

# Real-time deep hair matting on mobile devices

Alex Levenshtein\*, Cheng Chang\*, Edmund Phung\*, Irina Kezele\*, Wenzhangzhi Guo\*, Parham Aarabi\*†

\*ModiFace Inc.

†University of Toronto

**Abstract**—Augmented reality is an emerging technology in many application domains. Among them is the beauty industry, where live virtual try-on of beauty products is of great importance. In this paper, we address the problem of live hair color augmentation. To achieve this goal, hair needs to be segmented quickly and accurately. We show how a modified MobileNet CNN architecture can be used to segment the hair in real-time. Instead of training this network using large amounts of accurate segmentation data, which is difficult to obtain, we use crowd sourced hair segmentation data. While such data is much simpler to obtain, the segmentations there are noisy and coarse. Despite this, we show how our system can produce accurate and fine-detailed hair mattes, while running at over 30 fps on an iPad Pro tablet.

**Keywords**-hair segmentation; matting;augmented reality; deep learning; neural networks

## I. INTRODUCTION

Real-time image segmentation is an important problem in computer vision with a multitude of applications. Among them is the segmentation of hair for live color augmentation in beauty applications (Fig. 1). This use case, however, presents additional challenges. First, unlike many objects with simple shape, hair has a very complex structure. For realistic color augmentation, a coarse hair segmentation mask is insufficient. One needs a hair matte instead. Secondly, many beauty applications run on mobile devices or in web browsers, where powerful computing resources are not available. This makes it more challenging to achieve real-time performance. This paper addresses both these challenges and introduces a system that can accurately segment hair at over 30 fps on a mobile device.

In line with recent success of convolutional neural networks (CNNs) for semantic segmentation, our hair segmentation methods is based on CNNs. We make two main contributions. First, most modern CNNs cannot run in real-time even on powerful GPUs and may occupy a large amount of memory. Our target is real-time performance on a mobile device. In our first contribution we show how to adapt the recently proposed MobileNets [1] architecture for hair segmentation, which is both fast and compact enough to be used on a mobile device.

In the absence of detailed hair segmentation ground truth, we train our network on noisy and coarse crowd-sourced data. A coarse segmentation result, however, is insufficient for hair color augmentation purposes. For realistic color augmentation, an accurate hair matte is needed. In our

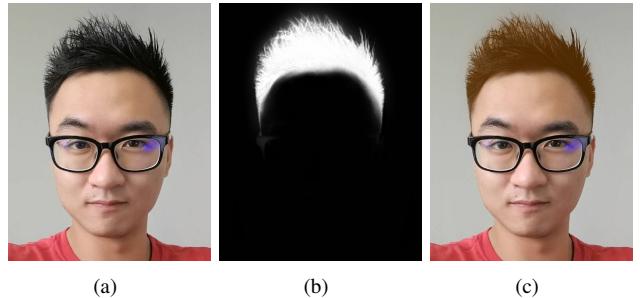


Figure 1. Automatic hair matting and coloring. (a) Input image. (b) Output hair matte produced by our method. (c) Recolored hair.

second contribution, we propose a method for obtaining accurate hair mattes in real-time without the need for accurate hair matte training data. First, we show how to modify our baseline network architecture to have the capacity for capturing fine-level details. Next, by adding a secondary loss function that promotes perceptually appealing matting results, we show that the network can be trained to yield detailed hair mattes using only coarse hair segmentation training data. We compare this approach to simple Guided Filter [2] post-processing and show that it yields more accurate and sharper results.

We evaluate our method, showing that it achieves state-of-the-art accuracy while running in real-time on a mobile device. In the remainder of the paper, we discuss related work in Sec. II, describe our approach in Sec. III, evaluate our method Sec. IV, and conclude in Sec. V.

