FabSoften: Face Beautification via Dynamic Skin Smoothing, Guided Feathering, and Texture Restoration

Sudha Velusamy, Rishubh Parihar, Raviprasad Kini, Aniket Rege
Samsung R&D Institute
Bangalore, India
{sudha.v, r.parihar, ravi.kini, a.devdattar}@samsung.com

Abstract

Face retouching is a widespread application in modern smartphone cameras with its high business value evidenced by its broad user base. We propose a real-time face softening approach that smooths blemishes in the facial skin region, followed by a wavelet band manipulation to restore the underlying skin texture, which produces a highly appealing 'beautified' face that retains its natural appearance. Softening is carried out by an attribute-aware dynamic smoothing filter that is guided by facial attributes, including the number of blemishes and coarseness of facial skin texture. The proposed solution is robust to wide variations in lighting conditions, skin nonuniformities, blemishes, the presence of facial accessories, and delicate hair-like regions. The method includes an explicit facial hair preservation module to preserve their delicate texture while smoothing blemishes. We perform a qualitative comparison of our proposed face softening approach with numerous state-of-the-art techniques and commercial products. We demonstrate the power of our method in producing beautified faces at a minimal performance cost, which enables smooth execution on low-power devices like smartphones.

1. Introduction

Face retouching is among the most popular applications available across the entire spectrum of modern smartphone cameras, influenced by its transformative capability on facial images. Its importance and market relevance is further indicated by the presence of many face beauty related applications in the top 100 applications on Google Play [3], such as *YouCam* [4], *B612* [2] and *ModiFace* [1]. While a universal definition of face beauty remains debatable, existing literature shows that face smoothing, also called face softening, is a primary component in improving facial attractiveness [14, 11].

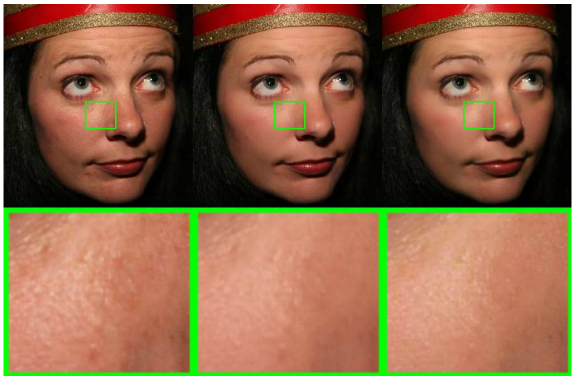
(a) Input (b) ModiFace [1] (c) FabSoften

Figure 1: A visual comparison of face beautification results.

Traditional face softening solutions apply general-purpose smoothing filters to address skin imperfections [32, 21, 7]. However, these methods do not differentiate between inherent facial factors that we wish to retain and the blemishes that we wish to smoothen. Here, facial factors refer to skin texture, facial features (such as nose, eyes, and lips), facial hair, and accessories. Smoothing with these filters thus results in over-smoothed faces that look artificial and unnatural.

In this work, we propose a novel face beautification approach that carries out dynamic skin smoothing for smoothing blemishes in the facial region, including wrinkles, spots, patchy reflections, and skin nonuniformities. Inspired by He and Sun [8], we have applied Guided Feathering to preserve hair-like regions within the face, such as the beard, mustache, and eyebrows. We aim to produce a softened face that looks natural and closely resembles the original face by retaining fine-grained skin texture and not altering the structures of essential facial features. To this end, we introduce a wavelet-based skin texture restoration module that selectively manipulates and combines the wavelet bands of the original and smoothed images to restore the fine-grained skin texture that was lost during smoothing. We design our approach to perform real-time face beautification that is capable of running on modern smartphone cameras. We henceforth refer to our method **Fa**ce **B**eautification via Dynamic Skin **S**m**o**othing, Guided **F**eathering and **T**exture Restoratio**n** as **FabSoften**. Figure 1 shows an exemplar face, and the face softening carried out by the commercial product ModiFace [1] compared to our proposed solution, FabSoften.

2. Related Work

Prior work in the area of face softening can be grouped into four dominant classes: 1) Edge-Aware Smoothing Filters; 2) Layer Decomposition Based Approaches; 3) Deep Learning-Based Approaches; 4) Generative Models. We will discuss each class of approach in detail below.

Significant work in the domain of noise filtering is present in the existing literature. Traditionally, face softening techniques utilized edge-aware smoothing filters that preserve facial feature structures while smoothing blemishes [34, 32, 21, 7]. Dating back to anisotropic smoothing, the edge-aware methods that were developed for face softening were categorized into two primary classes: local and global kernels. Local approaches like the Bilateral Filter [27] and Weighted Median Filter [34] were designed to consider image statistics within a local window, while global methods were designed to remove insignificant details without destroying dominant structures via optimizing a global energy function [36]. Representative global methods include $L0$ smoothing [32] and Fast Global Smoothing (FGS) [21], which introduced weighted tree filtering, where the weights are determined by distance, color, and pixel connectedness. Additional examples include Domain Transform (DT) [26] and Mutually Guided Filter [7].

Alternatively, several methods that decompose the face image into multiple layers to improve their perceived attractiveness have been proposed [28, 31, 19]. Lu *et al*. [31] first decompose a face image into three constituent layers: lighting, detail, and color. They maintain a learned detail layer dictionary of beautified face images and transform the detail layer of the input to the learned one. Utilizing the detail layer from the dictionary in this manner will result in smoothing the skin region well. However, in these methods, the restored skin texture details do not represent the original skin texture well. Boyadzhiev *et al*. [5] decompose an image into sub-bands based on frequency, amplitude, and the sign of the coefficients. They have introduced band-sifting operators to manipulate these coefficients and generate various plausible appearances for an object. They have also used these operators to produce variations in the appearance of oiliness, glow, wrinkles in the skin region.

