



## Article

# Real-time Segmentation and Classification of Facial Skin Tone Levels

 Ling Luo <sup>1,†</sup>, Dingyu Xue <sup>1,\*</sup> and  Xinglong Feng <sup>1,‡</sup>

<sup>1</sup> College of Information Science and Engineering, Northeastern University, Shenyang 110819, China; lingluo@stumail.neu.edu.cn

\* Correspondence: xuedingyu@mail.neu.edu.cn

‡ These authors contributed equally to this work.

Received: date; Accepted: date; Published: date

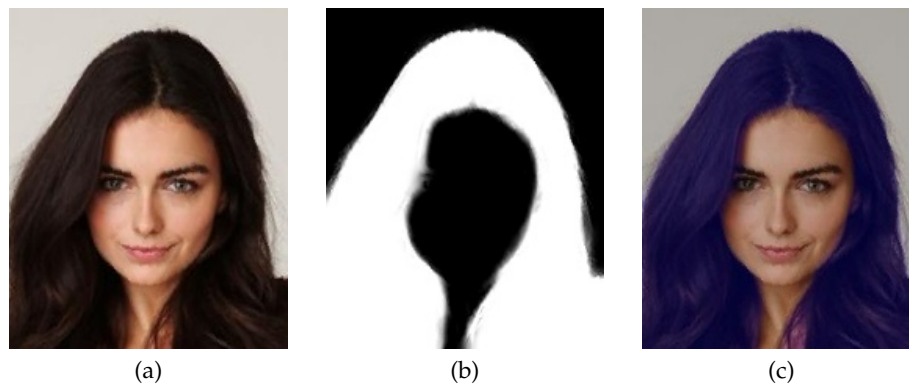


**Abstract:** Real-time semantic segmentation plays a crucial role in industrial applications, such as autonomous driving, the beauty industry, and so on. It is a challenging problem to balance the relationship between speed and segmentation performance. To address such a complex task, this paper introduces an efficient convolutional neural network (CNN) architecture named HLNet for devices with limited resources. Based on high-quality design modules, HLNet better integrates high-dimensional and low-dimensional information while obtaining sufficient receptive fields, which achieves remarkable results on three benchmark datasets. To our knowledge, the accuracy of skin tone classification is usually unsatisfactory due to the influence of external environmental factors such as illumination and background impurities. Therefore, we use HLNet to obtain accurate face regions, and further use color moment algorithm to extract its color features. Specifically, for a  $224 \times 224$  input, using our HLNet, we achieve 78.39% mean IoU on Figaro1k dataset at over 17 FPS in the case of the CPU environment. We further use the masked color moment for skin tone grade evaluation and approximate 80% classification accuracy demonstrate the feasibility of the proposed method. Code is available at: <https://github.com/JACKYLUO1991/Face-skin-hair-segmentaiton-and-skin-color-evaluation>.

**Keywords:** semantic segmentation; deep convolutional neural network; skin tone classification; color moment

## 1. Introduction

AR (Augmented Reality) technology has been widely used in various fields as a hot spot in recent years. Among them, 2D automatic hair-dyeing (as shown in Figure 1) attracts the most attention. However, its application is limited by several factors. First of all, because the hair has a very complex shape structure [1], it is quite difficult to handle accurate edge information. Although the existing semantic segmentation methods[2–4] have relatively high segmentation performance for simple objects, only a relatively rough mask can be obtained in the processing of hair segmentation. Secondly, almost all networks require GPUs with powerful computing capabilities that most mobile devices do not have, which greatly limits their usage scenarios. Thirdly, taking into account runtime limitations, Conditional Markov random fields (CRFs) [5] is not suitable for processing the edge (*e.g.* hair), so it is necessary to find an alternative. Taking all these factors into consideration, real-time hair dyeing faces enormous challenges. Besides, e-commerce and digital interaction with clients allows people to buy their favorite products without leaving home. Among them, the robust product recommendation function plays an important role. Automatic assessment of skin tone levels which makes it possible to personalize recommendation for beauty products. However, taking into account complex environmental factors, such as lighting, shadows, and imaging equipment can affect the evaluation of skin tone levels, in



**Figure 1.** Automatic hair dyeing exemplar. **(a)** Input RGB image. **(b)** The guided filter output of our proposed algorithm. **(c)** Final dyed rendering.

which case even an experienced skin therapist can hardly judge it with the naked eye. This paper is dedicated to solving the aforementioned problems using machine learning and fiery deep learning algorithms.

The advent of deep convolutional neural networks (DCNNs) has improved the performance of many tasks, the most significant of which is semantic segmentation. Semantic segmentation is an advanced visual task, whose goal is to assign dense labels to each image pixel. However, limited by bulky backbones, existing state-of-the-art (SOTA) models are usually not suitable for real-world applications. In this paper, we strive to balance the relationship between performance and efficiency, and provide a much simpler and more compact alternative for our segmentation task. To get accurate segmentation results, local and global context information should be considered simultaneously. Based on this observation, we propose a spatial and context information fusion framework called HLNet, which integrates high-dimensional and low-dimensional feature maps in parallel. While increasing the receptive field, it effectively alleviates the insufficient extraction of shallow features. Moreover, inspired by BiSeNet [6], the FFM module is used to re-encode feature channels using context to improve feature representation in a particular category. Extensive experiments confirm that HLNet achieves significant trade-off between efficiency and accuracy. Considering that background illumination is not conducive to identifying skin tone, we extract features (*a.k.a.* masked color moments) based on the segmented face and color moment algorithm [7]. The mask color moments are then input into a powerful Random Forest classifier [8] to evaluate a person's skin tone level. We verify the feasibility of the method on a manually labeled dataset.

In summary, our main contributions are as follows:

- We propose an efficient hair and face segmentation network that utilizes newly proposed modules to achieve real-time inference while guaranteeing performance.
- We propose a module called InteractionModule, which exploits multi-dimensional feature interactions to mitigate the weakening of spatial information as the network becomes deeper and deeper.
- A novel skin color level evaluation algorithm is proposed and obtains accurate results on manually labeled datasets.
- Our method achieves comparable results on multiple benchmark datasets.

## 2. Methodology

### 2.1. Related Work

#### 2.1.1. Lightweight model

Since pioneering work [2] based on deep learning, many high-quality backbones [9–12] have been derived. However, due to the requirements of computationally limited platforms (e.g., drones, autonomous driving, smartphone), people pay more attention to the efficiency of the network than just the performance.

