doi:10.20944/preprints201810.0756.v1

Realistic Hair Simulator for Skin lesion Images Using Conditional Generative Adversarial Network

M. Attia*, M. Hossny*, H. Zhou*, A. Yazdabadi†, H. Asadi‡ and S. Nahavandi*
*Institute for Intelligent Systems Research and Innovation (IISRI), Deakin University
†School of Medicine, Deakin University
‡School of Medicine, University of Melbourne

Abstract*—Automated skin lesion analysis is one of the trending fields that has gain attention among the dermatologists and health care practitioners. Skin lesion restoration is an essential pre-processing step for lesion enhancements for accurate automated analysis and diagnosis. Digital hair removal is a non-invasive method for image enhancement by solving the hair-occlusion artifact in previously captured images. Several hair removal methods were proposed for skin delineation and removal. However, manual annotation is one of the main challenges that hinder the validation of these proposed methods on large number of images or using benchmarking datasets for comparison purposes. In the presented work, we propose a realistic hair simulator based on context-aware image synthesis using image-to-image translation techniques via conditional adversarial generative networks for generation of different hair occlusions in skin images, along with ground-truth mask for hair location. In addition, we explored using three loss functions including L1-norm, L2-norm and structural similarity index (SSIM) to maximize the synthesis quality. For quantitatively evaluate the realism of image synthesis, the t-SNE feature mapping and Bland-Altman test are employed as objective metrics. Experimental results show the superior performance of our proposed method compared to previous methods for hair synthesis with plausible colors and preserving the integrity of the lesion texture

***Index Terms*—Dermoscopic Image, Skin Lesion, Melanoma, Simulation, Generative Adversarial Networks, Deep Learning**

I. Introduction

Several computer aided diagnosis systems were developed to help dermatologists and health-care practitioners in screening and early detection of skin cancer [1], [2], [3], [4]. Skin melanoma is diagnosed based on the morphological properties of the lesions. Thus, lesion diagnosis is highly subjective and prone to inter- and intra-personal variability. Therefore, the computational methods were introduced to eliminate the subjectivity and increase sensitivity and specificity [5]. These computational methods were based on clinical tools, widely used by the dermatologists, such as the ABCD method (A: asymmetry, B: border, C: color, D: diameter); Menzies's method; and the 7-point checklist rules. These methods diagnose the skin lesion based on their photometric and morphological features: color, shape and size [5], [6].

Hair occlusion is one of the main factors that affects the accuracy of melanoma detection and lesion delineation [7]. Therefore, image restoration are used as a pre-processing step to remove artifacts, then followed by lesion segmentation and feature extraction to enhance the diagnosis accuracy [6], [8]. Many digital hair removal methods were developed to tackle the image restoration problem. However, the previously adopted methods lack the generalizability due to the limited availability of annotated data as ground truth for training and validation [7], [8]. These hair removal methods used simulated hair with un-realistic morphology to validate their work. These simulated hair overlays suffer from odd colors, even-thickness, un-blended simulated hair or un-natural curvature.

The basic idea of hair simulation is to add artificial hair into hairless images which involves two major components: generating hair mask and simulating hair color [9]. The hair mask is to define the geometrical properties of hair that includes locations, distributions, quantity and thickness. It can be created from existing segmentation results or manually drawing with curves and lines. The hair appearance defines the photometric property of hair in skin images, which is our focus in this paper. The main concern of simulating hair color is the hair synthesis in a seamless manner without affecting the surrounding skin and lesion pixels.

In this work, we propose a realistic hair simulator to solve the benchmarking and annotation problem of skin hair delineation, based on the state-of-the-art image-to-image translation method using adversarial training [10]. This method can be used as an artifacts simulation method to test the sensitivity of the classification methods using controlled experiments. The generative adversarial learning technique is employed to model the conditional data distribution of the hair in the training images. Thus, they can be used for stochastic regression of the simulated hair, based on the the contextual content in the input images, in terms of skin and lesion color tone; and illumination. The main advantage of the proposed method is the ability to synthesize and embed skin hair that resembles real hair according to a predefined mask. The hair is generated based on the contextual information: the data distribution within the image; and the neighboring pixels with great visual plausibility. This proposed simulator can be used for validation of hair segmentation and inpainting methods.

A. Contributions

In this paper, a novel skin hair simulation method is proposed to synthesize various hair occlusions on skin lesion images where realistic synthesis is achieved while preserving lesion texture. This is achieved by using image-to-image translation technique via cGAN. Our main contributions are summarized as follows.

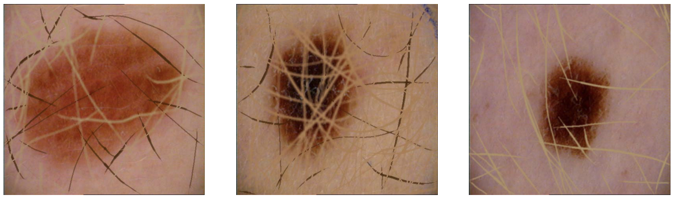


Fig. 1: Simulated hair by Mirzaalian *et al.* [9]. The simulated hair has unrealistic appearance with random colors.

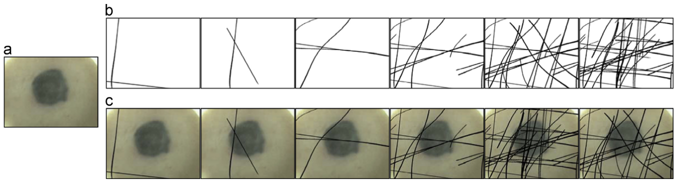


Fig. 2: Simulated hair by Xie *et al.* [16]. The simulated hair was simulated as random straight lines in black colors.

- To maintain the quality of the synthesized image, the proposed method applies the pixel synthesis to dilated hair masks instead of the traditional application of cGANs holistically. This, in return, allocates the majority of the learning capacity of the cGAN network to generate realistic hair, skin and lesion pixels in the predefined masks.
- In order to have a seamless blending with the lesion and skin pixels, the proposed method empowers L_1 and L_2 loss functions with a structural similarity image metric (SSIM) loss function. This allows the synthesized pixels to preserve the surrounding skin and lesion textures without the need for multi-scale image fusion or Gaussian blending filters.
- For quantitative evaluation of the quality of synthesized results: L_1, L_2, structural similarity image metric (SSIM), t-SNE feature mapping and Bland-Altman tests.

The rest of the paper is organized as follows. The efforts in the literature and related work are elaborated in Section II. In section III, we describe briefly the proposed method and data preparation method. Experiments and results were discussed in Section IV. Finally, conclusions are drawn in Section V.

II. Related Work

Despite the importance of the hair problem, it was not fully addressed in the literature with limited contributions in that domain [14]. Early attempts for hair synthesis were based on the external morphology of the hair under the assumption that the hair has thin architecture with varying width. Denton *et al.* proposed one of the earliest attempts for simulating hair and skin lesions [11]. In 2001, She *et al.* proposed a hair simulator for a single black hair image. They assumed that hair is 100 pixels long and 3 pixels wide. They used predefined hair orientations: horizontal, vertical and with 45 $\deg$ [15]. Then, they proposed a skin hair simulator as a part of skin lesion simulator to facilitate the analysis of the skin lesions. They simulated the hair as straight and curved black lines [12].

Mirzaalian *et al.* proposed a hair simulator based on second degree splines to generate curved hair masks. Then, they used Gaussian filtering to perform blending of simulated hair edges with the skin and lesion pixels. Afterwards, they colored the hair mask by random colors from a predefined dictionary, as shown in Fig. 1 [9]. Xie *et al.* used manual simulation technique for hair simulation using the Photoshop software. They applied random distortions on several curved lines and then they compiled them into combined masks, as shown in Fig. 2 [16].

