mathematics

MDPI

Article

Efficient and Low Color Information Dependency Skin Segmentation Model

Hojoon You [1], Kunyoung Lee [2], Jaemu Oh [1] and Eui Chul Lee [3,*]

[1] Department of AI & Informatics, Graduate School, Sangmyung University, Seoul 03016, Republic of Korea; 202132041@sangmyung.kr (H.Y.); 202231059@sangmyung.kr (J.O.)
[2] Department of Computer Science, Graduate School, Sangmyung University, Seoul 03016, Republic of Korea; 201933048@sangmyung.kr
[3] Department of Human-Centered Artificial Intelligence, Graduate School, Sangmyung University, Seoul 03016, Republic of Korea
* Correspondence: eclee@smu.ac.kr; Tel.: +82-2-781-7553

Abstract: Skin segmentation involves segmenting the human skin region in an image. It is a preprocessing technique mainly used in many applications such as face detection, hand gesture recognition, and remote biosignal measurements. As the performance of skin segmentation directly affects the performance of these applications, precise skin segmentation methods have been studied. However, previous skin segmentation methods are unsuitable for real-world environments because they rely heavily on color information. In addition, deep-learning-based skin segmentation methods incur high computational costs, even though skin segmentation is mainly used for preprocessing. This study proposes a lightweight skin segmentation model with a high performance. Additionally, we used data augmentation techniques that modify the hue, saturation, and values, allowing the model to learn texture or contextual information better without relying on color information. Our proposed model requires 1.09M parameters and 5.04 giga multiply-accumulate. Through experiments, we demonstrated that our proposed model shows high performance with an F-score of 0.9492 and consistent performance even for modified images. Furthermore, our proposed model showed a fast processing speed of approximately 68 fps, based on 3 × 512 × 512 images and an NVIDIA RTX 2080TI GPU (11GB VRAM) graphics card.

Keywords: skin segmentation; preprocessing; mobile vision transformer; low color information dependency

MSC: 68U10; 68T45

Citation: You, H.; Lee, K.; Oh, J.; Lee, E.C. Efficient and Low Color Information Dependency Skin Segmentation Model. *Mathematics* **2023**, *11*, 2057. https://doi.org/10.3390/math11092057

Academic Editor: Vladimir V. Arlazarov and Konstantin Bulatov

Received: 3 April 2023
Revised: 21 April 2023
Accepted: 25 April 2023
Published: 26 April 2023

1. Introduction

Skin segmentation is the task of detecting human skin regions in an image. It is a preprocessing method commonly used in various applications and is especially important in the field of biological systems and medicine [1]. Its applications include face detection, hand gesture recognition, and biosignal measurements such as remote photoplethysmography (rPPG) [2–4]. The performance of skin segmentation is important because it directly affects the performance of these applications, and it should be lightweight to avoid affecting the processing time of the entire process. For example, in rPPG measurement, information about a human heart is contained only in the human skin pixels, and elements such as the background or moving mouth, eyes, and hair across the face can contaminate the signal. Therefore, accurate skin segmentation is important for reliable measurement of rPPG signals [5,6].

Thresholding-based methods are the most commonly used skin segmentation approaches. This simple and quick technique segments the skin region by defining a limited

range of skin colors within a specific color space such as YCbCr or HSV [7–11]. However, several problems exist because skin segmentation is performed using only pixel color information. The first is the change in the illumination conditions. The range of skin color is limited, but the skin color can change owing to illumination. Changes in illumination conditions are more frequent in the real world than in laboratory environments, which significantly affects the performance of thresholding-based methods [7,12]. The second is the presence of pixels of the same color as skin. This also causes performance degradation in thresholding-based methods [8,13]. Finally, defining the range of skin colors perfectly is challenging because skin colors vary according to race or individual differences. Therefore, thresholding-based methods are unsuitable for applications where preprocessing performance is important.

Recently, owing to improvements in computer performance, learning-based methods that use machine or deep learning have attracted interest in various fields. In particular, deep-learning-based methods exhibit better performance than traditional methods, and real-time processing is possible. Therefore, studies are being actively conducted in most computer vision fields, as well as in skin segmentation. To the best of our knowledge, the best-performing deep-learning-based methods are Tarasiewicz's proposed method and Salah's proposed method [14,15]. Tarasiewicz's proposed method is based on U-Net and trained using the ECU dataset [16,17]. This method yielded a high F-score of 0.9230. Salah's proposed method classifies skin and nonskin pixels using a simple convolutional neural network (CNN). Their model was trained using the SFA dataset and showed a high F-score of 0.9765 in experiments [18].

Both Tarasiewiz's and Salah's methods showed an overperformance for each dataset; however, these methods also have problems. These problems are described in Section 2.2. In this study, we propose a high-performance method for solving these problems. The contributions of this study are as follows:

- We propose a lightweight skin segmentation method that is more suitable than previous methods for real-time application preprocessing.
- We used data augmentation techniques to reduce the color-information dependency of the model and demonstrated this experimentally.

2. Related Work

2.1. Thresholding-Based Method

As described above, the thresholding-based method defines the most suitable range of color for human skin and uses the color range to segment the skin region. This method is mainly used because it can perform skin segmentation with low processing times and does not require a training process or numerous device resources.

In Phung's study, skin was segmented by defining the range of Cb and Cr in the YCbCr color space [8]. The ranges are $Cb \in [75, 154]$ and $Cr \in [130, 180]$ as defined by the experiment. The authors of this paper explained that it quickly removed non-skin regions.