ENet [13] is the first lightweight network for real-time scene segmentation which does not apply any post-processing steps in an end-to-end manner. Zhao *et al.* [14] introduced a cascade feature fusion unit to quickly achieve high-quality segmentation. Howard *et al.* [15] proposed a compact encoder module based on a streamlined architecture that uses depthwise separable convolutions to build light-weight deep neural networks. Poudel *et al.* [16] combined spatial detail at high resolution with deep features extracted at lower resolution yielding beyond real-time effects. DFANet [17] starts from a single lightweight backbone and aggregates discriminative features through sub-network and sub-stage cascade respectively. Recently, LEDNet [18] has been proposed which channel split and shuffle are utilized in each residual block to greatly reduce computation cost while maintaining higher segmentation accuracy.

#### 2.1.2. Contextual information

Some details cannot be recovered during conventional up-sampling of the feature maps to restore the original image size. The design of skip connections [19] can alleviate this deficiency to some extent. Zhao *et al.* [11] proposed a pyramid pooling module that can aggregate context information from different regions to improve the ability to capture multi-scale information. Zhang *et al.* [20] designed a context encoding module to introduce global contextual information, which is used to capture the context semantics of the scene and selectively highlight the feature map associated with a particular category. Fu *et al.* [21] addressed the scene parsing task by capturing rich contextual dependencies based on spatial and channel attention mechanisms, which significantly improved the performance on numerous challenging datasets.

#### 2.1.3. Post processing

Generally, the quality of the above segmentation methods is obviously rough and requires additional post-processing operations. Post-processing mechanisms are usually able to improve image edge detail and texture fidelity, while maintaining a high degree of consistency with global information. Chen *et al.* [22] proposed a CRF post processing method that overcomes poor localization in a non-end-to-end way. CRFasRNN [5] considers the CRF iterative reasoning process as an RNN operation in an end-to-end manner. In order to eliminate the excessive execution time of CRF, Levinshtein *et al.* [23] presented a hair matting method with real-time performance on mobile devices.

#### 2.1.4. Color feature

Color Histograms [24] are widely used color features in many image retrieval systems which describe the proportion of different colors in the entire image. Calculating the color histogram requires dividing the color space into a number of small color intervals, each called a bin. This process is often referred to as color quantization. Since the color space such as HSV and LAB is more in line with people's subjective judgment on color similarity, RGB space is not generally used. The biggest disadvantage of this method is that it cannot represent the local distribution of colors in the image and the spatial location of each color.

Color Moment [7] proposed by Stricker and Orengo is another straightforward and effective color feature. The mathematical basis of this method is that any color distribution in an image can be represented by its moment. In addition, since the color distribution information is mainly concentrated in the low-order moment, only the first moment (mean), the second-order moment (variance) and third-order moment (skewness) of the color are sufficient to express the color distribution of the image. Another benefit of this approach is that it does not require vectorization of features compared to color histograms.

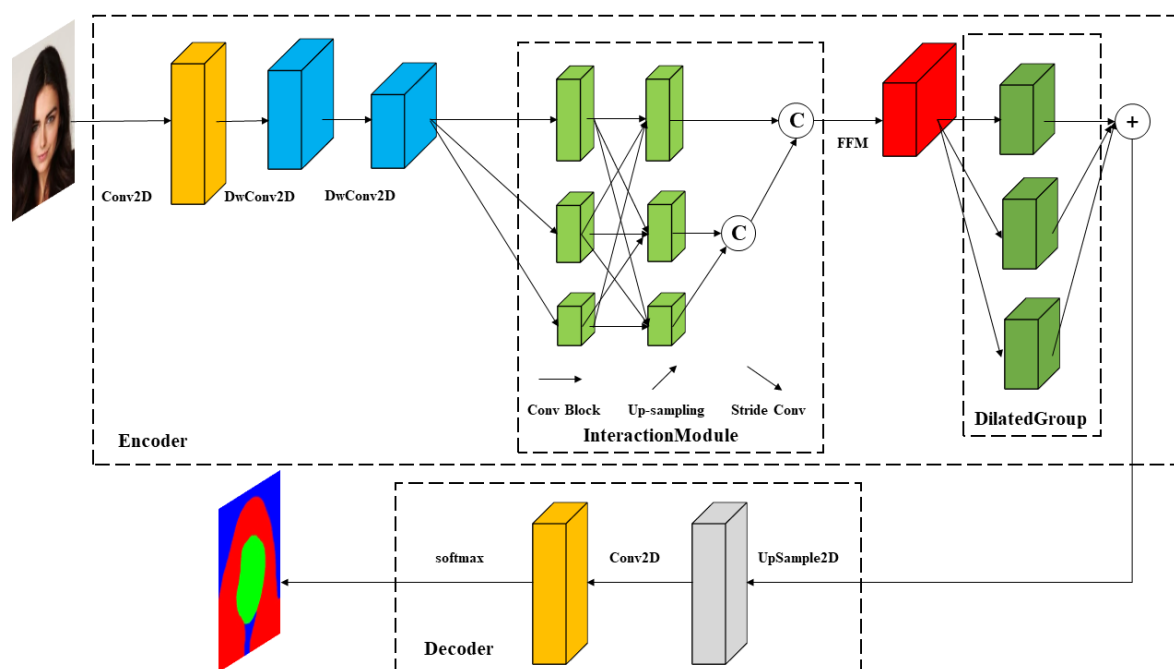
Color Correlogram [25] is also an expression of the color distribution of the images. This feature not only depicts the proportion of pixels in a certain color to the entire image, but also reflects the spatial correlation between pairs of different colors. However, this solution is too complicated in time.

Overall, our approach is closely related to asymmetric encoding and decoding structure. Furthermore, we employ masked color moment to sort the skin tone, which will be discussed in section 2.2.2.

## 2.2. Main Process

### 2.2.1. High-to-low dimension fusion network

The proposed HLNet network is inspired by HRNet [26] which maintains high-resolution representation through the whole process by connecting high-to-low resolution convolutions in parallel. Figure 2 illustrates the overall framework of our model. We experimentally prune the model parameters to increase the speed without excessive performance degradation. Furthermore, the existing SOTA modules [6,16,27,28] are reasonably combined to further improve the performance of the network. Table 1 gives an overall description of the modules involved in the network. The model consists of different kinds of convolution modules, bilinear up-sampling, bottlenecks, and other feature maps communication modules. In the following part, we will expand the above modules in detail.