Since the relatively recent advent of deep learning, there have been several deep neural network-based smoothing techniques developed for the problem of face softening [25, 15, 13]. For example, Xu *et al*. [15] proposed a Convolutional Neural Network (CNN) based method to predict smoothed image gradients, and then run an expensive step to reconstruct the smoothed image itself. Li *et al*. [13] employed a deep learning model to modify the geometry of the face shape to achieve higher attractiveness. Specifically, they use the back-propagated gradient flow to modify facial landmarks with the guidance of an attractiveness score. These techniques suffer from a lack of explainability of the network's decision, and also consume significant resources during inference runtime.

The growing popularity of generative models for a variety of image generation tasks [30] has recently extended to face editing. Diamant *et al*. [22] proposed a generative model that can beautify any face image conditioned on the desired level of beautification. Liu *et al*. [10] developed a two-stage deep network for automatic group photo beautification by editing facial expressions. Though these approaches tend to produce realistic results, there is very little user control in achieving beautification in the desired fashion. Chen *et al*. [33] designed a face attribute manipulation system in which the face image is decomposed into its semantic components for selective processing of each facial region. Region-specific processing provided the previously lacking control of editing each facial attribute separately. Nevertheless, employing generative models for face beautification results in synthesized skin textures, which is often unacceptable due to their unrealistic appearance [30]. Finally, mobile applications such as *B612* [2] and *Modiface* [1] are popular among digital consumers, and provide a considerable measure of control to the user over the beautification process. We, however, take inspiration from work done by Lee *et al*. [17], who perform skin smoothing for portrait images via segmenting the skin region, as detailed in Section 3.2. They utilize the segmentation mask to smooth the original image and blend this smoothed image with the original image for beautification.

Most of the above mentioned classical and neural network-based solutions focused on two primary problems: $i)$ Edge-preserving noise or blemish smoothing. $ii)$ Retaining facial skin texture. Our comparative evaluation of these facial beautification techniques, as shown in Section 4, proves the need for a unified framework to address the problem of face softening. In this work, we propose a method of face beautification via skin softening with restored skin texture. The primary contributions of this paper are: 1) Attribute-Aware Dynamic Filter (ADF) for smoothing facial blemishes, deep wrinkles, nonuniformities in skin texture, and reflections. 2) Wavelet-based Skin Texture Restoration (STR) to selectively manipulate the wavelet bands across the input and smoothed image. This module restores the fine-grained original skin texture to obtain a natural-looking softened face. 3) Guided Feathering (GF)

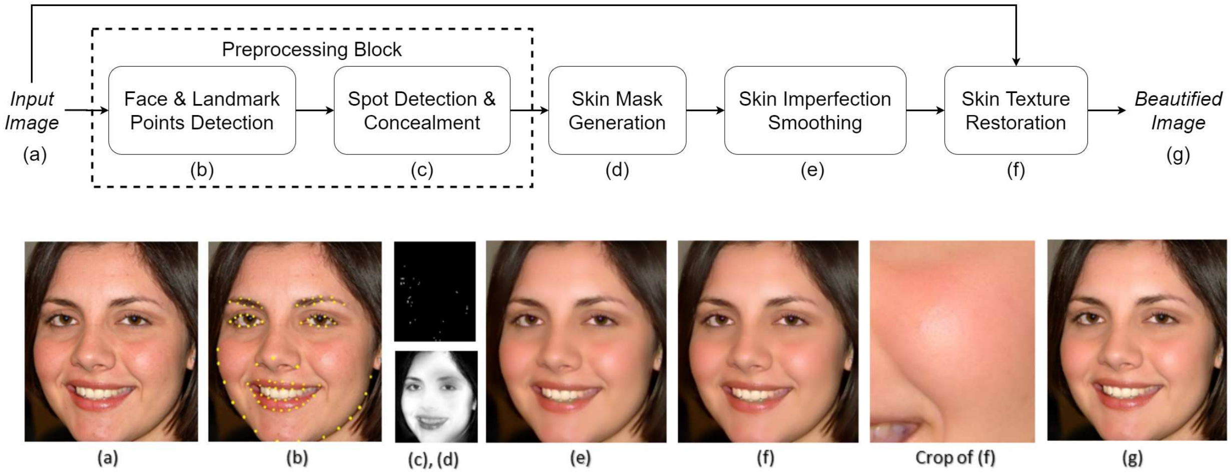

Figure 2: Top: a flow diagram of the proposed facial beautification pipeline. Bottom: a) Original image b) Detected facial landmarks[12] c) Detected spots d) Skin mask e) ADF: Result after skin smoothing f) STR: Result after texture restoration g) Final output.

technique to generate a refined skin-probability mask for hair-like regions (eyebrows, mustache, beard) to retain the sharpness of their delicate texture.

3. Proposed Method

Facial skin softening is a process of reducing the skin imperfections, without altering the underlying skin textures and facial features. After an initial preprocessing step to extract the face and remove large blemishes, we introduce three novel contributions: a) ADF Module for optimal facial smoothing. b) STR module for skin texture retention. c) GF module for hair region preservation. Figure 2 shows the flow diagram of the proposed skin-softening algorithm. These contributions are explained in more detail in this section.