Hajraoui et al. proposed a skin segmentation method using the RGB color space [9]. This method defines two conditions for skin segmentation and classifies pixels that satisfy both conditions as skin. The conditions are shown in Equations (1) and (2).

$$\begin{gathered} R > 95, G > 40, B > 20, \\ \max(R, G, B) - \min(R, G, B) > 15, \\ |R - G| > 15, R < G, R > B. \end{gathered} \tag{1}$$

$$0.36 \leq \frac{R}{R+G+B} \leq 0.465,$$
$$0.28 \leq \frac{G}{R+G+B} \leq 0.363. \quad (2)$$

Tao et al. proposed a method for skin segmentation using YCbCr and YIQ color spaces [10]. Pixels that satisfy both conditions are classified as skin pixels. The conditions are shown in Equations (3) and (4).

$$Cr \in [-Cg + 260, -Cg + 280],$$
$$Cg \in [85,135]. \quad (3)$$

$$I \in [15,90],$$
$$Q \in [-20,10]. \quad (4)$$

Kolkur et al. proposed a method for performing skin segmentation using the RGBA and YCbCr or HSV color spaces [11]. The ranges defined are shown in Equations (5) and (6).

$$0.5 \leq H \leq 50, 0.23 \leq S \leq 0.68,$$
$$R > 95, G > 40, B > 20, R > G, R > B,$$
$$|R - G| > 15, A > 15. \quad (5)$$

$$R > 95, G > 40, B > 20, R > G, R > B,$$
$$|R - G| > 15, A > 15, Cr > 135, Cb > 85, Y > 80,$$
$$Cr \leq (0.15862 \times Cb) + 20,$$
$$Cr \geq (0.3448 \times Cb) + 76.2069,$$
$$Cr \geq (-4.5652 \times Cb) + 234.5652,$$
$$Cr \leq (-2.2857 \times Cb) + 432.85. \quad (6)$$

Although many studies have been conducted on thresholding-based methods, the performance of skin segmentation varies greatly depending on the image because of the limitation of relying on the color information of the pixels. For example, Figure 1 shows the low performance of the thresholding-based method for images of black people or images with poor illumination conditions. Figure 1 shows an example using Equation (5), as defined by Kolkur.

(**a**) (**b**)

Figure 1. Example of thresholding-based method using Kolkur's proposed method: (**a**) Example of clean skin segmentation results; (**b**) Example of poor skin segmentation results.

2.2. Deep-Learning-Based Method

Deep-learning-based methods have become dominant in the study of computer vision since the proposal of CNN. Deep learning-based methods perform better than existing methods without requiring inconvenient handcrafted features. Therefore, CNN-based

studies have been actively conducted on skin segmentation. Kim et al. proposed a network-in-network (NiN) architecture and demonstrated a better performance than existing methods or other proposals based on VGGNet [19,20]. The NiN architecture was inspired by the inception module [21].

As shown in Figure 2, the NiN architecture comprises a cascade of CNNs with a filter size of one or more CNNs. The final feature map is obtained by concatenating the feature maps of each CNN. They modified the structure of the inception module to make it effective for image-to-image tasks. They removed the pooling layer of the inception module and added a CNN layer with a filter size of seven to maintain a large receptive field. This architecture required fewer parameters than models with the same depth. Their proposed model using the NiN architecture required fewer parameters than another proposed model based on VGGNet, with a total number of 2.3M parameters. They trained and evaluated their proposed model using the ECU dataset and evaluated the performance of the model using the Pratheepan and VT-SSAT dataset [22,23]. The performance of their proposed model using the NiN architecture had an F-score of 0.8917 for the ECU dataset and 0.8957 for the Pratheepan dataset.

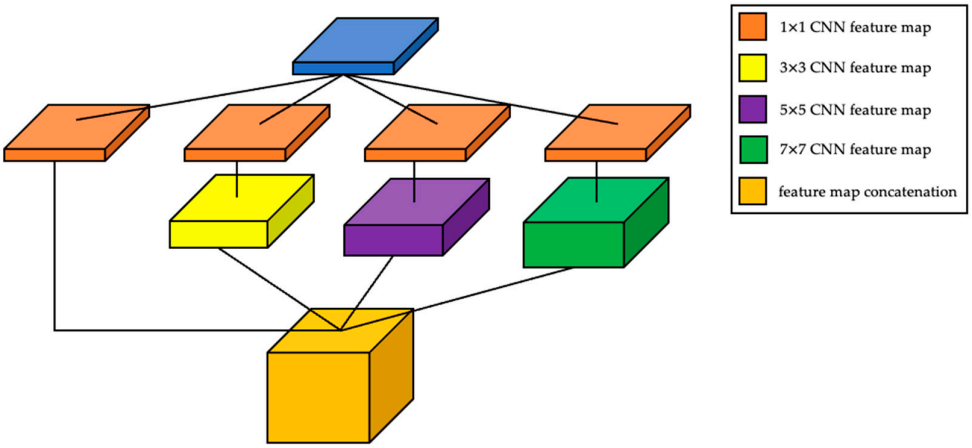

Figure 2. NiN architecture proposed by Kim.

Tarasiewicz et al. proposed Skinny based on U-Net, which is mainly used in medical image segmentation. Skinny is at a level deeper than U-Net in learning larger-sized features well. They added an inception module and a dense block to improve performance [24]. In addition, they attempted to solve the gradient vanishing problem, which is a major problem in deep networks, and reduced the number of parameters through dense blocks. They trained their model using the ECU dataset and evaluated its performance using the ECU and HGR datasets [25]. The HGR dataset is not used to train the model. The performance of Skinny was 0.9230 and 0.9494 for the ECU and HGR datasets, respectively. Skinny not only shows high performance compared to U-Net, but also requires 7.5M parameters, which is four times lower than U-Net. Furthermore, their model was shown experimentally using an NVIDIA RTX 2080Ti GPU (11GB VRAM) with 19 frames per second (fps) on 512×512 image.

Salah et al. proposed a skin segmentation method using a CNN as a class classifier to determine whether it is a skin pixel, instead of segmenting the whole or part of the image, as in the previous two methods. Their proposed method classifies the pixels individually and then uses several morphological operations to generate a final skin region mask. A summary of Salah's proposed method is presented in Figure 3. They modified the SFA dataset to be suitable for training the proposed model and experimented with it for evaluation using the SFA and Pratheepan datasets. Salah's model had an F-score of 0.9500 for the SFA dataset and 0.9765 for the Pratheepan dataset.