These simulation methods were useful at the early stages for validation of the proposed methods and overcome the shortage in the annotated testing images [17]. Nevertheless, they based their simulation techniques on utilization of simplified mathematical models, along with other assumptions, to synthesize skin hair. These assumptions leads to un-realistic visual appearance of the hair with less variability, as depicted in Table I. Additionally, the simulated hair produced by these methods are blended with skin and lesion pixels using low pass filtering and multi-scale image fusion. This, in return, does affect the integrity of the lesion which affects subsequent processing using available computer-based techniques for lesion segmentation and classification. In contrast to the surveyed methods, we propose generating the simulated hair along with surrounding tissues using generative adversarial convolutional neural networks.

Convolutional neural networks (CNNs) were successfully employed to solve various computer vision problems. They were able to achieve outstanding performance in a wide range of tasks, such as classification and semantic segmentation tasks. CNNs are optimized using the back-propagation algorithm to minimize an objective function that is directly related to the task. These objective functions are known as loss functions. Several optimizers and loss functions have been used to train CNNs for various tasks. The loss functions have to be differentiable, either fully or piece-wise differentiablity, to facilitate the back-propagation of the error term to optimize networks weights [18].

$\mathcal{L}_2$-norm is one of the popular methods that has been used extensively as an error function for the optimization of CNNs. Based on the conducted experiment by Zhao *et al.*, They concluded that L_2 is not the optimal loss function for training the networks for image synthesis task. The synthetic images suffer from blurriness due to utilization of $\mathcal{L}_2$-norm as an optimization objective [19]. The limitations of using $\mathcal{L}_2$-norm as a quality metric, were evident in the experiments conducted by Simoncelli *et al* in [20] which then laid the foundation to a more robust image similarity metric by Wang et al. in [21]. Afterwards, the structural similarity index was proposed as loss function for the optimization of convolutional neural networks (CNNs) during network training [19].

Generative adversarial networks have been successfully used for image synthesis tasks for generating benchmarking data and cross-modality synthesis [17], [10], [22], [23]. Generative adversarial networks (GANs) are based on training of two competing neural networks, also known as dueling neural networks, simultaneously through back propagation

TABLE I: Summarization for hair simulation methods in terms of Hair morphology, number of test images, availability of the implementation and the simulated hair color.

Method	Hair morphology	#images	Code	Hair color
Danton *et al.* [11]	straight lines with different orientation	-	N/A	monochrome black
She *et al.* [12]	straight lines with different orientations	16 synth.	N/A	monochrome black
Xie *et al.* [13]	straight and curved lines	40 real 200 synth.	N/A	monochrome black
Mirzaalian *et al.* [9]	Curved lines	40 real 90 synth.	A	colored

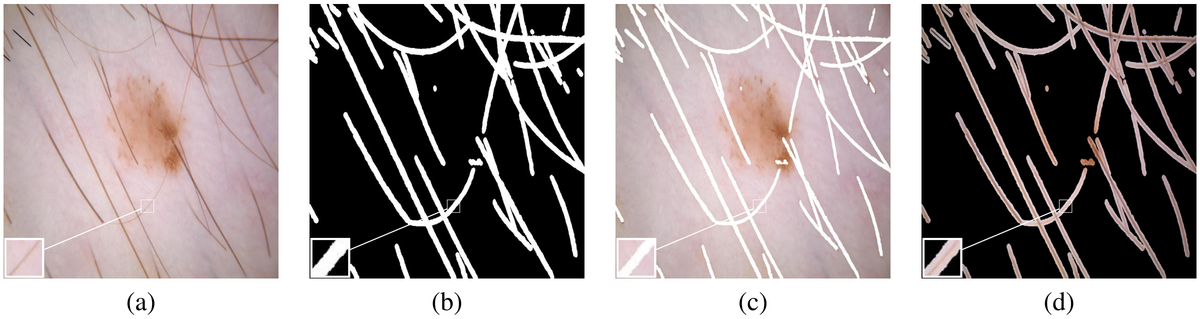

(a) (b) (c) (d)

Fig. 3: Data preparation step for ground truth generation. The hair is segmented from the input skin image (a) to generate he hair mask, in (b). We used this hair mask to replace the hair pixels with black pixels to generate the input training images, as in (c). The same hair mask is used to extract the hair locations as ground truth for network optimization, as in (d).

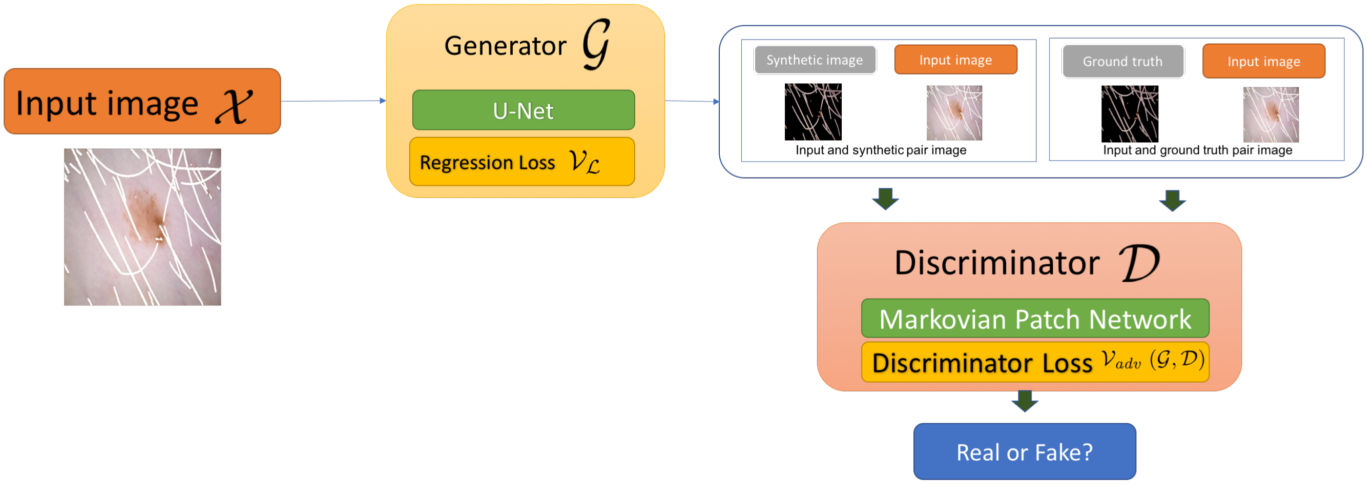

Input image $\mathcal{X}$
Generator $\mathcal{G}$
U-Net
Regression Loss $\mathcal{V}_{\mathcal{L}}$
Synthetic image
Input image
Input and synthetic pair image
Ground truth
Input image
Input and ground truth pair image
Discriminator $\mathcal{D}$
Markovian Patch Network
Discriminator Loss $\mathcal{V}_{adv}(\mathcal{G}, \mathcal{D})$
Real or Fake?

Fig. 4: The proposed architecture for the hair simulator. It consists of two networks: generator and discriminator. The generator is responsible for the generation of synthetic images to deceive the discriminator. while, the discriminator detects the synthetic images. The desired locations for hair is highlighted as white lines according the hair mask. The generator selectively embed the hair into these areas.

of the error using the adversarial loss. These two networks are a generator $\mathcal{G}$ and a discriminator $\mathcal{D}$. The generator is responsible for the creation of "synthetic" data based on the input data. Thus, it can be modelled as a mapping function for an RGB image $\mathcal{G} : \mathcal{G}_x \rightarrow \mathbb{R}^3$. While, the discriminator is required to differentiate between the real images from the forged synthetic data $\mathcal{G}_x$. The discriminator can be expressed as $\mathcal{D} : \mathcal{D}_{(x,\mathcal{G}(x))} \rightarrow (0, 1)$ where 0 is decision for fake forged data and 1 stands for real data [24], [25].