3.1. Preprocessing

Before employing the ADF for skin softening, we require a preprocessing step to generate a crude skin mask and remove large blemishes in the skin region. We consider a blemish to be a low-intensity imperfection in the skin that is localized to a small region. This step is essential as filtering does not eliminate large blemishes, and they require additional spatial concealment. We first detect the faces present in the image and their corresponding 68 facial landmark points using a landmark point detector [12]. We then join the landmark points on the boundary of each facial feature (eyes, nose, lips) using cubic curves to obtain an approximate outer contour of these facial features. To get the approximate face shape, we join the landmark points located at the facial boundary and complete the upper face region, which lacks landmarks, using an ellipse. To generate a binary skin mask, we fill skin regions with ones and non-skin features with zeros. To detect blemishes in the skin region, we first employ the Canny Edge detector [6] to localize strong edge patterns. We perform a depth-first traversal from these detected edge pixels for a depth of 200 to find strong, connected edges that are likely to represent blemish boundaries. If a loop is formed during this traversal, we consider it as a large blemish. From this point forward, we refer to these large blemishes as ”spots.” We further prune these spots using their edge magnitude and shape information to reject weak candidates, as shown in Fig. 2c. Finally, we conceal detected spots by replicating pixel values from their neighboring blemish-free skin patches.

3.2. Skin Mask Generation

Retaining the sharpness of facial features and fine hair-like regions such as eyebrows, eyelashes, beards, and mustaches are essential for natural-looking face beautification. Fine individual textures such as hair strands are very susceptible to blurring effects during face softening due to their proximity to adjacent equally thin textures. A great degree of care must thus be taken during face softening with larger kernels to prevent blurring artifacts.

Traditional skin-mask generation algorithms generally fall into three broad categories: 1) Color pixel classification [29, 24], which generally apply Bayesian classifiers with histogram techniques. These techniques often fail due to the high variability of skin coloration while segmenting out hair regions. 2) Gaussian Mixture Model (GMM) methods [17], which use facial feature alignment to create

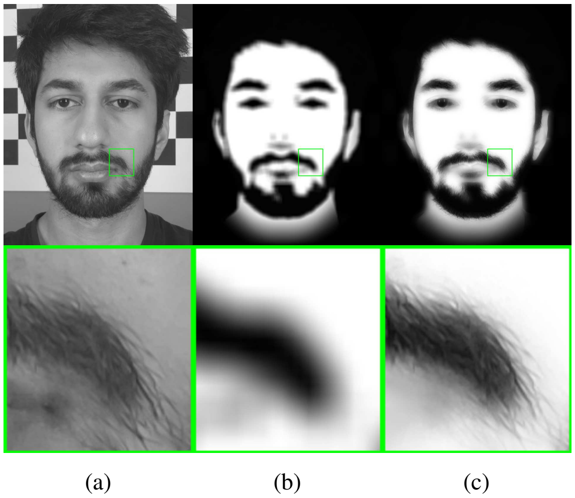

Figure 3: Skin mask refinement using Guided Feathering (GF). a) Original Image b) Approximate mask [17] c) Refined mask. The representative image was taken from a local, internally maintained database.

an approximate skin map, and perform skin/non-skin pixel classification via the GMM. This results in a skin probability mask which requires an additional edge-refinement step to distinguish fine facial features (such as hair regions) from the skin. 3) Deep learning-based facial feature segmentation approaches [20] [35]. These techniques employ Convolutional Neural Networks (CNNs) to perform pixel-accurate semantic segmentation over facial features. Specifically for beautification, Liang *et al.* [14] proposed adaptive, region-aware facial skin masks for facial skin beautification. This method suffers from hair regions lacking precision and requiring human intervention for hyper-parameter tuning. Our GF module retains fine hair textures while utilizing fixed parameters, thereby requiring no additional refinement steps, and is robust to skin color variation.

Inspired by Guided Filtering [8], we use our spot-concealed input face image from the preprocessing step as a guide to refine the skin-probability mask. This pixel-based approximate skin probability mask for skin and non-skin regions is computed as introduced by Lee *et al.* [17] and illustrated in Fig. 3b. Feathering results in a highly refined skin mask, as seen in Fig. 3c, with the hair regions restored. The window radius parameter r and the regularization parameter ϵ of the Guided Filter are tuned to smoothen strongly in skin regions and simultaneously preserve delicate feathery hair-like textures. In our experiments, we observed $r = 10$ and $\epsilon = 200$ values performed best for our purpose. We further observed that while using a higher value of r increased the intensity of smoothing, it caused intolerable blurring artifacts in hair regions. As we performed pixel-based approximate skin probability computation, some of the pixels located inside the eyes, lips, and inner mouth regions are also being classified wrongly as skin pixels because of similar local appearance. To address this, we have used the contours of these features as defined in Section 3.1 and filled the regions inside with zeros in the refined skin mask.

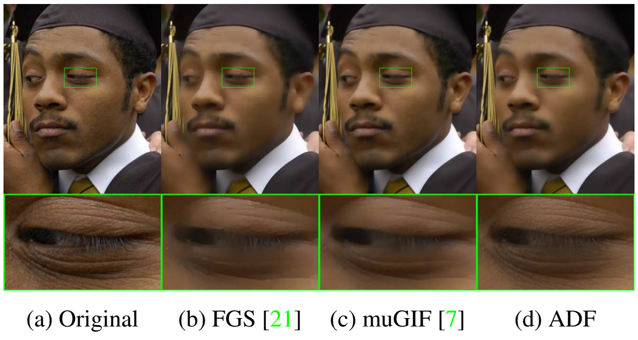

Figure 4: Comparison of Edge-aware filtering methods.