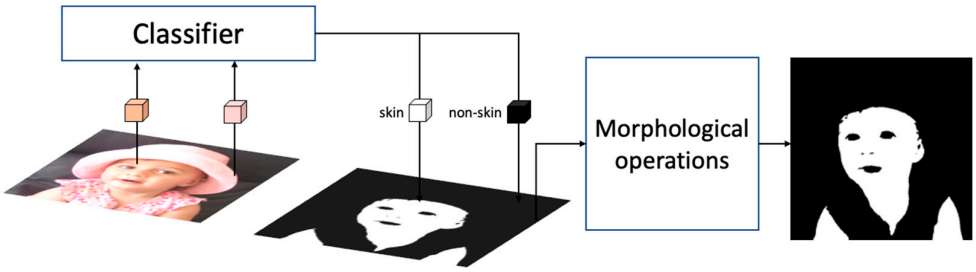

Figure 3. Summary of Salah's proposed method.

Deep-learning-based methods have shown better performance than thresholding-based methods. However, these methods exhibit several limitations. First, they did not consider changes in illumination conditions, as did the thresholding-based methods. Secondly, they did not consider the characteristics of skin segmentation, which are primarily used for preprocessing. It can be assumed that a processing time of 19 fps is sufficient for Skinny. However, when used for preprocessing, it is not fast enough to affect the processing time of the application process to be processed later. Salah's model requires few parameters, owing to the use of a simple CNN model. However, it relies on information from one pixel because it cannot learn the local information of neighboring pixels, which is an advantage of CNN. Because each pixel of the image requires computation of the model, the computational cost is not small compared with the model that uses the entire image. The computational costs of each method are listed in Table 1 as giga-multiple accumulations (GMACs). It is calculated as the sum of the number of $a \times b + c$ operations.

Table 1. The computational cost of each method. Computational costs were calculated based on 512 × 512 image.

	Kim's *	Skinny	Salah's	Ours
GMACs	674.4MMACs × 47 = 30.4278	18.37	34.96KMACs × 512 × 512 = 9.16	5.04

* The computational cost of Kim's model was multiplied by 47 because Kim's model was based on a patch of size 50 × 50 and moved 10 strides to obtain the final skin mask.

3. Method

In this section, we describe the architecture of the proposed method based on SINet and a mobile vision transformer (MobileViT) [26,27]. SINet is an extremely lightweight network used for portrait segmentation. One of the main contributions of SINet is information blocking. Information blocking can reduce typical segmentation errors by providing additional information to regions in which the model is uncertain (Section 3.1). We combined the SINet architecture with MobileViT to improve model performance. The MobileViT is a lightweight vision transformer that can simultaneously encode local and global information (Section 3.2). Therefore, we changed the encoder of SINet to a MobileViT block. However, this change caused the model to be heavy; therefore, we applied Simplified Channel Attention (SCA) to make the model lighter (Section 3.3) [28]. Finally, to reduce the color information dependence of the proposed model, we used the data augmentation technique proposed by Xu (Section 3.4) [29]. The overall architecture of the proposed model is illustrated in Figure 4.

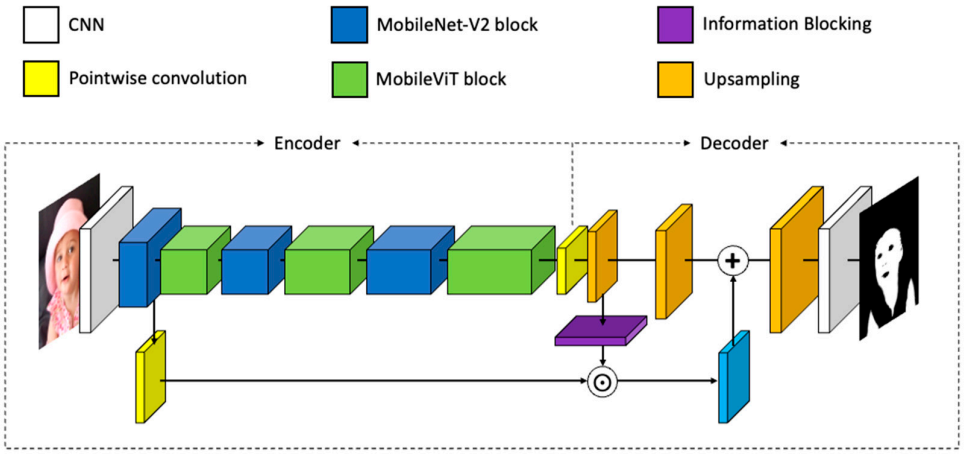

Figure 4. The overall architecture of our proposed model.

3.1. Information Blocking

Information blocking has been proposed to reduce errors in the segmentation model [26]. In the encoder–decoder structure used in SINet, the encoder loses details of the local information while extracting the feature maps. Thus, the segmentation model of this structure has low certainty at the boundary between the foreground and background. Owing to the disadvantage of losing the details of the information, several studies have often used skip connections to compensate [26]. However, using a skip connection not only provides useful information, but also unnecessary information that can act as noise. Thus, SINet applies information blocking, which provides additional information only for uncertain regions.

The equation of information blocking is shown in Equation (7):

$$M = 1 - \max\left(softmax(X_{low})\right),$$
$$I = X_{high} \odot M. \tag{7}$$

X_{low} is a feature map of the same size as the high-resolution feature map obtained by performing pointwise convolution and bilinear upsampling on the final feature maps of the encoder. X_{high} is a high-resolution feature map and $\odot$ is the mean element-wise product. The maximum softmax value in the feature maps can be considered as the confidence maps of the model for each pixel's class. By subtracting 1 from the confidence map and the element-wise product from the high-resolution feature, additional information can be provided only to low-confidence regions. This reduces the uncertainty of the model, thereby reducing the typical segmentation errors.