**Figure 2.** An overview of our asymmetric encoder-decoder network. Blue, red and green represents background, hair mask recolored and face mask recolored, respectively. In the dotted rectangle (also called InteractionModule), arrows in different directions represent different operations. "C" and "+" represent *Add* and *Concat* operations, respectively.

**Table 1.** HLNet consists of an asymmetric encoder and decoder. The whole network is mainly composed of standard convolution (Conv2D), deep separable convolution (DwConv2D), inverted residual bottleneck blocks, upsampling (UpSample2D) module and specially designed modules.

Stage	Type	Output Size
Encoder	-	$224 \times 224 \times 3$
	Conv2D	$112 \times 112 \times 32$
	DwConv2D	$56 \times 56 \times 64$
	DwConv2D	$28 \times 28 \times 64$
	InteractionModule	$28 \times 28 \times 128$
	FFM	$28 \times 28 \times 64$
	DilatedGroup	$28 \times 28 \times 32$
Decoder	UpSample2D	$224 \times 224 \times 32$
	Conv2D	$224 \times 224 \times 3$
	SoftMax	$224 \times 224 \times 3$

To preserve details as much as possible, the downsampling rate of the entire network is 1/8. In the first three layers, we refer to Fast-SCNN [16] to employ standard convolution and depth separable convolution for fast down-sampling in order to ensure low-level feature sharing. Depth separable convolution reduces the amount of model parameters effectively while achieving a comparable representation ability. The above convolution unified use stride 2 and  $3 \times 3$  spatial kernel size, followed by BN [29] and ReLU activation function.

According to FCOS [30], the low-dimensional detail information of the feature map promotes the segmentation of small objects, so we strengthen the model's ability to represent details by stacking low-dimensional layers. Interaction of high-resolution and low-resolution information facilitates learning of multi-scale information representation. In this part, we draw the above advantages and propose a information interaction module (InteractionModule) with feature maps of different resolutions to obtain elegant output results. For the backbone  $\Phi_n^i(x)$ , a stage process can be defined as  $\phi_n^i$ , where  $n$  and  $i$  represent the index and the width of the stage, respectively. The calculation process in the dotted rectangle can be formulated as:

$$\phi_n^i = \begin{cases} \text{Conv}(\phi_{n-1}^i), & n = 4, \\ \sum_{i=1}^M \text{Conv}(\phi_{n-1}^i), & n = 5, \\ \text{Concat}(\phi_{n-1}^1, \dots, \phi_{n-1}^M), & \text{otherwise} \end{cases} \quad (1)$$

where  $M$  is 3. *Conv* and *Concat* represent convolution operator and feature maps are stacked in the channel dimension, respectively. MobileNet v2 [27] takes advantage of residual block and deep separable convolution, which greatly reduces the calculation parameters while effectively avoiding gradient dispersion. The inverted residual block proposed by MobileNet v2 is utilized to improve the sparse parameter space by proper pruning. In particular, for  $\phi_n^i (i = 1, \dots, M)$ , the corresponding parameters  $\{k = 3, c = 64, t = 6, s = 1, n = 3\} \rightarrow \{k = 3, c = 96, t = 6, s = 2, n = 3\} \rightarrow \{k = 3, c = 128, t = 6, s = 4, n = 3\}$  are given in order, where  $k$ ,  $c$ ,  $t$ ,  $s$  and  $n$  denote the size of convolution kernel, the number of feature map channels, the channel multiplication factor, stride and the number of module repetitions, respectively. Next, feature maps of different scales are combined and exchanged by using a  $1 \times 1$  convolution, strided convolution or upsampling.  $1 \times 1$  convolution can well perform the dimensional increase and decrease of the feature map without significantly increasing the amount of parameters. The last part of the InteractionModule is implemented by using *Concat* in order to aggregate multi-scale features. Subsequently, following the FFM Attention [6], the model focuses more on channels that contain important features and suppress those that are not important. To capture multi-scale context information, we also introduce a multi-receptive field fusion block (*e.g.*, dilation rate is set to 2, 4, 8).

For simplicity, the decoder performs bilinear upsampling (transposed convolution layer can cause gridding artifacts [23]) directly on the  $28 \times 28$  feature map followed by a  $3 \times 3$  convolution to maintain that the number of channels and the number of categories are consistent. Subsequently, a softmax layer is connected for dense classification. In terms of loss function, we apply generalized dice loss (GDL) [31] to compensate for the segmentation performance of small objects, which is formulated as:

$$GDL_{loss} = 1 - \frac{2 \sum_{l=1}^L \omega_l \cdot \sum_{n=1}^N r_{ln} p_{ln}}{L \sum_{l=1}^L \omega_l \cdot \sum_{n=1}^N r_{ln} + p_{ln}} \quad (2)$$

$$\omega_l = \frac{1}{(\sum_{n=1}^N r_{ln})^2} \quad (3)$$

where  $p$  denotes the softmax output and  $r$  denotes the one-hot encoding of the ground truth.  $N$  and  $L$  represent the total number of pixels and the total number of categories, respectively. Equation 3 gives the expression of  $\omega_l$ , which is the category balance coefficient.

To pursue perceptual consistency and reduce the time complexity of running, we advocate the idea of Guided Filter [32,33] to achieve edge-preserving and denoising. Guided Filter can effectively suppress gradient-reversal artifacts and produce visually pleasing edge profiles. Given a guidance image  $I$  and filtering input image  $P$ , our goal is to learn a local linear model to describe the relationship between the former and the output image  $Q$  while seeking consistency between  $P$  and  $Q$  just like the role of Image Matting [34]. During the experiment,  $s$ ,  $r$ ,  $\zeta$  are empirically set to 4, 4, and 50, respectively.

## 2.2.2. Facial skin tone classification

---

### Algorithm 1: Segmentation-based inference algorithm for smoothed facial region extraction

---

**Input:** Given a dataset  $D = \{X_1, X_2, \dots, X_n\}$ ,  $X_i = \{x_1, x_2, \dots, x_m\}$ , where  $m$  denotes the total number of the image  $X_i$ ;  $y = f(X, \theta)$  represents the output of HLNet, and  $\theta$  is the weight of our fine-tuned network. For image processing, we use the Python bindings of the OpenCV library, named [cv2](#).