The generator tries to synthesize pseudo-realistic data to deceive the discriminator. On the other side, the discriminator tries to detect the synthetic images. The joint training, of both the generator and the discriminator, facilitates the training of the generator to produce synthetic images that match the distribution of the real data. Also, the generator is optimized using back propagation of the error of the loss function between the generated images and ground truth. In generative adversarial networks, the ground truth data and the generated synthetic data are fed into the discriminator to assess the quality of the synthetic images. Thus, the generator is optimized based on the interaction with the discriminator. This process is known as adversarial training, where the generator tries to minimize the loss between the generated images and the ground truth, while, the discriminator tries to detect the fake generated data [25],

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 1 November 2018 doi:10.20944/preprints201810.0756.v1

[10].

Costa *et al.* proposed one of the earliest attempts for adversarial training architectures to synthesize retinal images using image-to-image translation [26]. They were able to learn latent representations for retinal vascular architectures for the synthesis of realistic retinal images with accurate vascular structure, corresponding to the input vascular tree [17]. However, the reported results showed artifacts in the simulated images.

Several studies have been conducted to study the effect of using different loss functions to optimize the generator, such as $\mathcal{L}_1$-norm, $\mathcal{L}_2$-norm and Structural Similarity Index (SSIM) to reduce visual artifacts problem [27], [19]. In [19], Zhao *et al.* showed that $\mathcal{L}_1$ and $\mathcal{L}_2$-norm loss functions are capable of optimization of the low-level contextual information, namely, pixel values. Thus, the images were locally optimized with usually blurred details. Therefore, they do not have good visual perception quality [22]. The convolutional networks treat the images as a set of independent pixels without a penalty on the contextual information within the sub-regions [28]. Therefore, an error term is required to address the perception quality of the generated images. The SSIM loss layer is utilized to optimize the synthetic images globally, besides the local optimization using $\mathcal{L}_1$ and $\mathcal{L}_2$-norms [22], [23].

III. Materials and Methods

The proposed method consists of three mains steps: data preparation, hair synthesis and hair merging. For training and validation, lesion images with hair, Fig. 3-(a), are fed into Virtual shaver, proposed by Koehoorn *et al.* for hair mask preparation [29]. The extracted hair masks, Fig. 3-(b), are used to define the hair locations in the images. Based on the hair mask, candidate hair locations are filled by white color pixels, Fig. 3-(c). These white pixels are used as markers for the conditional generative adversarial network, as shown in Fig. 5, to generate hair in these predefined locations. For ground truth, the hair masks are used to extract the hair locations only and fill the other pixel locations with black color, as depicted in Fig. 3-(d). We adopted this methodology to focus the network layers on learning to synthesize pixels of the hair and adjacent tissues as well as facilitating a faster network convergence. Additionally, this method also avoids producing unnecessary artifacts usually generated by generative adversarial networks when operating on images holistically. The white labelled pixels ensure that the hair is inpainted only in the required location and according to the input mask.

During the training and validation, the output images are compared to the ground truth (Fig. 3-(d)) for network layers optimization using adversarial training as discussed in Section III-B. It is worth noting that the output synthetic hair images contain hair perfectly blended with the background. Thus, we replaced only the candidate locations, highlighted by white pixels in the input hair mask in Fig. 3-(b), to update the corresponding hair locations in the original image in Fig. 3-(a) with the synthetic hair from the output image, Fig 5, to generate the full synthetic image. We used pixel-wise replacement for the synthetic hair without any further post-processing or image blending methods to make sure that the skin lesions are not affected.

During the testing phase, we used hair-free images. Therefore, there is no need to use the hair segmentation method and we randomly choose hair masks. These random masks can be generated synthetically using free-hand, random splines methods [9] or pre-segmented masks.

A. Hair Data Preparation

Data preparation is one of the most important steps in training neural networks, where the annotation process is a challenging task. Manual annotation of the hair in skin lesion images is extremely tedious task. Therefore, we used an automated method proposed by Koehoorn *et al.* for ground truth preparation of the images obtained from the dataset used in ISBI 2017 challenge [6], [1], [30], [31]. This automated method was able to extract hair masks and repair the segmented hair regions, as shown in Fig. 3-(b). We used the generated mask (Fig. 3-(b)) in the training process. For the training step, we superimposed the mask as white pixels on the image to facilitate the learning process and accelerate network convergence to construct the training set $\mathcal{X}$. Using this technique, the network was able to synthesize the hair in the defined regions by the mask and keep the other regions of the image intact. For validation purposes, we superimposed the hair masks on the inpainted images and compared them to the original images to assess the quality of the synthesized hair against the original ones.

B. Conditional Generative Adversarial Neural Network

Conditional Generative Adversarial Neural Network is one of the popular implementations for image-to-image translation that is based on adversarial training for contextual image synthesis. It can be used for mapping an image from a domain to another. During the adversarial training process, the mapping function $\mathcal{G}$ is optimized based on the data distribution from the training images $\mathcal{X}$, along with ground truth images $\mathcal{Y}$. The ultimate goal for the optimization process is the ability of the generator to produce synthetic data $\mathcal{G}_x$ that can not be distinguished by the discriminator $\mathcal{D}$ from the real images $\mathcal{Y}$, as shown in Fig. 4.

During the training process, the generator parameters $\theta_\mathcal{G}$ are optimized to minimize the adversarial loss between the generator $\mathcal{G}$ and discriminator $\mathcal{D}$, in addition to the generator loss with $\mathcal{L}1$-norm or $\mathcal{L}2$-norm. The adversarial loss is the penalty of the detected fake images from the generator. Back propagation is utilized to optimize the loss function $\mathcal{G}^*$ with pairs of input images (x,y), where $x \in \mathcal{X}$ and $y \in \mathcal{Y}$, as follows in Eqn. 1:

$$\mathcal{G}^* = \arg\min_{G}\max_{D}\ \mathcal{V}_{adv}\ (\mathcal{G},\mathcal{D})\ +\ \lambda\mathcal{V}_{\mathcal{L}} \tag{1}$$

The first term of the optimization function is the adversarial loss $\mathcal{V}_{adv}(\mathcal{G},\mathcal{D})$, which represents the competition between the

doi:10.20944/preprints201810.0756.v1

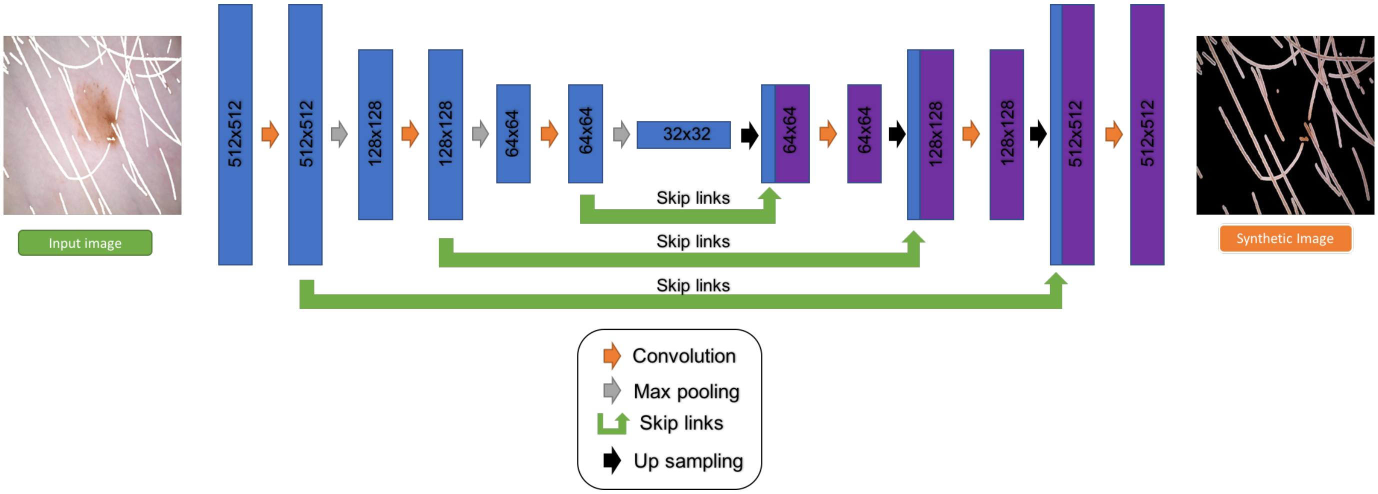


Fig. 5: U-Net architecture for the generator ($\mathcal{G}$) with skip links for concatenation of the deep encoded feature during the reconstruction. The skip links are responsible for re-utilization of deep encoded feature in the reconstruction.