## II. RELATED WORK

Similar to work on general image segmentation, hair segmentation work can be divided into two categories. The first category of approaches uses hand-crafted features for segmentation. Yacoob et al. [3] employ simple pixel-wise color models to classify hair. Analogous method is employed by Aarabi [4], while also making use of facial feature locations and skin information. Khan et al. [5] use more advanced features with random forests for classification. Such approaches, however, prove to be insufficiently robust for real-world applications.

For more spatially consistent segmentation results, a popular method is to formulate segmentation as random field inference. Lee's et al. [6] build a Markov Random Field

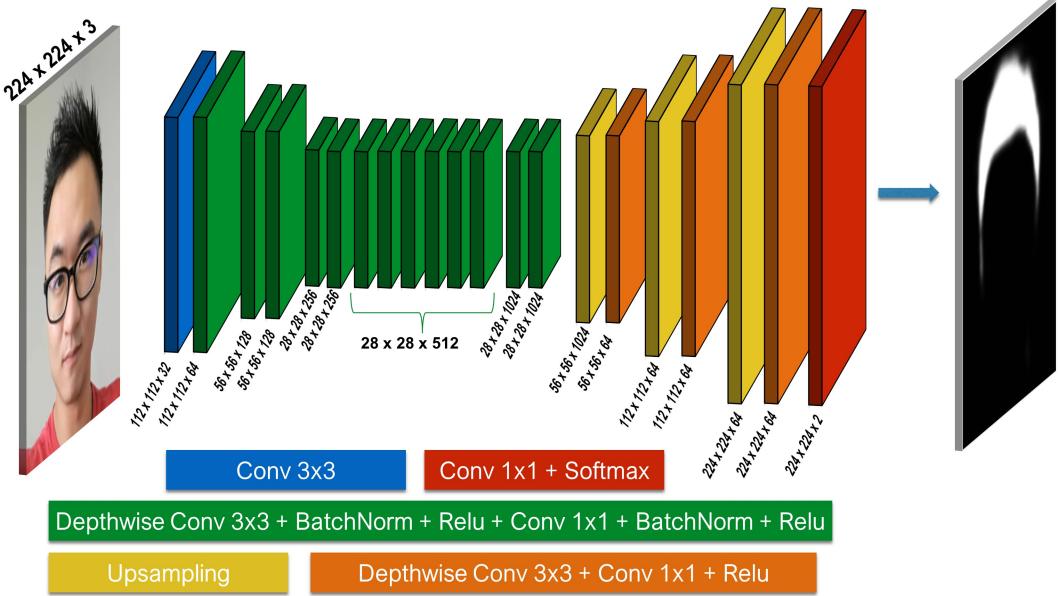


Figure 2. Fully Convolutional MobileNet Architecture for Hair Segmentation

over image pixels, while Huang et al. [7] build their model over superpixels instead. Wang has an alternative method [8], [9] where overlapping image patches are first segmented independently and then combined.

Recently, given the success of deep neural networks (DNNs) in many areas, including semantic segmentation, DNN-based hair segmentation methods have emerged. Guo and Aarabi [10] use a heuristic method to mine high-confidence positive and negative hair patches from each image, and train a separate DNN per image, which is then used to classify the remaining pixels. Inspired by recent success of fully convolutional networks (FCN) for semantic segmentation [11], Chai et al. [12] and Qin et al. [13] employ FCNs for hair segmentation. Due to the coarseness of raw FCN segmentation results, similar to [14], both methods post process the results using dense CRFs [15]. Additionally, [13] have an extra matting step to obtain high-detail hair mattes. Finally, Xu et al. [16] propose an end-to-end CNN architecture for generic image matting yielding state-of-the-art results.