**Output:** Smoothed facial region  $F_i$ .

```

1 Step 1:
2  $M_i^j = 0$ ;
3 for  $i = 1, 2, \dots, n$  do
4   for  $j = 1, 2, \dots, m$  do
5     if  $y_i^j == \text{index of face}$  then
6        $M_i^j = 255$ ;
7     end
8   end
9 end
10 Step 2:
11 for  $i = 1, 2, \dots, n$  do
12    $P_i(w, h) \leftarrow \text{cv2.erode} \leftarrow \min M_i(w + w', h + h')$ ;
13    $Q_i(w, h) \leftarrow \text{cv2.bilateralFilter} \leftarrow P_i(w, h)$ ;
14    $F_i(w, h) \leftarrow \text{cv2.bitwise\_and} \leftarrow (I_i(w, h), Q_i(w, h))$ 
15 end
```

---

The second stage is to classify facial skin tone. Usually for Asians, we divide it into porcelain white, ivory white, medium, yellowish and black. For skin tone features, due to the small feature space, it is not suitable to use DCNNs-based methods for feature extraction. Therefore, after repeated thinking and experimental trial and error, the scheme is selected to extract the color moment of the image as the features to be learned and put it into the classic machine learning algorithm for learning. Considering facial skin tone in complex scenes, background lighting has a incurable impact on the results. So we



employ image morphology algorithms and pixel-level operations to get rid of background interference. **Algorithm 1** summarizes the pseudo code of the extraction process. The pre-processed face image is used to extract the color moment features, which are then put into a powerful Random Forest classifier [8] for learning. Color moment can be expressed as:

$$\mu_i = \frac{1}{N} \sum_{j=1}^N p_{i,j} \quad (4)$$

$$\sigma_i = \left( \frac{1}{N} \sum_{j=1}^N (p_{i,j} - \mu_i)^2 \right)^{\frac{1}{2}} \quad (5)$$

$$s_i = \left( \frac{1}{N} \sum_{j=1}^N |p_{i,j} - \mu_i|^3 \right)^{\frac{1}{3}} \quad (6)$$

where  $p_{i,j}$  represents the probability of a pixel in the  $i$  channel with a value of  $j$ , and  $N$  represents the total number of pixels. Color feature  $F_{color} = [\mu_U, \sigma_U, s_U, \mu_V, \sigma_V, s_V, \mu_Y, \sigma_Y, s_Y]$ ,  $U, V, Y$  represent each channel of the image.

### 3. Experiments

#### 3.1. Implementation Details

Our experiments are conducted using Keras framework with Tensorflow backend. Standard mini-batch gradient descent (SGD) is employed as the optimizer with the momentum of 0.98, weight decay  $2e - 5$ , and batch size 64. We adopt the widely equipped “poly” learning rate policy in configuration where the initial rate is multiplied by  $(1 - \frac{iter}{total\_iter})^{power}$  with power 0.9 and initial learning rate is set as  $2.5e - 3$ . Data augmentation includes normalization, random rotation  $[-20, 20]$ , random scale  $[-20, 20]$ , random horizontal flip and random shift  $[-10, 10]$ . For fair comparison, all the methods are conducted on a server equipped with a single NVIDIA GeForce GTX1080 Ti GPU.

#### 3.2. Datasets

Data is the soul of deep learning, because it determines the upper limit of an algorithm to some extent. To ensure the robustness of the algorithm, it is necessary to construct a dataset with human faces in extreme situation such as large angles, strong occlusions, complex lighting changes, etc.

##### 3.2.1. Face and hair segmentation datasets

Labeled Faces in the wild (LFW) [35] dataset consists of more than 13,000 images on the Internet. We use its extension version (Part Labels) during the experiment which automatically labeled via a super-pixel segmentation algorithm. We adopt the same data partitioning method in [36] as 1,500 images in the training, 500 used in validation, and 927 used for testing.

Large-scale CelebFaces Attributes dataset (CelebA) [37] consisting of more than 200k celebrity images, each with multiple attributes. The main advantage of this dataset is that it combines large pose variations and background clutter making the knowledge learned from this dataset easier to satisfy demand of actual products. In the experiment we adopt CelebHair version <sup>1</sup> of CelebA in [36] which includes 3556 images. We use the same configuration as the original paper, that is, 20% for validation.

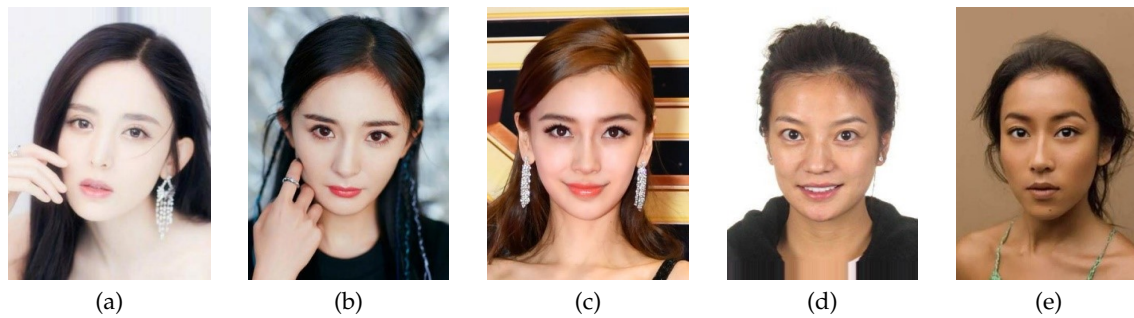
For the last dataset, we employ Figaro1k [38], which is dedicated to hair segmentation. It needs to be considered that the dataset is developed for general hair detection, many of which do not include faces, which is not conducive to subsequent experiments. In this case, we follow the pre-processing