generator to synthesize images that can deceive the discriminator and the ability of the discriminator to distinguish between the real and synthetic data, as shown in Eqn. 2.

$$\mathcal{V}_{adv}\ (\mathcal{G},\mathcal{D}) = \mathbb{E}_{x,y}[\log \mathcal{D}_{(x,y)}] + \mathbb{E}_x[1-\log \mathcal{D}_{(x,G(x))}] \quad (2)$$

where $\mathbb{E}_{x,y}\left[\log \mathcal{D}_{(x,y)}\right]$ is the expectation of the log-likelihood of discriminator output for the input "real" pair data (x,y), similarly, $\mathbb{E}_x\left[1-\log \mathcal{D}_{(x,G(x))}\right]$ is the expectation of the complementary term of the log-likelihood of the discriminator for the "fake" data $(x,G(x))$. The generator is optimized by loss layers to minimize the error between the generated image and the ground truth. In addition to the adversarial loss term, based on the discriminator loss.

Theoretically, the optimal state is achieved when the discriminator is incapable of discrimination between the real and fake data with $P_{synthetic} = P_{data}$. In other words, the confidence of the predictions of the discriminator is 0.5 [32].

C. Loss function

Loss functions are used to train the neural networks based on the calculated error between the network output and the ground truth. They are essential modules for network training to facilitate convergence to locally-optimal point in the weights space. To facilitate the training process, the loss function has to be differentiable function, either fully or piece-wise differntiable. The loss for an error function $\mathcal{E}$ between the generated image and ground truth at the pixel of the position (x,y) can be mathematically formulated as follows in Eqn. 3.

$$\mathcal{L} = \frac{1}{n \times m} \sum_{x\in n\ ,\ y\in m} ||\mathcal{E}_{(x,y)}|| \quad (3)$$

Mean square error "MSE" is one of the most popular cost functions that has been successfully used to train many deep neural networks for various tasks, such as regression. Also, this loss function is known as $\mathcal{L}_2$. It penalizes the squared error between the generated image and the ground truth. It can be computed as in Eqn. 4.

$$\mathcal{L}_2 = \frac{1}{n \times m} \sum_{x\in n\ ,\ y\in m} ||\mathcal{I}_{(x,y)} - \mathcal{G}_{(x,y)}||^2 \quad (4)$$

However, images generated from the MSE loss function suffer from poor quality in terms of human visual system "HVS" [19]. MSE loss function assumes that the individual pixel noise is independent from the neighbouring pixel within a local window. Also, it assumes that the noise in the image is white Gaussian noise. In [19], Zhao *et al.* stated that $\mathcal{L}_2$ tends to over penalize the error during the training, which reduce the visual quality [33], [19].

Mean Average Error "MAE" was introduced to suit the image regression problems. Also, it is known as $\mathcal{L}_1$. It is used to reduce blurring effect introduced by $\mathcal{L}_2$. It evaluates the absolute error between the output of the network and the ground truth, as shown in Eqn. 5.

$$\mathcal{L}_1 = \frac{1}{n \times m} \sum_{x\in n\ ,\ y\in m} ||\mathcal{I}_{(x,y)} - \mathcal{G}_{(x,y)}|| \quad (5)$$

Perceptually motivated image quality methods have been introduced to address the adaptation to the human visual system. Structural Similarity Index "SSIM" is one of the most popular methods that addresses the image quality from the human visual system prospective. Thus, it overcomes the limitations of both $\mathcal{L}_1$ and $\mathcal{L}_2$, which depend only on the error difference between the pixels of two images.

Wang *et al.* designed the structural similarity image metric "SSIM" to take in account the intrinsic properties of the human visual system. In SSIM, the quality of an image is tested against a reference image based on the variation within the illumination and color. It is evaluated as an index representing the similarity score between the image and the reference image. Thus, it has the range of minimum index of zero to maximum index of one. The SSIM index between two image x and y patches is calculated based on: the mean of the image patches, μ_x and μ_y; the variance within the patches: σ_x^2 and σ_y^2;

doi:10.20944/preprints201810.0756.v1

the covariance of the local patches: σ_{xy}. C_1 and C_2 are terms to stabilize the division with weak denominator Therefore, the SSIM loss function can be defined as in Eqn. 7.

$$\mathcal{L}_{SSIM} = 1 - \mathit{SSIM(x,y)} \tag{6}$$

$$= 1 - \frac{2\mu_x\mu_y + \mathcal{C}_1}{\mu_x^2 + \mu_y^2 + \mathcal{C}_1} \cdot \frac{2\sigma_{xy} + \mathcal{C}_2}{\sigma_x^2 + \sigma_y^2 + \mathcal{C}_2} \tag{7}$$

The $\mathcal{L}_1$ and $\mathcal{L}_2$ are used to penalize the pixel-wise error based on the difference between the generated and ground truth without taking into account the pixel neighbourhood. The perceptually based image quality metrics focus more on the local variation in the color and illumination. Thus, we are going to employ both of loss functions: $\mathcal{L}_1$ and $\mathcal{L}_2$; and we will combine them with $\mathcal{L}_{SSIM}$. These loss functions will be investigated. They will be added to the adversarial loss term resulting from the interaction between the generator and discriminator.

D. Training parameters

We based our implementation on the proposed model by Isola *et al.*, "pix2pix" in Pytorch framework [10], [34]. We trained the network on image-to-image translation for hair synthesis. We used the U-Net architecture for the generator $\mathcal{G}$, as shown in Fig. 5 [35], [10]. The image-to-image translation can be used for mapping of a 2-D image with high resolution in the RGB color space in specific domain to high resolution image in RGB color space in another domain. In U-Net, the deep encoded feature maps are concatenated to the corresponding layers in the decoder for more precise reconstruction of the output, as shown in Fig. 5. This will help in preserving the structure while changing the external features, i.e. preserving colors and edges while embedding the synthetic hair. In our experiments, we replaced the loss layer for the generator to $\mathcal{L}_1$, $\mathcal{L}_2$ and $\mathcal{L}_{SSIM}$. The model was trained using the ADAM optimizer with initial learning rate 0.002 for 200 epochs, with batch size set to 2. Also, Batch normalization was used for the convolution layers to accelerate the convergence of the generator.

IV. Experimental Evaluation and analysis

In this section, we will describe the used dataset, conducted experiments and results of the proposed method along with quantitative and qualitative analysis in comparison to the hair simulator proposed by Mirzaalian *et al.* [9].

A. Data

In our work, we used images from ISBI 2017 challenge for melanoma segmentation and classification. We used this dataset for training and validation [31]. It has 2000 images. It contains images with different artifacts and hair occlusion. Also, it has a set of images with no hair, that can be used for testing. We used 1900 images with hair occlusion and ruler markers for training and validation; and 100 images with no-hair for test. We had limited number of the test images as most of the dataset images have hair occlusion. We compared between the generated images and the ground truth to evaluate the loss function and back-propagated the error to optimize the networks weights.

