specifically designed to enhance the accuracy of imperfection localization by leveraging flow information. This process involves imperfection movement estimation, imperfection localization, and imperfection map refinement. The refinement is crucial for stable retouching in each target frame. To achieve this, we define the imperfections are the differences between the manually retouched

frames and raw frames, represented as M_{gt}. We apply a pretrained SpyNet [31] on M_{gt} to produce the ground truth imperfection flow, denoted as O_{gt}.

Our designed FIR module consists of an imperfection localization network N and learnable align factors α, β. Given frame X^i and its consecutive frame X^j, the flow estimation network S is trained to predict an imperfection displacement map $O^{i\to j}$, which is required to approximate the corresponding ground truth imperfection flow as accurately as possible. We evaluate the imperfection flow prediction by measuring the degree of consistency as follows:

$$\begin{aligned} O^{i\to j} &= S(X^i, X^j), \\ \mathcal{L}_{flow} &= \mathbb{E}_X\left[|O^{i\to j} - O_{gt}^{i\to j}|_1\right], \end{aligned} \tag{1}$$

By minimizing $\mathcal{L}_{flow}$, S is encouraged to focus more on the displacement of imperfections instead of background and normal skin regions between consecutive frames.

On the other hand, the imperfection localization network N is trained to detect imperfections from frame X^i together with the corresponding encoder feature f_r^i. We observe that the detection results on individual frames are unstable when there are significant changes in facial postures. To address this issue, we utilize the information from consecutive frames and the estimated imperfection flow $O^{i\to j}$ to refine the detection results. Specifically, the imperfection flow $O^{i\to j}$ will warp M^i into a flow-based imperfection map $\mathcal{M}^j$. Let $\mathcal{W}(\cdot,\cdot)$ denotes the warping operation, and we have:

$$\begin{aligned} M^i &= N(X^i, f_r^i), \\ M^j &= N(X^j, f_r^j), \\ \mathcal{M}^j &= \mathcal{W}(O^{i\to j}, M^i), \end{aligned} \tag{2}$$

where $\mathcal{M}^j$ denotes the warped imperfection map. Next, our designed align factors α and β are employed to align the flow-based imperfection map $\mathcal{M}^j$ with predicted imperfection map M^j. We utilized adaptive convolution blocks [35], denoted as $\theta(\cdot,\cdot)$, for an effective fusion of these features. The refinement operation is defined as follows:

$$\begin{aligned} \mathcal{M}_a^j &= \theta(\alpha * \mathcal{M}^j + \beta, \mathcal{M}^j), \\ \hat{M}^j &= \sigma(\theta\left(M^j, \mathcal{M}_a^j\right)), \end{aligned} \tag{3}$$

where $\hat{M}^j$ represents the refined imperfection map, $\mathcal{M}_a^j$ is the aligned map and σ represents the sigmoid function. All the refined imperfection maps will be stacked together to form the output, denoted as M. The elements in M are normalized in the range from 0 to 1, representing the refined estimation of the imperfections across frames.

3.3. Multi-frame Masked Attention Transformer

Our proposed MMA mechanism enables the latent transformer to modify the features f_t in the imperfection regions by leveraging the features f_r extracted from multiple frames as references. To achieve this, we use the obtained imperfection map M to weight the features of target and reference frames, and perform cross-attention computation between them. Let the number of reference frames be δ, the weighted target feature is used as query, and the weighted reference ones are used as keys and values, defined as follows:

$$\begin{aligned} Q_t &= W_q\left(f_t \otimes M_t + b_q\right), \\ K_r^i &= W_k\left(f_r^i \otimes (1 - M_r^i) + b_k\right), \\ V_r^i &= W_v\left(f_r^i \otimes (1 - M_r^i) + b_v\right), \end{aligned} \tag{4}$$

where M_t and M_r represent the refined imperfection map for target frame and reference frames correspondingly. Next, we perform the cross-attention computation to obtain the modification maps as follows:

$$\Delta_{f_t} = \text{softmax}\left(Q_t \cdot \sum_i^{\delta} K_r^i / \sqrt{\Lambda}\right) \cdot \sum_i^{\delta} V_r^i, \tag{5}$$

where Λ denotes the channel number of the features. In Eq.(4), we adopt the imperfection map M as a soft mask to isolate the regions to be edited, M and $(1 - M)$ represents the mask for imperfection regions and normal regions respectively. Based on this, we can realize the goal of replacing imperfection regions with normal skin by strengthening the interactions between the target features associated with imperfections and the reference features associated with normal skin. Furthermore, we use additive attention in Eq.(5) with linear complexity to aggregate the key and value vectors from multiple reference frame into global vectors, which is used to perform element-wise multiplication to induce temporal context information.