<sup>1</sup> <http://www.cs.ubbcluj.ro/~dadi/face-hair-segm-database.html>

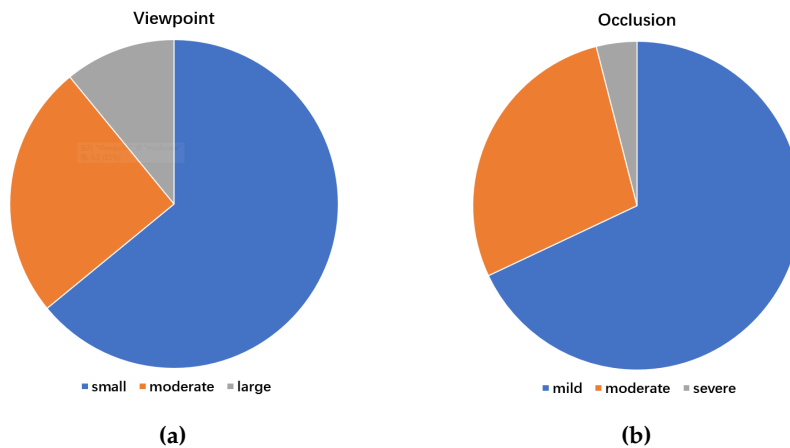
in [1], leaving 171 images for experiments. To better take advantage of batch training, offline data augmentation is adopted to expand the images ( $\times 10$ ).

### 3.2.2. Manually annotated dataset

One contribution of this work is a manually labeled facial skin tone rating dataset. For Asians, it is mainly divided into five categories: porcelain white, ivory white, medium, yellowish and black. In the process of labeling, three professionally trained makeup artists rated the face tone color using a voting mechanism. Our face data is collected from the web without conflicts of interest. The obtained image is filtered by an open source face detection library (such as MTCNN [39]) to remove images without detected faces, and the remaining ones are used for feature extraction and machine learning. The number of each category is 95, 95, 96, 93 and 94, samples are shown in Figure 3. Besides, their statistical distribution are plotted in Figure 4.



**Figure 3.** Manually marked face skin hue level sample after voting mechanism. From (a) to (e), it represents porcelain white, ivory white, medium, yellowish and black races. In order to have a consistent understanding of image classification criteria, only chinese actresses are used here.



**Figure 4.** Pie chart visualization. (a) viewpoint graph. According to the yaw angle, it is divided into small ( $|\theta| < 15^\circ$ ), moderate ( $15^\circ \leq |\theta| \leq 45^\circ$ ) and large ( $|\theta| > 45^\circ$ ). (b) Occlusion graph. Including ‘mild occlusion’ ( $< 20\%$ ), ‘severe occlusion’ ( $> 50\%$ ) and ‘moderate occlusion’ between the two items.

### 3.3. Evaluation Metrics

All segmentation experiments are applied to mean-interesection-over-union (mIoU) criterion. The definition of mIoU is as follows:

$$mIoU = \frac{1}{1+k} \sum_{i=0}^k \frac{p_{ii}}{\sum_{j=0}^k p_{ij} + \sum_{j=0}^k p_{ji} - p_{ii}} \quad (7)$$



where  $k + 1$  is the number of classes (including background),  $p_{ij}$  indicates the number of pixels that belong to category  $i$  but have been misjudged as category  $j$ . For more metrics, please refer to [2].

## 4. Results and Discussion

### 4.1. Segmentation Results

In this section, we carry on the experiments to demonstrate the potential of our segmentation architecture in terms of accuracy and efficiency trade-off.

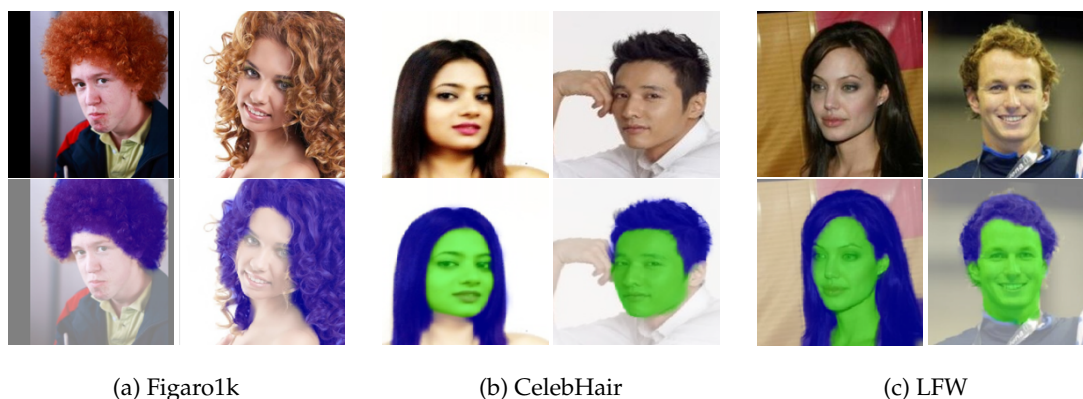
#### 4.1.1. Overall comparison

We use four FCN [2] introduced metrics to evaluate the performance of our algorithm. Subsequently, comparative experiments across different datasets with the outstanding UNet variant [36] are constructed. Unless otherwise stated, the input resolution is  $224 \times 224$ . The training continues for 200 epochs, after which the model will become saturated. Table 2 reports the qualitative results.

**Table 2.** Segmentation performance on LFW, CelebHair and Figaro1k test sets. "OC" denotes the number of output channels. All values are in %. Moreover, the best one is highlighted in bold.

Metric	LFW (OC=3)		CelebHair (OC=3)		Figaro1k (OC=2)	
	U-Net	HLNet	U-Net	HLNet	U-Net	HLNet
mIoU	83.46	<b>83.81</b>	88.56	<b>89.55</b>	77.75	<b>78.39</b>
fwIoU	<b>92.75</b>	90.28	91.79	<b>91.98</b>	83.01	<b>83.12</b>
pixelAcc	<b>95.83</b>	94.69	95.54	<b>96.08</b>	90.28	<b>90.73</b>
mPixelAcc	88.84	<b>90.35</b>	93.61	<b>94.49</b>	84.72	<b>84.93</b>

Experimental results show that HLNet outstanding the trimmed U-Net (tU-Net) [36] by a large margin, except for LFW dataset. However, one drawback of fast down-sampling is that the feature extraction for the shallow layers is not sufficient. As we know, shallow features contribute to extracting texture and edge details, so our HLNet is slightly worse than the tU-Net in LFW dataset (LFW facial details are blurry than others). Furthermore, considering the latency time, we reach 60 ms per image on an Intel Core i5-7500U CPU without any tricks. We can further reach no more than 10 ms under GPU. Comparing tU-Net with HLNet (8 ms vs  $7.2 \pm 0.3$  ms) shows that the latter is more efficient, while performance is more remarkable. This conclusion suggests that we can further apply this framework to the edge and embedded devices with small memory and battery budget. The qualitative analysis results are shown in Figure 5. Post-processing employs Guided Filter to achieve more realistic edge results.