B. Experiment setup

We conducted qualitative and quantitative analysis of the synthetic images. We carried out two experiments: comparison with ground truth and with state-of-the-art hair simulator. In the ground truth comparison experiment, we evaluated the simulation results against real images with hair. We inpainted the hair, then, we super imposed the hair mask again. Then, we fed the masked images into trained generator. We compared the results with original images as a ground truth. In the second evaluation experiment, we compared the synthesis results against the proposed method in [9]. For quantitative analysis, we calculated the mean-square error and structural similarity index "SSIM" for the synthesized images using the proposed method, in comparison to the real images. In addition, we used t-distributed stochastic neighbor embedding "t-SNE" to visualize the data distribution of the synethetic images, real images and the baseline results.

C. Synthesis results

We have tested the proposed model on skin images with no hair to evaluate the quality of the synthesized images. We compared the performance of the hair generator in the proposed network using different images and different hair masks that vary from light to dense, as shown in Fig. 6-(a-f). Also, we simulated hair of skin-free images, as shown in Fig. 7-(a-f). We simulated hair with different patterns: short and long hair. Also, we synthesized ruler markers as artifacts, as shown in Fig. 6-(d) and (f).

In Fig. 6, the realistic hair is simulated according to each hair mask precisely. However, the generator with $\mathcal{L}$1-norm regularization synthesized hair with faded color and does not have sharp distinctive edges, as shown in Fig. 6-(d) and (e), and they have artifacts in terms of hair embedding in the background, as shown in Fig. 6-(a). In addition, the generated hair using $\mathcal{L}_1$ suffers from discontinuity, which makes hair appears as dotted lines, as shown in Fig. 6-(b) and (c). Color bleeding is one of the major artifacts that affects the performance of the generator with $\mathcal{L}_2$-norm, as depicted in Fig. 6-(e) and (f).

The SSIM loss layer in the regularization helped in solving the bleeding and blurred edges problem, as it penalizes the smoothness of the neighboring patches within the windows as it takes in account not only the pixels value but also the relationship to the neighboring pixels. This loss term was capable of solving the aforementioned artifacts and enhance the performance of the $\mathcal{L}_2$, as shown in Fig. 6-(a-e). In Fig. 6-(b), the simulated hair has a realistic appearance with varying diameter as the hair grows longer, it looks thin near the ends.

Also, the generator was capable of simulating hair based on dense masks with thicker hair on hair-free images, as depicted in Fig. 7. The qualitative results show the ability of the proposed methods to adapt the colors of the hair to the image context and colors. Also, the simulated hair has varying sizes with thinner ends. The synthesis results using the SSIM+$\mathcal{L}_2$ have better visual quality with sharper edges, as shown in Fig. 7-(d) and (e). We were able to simulate

doi:10.20944/preprints201810.0756.v1

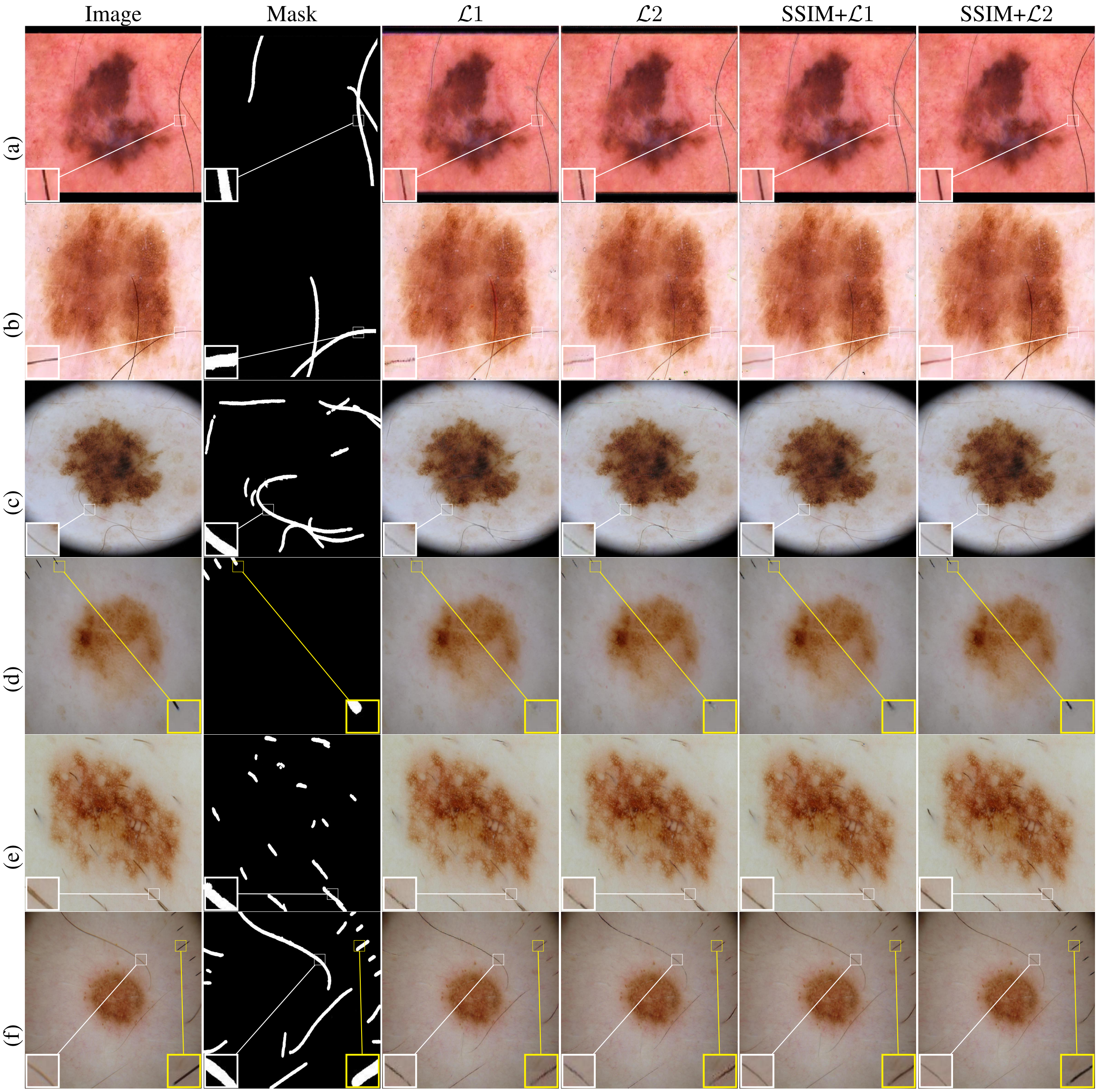


Fig. 6: Qualitative results for hair simulation using different objective functions: $\mathcal{L}_1$, $\mathcal{L}_2$, SSIM+$\mathcal{L}_1$ and SSIM+$\mathcal{L}2$. Based on the masks for hair delineation, the hair locations were inpainted. Then, using same hair mask, the hair is simulated using different optimization functions. The synthesized images were compared to the original image for validation. The white spies are for hair, while, the yellow ones are for ruler markers.

ruler markers effectively on different background, as shown in Fig. 6-(d) and (f).

We estimated the error for the proposed method between the synthetic images and the real images using different loss functions: $\mathcal{L}_1$, $\mathcal{L}_2$, SSIM+$\mathcal{L}_1$, and SSIM+$\mathcal{L}_2$. The loss function based on SSIM + $\mathcal{L}_2$ achieved the highest quality quantitatively, as shown in Table II. This superior quantitative image quality aligns with the qualitative results depicted in Fig. 6 and Fig. 7. It is worth noting that the generator network which was optimized using $\mathcal{L}_1$ produced images with higher quality than images generated using the generator with $\mathcal{L}_2$. These results agree with the experiments conducted by Zhao *et al.* [19].