Considering that the background and normal skin in the target frame should be preserved properly, the original feature maps are weighted combined with the modification maps as follows:

$$\hat{f}_t = f_t \otimes (1 - M_t) + \Delta_{f_t} \otimes M_t, \tag{6}$$

where $\hat{f}_t$ denotes the retouched feature, can be fed into the subsequent transformer block as input target feature for further refinement. In this process, the masked attention mechanism enables effective information exchange across frames, and the imperfections in the target frame are progressively suppressed and replaced with the content learnt from normal skin.

3.4. Model Training

The training loss functions of VRetouchEr involves the following three aspects: imperfection flow estimation, imperfection localization and retouching evaluation. As illustrated in Subsection 3.2, we employ a pretrained SpyNet

to produce the ground truth of imperfection flow estimation, and the network S takes consecutive frames as input and predicts the flow as accurately as possible. In addition, the flow is used to improve the imperfection localization on each frame, and we evaluate the precision by measuring the discrepancy between the prediction M and the ground truth imperfections M_{gt} as follows:

$$\mathcal{L}_{imp} = \mathbb{E}_X \|\mathcal{I}(M) - M_{gt}\|_1, \tag{7}$$

where $\mathcal{I}$ denotes a learnable layer to align M's channel number with M_{gt}. We optimize N and S by minimizing $\mathcal{L}_{imp}$ together with the flow estimation loss $\mathcal{L}_{flow}$ defined in Eq.(2), and the formulation is expressed as follows:

$$\min_{S,N} \mathcal{L}_{flow} + \mathcal{L}_{imp}. \tag{8}$$

Let $\hat{y} = G(\hat{f}_t)$ denote the retouched frame synthesized by VRetouchEr. We perform retouching on every raw frame and combine the results together to get the retouched sequence $\hat{Y}$, which is evaluated by measuring the degree of consistency with the manually retouched frames Y as follows:

$$\mathcal{L}_{con} = \mathbb{E}_X \left[\zeta \|Y - \widehat{Y}|_1 + \|\mathcal{V}(Y) - \mathcal{V}(\widehat{Y})\|_2^2\right], \tag{9}$$

where ζ denotes a weighting factor, and $\mathcal{V}(\cdot)$ represents the features extracted from a pre-trained VGG-19. Considering that high-fidelity content synthesis can benefit from adversarial training, we thus adopt the real-fake discrimination loss formulated as follows:

$$\begin{aligned} \mathcal{L}_{adv}^{synt} &= \mathbb{E}_X[\log(D(\hat{Y}))] \\ \mathcal{L}_{adv}^{real} &= \mathbb{E}_X[\log(1 - D(X))] + \mathbb{E}_Y[\log(D(Y))], \end{aligned} \tag{10}$$

where discriminator D takes the sequence as input, and is trained to distinguish the manually retouched frames from the synthesized ones. By integrating the loss functions of retouching evaluation and adversarial training, the optimization formulation of the constituent networks is expressed as follows:

$$\begin{aligned} &\min_{E,T,G} \mathcal{L}_{con} + \mathcal{L}_{adv}^{syn}, \\ &\max_{D} \mathcal{L}_{adv}^{real}. \end{aligned} \tag{11}$$

The constituent networks of the proposed VRetouchEr are jointly optimized from scratch.

4. Experiment

In this section, we perform extensive experiments to assess the effectiveness of VRetouchEr in face video retouching. We first present the details of the datasets and the experimental setup. Next, we compare our VRetouchEr with previous state-of-the-art face retouching methods quantitatively and qualitatively. Finally, we provide insights by analyzing the design elements of VRetouchEr.

4.1. Experimental Settings

Datasets. Due to the difficulty in collecting paired raw-retouched image sequences, we utilized the Flickr-Face-HQ-Retouching (FFHQR) dataset [33] to construct emulated video data. The training/validation/test data in FFHQR consists of 56k/7k/7k raw-retouched image pairs. For each image pair $\{x, y\}$ in FFHQR, we perform random cropping, flipping and translation on both x and y to generate a pair of image sequences $\{X, Y\}$, and the resulting dataset is referred to FFHQR-Seq. In addition, we build another Manually Retouching Face Video dataset, referred to as **MRFV**, which contains 200 in-the-wild portrait videos with different types of facial imperfections, and each of them contains at least 500 frames. We employ multiple retouchers to manually retouch each frame of these videos, and evaluate the retouching performance of our VRetouchEr and the competing methods trained on FFHQR-Seq.