3.3. Skin Imperfection Smoothing

For smoothing skin imperfections, it is essential to design an appropriate smoothing filter that preserves facial features like eyes, lip-lines, eyebrows, while smoothing out skin imperfections alone. For this purpose, we experimented with several edge-preserving filters, including the Fast Global Smoothing Filter (FGS) [21], Guided Filter [8], and Mutually Guided Image Filter (muGIF) [7]. Figure 4 shows a visual comparison of these filters compared to the ADF module applied in the skin smoothing context. When the smoothing strength is low, the FGS Filter performs poorly and retains a significant number of blemishes. Increasing the smoothing strength creates flat patches in the skin region. This effect is especially noticeable when the face is illuminated unevenly. The muGIF provides a better balance between preserving details and skin smoothing than the FGS filter. However, both these filters perform poorly in hair regions by blurring finer details, as seen in Fig. 4b and 4c. We utilize an Attribute-aware Dynamic Guided Filter (ADF) that strikes the optimal balance between good skin smoothing and preserving fine details in hair regions, as shown in Fig. 4d. We modify the original Guided Filter proposed by He *et al.* [8], as detailed below. The Guided Filter provides a degree of control over the amount of smoothing and kernel size via its parameters ϵ and r, respectively. It divides an image into windows with radius r and performs smoothing over these windows.

We introduce the ADF module wherein the parameters r and ϵ are not fixed across image windows but are computed based on input image statistics. The parameter r is the window radius and represents the influence window of a kernel centered at pixel i. A higher value of r implies the influence of a pixel i is spread over a large spatial location, which results directly in a higher amount of smooth-

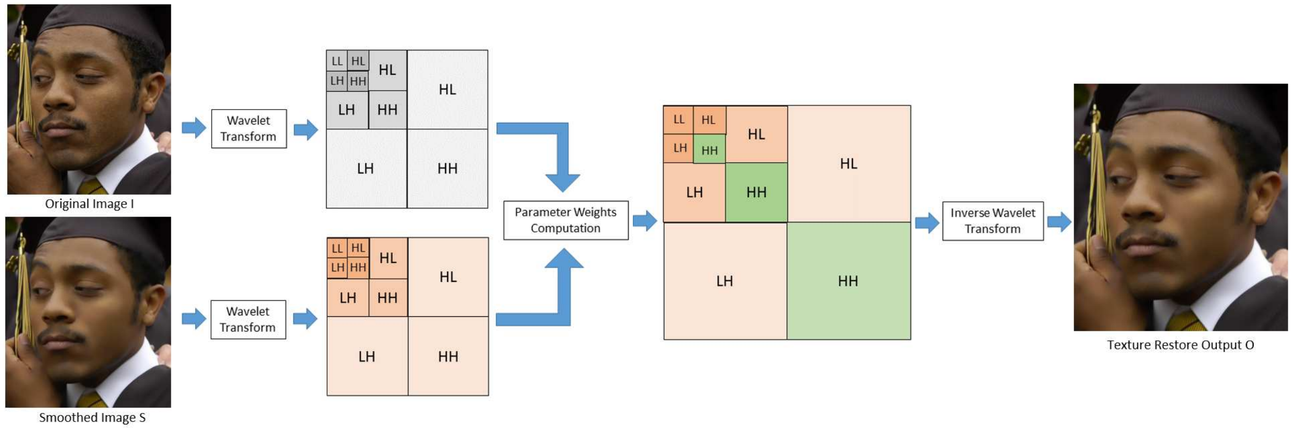

Figure 5: STR pipeline: skin texture restoration using wavelet domain processing.

ing. In the case of face smoothing, we desire more smoothing at uniform skin regions like the cheeks and forehead. On the other hand, for facial features like the eyes, lips, and nose that have clearly defined boundaries and fine textures, less or no smoothing is desired. To this end, we have used the probability-based skin mask computed in Section 3.2 to select the kernel radius for every window. Specifically, for each window, we defined the window radius as a linear transformation of the skin probability at the window's central pixel location i:

$$r_i = \alpha_r * (P_{skin}(i)) + \beta_r \tag{1}$$

where $P_{skin}(i)$ is the skin probability at central pixel location i. If the central pixel is a skin pixel, then $P_{skin}(i)$ will be close to 1, and its corresponding window radius will be larger as desired for uniform skin regions. On the other hand, if the central pixel is not a skin pixel, then $P_{skin}(i)$ will be close to 0, then the window radius will be a smaller value near β_r, achieving minimal smoothing in these regions.

The regularization parameter ϵ controls the amount of smoothing. A higher ϵ value results in a higher level of smoothing. We have used a fixed global value of ϵ based on the number of spots present in the face, as calculated in the preprocessing step in Section 3.1. Recognizing that a higher number of spots calls for additional smoothing, we defined ϵ to be the linear transformation of the number of spots:

$$\epsilon = \alpha_\epsilon * (\text{num}_{\text{spots}}) + \beta_\epsilon \tag{2}$$

This formulation results in a higher amount of smoothing in faces with more spots as desired. We have set α_r = 10 and β_r = 10 for all our experiments. For the ϵ computation, we have kept α_ϵ = 5 and β_ϵ = 100. We provide an exemplar image from the Helen dataset [16] in Fig. 6 to visually illustrate the effect of varying control parameters α_r and α_ϵ on smoothing. This particular image had $\text{num}_{\text{spots}} = 60$.

3.4. Skin Texture Restoration

As mentioned previously in Section 3.3, the fundamental shortcoming of applying smoothing filters to the face beautification problem is that they wipe clean the essential, fine-grained texture in skin regions. To address this concern, we have included a wavelet-based STR in our face softening pipeline. We employ wavelet domain im-