3.2. MobileViT

MobileViT is a type of Vision Transformer (ViT) suitable for low-resource devices such as mobile devices [30]. ViT, proposed by Dosoviskiy, showed state-of-the-art (SOTA) performance by dividing images into patches and feeding them as inputs to a vanilla transformer [31]. However, ViT lacks inductive biases compared to CNN. Therefore, it is large-scale dataset-dependent and requires strong regularization. In contrast, MobileViT has the same properties as convolution because it processes local information using a CNN and global information using a transformer. Furthermore, because of this, it has a sufficient capacity to learn visual representations, allowing the model to be lighter and faster. Therefore, MobileViT is suitable for segmentation tasks that require simultaneous handling of local and global information; therefore, we replaced the encoder of SINet with the MobileViT block to improve the performance of skin segmentation. In addition, a gate depthwise convolution feed-forward network (GDFN) was used instead of the simple

feed-forward network of MobileViT blocks [32]. The GDFN is useful for learning local image structures and allows hierarchical models to focus on fine details using the gate mechanism. The GDFN formula is given by Equation (8):

$$\begin{aligned} \hat{X} &= W_p Gating(X) + X, \\ Gating(X) &= \phi(W_d W_p LN(X)) \odot W_d W_p LN(X). \end{aligned} \tag{8}$$

W_p indicates pointwise convolution. W_d represents depthwise convolution. LN indicates layer normalization [33]. ϕ means gaussian error linear units (GELU) [34]. In addition, the activation functions of MobileViT and SINet were replaced by GELU.

3.3. Simplified Channel Attention

In this study, SCA was applied to make MobileViT lighter. SCA is an attention mechanism that simplifies Channel Attention (CA) [35]. The SCA equation is shown in Equation (9):

$$SCA(X) = X * Wpool(X) \tag{9}$$

where $*$ indicates a channel-wise product. W means convolution. $pool$ refers to global average pooling. Through experiments, the authors demonstrated that there was no performance loss in the denoising task compared with CA. In addition, the original MobileViT required the process of unfolding $X \in \mathbb{R}^{H \times W \times C}$ to $X_U \in \mathbb{R}^{P \times N \times d}$, and then folding to the original dimension to apply Attention while maintaining positional information. Here, $P = w \times h$ means the size of patch, and $N = \frac{H \times W}{P}$ means number of patches. However, SCA did not require them, so we removed those processes. A comparison of the computational costs of SINet and MobileViT is presented in Table 2. The proposed model requires 5.04 GMACs and requires the number of 1.09M parameters. The details of the model are listed in Table 3.

Table 2. The computational cost of SINet with MobileViT.

	MobileViT + SINet	**MobileViT + GDFN + SINet**	**MobileViT + GDFN + SCA + SINet**
GMACs	5.88	6.17	5.04

Table 3. The details of the model.

Input	**Operation**	**Output**
[3 × 512 × 512]	3 × 3 CNN Batch normalization	[16 × 512 × 512]
[16 × 512 × 512]	MobileV2 *	[16 × 256 × 256]
[16 × 256 × 256]		[24 × 128 × 128]
[24 × 128 × 128]		[24 × 128 × 128]
[24 × 128 × 128]		[24 × 128 × 128]
[24 × 128 × 128]		[48 × 64 × 64]
[48 × 64 × 64]	MobileViT block	[48 × 64 × 64]
[48 × 64 × 64]	MobileV2	[64 × 64 × 64]
[64 × 64 × 64]	MobileViT block	[64 × 64 × 64]
[64 × 64 × 64]	MobileV2	[80 × 64 × 64]
[80 × 64 × 64]	MobileViT block	[80 × 64 × 64]
[80 × 64 × 64]	1 × 1 CNN Batch normalization	[320 × 64 × 64]
[320 × 64 × 64]	1 × 1 CNN Upsampling Batch normalization	[2 × 128 × 128]

$[2 \times 128 \times 128] \times 2$	Information Blocking	$[2 \times 128 \times 128]$
$[2 \times 128 \times 128]$	Upsampling Batch normalization	$[2 \times 256 \times 256]$
$[2 \times 256 \times 256]$	Upsampling 3×3 CNN	$[2 \times 512 \times 512]$

* MoblieV2 is MobileNet-V2 block proposed by Sandler [36].

3.4. Xu's Data Augmentation

Existing skin segmentation methods rely on color information. This can cause performance degradation in real-world environments. To address this problem, Xu proposed a novel data augmentation technique. Xu noted that the color information dependence of deep-learning-based methods is due to the skin color bias of the dataset. Because most datasets are biased towards bright skin tones, the proposed method addresses this bias by modifying hue, saturation, value channels of the image. The hue channel of the images was rotated by 60°, the saturation channels decayed at ratios of (0.8, 0.6, 0.4, 0.2, 0.0), and the value channel changed at ratios of (1.0, 0.8, 0.6, 0.4, 0.2). The authors of demonstrated performance improvements in skin-type and race-group images through experiments using this method. We also experimentally demonstrate that the color information dependence is reduced compared to other methods that do not use this method. Examples of modified images are shown in Figure 5.

(**a**) (**b**) (**c**)

Figure 5. Example of Xu's method: (**a**) Example images of hue channel rotation; (**b**) Example images of saturation channel decay; (**c**) Example of images of value channel change.

4. Experiments

4.1. Implementation Details

The training setting of the proposed model mostly follows that of MobileViT. Similar to SINet's training, only the encoder part was trained 200 epochs with batch size 8, and then the whole model was trained 100 epochs with batch size 4. The weight of the model was initialized using a truncated normal distribution [37]. The loss function used the mean of cross entropy and DICE coefficient with reference to the experimental results of the Skinny model. To learn the boundary better, the part that calculates the loss of the model using only the boundary component was added in the same way as SINet. The equation for the loss function is shown in Equation (10):

$$Loss = \frac{1}{2}\sum_{i}^{n} CE(y_i, \hat{y}_i) + DICE(y_i, \hat{y}_i) + \frac{\lambda}{2}\sum_{j}^{k}\{CE\left(y_j^b, \widehat{y_j^b}\right) + DICE\left(y_j^b, \widehat{y_j^b}\right)\} \quad (10)$$

where n means the number of pixels in the image. y_i, $\hat{y}_i$ denote the labels of the i-th pixel and the predicted label, respectively. y_j^b, $\widehat{y_j^b}$ denote the labels of the j-th pixel and the predicted label of the boundary component image for the input image, respectively. λ means ratio that control balance of boundary loss term. CE is the cross-entropy loss function and $DICE$ is the DICE loss function using the DICE coefficient. AdamW was used as the optimizer for the model [38]. The initial learning rate of the model was 0.0002. The learning rate was increased to 0.002 by five epochs when training only the encoder and by ten epochs when training the entire model, and then lowered to 0.0002 through a cosine annealing schedule [39]. Finally, an L2 weight decay of 0.01 was used. The model was implemented using PyTorch and NVIDIA RTX 2080Ti GPU (11GB VRAM) graphics cards.