Training Details. The images in both training and test sequences are resized to 512×512 for a fair comparison with the existing face retouching methods. In the training process, the parameters of VRetouchEr are updated by the Adam optimizer [20] with the learning rate of 2×10^{-4}. There are a total of 400k training iterations, the batch size is set to 1, and the hyper-parameter ζ in Eq.(9) is set to 10. We implement VRetouchEr by using PyTorch and train it on single NVIDIA GeForce GTX 3090 GPU.

Evaluation Protocols. In the experiments, we implement all the competing methods by using the open source codes. We adopt the Soft Intersection over Union Loss (Soft-IoU) to measure the correctness of imperfection localization. To measure the consistency between the synthesized image sequences and the manually retouched ones, we adopt the widely used metrics: Peak Signal-to-Noise Ratio (PSNR), Structure Similarity Index Measure (SSIM) and the Learnt Perceptual Image Patch Similarity (LPIPS) [49]. Furthermore, we quantitatively assess the diversity and the degree of realism of synthesized image sequences in terms of Video Fréchet Inception Distance (VFID) [40].

4.2. Comparison to State-of-the-arts

We perform a comprehensive comparative analysis between VRetouchEr and representative state-of-the-art methods, including a typical image translation method: Pix2PixHD [41], generic image restoration methods: MPRNet [48] and RestoreFormer [42], a blind face restoration method: GPEN [47], video enhancement methods: BIPNet [11] and ProPainter [51], and face retouching methods: AutoRetouch [33] and BPFRe [45]. We train BIPNet, ProPainter and VRetouchEr on FFHQR-Seq, and the remaining methods on FFHQR. All the trained models are evaluated on the test data of FFHQR-Seq and MRFV. We quantitatively assess the retouch performance of the meth-

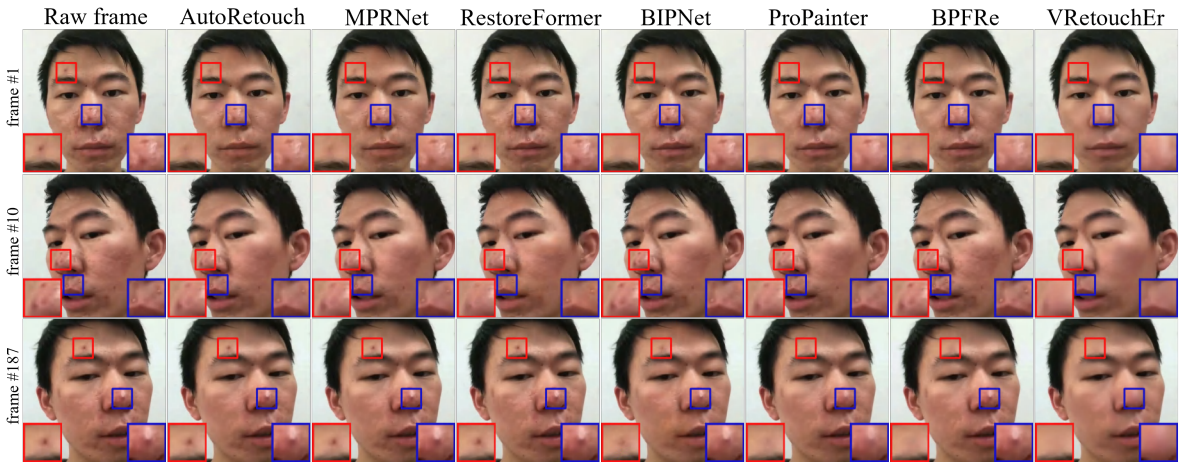

Figure 3. Visual comparison between VRetouchEr and the competing methods on representative frames of an in-the-wild videos.

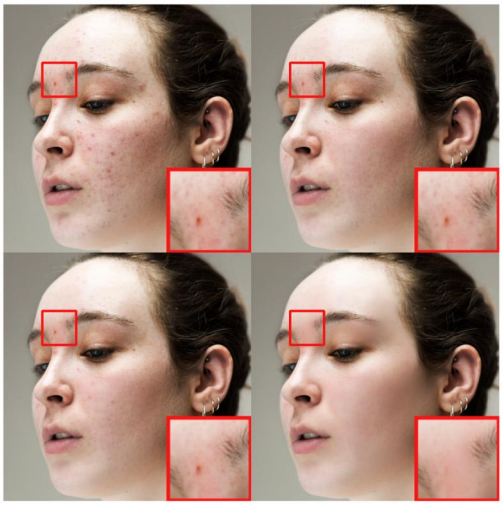

Figure 4. Representative high-quality retouching results. From top left to bottom right, the images are the raw frame and the retouching results of BPFRe, ProPainter and VRetouchEr, respectively.