D. Comparison to the Baseline Method

To validate the proposed architecture, the baseline uses artificial colors for hair. Thus, it is extremely hard to use the mean-square error and SSIM metrics for comparison. Therefore, we used t-distributed stochastic neighborhood embedding technique “t-SNE” to study the data distribution in the synthetic images compared to the real image and the images in the baseline.

doi:10.20944/preprints201810.0756.v1

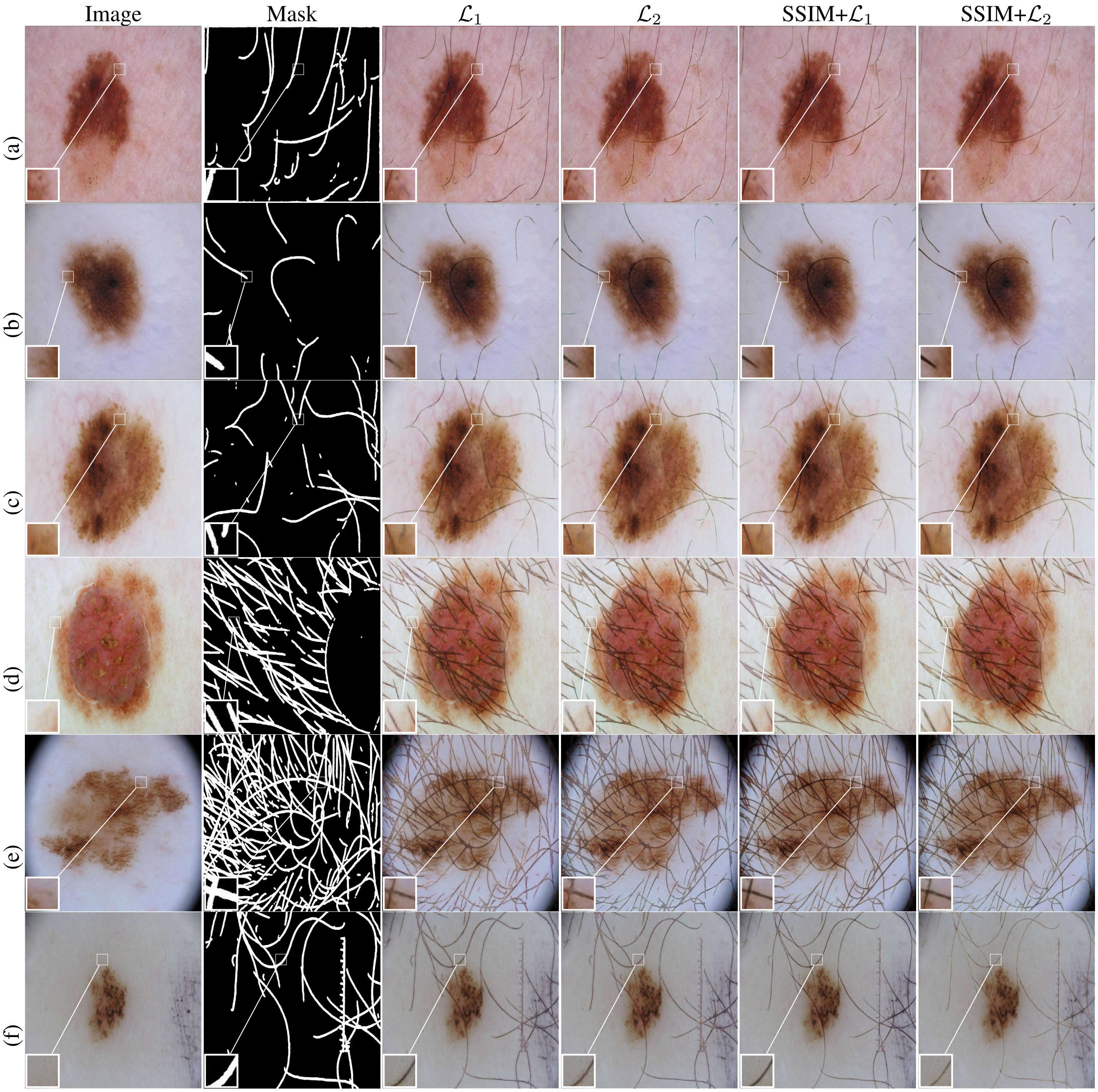


Fig. 7: Qualitative synthesis results for hair free-images using hair mask. The simulated hair has realistic appearnace on hair-free images, as shown in images (a-f). The images generated using the combined loss function $SSIM$ and $\mathcal{L}_2$ have superior quality compared to the methods.

The t-SNE method is one of the popular methods for data visualization of high dimensional data by projecting these data samples into two data pairs. This method is based on computing the similarity between the data points using the pair-wise distance metrics, such as euclidean distance. The samples are picked based on the probability distribution of data points similarity. Therefore, similar points are having higher probabilities to be picked and placed next to each other. This placement process is done according to the Kullback-Leiber divergence between the two data distributions [36].

In this work, we used t-SNE to project the high dimensional image feature vector into 2-D space [37], [36]. These image feature vectors were encoded using convolutional neural network pretrained on melanoma detection. We adopted the same network proposed by Yu *et al.* [38]. They proposed a network based on deep residual network of 18 layers [39], [38]. This network is trained to extract hierarchical features from skin images to facilitate melanoma detection. These features including the colors and edges present in skin images.

Using a pretrained network, we extracted the features encoded by the network layers except for the last layer, namely fully connected layer. These extracted features are vectors of size 2048x1 for each image. We then projected these feature vectors, using t-SNE, into a 2D space to visualize the data

TABLE II: Quantitative results for the proposed method. We calculated the mean-square error and structural similarity index for the proposed method, in compared to the real images, for the different loss function used in the network training. For mean-square error, the lower the better. Unlike the SSIM error metric, the higher the better.

Loss function	MSE	SSIM
$\mathcal{L}_1$	10.63	0.91
$\mathcal{L}_2$	10.60	0.91
SSIM+$\mathcal{L}_1$	9.71	0.92
SSIM+$\mathcal{L}_2$	**8.59**	**0.93**

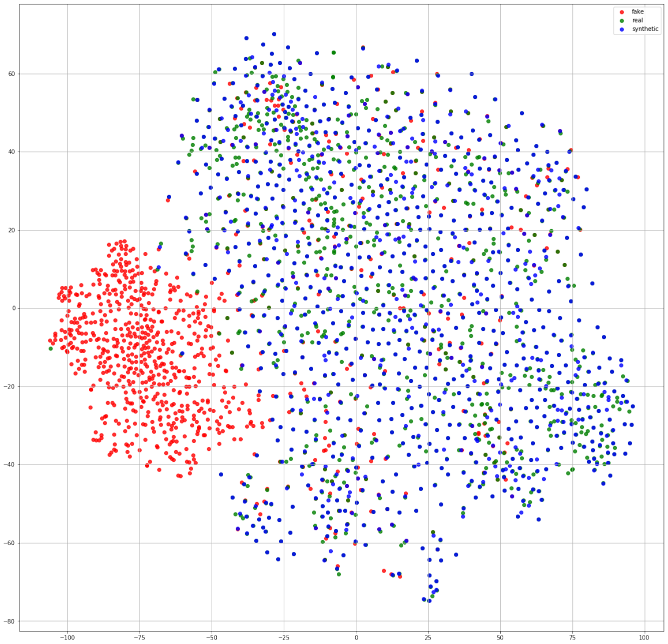


Fig. 8: High dimensional data visualization using t-SNE [36]. We projected the deep encoded image features of the same base images into 2-D space to study the data distribution in the images. Red data points represent the image with synthetic hair using the base-line method [9]. Blue data points represent the data distribution for synthetic image by the proposed method, while green data points are the real image with hair.

distribution in the images. As shown in Fig. 8, it was found that the majority of the baseline image features is segregated away from the real images features, which reflects the big difference in data distribution between the baseline images and real images [9]. While, the synthetic images using the proposed method with loss function $SSIM + \mathcal{L}_2$ follow the same data distribution, as depicted in Fig. 8. It is worth noting that some of the simulated hair by the baseline method have the same distribution as shown in Fig. 8 by the red dots within the vicinity of the green dots that represent the real image distribution.