Our approach follows this trend, while addressing several issues. All the aforementioned methods [12], [13], [16] build upon the VGG16 architecture [17]. A single forward pass in VGG16 takes around 100ms even on a powerful GPU, and much longer on a mobile device. Adding dense CRF inference and matting further increases the run-time. Moreover, VGG16 occupies approximately 500MB of memory, which is too much for mobile applications. Instead, we show how to adapt the recently proposed compact MobileNets architecture [1] for segmentation and yield real-time matting results without expensive post processing methods. Finally,

while it may be possible to obtain detailed hair matting data using semi-automatic labeling techniques [16], we show how to train our network without the need for such data.

### III. APPROACH

This section describes our contributions in detail. Firstly, we describe our modifications to the original MobileNet [1] architecture and the challenges of obtaining training data for hair segmentation. Secondly, we illustrate our method for real-time hair matting without the use of matting training data.

#### A. Fully Convolutional MobileNet for Hair Segmentation

Inspired by [12], [13], we first tried to use a modified VGG16 network [17] for hair segmentation, however a forward pass through the network took more than 2 seconds per frame and the network occupied about 500MB of memory. This was incompatible with our real-time mobile use case and therefore we use MobileNets [1] instead, which are faster and more compact.

We modified the original MobileNet architecture into a fully convolutional network for segmentation, which we name HairSegNet. First, we remove the last three layers: Avg Pool, FC, and Softmax (refer to Table 1 in [1]). Next, similar to [12], [14], to preserve fine details we increase the output feature resolution by changing the step size of the last two layers with step size of 2 to 1. Due to our use of pre-trained weights on ImageNet, we dilate all the kernels for the layers with updated resolution by their scale factor w.r.t. their original resolution. Namely, kernels for layers that increased by a factor of 2 are dilated by 2 and kernels for

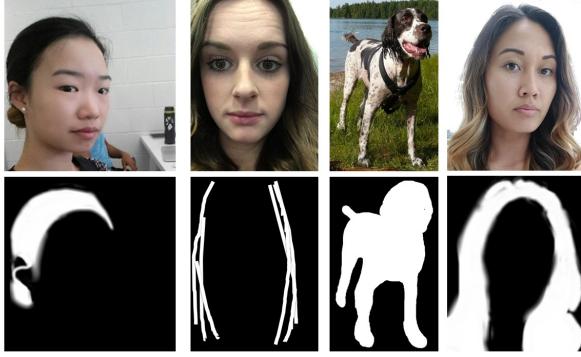


Figure 3. Crowd-sourced training data for hair segmentation. Top: images. Bottom: masks. The data is noisy and coarse, with some images having poor masks (2nd) and some non-face images (3rd).

layers that increased by a factor of 4 are dilated by 4. This yields a final resolution of  $28 \times 28$ .

Next, we build a decoder that takes the above CNN features as input and upsamples them to a hair mask at the original  $224 \times 224$  resolution. We tried upsampling using transposed convolution layers, but saw gridding artifacts in the resulting masks. Therefore, upsampling is performed by a simplified version of an inverted MobileNet architecture. At each stage, we upsample the previous layer by a factor of 2 by replicating each pixel in a  $2 \times 2$  neighborhood. Then, we apply separable depthwise convolution, followed by pointwise  $1 \times 1$  convolutions with 64 filters, followed by ReLu. The number of filters does not have a large effect on accuracy, with 64 filters yielding a slightly better performance based on our experiments (see Sec. IV). The previous block is repeated three times, yielding a  $224 \times 224 \times 64$  output. We conclude by adding a  $1 \times 1$  convolution with softmax activation and 2 output channels for hair / non-hair. The network is trained by minimizing the binary cross entropy loss  $L_M$  between predicted and ground truth masks. The full architecture is illustrated in Fig. 2.

The resulting architecture is considerably more compact than VGG16, occupying only 15MB. More importantly, a forward pass takes 300ms when implemented in Tensorflow on iPad Pro. Using the recently released optimized CoreML library [18] from Apple, this time can be further reduced to 60ms per frame.