Table 1. Quantitative comparison between VRetouchEr and competing methods on FFHQR-Seq.

Methods	FFHQR-Seq			
	PSNR↑	SSIM↑	LPIPS ↓	VFID↓
Raw pairs	37.50	0.9704	0.0337	10.779
Pix2PixHD [41]	37.63	0.9714	0.0303	9.252
AutoRetouch [33]	38.27	0.9771	0.0261	8.941
MPRNet [48]	38.33	0.9742	0.0267	8.231
GPEN [47]	38.27	0.9739	0.0256	8.125
RestoreFormer [42]	38.46	0.9733	0.0232	7.931
BIPNet [11]	38.14	0.9711	0.0275	8.241
ProPainter [51]	38.54	0.9768	0.0240	7.872
BPFRe [45]	38.69	0.9774	0.0219	7.604
VRetouchEr	**39.75**	**0.9813**	**0.0169**	**6.375**

ods per frame, and report their average PSNR, SSIM, LPIPS and VFID scores in Tables 1-2. On FFHQR-Seq, the competing methods achieve similar retouching performance except Pix2PixHD, and our VRetouchEr outperforms them in terms of all the metrics. In particular, VRetouchEr achieves the PSNR score of 39.75, which is higher than that of the second best method: BPFRe, by 1.06 dB. On MRFV, VRetouchEr also achieves superior retouching performance over the competing methods. The VFID score of VRetouchEr reaches 10.368, which is lower than that of BPFRe by a significant improvement of about 44.77 percentage points.

Table 2. Quantitative comparison between VRetouchEr and competing methods on MRFV.

Methods	MRFV			
	PSNR↑	SSIM↑	LPIPS ↓	VFID↓
Raw pairs	32.07	0.9054	0.0891	41.830
Pix2PixHD [41]	33.51	0.9128	0.0739	38.403
AutoRetouch [33]	35.03	0.9211	0.0502	31.244
MPRNet [48]	35.82	0.9327	0.0491	29.893
GPEN [47]	35.71	0.9281	0.0463	30.135
RestoreFormer [42]	35.76	0.9299	0.0516	29.785
BIPNet [11]	35.26	0.9231	0.0593	33.897
ProPainter [51]	36.27	0.9384	0.0452	20.344
BPFRe [45]	36.32	0.9427	0.0401	18.772
VRetouchEr	**37.63**	**0.9530**	**0.0357**	**10.368**

In Figure 3, we perform the detailed visual comparison. We can make the following observations: Most of the competing methods fail to remove the marked acnes. BPFRe falls short in maintaining the stability of retouching in videos, since it is designed for single-image retouching. Although ProPainter is specifically designed for video enhancement, it fails to achieve satisfactory retouching results due to lack of imperfection localization. In contrast, our VRetouchEr is able to consistently deliver stable and high-quality retouching outcomes. In Figure 4, we provide visual examples to further compare the performance of our framework with the main competing methods in removing dense

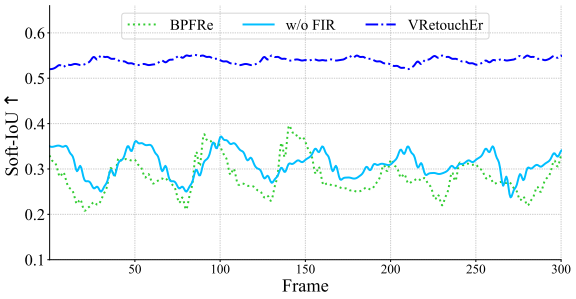

Figure 5. Quantitative comparison in imperfection localization per frame. VRetoucher achieves a more precise and stable result than 'VRetoucher w/o FIR' and BPFRe.

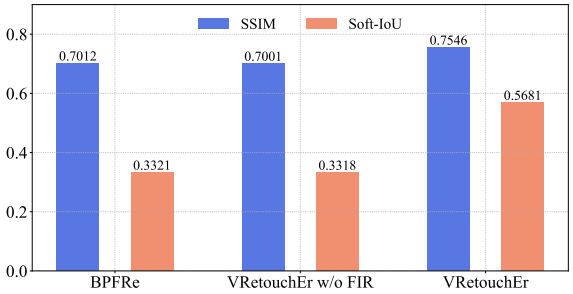

Figure 6. Quantitative comparison in imperfection localization on MRFV dataset.

imperfections. These results highlight the VRetouchEr's capability of face retouching in videos even without any manually retouched videos as training data.