Furthermore, we also studied the correlation of the t-SNE feature components of the synthetic and real data as shown in Fig. 9. While the extracted features from synthetic images using baseline method [9] do, sometimes, correlate with the feature components of the real data, these features do suffer from several outliers that affects the overall regression equation with an r^2 goodness of fit of 0.05. The features extracted from the synthetic images using the proposed method, on the other hand, does have a strong linear correlation with the features extracted from the real data with an r^2 goodness of fit of 0.92.

In addition to the correlation analysis, we also studied the limits of agreement between features extracted from real data with those extracted from synthetic data using the proposed method and the baseline method by Mirzaalian *et al.* in [9]. The deviations of the extracted features from synthetic images with the base line method do have wide limits of agreement from -85 to 130. The same features of images synthesized using the proposed method shows a much tighter limits of agreement from -0.33 to 0.35.

V. Conclusion

In the presented work, we proposed a novel technique for realistic hair simulation using generative adversarial networks. The proposed architecture consists of two networks: the generator and the discriminator. The generator is responsible for hair synthesis to fool the discriminator, while, the discriminator is responsible for detection of the forgery. The generator is optimized based on the data distribution within the input images and trained to embed the hair on the skin based on the skin image context. Thus, this improves the quality of the hair synthesis. The generator synthesizes hair in the marked areas with white hair-like structure. This step ensures that hair is only synthesized in the predefined areas. The output hair is well-blended with the skin regions in seamless manner without usage of any blending techniques and thus preserves the textural integrity of the surrounding skin and lesion pixels.

The proposed method facilitates the synthesis of hair occlusions to the hair images. It can also be used to validate hair segmentation and hair inpainting methods. We studied the effect of using different loss functions in training the network to improve the perceptual quality of the image. Our proposed method has more realistic appearance compared to the other methods with ability to simulate thin and thick for dark and light hair. Also, the network can simulate overlapping hair effectively. Unlike the images generated by the baseline method [9], the synthetic hair images have similar distribution of feature values to the real images as confirmed by the correlation and limits of agreement graphs.

References

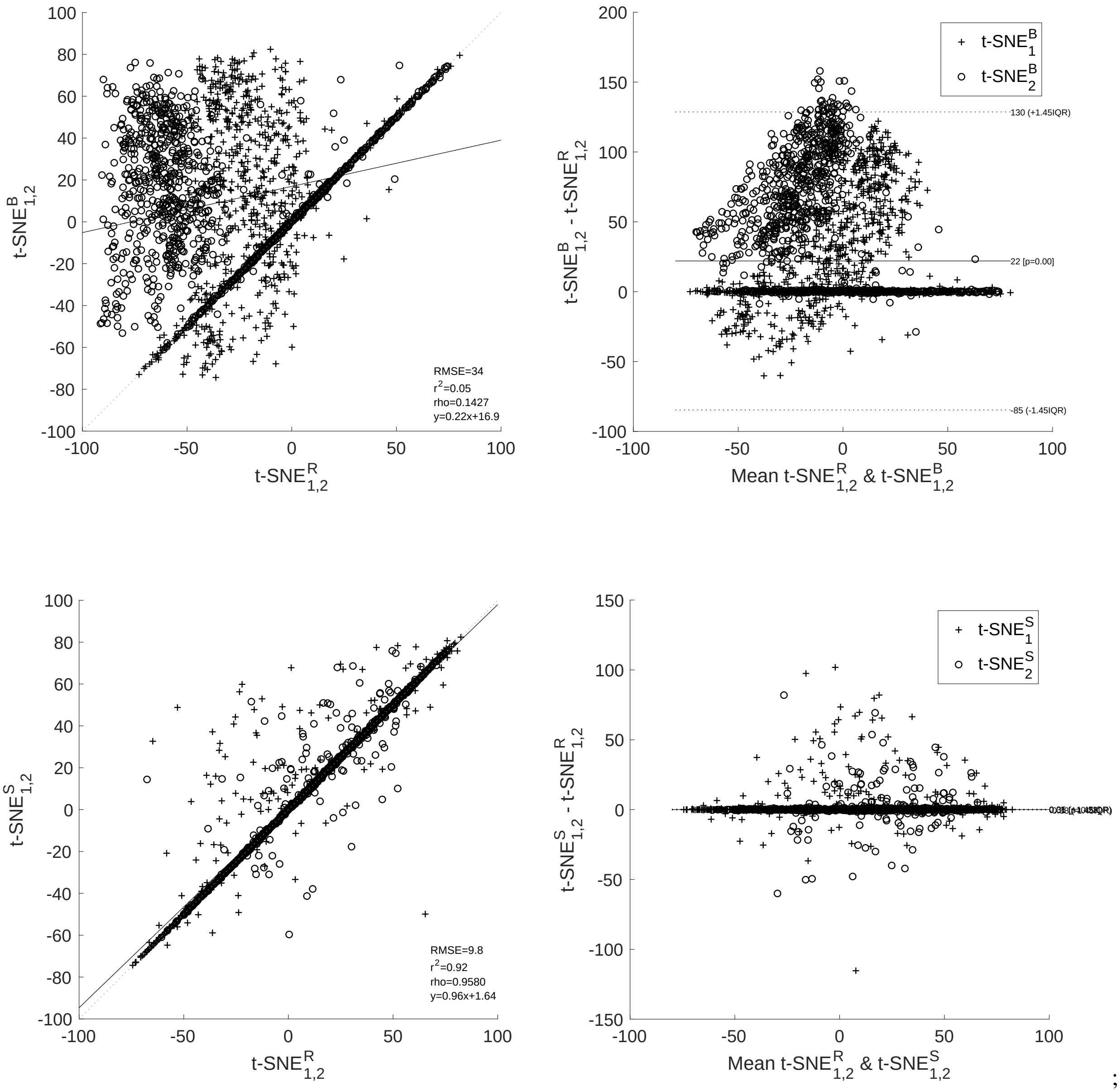
[1] S. Pathan, K. G. Prabhu, P. Siddalingaswamy, Techniques and algorithms for computer aided diagnosis of pigmented skin lesions—a review, Biomedical Signal Processing and Control 39 (2018) 237–262.

[2] F. Xie, H. Fan, Y. Li, Z. Jiang, R. Meng, A. Bovik, Melanoma classification on dermoscopy images using a neural network ensemble model, IEEE Transactions on Medical Imaging 36 (3) (2017) 849–858. doi:10.1109/TMI.2016.2633551.

[3] R. Kasmi, K. Mokrani, Classification of malignant melanoma and benign skin lesions: implementation of automatic abcd rule, IET Image Processing 10 (6) (2016) 448–455. doi:10.1049/iet-ipr.2015.0385.

[4] T. Y. Satheesha, D. Satyanarayana, M. N. G. Prasad, K. D. Dhruve, Melanoma is skin deep: A 3d reconstruction technique for computerized dermoscopic skin lesion classification, IEEE Journal of Translational Engineering in Health and Medicine 5 (2017) 1–17. doi:10.1109/JTEHM.2017.2648797.

doi:10.20944/preprints201810.0756.v1



;

Fig. 9: Statistical analysis results for the image features encoded by t-SNE, for the real images with hair “R”, baseline images by Mirzaalian [9] *et al.* “B” and the synthetic images by proposed method “S”. There is an evident strong linear correlation between the features extracted from the real $(t-SNE^R_{1,2})$ and the synthetic $(t-SNE^S_{1,2})$ images. On the other side, the extracted features do not exhibit any correlation between the real $(t-SNE^R_{1,2})$ and baseline $(t-SNE^B_{1,2})$ images [9].