4.2. Datasets

We used the ECU datasets for training and evaluation. The ECU dataset was collected by Edith Cowan University for facial detection and skin segmentation. A total of 4000 color images were obtained. Of these, 1% were obtained through digital cameras, whereas the remainder were collected online between 2002 and 2003. They have tried to secure diversity in various ways. Therefore, images of various skin colors were collected and consisted of images of all exposed skin areas, such as the neck and arms, not only facial skin. The illumination conditions also included images acquired in indoor and outdoor environments. The data used for the training were the same as those used by Tarasiewicz. A total of 1750 images were used for training, 250 for verification, and 2000 for evaluation. At this time, 26,250 images were used for training which increased 15 times due to the data augmentation technique.

Additionally, we used only the Pratheepan dataset for the evaluation. The Pratheepan dataset contains images for skin segmentation randomly collected using Google. The dataset consisted of 32 images of faces with simple backgrounds and 46 images of multiple people with complex backgrounds, totaling 78 images.

4.3. Performance for ECU and Pratheepan Datasets

The precision, Recall, and F-score were used in all the experiments. The evaluation metrics were calculated by averaging the true positive (TP), true negative (TN), false positive (FP), and false negative (FN) for each pixel in the test dataset images. Table 4 shows the performance of the model for the ECU and Pratheepan datasets. Table 5 shows the confusion matrix of our proposed model. Salah's model has no open code or model weights; thus, no experimental results are available for the ECU dataset. For the ECU dataset, our proposed model showed a better performance than Skinny, the best performing model. For the Pratheepan dataset, Salah's proposed model performed the best. Our model performed second best. Examples of skin mask images from the Pratheepan dataset are shown in Figure 6.

Table 4. Performance of the model for ECU dataset and Pratheepan dataset.

Method	ECU			Pratheepan		
	Precision	Recall	F-Score	Precision	Recall	F-Score
Kim's	0.8720	0.9122	0.8917	0.9003	0.8912	0.8957
Skinny	0.9253	0.9299	0.9230	0.8672	0.7475	0.8333
Salah's	-	-	-	0.9801	0.9600	0.9765
SINet *	0.9230	0.9486	0.9333	0.8476	0.8168	0.8178
Ours	0.9574	0.9459	0.9501	0.9133	0.9041	0.9055

* SINet was trained using the same dataset as our proposed model.

Table 5. Confusion matrix of our proposed model.

Predicted Values	ECU		Pratheepan	
	Positive	Negative	Positive	Negative
Positive	99,505,194	3,964,144	3,451,822	225,831
Negative	5,533,018	410,872,818	256,364	16,256,424

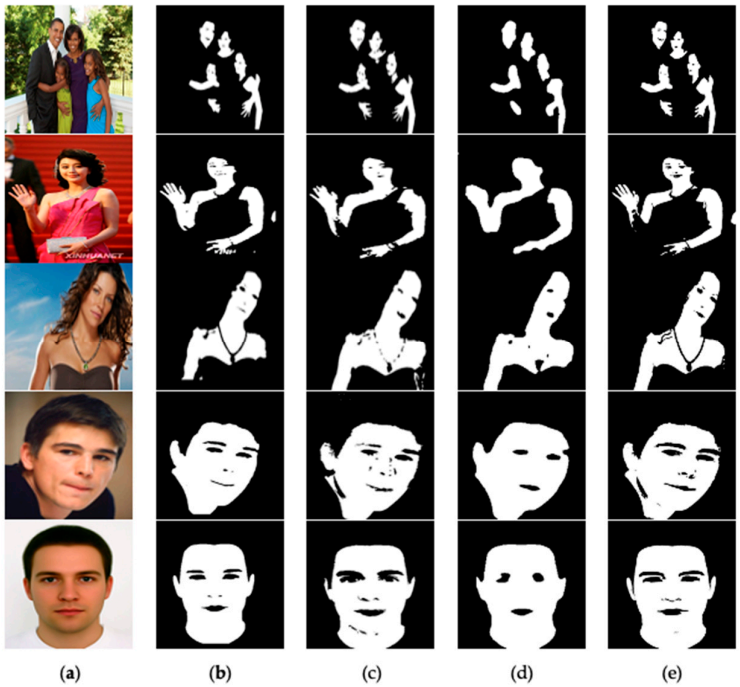

Figure 6. Examples of skin mask image for Pratheepan dataset: (**a**) Input; (**b**) Skinny; (**c**) Salah's; (**d**) SINet; (**e**) Ours.

4.4. Performance on Images Modified by the Xu's Method

Subsequent experiments used only the ECU dataset and compared the proposed model with Skinny and SINet. Table 6 shows the performance of the experiment, in which the test dataset was modified using Xu's method. In the case of Skinny, the F-score was approximately 34% compared to when the image was not modified. However, the proposed model only reduces the F-score by approximately 2%. SINet trained with the same dataset did not exhibit significant performance degradation. In addition, Skinny performed poorly on images with modified hues. In contrast, our proposed model showed that the performance was constant for any modification, demonstrating that the model has a low color information dependency owing to Xu's proposed data augmentation technique. Examples of the experimental results are shown in Figure 7.

Table 6. Performance of the model for modified images.