4.3. User Study

We further perform a user study to evaluate VRetouchEr and the competing methods in terms of human perception. The models trained on FFHQR/FFHQR-Seq are applied to a total of 40 raw videos randomly sampled from MRFV. There are 50 participants, each of which is presented with the synthesized videos and required to carefully compare and sort the retouching results, according to the criteria including overall visual quality, effectiveness of imperfection removal, naturalness, and consistency. Table 3 shows the ranking results, and we are able to draw the conclusion regarding the superior performance of VRetouchEr against the competing methods.

4.4. Flow-based Imperfection Refinement

We observed that our FIR module plays in an important role in facilitating face retouching in videos. Different from

Table 3. The voting result (%) of user study on MRFV.

Method	Rank-1	Rank-2	Rank-3	Rank-4	Rank-5	Rank-6
VRetouchEr	**88.45**	11.05	0.30	0.20	0.00	0.00
BPFRe [45]	11.15	**63.15**	19.35	5.05	1.20	0.10
ProPainter [51]	0.25	18.20	**61.60**	16.30	3.15	0.50
MPRNet [48]	0.05	6.05	11.65	**59.25**	6.40	16.60
RestoreFormer [42]	0.05	1.50	6.10	11.25	**61.30**	19.80
BIPNet [11]	0.05	0.05	1.00	7.95	27.95	**63.00**

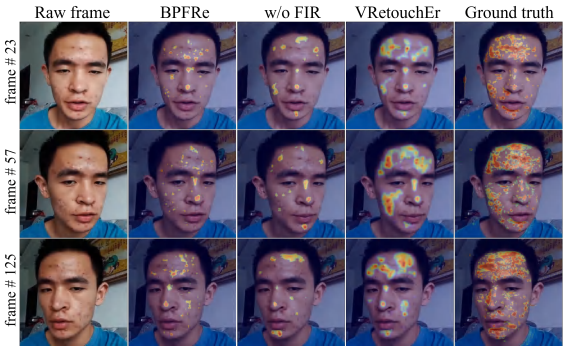

Figure 7. Visual comparison in imperfection localization.

BPFRe detecting facial imperfections on each frame separately, we estimate imperfection flow between consecutive frames followed by imperfection refinement. The displacement information is useful for refining and stabilizing the imperfection maps of the video to be edited. In Figure 5, we quantitatively evaluate imperfection localization results on a video sequence in terms of Soft-IoU per-frame between the results and the ground truth imperfection maps M_{gt} as defined in Sec. 3.2. In Figure 6, we perform the evaluation of imperfection localization between the results and M_{gt} in terms of mean SSIM and mean Soft-IoU on MRFV dataset. When disabling FIR, the variant 'VRetouchEr w/o FIR' performs single-image imperfection localization, and has comparable performance with BPFRe. We can find that FIR leads to a significant and stable improvement for cross-frame imperfection localization. We further visualize the imperfection maps of BPFRe, 'VRetouchEr w/o FIR' and VRetouchEr in Figure 7, and observe that VRetouchEr can stably locate the most imperfections, and its results are more consistent with the ground truth.

4.5. Impact of the Number of Reference Frames

The existing face retouching methods suffer from the problem of overlooking temporal context information in handling videos. As another important module of VRetouchEr, we consider that our designed MMA mechanism also contributes to the superior performance of VRetouchEr. To verify this, we investigate the impact of the number of reference frames δ on the final retouching performance. In Figure 8, we plot the VFID scores of VRetouchEr on MRFV during training. For the case of δ=1, we perform single-image retouching on each frame. One can observe that increasing the value of δ for MMA indeed leads to a significant improvement. In particular, VRetouchEr achieves the best retouching performance when δ=6. Beyond this value, VRetouchEr's performance becomes stable. In Figure 9, we visualize the retouching results to further verify this observation. This insight provides a valuable guidance for optimizing the performance of VRetouchEr in the face video retouching task.

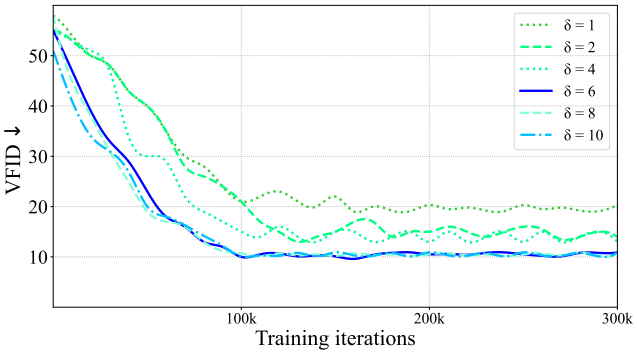

Figure 8. Quantitative results of VRetouchEr with different δ.