Training deep neural networks requires a large amount of data. While there are large datasets for general semantic segmentation, these datasets are much less popular for hair segmentation. Moreover, unlike some objects like cars, which have a relatively simple shape, hair shape is very complex. Therefore, obtaining precise ground truth segmentation for hair is even more challenging.

To cope with this challenge we use a pre-trained network on ImageNet and fine-tune the entire network on hair segmentation data. Nevertheless, several thousands of training

images are still needed. We crowd-source such data using a hair coloring app where users have to manually mark their hair. While getting this data is cheap, the resulting hair segmentation labels are very noisy and coarse. Fig. 3 illustrates this issue. Note that in the 2nd image hair was labeled very sparsely, while in the 3rd image a photograph of a pet was submitted. We manually clean this data by only keeping the images of human faces with sufficiently good hair masks. This is considerably faster than marking the hair from scratch or fixing incorrect segmentations.

### B. Hair matting

In our second contribution we show how to obtain accurate hair matting results. We solve the matting problem using a CNN, which we name HairMatteNet, in an end-to-end manner. Such an approach faces two challenges. First, we need an architecture with the capacity to learn high resolution matting details. The network in Sec. III-A may not be suitable since the results are still generated by incremental upsampling of relatively low-res layer ( $28 \times 28$ ). Secondly, our CNN needs to learn hair matting using only coarse segmentation training data.

We address the first issue by adding skip connections between layers in the encoder and corresponding layers in the decoder, similar to many modern network architectures [11]. This way, shallower layers in the encoder, which contain high-res but weak features are combined with low-res but powerful features from deeper layers. The layers are combined by first applying a  $1 \times 1$  convolution to the incoming encoder layers to make the output depth compatible with the incoming decoder layers (64 for the three outer skip connections and 1024 for the inner skip connection) and then merging the layers using addition. For each resolution, the deepest encoder layer at that resolution is taken for skip connection.

The second issue is addressed by adding a loss function that promotes perceptually accurate matting output. Motivated by the alpha matting evaluation work of Rhemann et al. [19], our secondary loss measures the consistency between image and mask edges. It is minimized when the two agree. Specifically, we define our mask-image gradient consistency loss to be:

$$L_C = \frac{\sum M_{mag} [1 - (I_x M_x + I_y M_y)^2]}{\sum M_{mag}}, \quad (1)$$

where  $(I_x, I_y)$  and  $(M_x, M_y)$  are the normalized image and mask gradients respectively, and  $M_{mag}$  is the mask gradient magnitude. This loss is added to the original binary cross entropy loss with a weight  $w$ , making the overall loss  $L = L_M + wL_C$ . The combination of the two losses maintains the balance between being true to training masks while generating masks that adhere to image edges. Fig. 4 illustrates our new architecture and the combination of the two loss functions.