[5] A. Esteva, B. Kuprel, R. A. Novoa, J. Ko, S. M. Swetter, H. M. Blau, S. Thrun, Dermatologist-level classification of skin cancer with deep neural networks, Nature 542 (7639) (2017) 115.

[6] J. Koehoorn, A. C. Sobiecki, D. Boda, A. Diaconeasa, S. Doshi, S. Paisey, A. Jalba, A. Telea, Automated digital hair removal by threshold decomposition and morphological analysis, in: International Symposium on Mathematical Morphology and Its Applications to Signal and Image Processing, Springer, 2015, pp. 15–26.

[7] I. Lee, X. Du, B. Anthony, Hair segmentation using adaptive threshold from edge and branch length measures, Computers in Biology and Medicine 89 (2017) 314 – 324. doi:https://doi.org/10.1016/j.compbiomed.2017.08.020.

[8] Q. Abbas, M. E. Celebi, I. F. García, Hair removal methods: a comparative study for dermoscopy images, Biomedical Signal Processing and Control 6 (4) (2011) 395–404.

[9] H. Mirzaalian, T. K. Lee, G. Hamarneh, Hair enhancement in dermoscopic images using dual-channel quaternion tubularness filters and mrf-based multilabel optimization, IEEE Transactions on Image Processing 23 (12) (2014) 5486–5496.

[10] P. Isola, J.-Y. Zhu, T. Zhou, A. A. Efros, Image-to-image translation with conditional adversarial networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 1125–1134.

[11] W. Denton, A. Duller, P. Fish, Synthesis of skin images for image processing validation, in: Proceedings of 9th International Conference of Biomedical Engineering, 1997, pp. 445–447.

[12] Z. She, A. Duller, Y. Liu, P. Fish, Simulation and analysis of optical skin lesion images, Skin Research and Technology 12 (2) (2006) 133–144.

doi:10.20944/preprints201810.0756.v1

[13] F. Xie, A. C. Bovik, Automatic segmentation of dermoscopy images using self-generating neural networks seeded by genetic algorithm, Pattern Recognition 46 (3) (2013) 1012–1019. doi:10.1016/j.patcog.2012.08.012.
URL http://dx.doi.org/10.1016/j.patcog.2012.08.012

[14] F.-Y. Xie, S.-Y. Qin, Z.-G. Jiang, R.-S. Meng, Pde-based unsupervised repair of hair-occluded information in dermoscopy images of melanoma, Computerized Medical Imaging and Graphics 33 (4) (2009) 275 – 282. doi:https://doi.org/10.1016/j.compmedimag.2009.01.003.

[15] Z. She, P. J. Fish, A. W. Duller, Improved approaches to hair removal from skin image, in: Medical Imaging 2001: Image Processing, Vol. 4322, International Society for Optics and Photonics, 2001, pp. 492–500.

[16] F. Xie, Y. Li, R. Meng, Z. Jiang, No-reference hair occlusion assessment for dermoscopy images based on distribution feature, Computers in biology and medicine 59 (2015) 106–115.

[17] A. F. Frangi, S. A. Tsaftaris, J. L. Prince, Simulation and synthesis in medical imaging, IEEE Transactions on Medical Imaging 37 (3) (2018) 673–679. doi:10.1109/TMI.2018.2800298.

[18] C. Dong, C. C. Loy, K. He, X. Tang, Image super-resolution using deep convolutional networks, IEEE transactions on pattern analysis and machine intelligence 38 (2) (2016) 295–307.

[19] H. Zhao, O. Gallo, I. Frosio, J. Kautz, Loss functions for image restoration with neural networks, IEEE Transactions on Computational Imaging 3 (1) (2017) 47–57.

[20] Z. Wang, E. P. Simoncelli, Maximum differentiation (mad) competition: A methodology for comparing computational models of perceptual quantities, Journal of Vision 8 (12) (2008) 8–8.

[21] Z. Wang, A. C. Bovik, H. R. Sheikh, E. P. Simoncelli, Image quality assessment: from error visibility to structural similarity, IEEE transactions on image processing 13 (4) (2004) 600–612.

[22] J. Johnson, A. Alahi, L. Fei-Fei, Perceptual losses for real-time style transfer and super-resolution, in: European Conference on Computer Vision, Springer, 2016, pp. 694–711.

[23] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, A. A. Efros, Context encoders: Feature learning by inpainting, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 2536–2544.

[24] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, in: Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, K. Q. Weinberger (Eds.), Advances in Neural Information Processing Systems 27, Curran Associates, Inc., 2014, pp. 2672–2680.

[25] M. Mirza, S. Osindero, Conditional generative adversarial nets, arXiv preprint arXiv:1411.1784.

[26] P. Costa, A. Galdran, M. I. Meyer, A. M. Mendonça, A. Campilho, Adversarial synthesis of retinal images from vessel trees, in: International Conference Image Analysis and Recognition, Springer, 2017, pp. 516–523.

[27] C. Godard, O. Mac Aodha, G. Brostow, Unsupervised monocular depth estimation with left-right consistency, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2017, pp. 6602–6611.

[28] C. Li, M. Wand, Combining markov random fields and convolutional neural networks for image synthesis, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 2479–2486.

[29] J. Koehoorn, A. Sobiecki, P. Rauber, A. Jalba, A. Telea, Effcient and effective automated digital hair removal from dermoscopy images, Mathematical Morphology-Theory and Applications 1 (1) (2016) 1–10.

[30] D. Gutman, N. C. F. Codella, E. Celebi, B. Helba, M. Marchetti, N. Mishra, A. Halpern, Skin lesion analysis toward melanoma detection: A challenge at the international symposium on biomedical imaging (ISBI) 2016, hosted by the international skin imaging collaboration (ISIC), CoRR abs/1605.01397.
URL http://arxiv.org/abs/1605.01397

[31] M. Berseth, Isic 2017-skin lesion analysis towards melanoma detection, arXiv preprint arXiv:1703.00523.

[32] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, A. A. Bharath, Generative adversarial networks: An overview, IEEE Signal Processing Magazine 35 (1) (2018) 53–65.

[33] Z. Wang, A. C. Bovik, Mean squared error: Love it or leave it? a new look at signal fidelity measures, IEEE signal processing magazine 26 (1) (2009) 98–117.

[34] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, A. Lerer, Automatic differentiation in pytorch.

[35] O. Ronneberger, P. Fischer, T. Brox, U-net: Convolutional networks for biomedical image segmentation, in: International Conference on Medical image computing and computer-assisted intervention, Springer, 2015, pp. 234–241.

[36] L. v. d. Maaten, G. Hinton, Visualizing data using t-sne, Journal of machine learning research 9 (Nov) (2008) 2579–2605.

[37] Y. Bengio, Learning deep architectures for ai, Foundations and Trends in Machine Learning 2 (1) (2009) 1–127. doi:10.1561/2200000006.
URL http://dx.doi.org/10.1561/2200000006

[38] L. Yu, H. Chen, Q. Dou, J. Qin, P. A. Heng, Automated melanoma recognition in dermoscopy images via very deep residual networks, IEEE Transactions on Medical Imaging 36 (4) (2017) 994–1004. doi:10.1109/TMI.2016.2642839.

[39] K. He, X. Zhang, S. Ren, J. Sun, Identity mappings in deep residual networks, in: European Conference on Computer Vision, Springer, 2016, pp. 630–645.