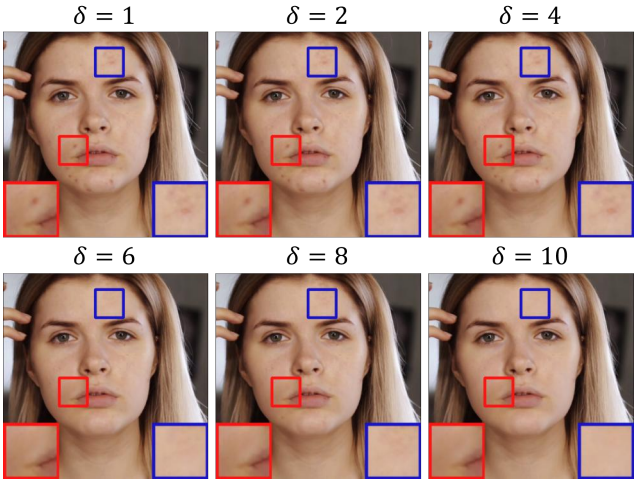

Figure 9. The retouching results of VRetouchEr with different δ.

4.6. Ablation Study

The two proposed modules: FIR and MMA, distinguish the proposed VRetouchEr from the existing works. We perform a set of ablative experiments to investigate the impact of FIR and MMA on the retouching performance on FFHQR-Seq. Specifically, we build the Base model of VRetouchEr by disabling both FIR and MMA, and find that VRetouchEr significantly outperforms the base model as shown in Table 4. Additionally, we build two variants by disabling FIR and MMA, and the resulting models are referred to as 'VRetouchEr w/o FIR' and 'VRetouchEr w/o MMA', respectively. In 'VRetouchEr w/o FIR', the maps produced by the imperfection localization network N are directly fed into the latent transformer. As analyzed in Sec.4.4, disabling FIR leads to unstable imperfection localization, which ultimately degrades the retouching performance on videos. As shown in Table 4, the PSNR score of 'VRetouchEr w/o FIR' is 38.74 dB, which is worse than that of VRetouchEr by 1.01 dB. When disabling MMA, we also observe a significant performance drop of about 40.83 percentage points in terms of LPIPS. Qualitative results in Figure 10 also prove the effectiveness of FIR and MMA.

Table 4. Quantitative results of ablative models.

Variants	FFHQR-Seq			
	PSNR↑	SSIM↑	LPIPS ↓	VFID↓
Base model	38.13	0.9738	0.0264	8.671
VRetouchEr w/o MMA	38.67	0.9758	0.0238	7.534
VRetouchEr w/o FIR	38.74	0.9769	0.0207	7.312
VRetouchEr	**39.75**	**0.9813**	**0.0169**	**6.375**

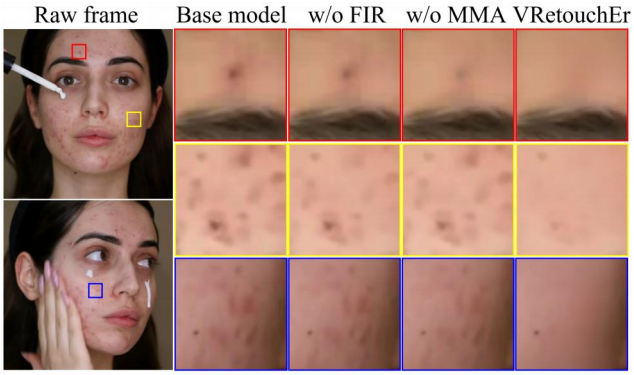

Figure 10. Representative retouching results of ablative models.

5. Conclusion

In this work, we propose a latent retouching transformer for removing facial imperfections in videos. Different from the existing methods that focus on single-image retouching, we leverage the temporal context information to significantly facilitate face retouching in videos. Toward this end, we perform imperfection flow estimation to obtain the displacement information between consecutive frames, and further refine the imperfection localization by fusing motion-based and frame-based predictions. By injecting the imperfection maps as side information into a latent retouching transformer, we adopt the masked attention computation over multiple frames, such that the features of normal skin from different frames can be leveraged to substitute for that of imperfections in the target frame. Extensive experiments are performed to demonstrate the superior capability of the proposed approach over state-of-the-art face retouching methods in stabilizing the retouching performance over video frames.

6. Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (Project No. 62072189), in part by the Research Grants Council of the Hong Kong Special Administration Region (Project No. CityU 11206622), in part by the GuangDong Basic and Applied Basic Research Foundation (Project No. 2020A1515010484, 2022A1515011160), and in part by TCL Science and Technology Innovation Fund (Project No. 20231752).