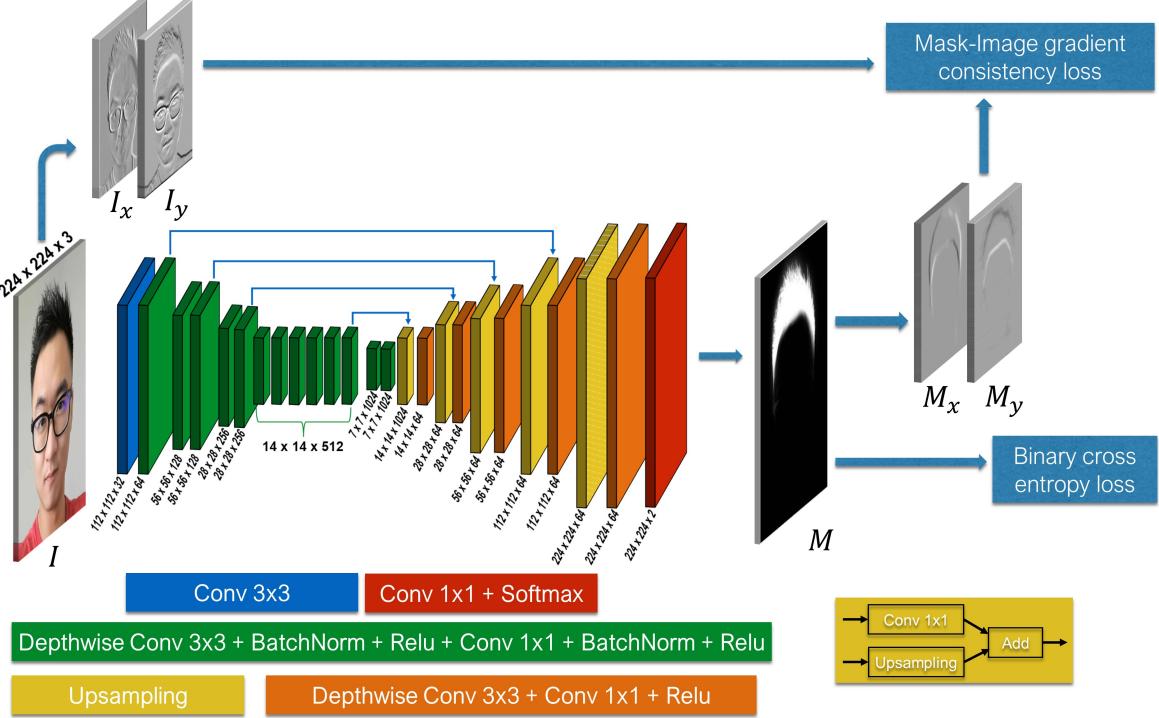


Figure 4. Fully Convolutional MobileNet Architecture for Hair Matting. Skip connections are added to increase the network capacity for capturing high resolution detail. Mask-image gradient consistency loss is added alongside with standard binary cross entropy loss to promote detailed matting results.

We compare HairMatteNet to simple post-processing of our coarse segmentation mask (HairSegNet) with a Guided Filter [2]. Qin et al. [13] used a similar approach but employed a more advanced matting method that is not fast enough for real-time applications on mobile devices. Guided Filter is an edge-preserving filter and has a linear runtime complexity w.r.t. the image size. It takes only 5ms to process a  $224 \times 224$  image on iPad Pro. Fig. 5 compares the masks with (c) and without (b) the filter. The former is clearly capturing more details, with individual hair strands becoming apparent. However, the filter adds detail only locally near the edges of the mask from CNN. Moreover, the edges of the refined masks have a visible halo around them, which becomes even more apparent when the hair color has lower contrast with its surroundings. This halo causes color bleeding during hair recoloring. HairMatteNet yields sharper edges (Fig. 5d) and captures longer hair strands, without the unwanted halo effect seen in Guided Filter post-processing.

As an additional bonus, HairMatteNet runs twice as fast compared to HairSegNet, taking only 30ms per frame on a mobile device and without the need for an extra post-processing matting step. Due to our use of skip connections, that help with capturing high resolution detail, HairMatteNet maintains the original MobileNet encoder structure with the deepest layers having  $7 \times 7$  resolution. These layers have many depth channels (1024) and become very expensive to

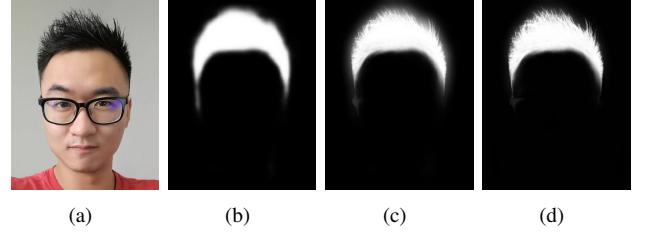


Figure 5. Hair segmentation and matting. (a) Input image. (b) HairSegNet. (c) HairSegNet + Guided Filter. (d) HairMatteNet.

process with increased resolution. Having a  $7 \times 7$  resolution makes processing much faster compared to the  $28 \times 28$  resolution in HairSegNet.