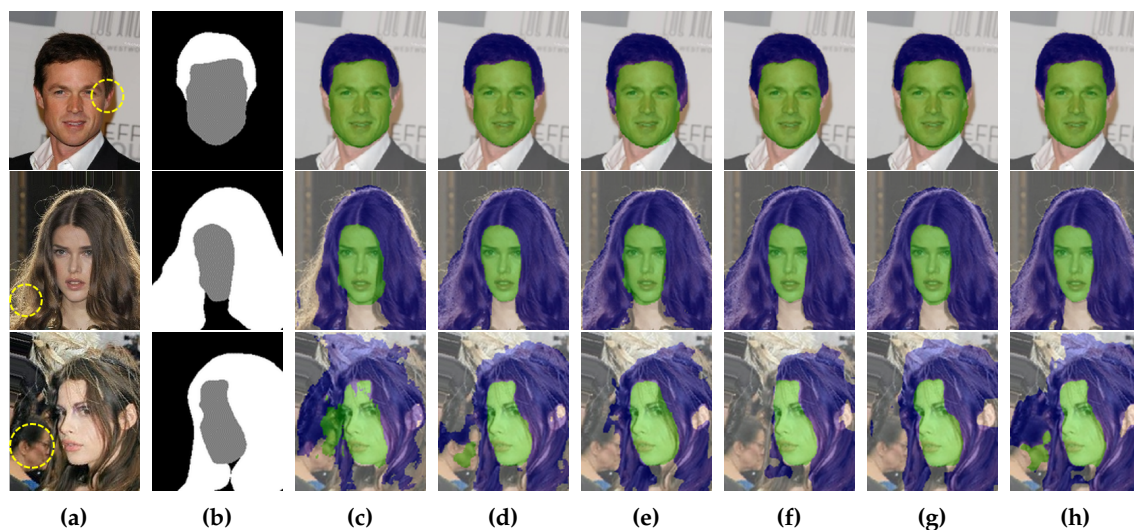
**Figure 5.** Samples of hair and face segmentations on different datasets.

**Table 3.** Comparison with SOTA approaches on CelebHair test set in terms of segmentation accuracy and execution efficiency. "+" indicates fine-tuning from LFW. 0.5 represents the contraction factor.

Model	#Param (M)	FPS	FLOPs (G)	mIoU (%)
ENet [13]	<b>0.36</b>	8.24	0.94	89.97
LEDNet [18]	2.3	6.44	3.28	88.63
Fast-SCNN [16]	1.6	<b>20.35</b>	0.41	87.14
MobileNet(0.5) + UNet [15]	0.37	5.80	0.75	86.08
DFANet [17]	0.42	17.72	<b>0.08</b>	82.88
<b>HLNet (ours)</b>	1.2	12.23	0.94	90.32
<b>HLNet (ours) +</b>	1.2	12.23	0.94	<b>90.98</b>

#### 4.1.2. Comparison with SOTA lightweight networks

In this subsection, we compare our algorithm with several state-of-the-art (SOTA) lightweight networks including ENet [13], LEDNet [18], Fast-SCNN [16], MobileNet [15] and DFANet [17] on CelebHair test set. For fair comparison, we re-implement the above networks under the same hardware configuration without any fine-tuning or fancy tuning techniques. It should be noted that the framework implementation is slightly different from the original, so the results may be slightly different, but the overall performance deviation is within the acceptable range. Since ENet has a downsampling rate of 32, we resize the input to  $256 \times 256$ . In addition, we measure FPS in our CPU environment, which takes an average of 200 forward propagations.



**Figure 6.** Qualitative comparison results with other SOTA methods. From left to right are input images, ground truth, segmentation outputs from DFANet [17], ENet [13], MobileNet [15], LEDNet [18], Fast-SCNN [16] and our HLNet. From top to bottom, the difficulty of segmentation increases in turn.

From table 3 and Figure 6, we see that our proposed method is more accurate than other methods. Compared with the sub-optimal ENet, our method improves mIoU by 0.35%, while the FPS is half higher. Although DFANet has  $2\times$  less parameters, as well as  $11\times$  less FLOPs than our HLNet, it delivers poor segmentation accuracy of 7.44% in terms of mIoU. We conjecture that this is due to DFANet's overdependence on pre-trained lightweight backbones. From Figure 6(c), it can be clearly observed that the DFANet has a serious misclassification on pixels. MobileNet's situation is consistent with DFANet. In particular, our HLNet is 3.18% higher than Fast-SCNN in terms of

accuracy, and the parameters are reduced by 0.4M. Excessive deep separable convolutions affect its performance, and even if this reduces time delay and computational complexity (FLOPs), it gets insufficient generalization capabilities. Compare the last line of Figure 6(g) and figure 6(h), which contains the second person (the latter one), even if the Ground Truth does not mark it. Benefiting from the rich context captured by the DilatedGroup, our method can roughly segment it. Moreover, compared with other methods, with the help of the introduced InteractionModule, HLNet has an advantage in detail processing of multi-scale objects (*i.e.* hairline). The whole experiment demonstrate that our HLNet achieve the best trade-off between accuracy and efficiency.

#### 4.1.3. Ablation study

We conduct the ablation experiments on Figaro1k test dataset, and follow the same training strategy for the fairness of the experiments. We mainly evaluate the impact of InteractionModule (IM) and DilatedGroup (DG) components on the results, as shown in Table 4. IM without information exchange (connected using ‘Upsampling’ and ‘concat’) and a  $3 \times 3$  convolution with a rate of 1 are used to replace the corresponding components as *baseline*. When we append DG and IM modules respectively, mIoU increases by 1.54% and 3.19% relative to the *baseline*. When we apply two modules at the same time, mIoU increases dramatically by 4.26%. The obvious performance gains illustrates the efficiency of our proposed modules.