References

[1] Kaoru Arakawa. Nonlinear digital filters for beautifying facial images in multimedia systems. In *Proc. IEEE International Symposium on Curcuits and Systems*, 2004. 2

[2] Nazre Batool and Rama Chellappa. Detection and inpainting of facial wrinkles using texture orientation fields and Markov random field modeling. *IEEE Transactions on Image Processing*, 23(9):3773–3788, 2014. 2

[3] Goutam Bhat, Martin Danelljan, Luc van Gool, and Radu Timofte. Deep burst super-resolution. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, June 2021. 2

[4] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *Proc. European Conference on Computer Vision*, 2020. 2

[5] Hanting Chen, Yunhe Wang, Tianyu Guo, Chang Xu, Yiping Deng, Zhenhua Liu, Siwei Ma, Chunjing Xu, Chao Xu, and Wen Gao. Pre-trained image processing transformer. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2021. 2

[6] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. StarGAN: unified generative adversarial networks for multi-domain image-to-image translation. In *Proc. IEEE conference on Computer Vision and Pattern Recognition*, pages 8789–8797, 2018. 2

[7] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. StarGAN v2: diverse image synthesis for multiple domains. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, June 2020. 2

[8] Edo Collins, Raja Bala, Bob Price, and Sabine Susstrunk. Editing in style: uncovering the local semantics of GANs. In *Proc. IEEE conference on Computer Vision and Pattern Recognition*, 2020. 2

[9] Jocob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In *arXiv:1801.04805*, 2018. 2

[10] Alexy Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. 2

[11] Akshay Dudhane, Syed Waqas Zamir, Salman Khan1, Fahad Shahbaz Khan, and Ming-Hsuan Yang. Burst image restoration and enhancement. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, June 2022. 2, 5, 6, 7

[12] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2021. 2

[13] Lore Goetschalckx, Alex Andonian, Aude Oliva, and Philip Isola. GANalyze: toward visual definitions of cognitive image properties. In *Proc. International Conference on Computer Vision*, 2019. 2

[14] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Proc. Neural Information Processing Systems*, 2014. 2

[15] Jinjin Gu, Yujun Shen, and Bolei Zhou. Image processing using multi-code GAN prior. In *Proceedings of IEEE conference on computer vision and pattern recognition*, pages 3012–3021, 2020. 2

[16] Xueqi Hu, Qiusheng Huang, Zhengyi Shi, Siyuan Li, Changxin Gao, Li Sun, and Qingli Li. Style transformer for image inversion and editing. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2022. 2

[17] Philip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 1, 2

[18] Ali Jahanian, Lucy Chai, and Phillip Isola. On the "steerability" of generative adversarial networks. In *arXiv:1907.07171*, 2019. 2

[19] Taeksoo Kim, Moonsu Cha, Hyunsoo Kim, Jung Kwon Lee, and Jiwon Kim. Learning to discover cross-domain relations with generative adversarial networks. In *Proc. International Conference on Machine Learning*, 2017. 2

[20] Diederik P. Kingma and Jimmy Lei Ba. Adam: a method for stochastic optimization. In *Proc. International Conference on Learning Representation*, 2015. 5

[21] Biwen Lei, Xiefan Guo, Hongyu Yang, Miaomiao Cui, Xuansong Xie, and Di Huang. Abpn: Adaptive blend pyramid network for real-time local retouching of ultra high-resolution photo. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, June 2022. 1, 3

[22] Tommer Leyvand, Daniel Cohen-Or, Gideon Dror, and Dani Lischinski. Data-driven enhancement of facial attractiveness. In *Proc. ACM Conference on Special Interest Group on Computer Graphics and Interactive Techniques*, 2008. 2

[23] Ang Li, Shanshan Zhao, Xingjun Ma, Mingming Gong, Jianzhong Qi, Rui Zhang, Dacheng Tao, and Ramamohanarao Kotagiri. Short-term and long-term context aggregation network for video inpainting. In *Proc. European Conference on Computer Vision*, 2022. 1

[24] Zhen Li, Cheng-Ze Lu, Jianhua Qin, Chun-Le Guo1, and Ming-Ming Cheng. Towards an end-to-end framework for flow-guided video inpainting. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, June 2022. 1, 2

[25] Jing Lin, Yuanhao Cai, Xiaowan Hu, Haoqian Wang, and Youliang Yan. Flow-guided sparse transformer for video deblurring. In *Proc. International Conference on Machine Learning*, 2022. 1, 2

[26] Jing Lin, Xiaowan Hu, Yuanhao Cai, and Haoqian Wang. Unsupervised flow-aligned sequence-to-sequence learning for video restoration. In *Proc. International Conference on Machine Learning*, 2022. 2