#### IV. EXPERIMENTS

We evaluate our method on three datasets. First is our crowd-sourced dataset, consisting of 9000 training, 380 validation, and 282 testing images. All three subsets include the original images and their flipped versions. Since our target is hair matting on mobile devices, we pre-process our data by detecting the face and cropping a region around it based on the scale expected for typical selfies.

To compare our method to existing approaches, we evaluate two public datasets: LFW Parts dataset [20] and the hair dataset of Guo and Aarabi [10]. The former consists of

2927  $250 \times 250$  images, with 1500 training, 500 validation, and 927 test images. Pixels are labeled into three categories: hair, skin, and background, generated at the superpixel level. The latter consists of 115 high-resolution images. Since it contains too few images to train on, we use our crowd-sourced training data when evaluating on this set. To make this dataset consistent with our training data, we pre-process it in a similar manner (using face detection and cropping), adding flipped images as well. Since in a few cases faces were not detected, the resulting dataset consists of 212 images.

Training is done using a batch size of 4 using the Adadelta [21] method in Keras [22], with learning rate 1.0,  $\rho = 0.95$ , and  $\epsilon = 1e - 7$ . We use L2 regularization with the weight  $2 \cdot 10^{-5}$  for convolution layers only. Depthwise convolution layers and the last convolution layer are not regularized. We set the loss balancing weight to  $w = 0.5$ . In the three-class LFW data, only the hair class is contributing to the mask-image gradient consistency loss. We train our model for 50 epochs and select the best performing epoch using validation data. Training on crowd-sourced dataset takes 5 hours on Nvidia GeForce GTX 1080 Ti GPU and less than an hour on LFW Parts due to much smaller training set size.

#### A. Quantitative evaluation

For quantitative performance analysis, we measure the F1-score, Performance [10], IoU, and Accuracy, averaged across all test images. To measure the consistency of image and hair mask edges, we also report the mask-image gradient consistency loss (Eqn. 1). Recall that during the manual clean-up in Sec. III-A we only filtered images rather than correcting the masks. As a result, the quality of our hair annotation is still poor. Therefore, prior to evaluation on our crowd-sourced data, we manually corrected the test masks, spending no more than 2 minutes per annotation. This yielded slightly better ground truth. Three variants of our method are evaluated on this relabeled data. Table I shows our results. All three methods perform similarly w.r.t. the ground truth comparison measures, however, HairMatteNet is the clear winner in the gradient consistency loss category, indicating that its masks adhere much better to image edges.

On the LFW Parts dataset, we report an on-par performance with the best performing method in Qin et al. [13], but achieve it in real-time on a mobile device. We use only the accuracy measure for evaluation since it is the only measure used in [13]. Arguably, especially since LFW Parts was annotated at the superpixel level, the ground truth there may not good enough for high-accuracy analysis. On the dataset of Guo and Aarabi [10] we report an F1-score of 0.9376 and Performance of 0.8253. We re-ran HNN, the best performing method in [10], on this post-processed dataset and obtained similar performance to that reported by the authors, with F1-score of 0.7673 and Performance of 0.4674.

Model	F1	Perf.	IoU	Acc.	Grad-cons.
Crowd-sourced dataset					
HairSegNet	0.9212	0.7833	0.8564	0.9624	0.2696
HairSegNet + GF	0.9251	0.7964	0.8637	0.9646	0.1803
HairMatteNet	0.9219	0.7945	0.8589	0.9619	0.0533
LFW Parts dataset [20]					
HairMatteNet	NA	NA	NA	0.965	NA
[13] FCN+CRF	NA	NA	NA	0.9676	NA
HairMatteNet SP	NA	NA	NA	0.9769	NA
[13] FCN+CRF SP	NA	NA	NA	0.9732	NA
Guo and Aarabi dataset [10]					
HairMatteNet	0.9376	0.8253	0.8848	0.964	0.0532
HNN [10]	0.7673	0.4674	0.6454	0.8793	0.2732