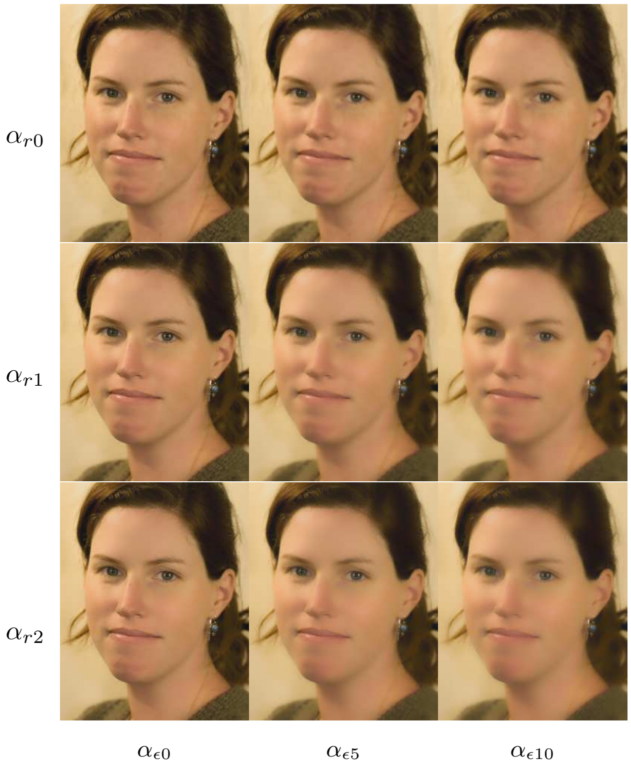

Figure 6: A visual demonstration of the effect of variation of α_r = 0,1,2 and α_ϵ = 0,5,10 on smoothing.

age decomposition as it effectively separates low-frequency and high-frequency components of the image. This choice was validated during experimentation as we observed that fine-skin textures were present in high-frequency bands while blemishes and skin nonuniformities were dominant in low-frequency bands. This separation of components in the wavelet domain helped orthogonalize our softening pipeline. We describe the STR algorithm in detail below:

Given an input image I, we first applied a strong smoothing filter to obtain a filtered output S via the ADF as defined in Section 3.3, and the result is illustrated in Fig. 4d. We then transformed both I and S to their corresponding wavelet representations I_w and S_w.

For each wavelet level l, we calculated a weighted average between the corresponding bands $I_w^l(LL, HH, HL, HH)$ and $S_w^l(LL, HH, HL, HH)$ to obtain wavelet layers of the desired output $O_w^l(LL, HH, HL, HH)$, based on the following equations:

$$O_w^l(LL) = \alpha_{LL}^l * I_w^l(LL) + (1 - \alpha_{LL}^l) * S_w^l(LL) \tag{3}$$

$$O_w^l(HH) = \alpha_{HH}^l * I_w^l(HH) + (1 - \alpha_{HH}^l) * S_w^l(HH) \tag{4}$$

$$O_w^l(HL) = \alpha_{HL}^l * I_w^l(HL) + (1 - \alpha_{HL}^l) * S_w^l(HL) \tag{5}$$

$$O_w^l(LH) = \alpha_{LH}^l * I_w^l(LH) + (1 - \alpha_{LH}^l) * S_w^l(LH) \tag{6}$$

In the course of our experimentation, we observed that the HH band contains a large proportion of the skin texture information of an image. The HL and LH bands do contain some texture information; however, these bands primarily contain small blemishes and spots, which we wish to eliminate from our processed output O_w. To generate O_w, we thus compute a weighted average between the HH bands of S_w^l and I_w^l, whereas all other bands (HL, LH, LL) are directly copied from S_w. We used values of $\alpha_{LL}^l = 0$, $\alpha_{LH}^l = 0$, $\alpha_{HL}^l = 0$, while α_{HH}^l is computed using Eq. 7. While we have used a constant value for α_{HL}, α_{LH}, and α_{LL} for simplification, we leave their intelligent computation based on facial statistics for future work.

To gain control over the amount of texture we add, we have defined α_{HH} as a linear function of texture intensity, which we define in Eq. 8. To achieve beautification that looks natural and not over-processed, we require the texture content in skin regions of our output O to resemble the texture content of the original image I. We have defined α_{HH}^l as follows:

$$\alpha_{HH}^l = \gamma * \mathrm{TI} + \beta \tag{7}$$

here $\gamma = 0.5$, and $\beta = 1.0$. TI is a texture intensity metric which we define as follows:

$$TI = \frac{\sum_x \sum_y |I(x,y) - I_b(x,y)| * P_{skin}(x,y)}{255} \tag{8}$$

Here I is the original image, and I_b is the output of bilateral filter [27] applied on all pixel locations (x, y) over I. The difference between the bilateral smoothed output and the original image captures the texture details present. We further perform pixel-wise multiplication with the skin probability mask from Section 3.2 to obtain texture details in only the skin region. Finally, we aggregate these values and normalize to the input image intensity range $(0, 1)$. By definition, TI is a scalar global image metric, which represents the amount of original texture present in the skin region.

As the features captured in the wavelet domain become more coarse as we propagate towards higher layers, we reduce the value of α_{HH}^l by a factor of two in every successive wavelet layer, from the $l = 1$ to $l = 4$. Using different weights for each level and band provides significant control over the amount of texture that one desires to retain. To obtain the final output image O, we carry out an inverse wavelet transform on processed wavelet representation O_w as computed in Eq. 3 - 6. The entire pipeline of the STR is illustrated in Fig. 5.

In summary, the power of the STR module comes from taking more information from the lower frequency levels of S_w and less information from the lower frequency levels of I_w to obtain O_w and vice versa for high-frequency bands. This results in a fusion of the uniform skin region in the filtered output S (which are represented by low-frequency levels) with the texture present in the input image I (which is represented by high-frequency levels). This fusion creates a filtered output S, where the facial nonuniformities present in low-frequency bands of the original image are smoothed out while the skin texture present in high-frequency bands is retained.

4. Experiments and Results

We have compared our results with two state-of-the-art edge-preserving filters $i)$ Mutually Guided Image Filtering (muGIF) filter [7] and $ii)$ Fast Global Smoothing filter (FGS) [21]. We have also compared the proposed method with popular commercial face beautification applications B612 [2] and ModiFace [1]. For our evaluation, we have used OpenCV [9] library for FGS, and the source code provided by Guo *et al.* for muGIF [7]. We have used the Android mobile applications for B612 [2] and ModiFace [1] from the Google Play Store [3] to generate results. A qualitative comparison of the softened outputs of these methods, along with our proposed FabSoften solution, is shown in Fig. 7.