**Table 4.** The effect of our proposed InteractionModule (IM) and DilatedGroup (DG) which is evaluated on Figaro1k test set.

Method		mIoU
IM	DG	
		74.13
	✓	75.67
✓		77.32
✓	✓	<b>78.39</b>

#### 4.2. Facial skin tone classification results

In the second phase of the experiment, we construct comparative studies to compare the influence of different color spaces and different experimental protocols on the results.

As shown in Table 5, we report the accuracy of facial skin color classification. The best results are obtained using the YCrCb space with color moment backend. It should be noted that before putting into the classifier, the data needs to be oversampled first to ensure that the number of samples between different categories is consistent. We simply split the dataset into 8 : 2 for training and testing, and then use the powerful Random forest classifier for training. Figure 7 gives the confusion matrix for this configuration. It can be observed from the figure that the main mistakes are between adjacent categories, and this situation also plagues a trained professional makeup artist when labeling data.

**Table 5.** Classification accuracy of different methods in different color spaces. PCA stands for principal component analysis.

Method	RGB	HSV	YCrCb
Histogram (8 bins)	75%	78%	73%
Histogram with PCA (256 bins)	77%	-	-
Color Moment	73%	77%	<b>80%</b>

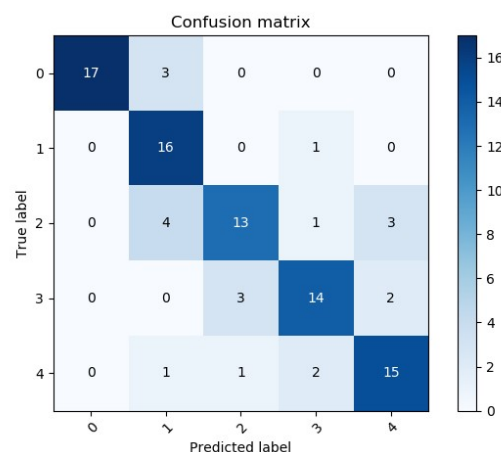


Figure 7. Multi-classification confusion matrix.

## 5. Conclusions

In this paper, we propose a fully convolutional network to solve the real-time segmentic segmentation problem, so as to achieve a trade-off between speed and performance. In particular, we propose InteractionModule, which combines features at different scales to alleviate the lack of spatial details due to network deepening. DilatedGroup is adopted to increase the receptive field in order to capture multi-scale contexts. Through extensive comparison and ablation experiments, we prove the feasibility and generalization of our method. Next, we propose a method for extracting skin tone features, which extracts masked facial tone features and throws them into a Random Forest classifier for classification. 80% classification accuracy demonstrate the effectiveness of the proposed solution.

The purpose of this work is to apply our algorithm to real-time dyeing, face swap, skin tone rating system, and skin care product recommendation based on skin tone levels in realistic scenarios. As a future work, we plan to further explore color features to improve classification accuracy.

**Author Contributions:** Conceptualization, L.L. and D.X.; methodology, L.L. and D.X.; software, L.L.; validation, L.L. and X.L.; formal analysis, L.L.; investigation, L.L. and X.L.; resources, D.X.; data curation, L.L.; writing—original draft preparation, L.L.; writing—review and editing, D.X.; visualization, L.L.; supervision, D.X.; project administration, D.X.; funding acquisition, D.X. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the National Natural Science Foundation of China grant number 61773104.

**Acknowledgments:** This work is done when Ling Luo was an intern at Meidaojia Research, Beijing, P.R. China. Thanks their support for this work.

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

## References

1. Muhammad, U.R.; Svanera, M.; Leonardi, R.; Benini, S. Hair detection, segmentation, and hairstyle classification in the wild. *Image. Vision. Comput.* **2018**, *71*, 25–37.
2. Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. *IEEE Trans. Pattern. Anal. Mach. Intell.* **2014**, *39*, 640–651.
3. 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 2017; pp. 234–241.
4. Yu, F.; Koltun, V. Multi-scale context aggregation by dilated convolutions, 2015. <https://arxiv.org/abs/1511.07122>, arxiv. Accessed April 30, 2016.
5. Zheng, S.; Jayasumana, S.; Romera-Paredes, B.; Vineet, V.; Su, Z.; Du, D.; Huang, C.; Torr, P.H. Conditional random fields as recurrent neural networks.; In Proceedings of the IEEE international conference on computer vision, Santiago, Chile, 11–18 December 2015; pp. 1529–1537.