Modification	Skinny			SINet			Ours		
	Precision	**Recall**	**F-Score**	**Precision**	**Recall**	**F-Score**	**Precision**	**Recall**	**F-Score**
Hue	0.1845	0.0675	0.0834	0.9180	0.9419	0.9265	0.9519	0.9380	0.9428
Saturation	0.8857	0.8686	0.8640	0.9198	0.9484	0.9311	0.9539	0.9455	0.9480
Value	0.8978	0.8410	0.8468	0.9191	0.9500	0.9316	0.9559	0.9453	0.9489
Total	0.6560	0.5923	0.5980	0.9190	0.9468	0.9298	0.9539	0.9430	0.9466

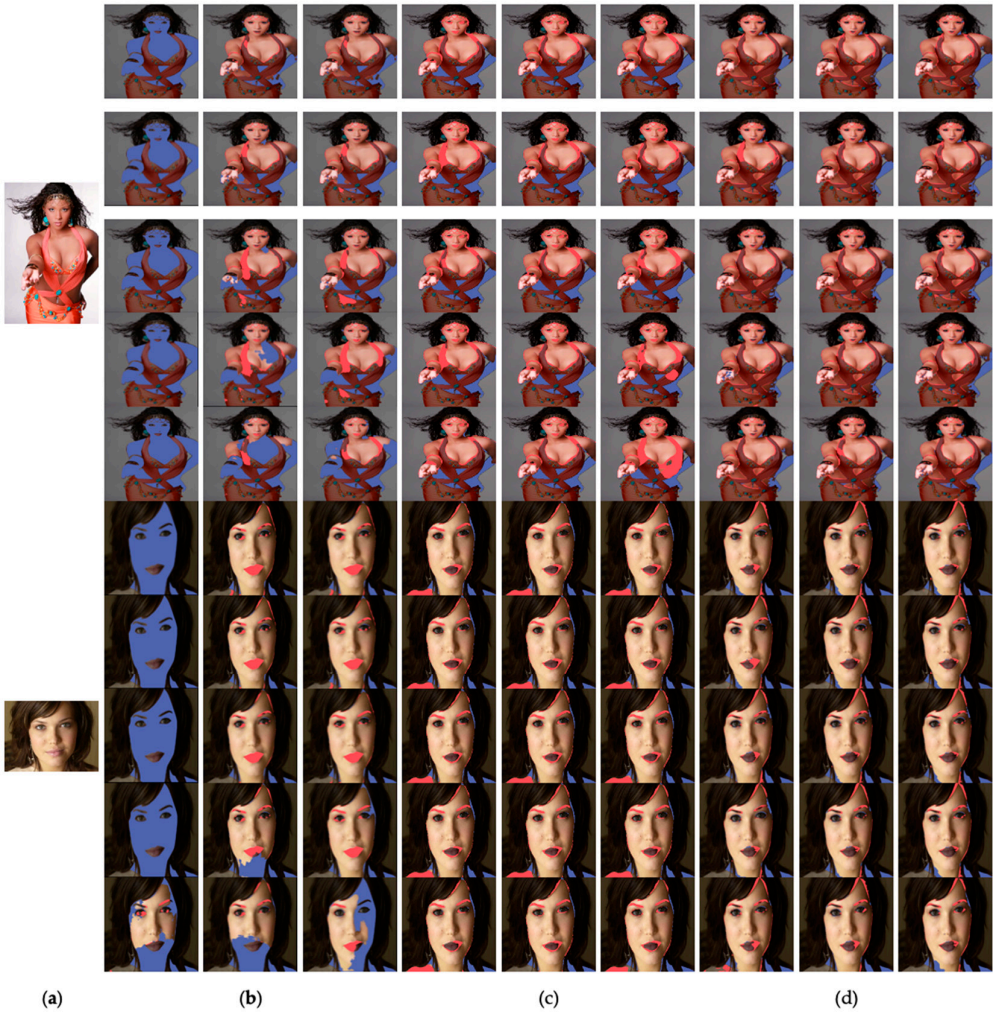

Figure 7. Examples of result image for modified image by Xu's method. First, second, and third columns are examples of result images for image modified by hue, saturation, and value, respectively (red: FP, blue: FN): (**a**) Input; (**b**) Skinny; (**c**) SINet; (**d**) Ours.

4.5. Performance for Gray Scale Images

The performances of the grayscale images are shown in Table 7. The models used in the experiment were not trained using grayscale images. The performance on grayscale images was also higher in our proposed model than in the other models. Example images from the experiment are shown in Figure 8.

Table 7. Performance of the model for gray scale images.

Metric	Skinny	SINet	Ours
Precision	0.9349	0.8815	0.8819
Recall	0.4405	0.7661	0.8288
F-score	0.5692	0.7855	0.8419