Table I  
QUANTITATIVE EVALUATION

Depth	F1	Perf.	IoU	Acc.
16	0.9202	0.7878	0.8559	0.9585
32	0.9217	0.8001	0.8584	0.9581
64	0.9229	0.7939	0.8608	0.9604
128	0.9225	0.7937	0.8605	0.9596

Table II  
DECODER LAYER DEPTH EXPERIMENTS ON VALIDATION DATA

#### B. Qualitative evaluation

We evaluate our method on publicly available selfie images for qualitative analysis. Results can be seen in Fig. 6. HairSegNet (Fig. 6b) yields good but coarse masks. HairSegNet with Guided Filter (Fig. 6c) produces better masks but with an undesirable blur around hair boundaries. The most accurate and sharpest results are achieved by HairMatteNet (Fig. 6d). A failure mode of both Guided Filter post-processing and HairMatteNet is their under-segmentation of hair-like objects in the vicinity of hair, such as eyebrows in case of dark hair or bright background for light hair. In addition, highlights inside the hair can cause the hair mask from HairMatteNet to be non-homogeneous, which is especially apparent in the last three examples in column (h).

#### C. Network architecture experiments

**Decoder layer depth** Using our validation data, we have experimented with number of decoder layer channels, but observed that it does not have a large effect on accuracy. Table II illustrates our experiments with the number of channels in the decoder, with 64 channels yielding the best results according to most measures. These experiments were done using the skip connections architecture in Fig. 4 without using the gradient consistency loss.

**Input image size** Howard et al. [1] observed that MobileNets perform better given higher image resolution. Given our goal of accurate hair matting, we experimented with increasing the resolution beyond  $224 \times 224$ , which is the highest resolution MobileNet were trained on ImageNet. For the 2nd and 3rd image in Fig. 6, Fig. 7 shows qualitative comparison of masks inferred using HairMatteNet from