6. Yu, C.; Wang, J.; Peng, C.; Gao, C.; Yu, G.; Sang, N. Bisenet: Bilateral segmentation network for real-time semantic segmentation.; In Proceedings of the European conference on computer vision, Munich, Germany, 8–14 September 2018; pp. 325–341.
7. Stricker, M.A.; Orengo, M. Similarity of color images.; In Proceedings of the International Society for Optics and Photonics, San Jose, CA, USA, 23 March 1995; pp. 381–392.
8. Pal, M. Random forest classifier for remote sensing classification. *Int. J. Remote. Sens.* **2005**, *26*, 217–222.
9. Chen, L.C.; Papandreou, G.; Schroff, F.; Adam, H. Rethinking atrous convolution for semantic image segmentation, 2017. <https://arxiv.xilesou.top/abs/1706.05587>, arxiv. Accessed December 5, 2017.
10. Zheng, S.; Jayasumana, S.; Romera-Paredes, B.; Vineet, V.; Su, Z.; Du, D.; Huang, C.; Torr, P.H. The one hundred layers tiramisu: Fully convolutional densenets for semantic segmentation.; In Proceedings of the IEEE conference on computer vision and pattern recognition workshops, Hawaii, USA, 21–27 July 2017; pp. 11–19.
11. Zhao, H.; Shi, J.; Qi, X.; Wang, X.; Jia, J. Pyramid scene parsing network.; In Proceedings of the IEEE conference on computer vision and pattern recognition, Hawaii, USA, 21–26 July 2017; pp. 2881–2890.
12. Lin, G.; Milan, A.; Shen, C.; Reid, I. Refinenet: Multi-path refinement networks for high-resolution semantic segmentation.; In Proceedings of the IEEE conference on computer vision and pattern recognition, Hawaii, USA, 21–26 July 2017; pp. 1925–1934.
13. Paszke, A.; Chaurasia, A.; Kim, S.; Culurciello, E. Enet: A deep neural network architecture for real-time semantic segmentation, 2016. <https://arxiv.xilesou.top/abs/1606.02147>, arxiv. Accessed June 7, 2016.
14. Zhao, H.; Qi, X.; Shen, X.; Shi, J.; Jia, J. Icnet for real-time semantic segmentation on high-resolution images.; In Proceedings of the European Conference on Computer Vision, Munich, Germany, 8–14 September 2018; pp. 405–420.
15. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. Mobilenets: Efficient convolutional neural networks for mobile vision applications, 2017. <https://arxiv.xilesou.top/abs/1704.04861>, arxiv. Accessed April 17, 2017.
16. Poudel, R.P.; Liwicki, S.; Cipolla, R. Fast-SCNN: fast semantic segmentation network, 2019. <https://arxiv.xilesou.top/abs/1902.04502>, arxiv. Accessed February 12, 2019.
17. Li, H.; Xiong, P.; Fan, H.; Sun, J. Dfanet: Deep feature aggregation for real-time semantic segmentation.; In Proceedings of the IEEE conference on computer vision and pattern recognition, Los Angeles, USA, 16–20 June 2019; pp. 9522–9531.
18. Wang, Y.; Zhou, Q.; Liu, J.; Xiong, J.; Gao, G.; Wu, X.; Latecki, L.J. LEDNet: A Lightweight Encoder-Decoder Network for Real-Time Semantic Segmentation, 2019. <https://arxiv.xilesou.top/abs/1905.02423>, arxiv. Accessed May 13, 2019.
19. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition.; In Proceedings of the IEEE conference on computer vision and pattern recognition, Las Vegas, USA, June 26–July 1 2016; pp. 770–778.
20. Zhang, H.; Dana, K.; Shi, J.; Zhang, Z.; Wang, X.; Tyagi, A.; Agrawal, A. Context encoding for semantic segmentation.; In Proceedings of the IEEE conference on computer vision and pattern recognition, Salt Lake City, USA, 19–21 June 2018; pp. 7151–7160.
21. Fu, J.; Liu, J.; Tian, H.; Li, Y.; Bao, Y.; Fang, Z.; Lu, H. Dual attention network for scene segmentation.; In Proceedings of the IEEE conference on computer vision and pattern recognition, Los Angeles, USA, 16–20 June 2019; pp. 3146–3154.
22. Chen, L.C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; Yuille, A.L. Semantic image segmentation with deep convolutional nets and fully connected crfs, 2014. <https://arxiv.xilesou.top/abs/1412.7062>, arxiv. Accessed June 7, 2016.
23. Levinshtein, A.; Chang, C.; Phung, E.; Kezele, I.; Guo, W.; Aarabi, P. Real-time deep hair matting on mobile devices.; In Proceedings of the Conference on Computer and Robot Vision, Toronto, Canada, 8–10 May 2018; pp. 1–7.
24. Swain, M.J.; Ballard, D.H. Color indexing. *Int. Comput. Vision.* **1991**, *7*, 11–32.
25. Huang, J.; Kumar, S.R.; Mitra, M.; Zhu, W.J.; Zabih, R. Image indexing using color correlograms.; In Proceedings of the IEEE conference on computer vision and pattern recognition, San Juan, Puerto Rico, 17–19 June 1997; pp. 762–768.
26. Sun, K.; Xiao, B.; Liu, D.; Wang, J. Deep high-resolution representation learning for human pose estimation, 2019. <https://arxiv.xilesou.top/abs/1902.09212>, arxiv. Accessed February 25, 2019.



27. 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, USA, 19–21 June 2018; pp. 4510–4520.
28. Chollet, F. Xception: Deep learning with depthwise separable convolutions.; In Proceedings of the IEEE conference on computer vision and pattern recognition, Hawaii, USA, 21–26 July 2017; pp. 1251–1258.
29. Ioffe, S.; Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift, 2015. <https://arxiv.xilesou.top/abs/1502.03167>, arxiv. Accessed March 2, 2015.
30. Tian, Z.; Shen, C.; Chen, H.; He, T. FCOS: Fully Convolutional One-Stage Object Detection, 2019. <https://arxiv.xilesou.top/abs/1904.01355>, arxiv. Accessed August 20, 2019.
31. Sudre, C.H.; Li, W.; Vercauteren, T.; Ourselin, S.; Cardoso, M.J. Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations.; In Proceedings of the Deep learning in medical image analysis and multimodal learning for clinical decision support, Québec City, QC, Canada, 14 September 2017; pp. 240–248.
32. He, K.; Sun, J.; Tang, X. Guided image filtering.; In Proceedings of the European conference on computer vision, Crete, Greece, 5–11 September 2010; pp. 1–14.
33. He, K.; Sun, J. Fast guided filter, 2015. <https://arxiv.xilesou.top/abs/1505.00996>, arxiv. Accessed May 5, 2015.
34. Levin, A.; Lischinski, D.; Weiss, Y. A closed-form solution to natural image matting. *IEEE Trans. Pattern. Anal. Mach. Intell.* **2007**, *30*, 228–242.
35. Kae, A.; Sohn, K.; Lee, H.; Learned-Miller, E. Augmenting CRFs with Boltzmann machine shape priors for image labeling.; In Proceedings of the IEEE conference on computer vision and pattern recognition, Sydney, Australia, 1–8 December 2013; pp. 2019–2026.
36. Borza, D.; Ileni, T.; Darabant, A. A deep learning approach to hair segmentation and color extraction from facial images.; In Proceedings of the International Conference on Advanced Concepts for Intelligent Vision Systems, Poitiers, France, 24–27 September 2018; pp. 438–449.
37. Yang, S.; Luo, P.; Loy, C.C.; Tang, X. From facial parts responses to face detection: A deep learning approach.; In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 11–18 December 2015; pp. 3676–3684.
38. Svanera, M.; Muhammad, U.R.; Leonardi, R.; Benini, S. Figaro, hair detection and segmentation in the wild.; In Proceedings of the IEEE International Conference on Image Processing, Phoenix, Arizona, USA, 25–28 September 2016; pp. 3676–3684.
39. Zhang, K.; Zhang, Z.; Li, Z.; Qiao, Y. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE. Signal. Process. Lett.* **2016**, *23*, 1499–1503.



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).