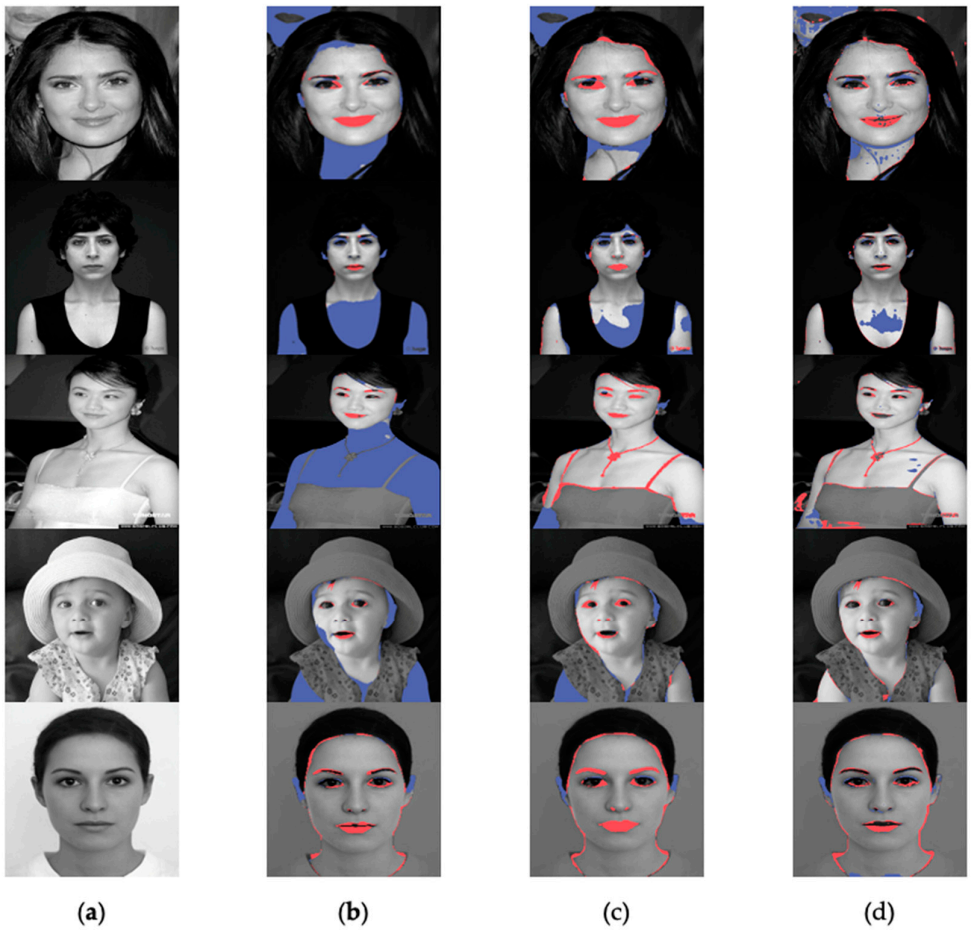

Figure 8. Examples of result image for gray scale image: (**a**) Input; (**b**) Skinny; (**c**) SINet; (**d**) Ours.

5. Discussion

The proposed model exhibited the best performance for the ECU dataset in the experiment, and the computational cost was 77% lower than that of the previous highest-performance model, Skinny. For the Pratheepan dataset, our model showed a slightly lower performance than that of Salah's proposed model. However, the proposed model is more efficient because the computational cost is approximately 44% lower, whereas the

performance decreases by 7%. Figure 9 shows the relationship between the computational cost and F-score, and the size of the circle is proportional to the number of parameters of the model.

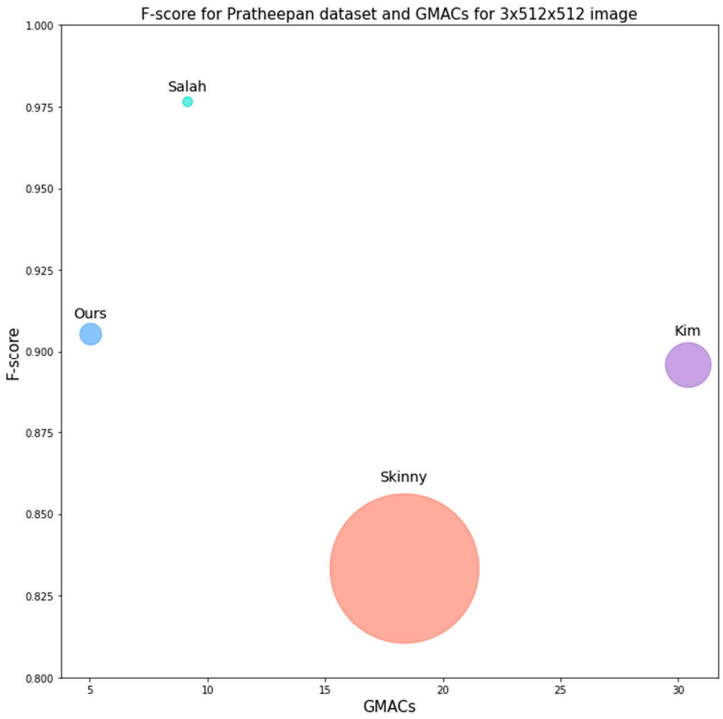

Figure 9. F-score and GMACs for the model's for Pratheepan dataset. The size of the circle is proportional to the number of parameters.

The proposed model is efficient, with a high performance and low computational cost and parameters. This is useful in applications involving devices with limited resources such as embedded or mobile devices. It can also be used for preprocessing applications that require real-time processing because it has a fast processing speed of 68 fps based on $3 \times 512 \times 512$ images and an NVIDIA RTX 2080TI GPU (11GB VRAM) graphics card.

6. Conclusions

In this study, we propose an efficient MobileViT-based skin segmentation model with low color dependency. The proposed model shows high performance in experiments on the ECU and Pratheepan datasets but requires a lower computational cost and number of parameters than the existing model. In addition, we demonstrate that our proposed model is less dependent on color information, with no significant performance degradation, even in hue, saturation, value-modified, or grayscale images.

The model proposed in this study has a lower computational cost than existing models. However, it does not have sufficient performance improvement compared to the significantly heavier SINet, owing to architectural changes in SINet. In the future, we will study ways to improve this to maintain the performance and make the model more lightweight, similar to SINet.

Author Contributions: Conceptualization, E.C.L. and H.Y.; methodology, H.Y.; software, H.Y.; validation, J.O. and K.L.; formal analysis, H.Y. and J.O.; investigation, K.L.; data curation, H.Y. and J.O.; writing—original draft preparation, H.Y.; writing—review and editing, E.C.L. and K.L.; visualization, H.Y.; supervision, E.C.L.; project administration, E.C.L. All authors have read and agreed to the published version of the manuscript.

Funding: This paper was supported by Field-oriented Technology Development Project for Customs Administration through National Research Foundation of Korea (NRF) funded by the Ministry of Science & ICT and Korea Customs Service (2022M3I1A1095155).