For the task of blemish removal, we observe that the FGS filter removes blemishes well, but fails to retain essential skin textures and facial feature structures as highlighted in green boxes in Fig. 7. Although muGIF retains more facial feature structures than the FGS filter, it still produces unnatural outputs, as seen in Fig. 7. ModiFace and $B612$ do a better job in smoothing skin region and retaining the skin-

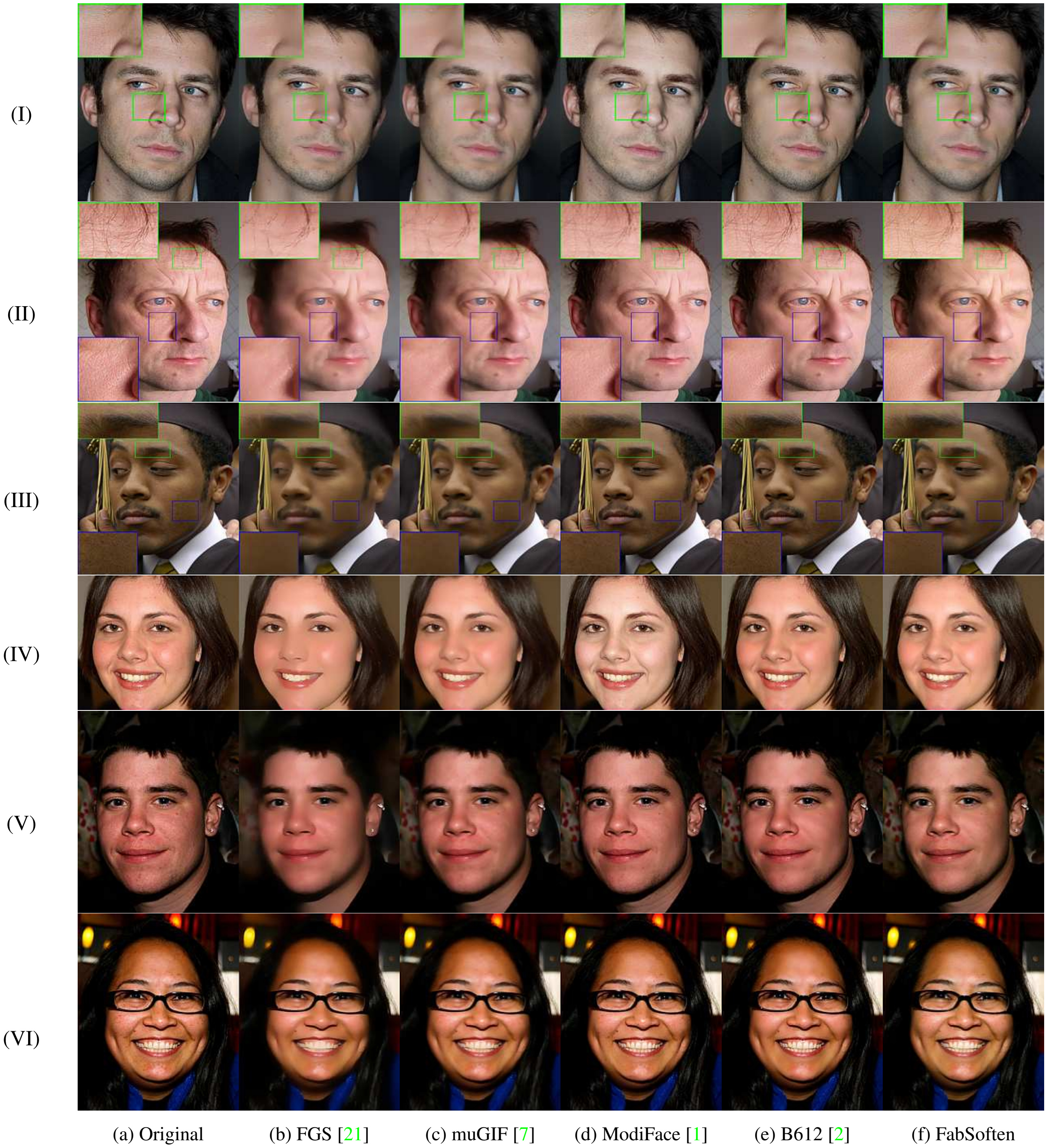

Figure 7: A qualitative comparison of the proposed method with popular face enhancement systems. We crop zoomed-in portions of the image to highlight each method's performance on skin texture retainment and preserving hair regions.

texture than edge-preserving filters, which are not explicitly designed for this purpose. In the green box in Fig. 7(I) as well as Fig. 7(IV), we observe FabSoften performs best in removing small blemishes.