[27] Huan Ling, Karsten Kreis, Daiqing Li, Seung Wook Kim, Antonio Torralba, and Sanja Fidler. EditGAN: high-precision semantic image editing. In *Proc. Neural Information Processing Systems*, 2021. 2

[28] Uri Lipowezky and Sarah Cahen. Automatic freckles detection and retouching. In *Proc. IEEE Convention of Electrical and Electronics Engineers in Israel*, 2008. 2

[29] Ming-Yu Liu, Thomas Breuel, and Jan Kautz. Unsupervised image-to-image translation networks. In *Proc. Neural Information Processing Systems*, 2017. 2

[30] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: hierarchical vision transformer using shifted windows. In *Proc. International Conference on Computer Vision*, 2021. 2

[31] Anurag RanjanMichael and J. Black. Optical flow estimation using a spatial pyramid network. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, June 2017. 4

[32] Elad Richardson, Yuval Alaluf, Or Patashnik, Yotam Nitzan, Yaniv Azar, Stav Shapiro, and Daniel Cohen-Or. Encoding in style: a StyleGan encoder for image-to-image translation. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 2287–2296, 2021. 2

[33] Alireza Shafaei, James J. Little, and Mark Schmidt. AutoRetouch: automatic professional face retouching. In *Proc. IEEE Winter Conference on Applications of Computer Vision*, 2021. 1, 2, 5, 6

[34] Yujun Shen, Jinjin Gu, Xiaoou Tang, and Bolei Zhou. Interpreting the latent space of GANs for semantic face editing. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 2

[35] Hang Su, Varun Jampani, Deqing Sun, Orazio Gallo, Erik Learned-Miller, and Jan Kautz. Pixel-adaptive convolutional neural networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 4

[36] Omer Tov, Yuval Alaluf, Yotam Nitzan, Or Patashnik, and Daniel Cohen-Or. Designing an encoder for StyleGan image manipulation. *ACM Transactions on Graphics*, 40(4):1–14, 2021. 2

[37] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proc. Neural Information Processing Systems*, 2017. 2

[38] Sudha Velusamy, Rishubh Parihar, Raviprasad Kini, and Aniket Rege. FabSoften: face beautification via dynamic skin smoothing, guided feathering and texture restoration. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition Workshop*, 2020. 2

[39] Huiyu Wang, Yukun Zhu, Hartwig Adam, Alan Yuille, and Liang-Chieh Chen. MaX-DeepLab: end-to-end panoptic segmetation with mask transformers. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2021. 2

[40] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Guilin Liu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. Video-to-video synthesis. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, June 2018. 5

[41] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional GANs. In *Proc. IEEE conference on Computer Vision and Pattern Recognition*, pages 8798–8807, 2018. 5, 6

[42] Zhouxia Wang, Jiawei Zhang, Runjian Chen, Wenping Wang, and Ping Luo. RestoreFormer: high-quality blind face restoration from undegraded key-value pairs. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2022. 2, 5, 6, 7

[43] Pengxu Wei, Yujing Sun, Xingbei Guo, Chang Liu, Guanbin Li, Jie Chen, Xiangyang Ji, and Liang Lin. Towards real-world burst image super-resolution: Benchmark and method. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, June 2023. 2

[44] Haiping Wu, Bin Xiao, Noel Codella, Mengchen Liu, Xiyang Dai, Lu Yuan, and Lei Zhang. CvR: introducing convolutions to vision transformers. *arXiv:2103.15808*, 2021. 2

[45] Lianxin Xie, Wen Xue, Zhen Xu, Si Wu, Zhiwen Yu, and Hau San Wong. Blemish-aware and progressive face retouching with limited paired data. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, June 2023. 1, 3, 5, 6, 7

[46] Rui Xu, Xiaoxiao Li, Bolei Zhou, and Chen Change Loy. Deep flow-guided video inpainting. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 2

[47] Tao Yang, Peiran Ren, Xuansong Xie, and Lei Zhang. GAN prior embedded network for blind face restoration in the wild. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2021. 1, 2, 5, 6

[48] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, Ming-Huang Yang, and Ling Shao. Multi-stage progressive image restoration. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2021. 1, 2, 5, 6, 7

[49] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 5

[50] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip H.S. Torr, and Li Zhang. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021. 2

[51] Shangchen Zhou, Chongyi Li, Kelvin C.K. Chan, and Chen Change Loy. Propainter: Improving propagation and transformer for video inpainting. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, June 2023. 1, 2, 5, 6, 7

[52] Jun-Yan Zhu, Taesung Park, Philip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proc. International Conference on Computer Vision*, 2017. 2

[53] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable DETR: deformable transformers for end-to-end object detection. In *International Conference on Learning Representations*, 2021. 2