Data Availability Statement: Since our study used public open datasets, the data can be accessed through the website that provides the datasets.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Harsha, B.K. Skin Detection in images based on Pattern Matching Algorithms-A Review. In Proceedings of the International Conference on Inventive Computation Technologies(ICICT), Coimbatore, India, 26–28 February 2020.
2. Pujol, F.A.; Pujol, M.; Jimeno-Morenilla, A.; Pujol, M.J. Face detection based on skin color segmentation using fuzzy entropy. *Entropy* **2017**, *19*, 26.
3. Jalab, H.A.; Omer, H.K. Human computer interface using hand gesture recognition based on neural network. In Proceedings of the National Symposium on Information Technology(NSITNSW), Riyadh, Saudi Arabia, 17–19 February 2015.
4. Casado, C.A.; López, M.B. Face2PPG: An unsupervised pipeline for blood volume pulse extraction from faces. *arXiv* **2022**, arXiv:2202.04101.
5. Scherpf, M.; Emst, H.; Misera, L.; Schmidt, M. Skin Segmentation for Imaging Photoplethysmography Using a Specialized Deep Learning Approach. In Proceedings of the Computing in Cardiology (CinC), Brno, Czech Republic, 13–15 September 2021.
6. De Haan, G.; Jeanne, V. Robust pulse rate from chrominance-based rPPG. *IEEE Trans. Biomed. Eng.* **2013**, *60*, 2878–2886.
7. Naji, S.; Jalab, H.A.; Kareem, S.A. A survey on skin detection in colored images. *Artif. Intell. Rev.* **2019**, *52*, 1041–1087.
8. Phung, S.L.; Bouzerdoum, A.; Chai, D. A novel skin color model in ycbcr color space and its application to human face detection. In Proceedings of the International on Image Processing, Rochester, NY, USA, 22–25 September 2002.
9. Hajraoui, A.; Sabri, M. Face detection algorithm based on skin detection, watershed method and gabor filters. *Int. J. Comput. Appl.* **2014**, *94*, 33–39.
10. Tao, L. An FPGA-based parallel architecture for face detection using mixed color models. *arXiv* **2014**, arXiv:1405.7032.
11. Kolkur, S.; Kalbande, D.; Shimpi, P.; Bapat, C.; Jatakia, J. Human skin detection using RGB, HSV and YCbCr color models. *arXiv* **2017**, arXiv:1708.02694.
12. Störring, M. Computer Vision and Human Skin Colour. Ph.D. Thesis, Aalborg University, Aalborg, Denmark, 2004.
13. Kakumanu, P.; Makrogiannis, S.; Bourbakis, N. A survey of skin-color modeling and detection methods. *Pattern Recognit.* **2007**, *40*, 1106–1122.
14. Tarasiewicz, T.; Nalepa, J.; Kawulok, M. Skinny A lightweight u-net for skin detection and segmentation. In Proceedings of the IEEE International Conference on Image Processing(ICIP), Abu Dhabi, United Arab Emirates, 25–28 October 2020.
15. Salah, K.B.; Othmani, M.; Kherallah, M. A novel approach for human skin detection using convolutional neural network. *Vis. Comput.* **2022**, *38*, 1833–1843.
16. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. In Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention, Munich, Germany, 5–9 October 2015.
17. Phung, S.L.; Bouzerdoum, A.; Chai, D. Skin segmentation using color pixel classification: Analysis and comparison. *IEEE Trans. Pattern Anal. Mach. Intell.* **2005**, *27*, 148–154.
18. Casati, J.P.B.; Moraes, D.R.; Rodrigues, E.L.L. SFA: A human skin image database based on FERET and AR facial images. In Proceedings of the IX Workshop on Computational Vision—WVC 2013, Rio de Janeiro, Brazil, 3–5 June 2013.
19. Kim, Y.; Hwang, I.; Cho, N.I. Convolutional neural networks and training strategies for skin detection. In Proceeding of the IEEE International Conference on Image Processing(ICIP), Beijing, China, 17–20 September 2017.
20. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
21. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–15 June 2015.
22. Tan, W.R.; Chan, C.S.; Yogarajah, P.; Condell, J. A fusion approach for efficient human skin detection. *IEEE Trans. Ind. Inform.* **2011**, *8*, 138–147.
23. Abdallah, A.S.; Bou El-Nasr, M.A.; Abbott, A.L. A new color image database for benchmarking of automatic face detection and human skin segmentation techniques. *Int. J. Comput. Inf. Eng.* **2007**, *1*, 3782–3786.
24. Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K.Q. Densely connected convolutional network. In Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017.
25. Kawulok, M.; Kawulok, J.; Nalepa, J.; Smolka, B. Self-adaptive algorithm for segmenting skin region. *EURASIP J. Adv. Signal Process.* **2014**, 170.
26. Park, H.; Siosund, L.; Yoo, Y.; Monet, N.; Bang, J.; Kwak, N. Sinet: Extreme lightweight portrait segmentation networks with spatial squeeze module and information blocking decoder. In Proceeding of the IEEE/CVF Winter Conference on Applications of Computer Vision, Snowmass Village, CO, USA, 1–5 March 2020.

27. Mehta, S.; Rastegari, M.; Mobilevit: Light-weight, general-purpose, and mobile friendly vision transformer. *arXiv* **2021**, arXiv:2110.02178.
28. Chen, L.; Chu, X.; Zhang, X.; Sun, J. Simple baselines for image restoration. In Proceeding of the Conference on Computer Vision—ECCV, Tel Aviv, Israel, 23–27 October 2022.
29. Xu, H.; Sarkar, A.; Abbott, A.L. Color Invariant Skin Segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 19–24 June 2022.
30. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Houlsby, N. An image is worth 16×16 words: Transformers for image recognition at scale. *arXiv* **2020**, arXiv:2010.11929.
31. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In *Advances in Neural Information Processing Systems*; MIT Press: Cambridge, MA, USA, 2017; p. 30.
32. Zamir, S.W.; Arora, A.; Khan, S.; Hayat, M.; Khan, F.S.; Yang, M.H. Restormer: Efficient transformer for high-resolution image restoration. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 19–24 June 2022.
33. Ba, J.L.; Kiros, J.R.; Hinton, G.E. Layer normalization. *arXiv* **2016**, arXiv:1607.06450.
34. Hendrycks, D.; Gimpel, K. Gaussian error linear units (gelus). *arXiv* **2016**, arXiv:1606.08415.
35. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018.
36. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.C. Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018.
37. Hanin, B.; Rolnick, D. How to start training: The effect of initialization and architecture. In *Advances in Neural Information Processing Systems*; MIT Press: Cambridge, MA, USA, 2018; p. 31.
38. Loshchilov, I.; Hutter, F. Decoupled weight decay regularization. *arXiv* **2017**, arXiv:1711.05101.
39. Loshchilov, I.; Hutter, F. Sgdr: Stochastic gradient descent with warm restarts. *arXiv* **2016**, arXiv:1608.03983.