For the task of skin texture retention, we observe from Fig. 7 that FabSoften outperforms other methods, as highlighted further in the green box crops. $B612$ performs the next best at retaining skin texture but noticeably contains washed out patches without visible texture, as shown in Fig. 7(I) and (II). As seen in the green box crops of Fig. 7(II)

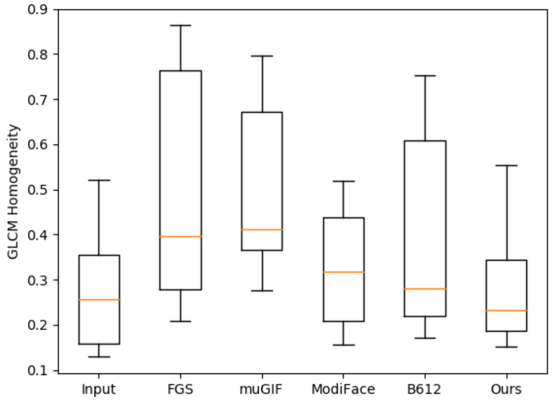

Figure 8: GLCM Homogeneity: A box-plot of the texture homogeneity of all evaluated methods with the input image.

and (III), all methods except FabSoften and $B612$ create blurring artifacts in hair regions. These artifacts give the output an increasingly artificial and over-processed appearance. Analyzing varied conditions of lighting, we observed that FabSoften and $B612$ were able to retain features well while smoothing the skin region, as shown in Fig. 7(V). We further observed our method to perform well in the presence of facial accessories, as seen in Fig. 7(VI), where the eyeglasses have been retained. All images used in this paper, including those in Fig. 7, are from the Helen dataset [16] unless otherwise mentioned.

To quantitatively evaluate the texture restoration capability of the STR module, we used the Gray-Level Co-Occurrence Matrix (GLCM) [23], which represents texture with statistical features. We randomly sampled 48 images from the Helen dataset [16] and cropped ten non-overlapping patches containing only skin pixels for each of these samples. We then computed the GLCM homogeneity feature for each patch and averaged them to get the homogeneity of the given image. From an analysis of the GLCM homogeneity box-plot in Fig. 8, FabSoften was able to achieve similar texture homogeneity distribution with the input image, thereby demonstrating the power of the STR module.

As a qualitative comparison, we performed a user study to evaluate the visual performance of the methods shown in Fig. 7, similar to that followed by Li *et al.* [18]. A total of 80 volunteers participated in this study. We randomly sampled 200 images from the Helen dataset [16] and applied five softening methods to generate the outputs: FGS [21], muGIF [7], ModiFace [1], B612 [2] and the proposed method FabSoften. We presented these five outputs in a random order to our survey volunteers and asked them to rank the results considering the perceived quality of the image, realism of the image, and the quality of the image's skin texture. Rank 1 represents the best beautification result, and Rank 5 represents the worst beautification result for any given image. The results of the study are shown in Table 1. This study showed that the proposed method ranked first 59 percent of the time, followed by $B612$, which was chosen 35 percent of the time. The edge-preserving filters FGS and muGIF were ranked very low due to their artificial appearance, which is caused by washed-out texture. One of the open challenges with FabSoften is its difficulty in obtaining perfect skin masks for side profile faces due to possible errors in landmark point detection as well as contrast variations near face sides.

Table 1: Results of the User Study.

Methods	**Rank 1**	**Rank 2**	**Rank 3**	**Rank 4**	**Rank 5**
FGS [21]	0.0	0.89	1.32	9.72	88.07
muGIF [7]	2.34	5.47	39.56	46.08	6.55
ModiFace [1]	3.37	6.45	47.68	37.3	5.38
B612 [2]	35.28	52.71	7.05	4.96	0.0
FabSoften	59.01	34.66	4.39	1.94	0.0

FabSoften is optimized to run in under 90 milliseconds, executing preprocessing, skin mask generation, skin imperfection smoothing, and texture restoration. FabSoften was tested on 12-megapixel images on a Samsung Galaxy S10 running a Qualcomm Snapdragon 855 processor.

5. Conclusions

We presented a new method of face softening that removes facial blemishes while retaining skin textures to produce highly appealing faces. Our proposed FabSoften solution is either comparable or superior to both the state-of-the-art edge-preserving filters and commercial face softening applications in the following aspects: $a)$ Blemish removal while retaining facial feature sharpness. $b)$ Restoring skin texture as well as fine hair texture $c)$ Handling varied lighting conditions. The low-cost performance and control provided by FabSoften are highly suitable for deployment on smartphone devices. We leave the incorporation of facial intelligence in the form of face shape, skin type, age, and gender in our face softening solution as future work.

References

[1] Modiface - augmented reality, 2019. Accessed on: Feb 22, 2020. [Online]. Available: `http://modiface.com/`. 1, 2, 6, 7, 8

[2] B612 beauty and filter camera, 2020. Accessed on: Feb 22, 2020. [Online]. Available: `https://b612.snow.me/`. 1, 2, 6, 7, 8

[3] Google play, 2020. Accessed on: Feb 22, 2020. [Online]. Available: `https://play.google.com/store/apps/top?hl=en`. 1, 6

[4] Youcam perfect - best selfie camera & photo editor, 2020. Accessed on: Feb 29, 2020. [Online]. Available: `https:`

//play.google.com/store/apps/details?id=com.cyberlink.youperfect&hl=en_IN. 1

[5] Ivaylo Boyadzhiev, Kavita Bala, Sylvain Paris, and Edward Adelson. Band-sifting decomposition for image-based material editing. *ACM Transactions on Graphics (TOG)*, 34(5):1–16, 2015. 2

[6] Lijun Ding and Ardeshir Goshtasby. On the canny edge detector. *Pattern Recognition*, 34(3):721 – 725, 2001. 3

[7] Xiaojie Guo, Yu Li, Jiayi Ma, and Haibin Ling. Mutually guided image filtering. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 42:694–707, 2020. 1, 2, 4, 6, 7, 8

[8] K. He and J. Sun. Guided image filter. In *In Proc. Trans. On Pattern Recg. and Machine Intelli.2013*. 1, 4

[9] Itseez. Open source computer vision library. https://github.com/itseez/opencv, 2015. 6

[10] L. Ji, L. Shuai, S. Wenfeng, L. Liang, Q. Hong, and H. Aimin. Automatic beautification for group-photo facial expressions using novel bayesian gans. In *Intl. Conf. on Artificial Neural Nw*, pages 760–770. Springer, 2018. 2

[11] Benedict C Jones, Anthony C Little, Ian S Penton-Voak, Bernard P Tiddeman, D Michael Burt, and David I Perrett. Facial symmetry and judgements of apparent health: Support for a "good genes" explanation of the attractiveness–symmetry relationship. *Evolution and human behavior*, 22(6):417–429, 2001. 1

[12] V. Kazemi and J. Sullivan. One millisecond face alignment with an ensemble of regression trees. *IEEE Conf. On Computer Vision and Pattern Recog.(CVPR)*, page 1867–1874, 2014. 3

[13] L. Liu S. Xiangbo Y. Shuicheng L. Jianshu, C. Xiong. Deep face beautification. In *MM '15 Proc. of the 23rd ACM intl. conf. on Multimedia*, pages 793–794, 2015. 2

[14] L. Jin L. Liang and X. Li. Facial skin beautification using adaptive region-aware masks. *IEEE Transactions on Cybernetics*, 44(12):2600–2612, 2014. 1, 4

[15] Q. Yan R. Liao L. Xu, J. Ren and J. Jia. Deep edge-aware filters. In *Proc. of the 32nd Intl. Conf. on Machine Learning*, volume 30(6), page 1669–1678, 2015. 2

[16] Vuong Le, Jonathan Brandt, Zhe Lin, Lubomir Bourdev, and Thomas S Huang. Interactive facial feature localization. In *European conference on computer vision*, pages 679–692. Springer, 2012. 5, 8

[17] C. Lee, M. T. Schramm, M. Boutin, and J. P. Allebach. An algorithm for automatic skin smoothing in digital portraits. *Intl. Conf. Image Proc., 2009*. 2, 3, 4

[18] Tingting Li, Ruihe Qian, Chao Dong, Si Liu, Qiong Yan, Wenwu Zhu, and Liang Lin. Beautygan: Instance-level facial makeup transfer with deep generative adversarial network. In *Proceedings of the 26th ACM international conference on Multimedia*, pages 645–653, 2018. 8

[19] L. Lingyu, L. Deng, and J. Lianwen. Facemore: A face beautification platform on the cloud. In *IEEE Intl. Conf. on Systems, Man, and Cybernetics*, pages 1798–1803. IEEE, 2015. 2

[20] Ping Luo, Xiaogang Wang, and Xiaoou Tang. Hierarchical face parsing via deep learning. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2480–2487. IEEE, 2012. 4

[21] D. Min, S. Choi, J. Lu, B. Ham, K. Sohn, and M. N. Do. Fast global image smoothing based on weighted least squares. In *IEEE Trans. Image Procsessing*, volume 12, pages 5638–5653, 2014. 1, 2, 4, 6, 7, 8

[22] D. Nir, Z. Dean, B. Chaim, S. Eli, and B. Alex M. Beholder-gan: Generation and beautification of facial images with conditioning on their beauty level. *arXiv preprint arXiv:1902.02593*, 2019. 2

[23] L. GuruKumar P. Mohanaiah, P. Sathyanarayana. Image texture feature extraction using glcm approach. *International Journal of Scientific and Research Publications*, 3(5), 2013. 8

[24] S. L. Phung, A. Bouzerdoum, and D. Chai. Skin segmentation using color pixel classification: Analysis and comparison. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(1):148–154, 2005. 3

[25] Gang Hua Baoquan Chen David Wipf Qingnan Fan, Jiaolong Yang. A generic deep architecture for single image reflection removal and image smoothing. *IEEE Intl. Conf. on Computer Vision*, pages 54–63, 2017. 2

[26] M. M.Oliveira S. L. Eduardo. Domain transform for edge-aware image and video processing. In *In Proc. of ACM SIGGRAPH*, pages 1–12, 2011. 2

[27] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. 1998. 2, 6

[28] Tsung-Shian Huang Wen-Chieh Lin Jung-Hong Chuang Tsung-Ying Lin, Yu-Ting Tsai. Exemplar-based freckle retouching and skin tone adjustment. *Comput. Graph*, pages 54–63, 2019. 2

[29] Z. Wang and L. Tong. A face detection system based skin color and neural network. *Intl. Conf. on Computer Science and Software Engg.*, pages 961–964, 2008. 3

[30] C. WenTing, X. Xinpeng, J. Xi, and S. Linlin. Texture deformation based generative adversarial networks for face editing. *arXiv preprint arXiv:1812.09832*, 2018. 2

[31] L. Xi, C. Xiaobin, X. Xiaohua, H. Jian-Fang, and Z. Wei-Shi. Facial skin beautification via sparse representation over learned layer dictionary. In *2016 Intl. Joint Conf. on Neural Networks (IJCNN)*, pages 2534–2539. IEEE, 2016. 2

[32] L. Xu, C. Lu, Y. Xu, and J. Jia. Image smoothing via l0 gradient minimization. In *In Proc. of ACM SIGGRAPH Asia Conf.*, volume 30(6), 2011. 1, 2

[33] C. Ying-Cong, S. Xiaohui, L. Zhe, L. Xin, I. Pao, and J. Jia. Semantic component decomposition for face attribute manipulation. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recog.*, pages 9859–9867, 2019. 2

[34] Q. Zhang, L. Xu, and J. Jia. 100+ times faster weighted median filter (wmf). In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, volume 30(6), pages 2830–2837, 2014. 2

[35] Yisu Zhou, Xiaolin Hu, and Bo Zhang. Interlinked convolutional neural networks for face parsing. In *International symposium on neural networks*, pages 222–231. Springer, 2015. 4

[36] F. Zhu, Z. Liang, X. Jia, L. Zhang, and Y. Yu. A benchmark for edge-preserving image smoothing. *IEEE Transactions on Image Processing*, 28:3556–3570, 2019. 2