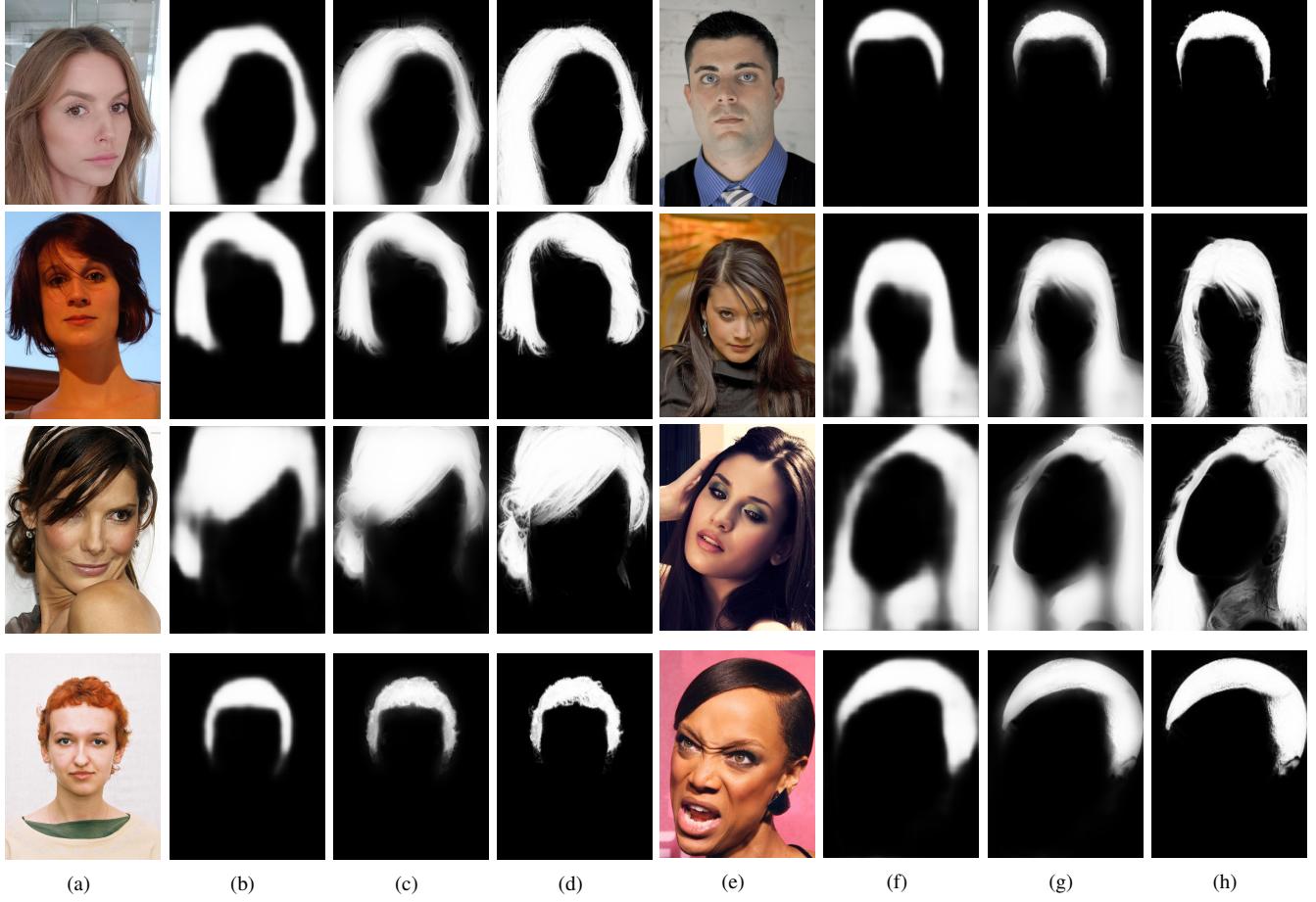


Figure 6. Qualitative evaluation. (a,e) Input image. (b,f) HairSegNet. (c,g) HairSegNet + Guided Filter. (d,h) HairMatteNet.

$224 \times 224$  images vs.  $480 \times 480$  images. The  $480 \times 480$  results look more accurate around the hair edges, with longer hair strands being captured (e.g., the long hair strand falling on the nose in the first image). However, the issues mentioned in the previous section are emphasized as well, with more of the hair mask bleeding into non-hair regions and the inside of the mask becoming non-homogeneous due to hair highlights. In addition, processing a larger image is significantly more expensive.

## V. SUMMARY

This paper presented a hair matting method with real-time performance on mobile devices. We have shown how, given noisy and coarse data, a modified MobileNet architecture is trained to yield accurate matting results. While we apply the proposed architecture for hair matting, it is general and can be applied to other segmentation tasks. In future work we will explore fully automatic methods for training from noisy data without the need for manual filtering. In addition, we will explore further improvements to matting quality, such as capturing longer hair strands and segmenting light hair; all this while keeping the hair mask homogeneous, preventing

it from bleeding into non-hair regions, and maintaining our real-time performance on mobile devices.

## REFERENCES

- [1] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, “Mobilenets: Efficient convolutional neural networks for mobile vision applications,” *arXiv preprint arXiv:1704.04861*, 2017.
- [2] K. He, J. Sun, and X. Tang, “Guided image filtering,” *TPAMI*, vol. 35, no. 6, pp. 1397–1409, 2013.
- [3] Y. Yacoob and L. S. Davis, “Detection and analysis of hair,” *TPAMI*, vol. 28, no. 7, pp. 1164–1169, 2006.
- [4] P. Aarabi, “Automatic segmentation of hair in images,” in *ISM*, 2015, pp. 69–72.
- [5] K. Khan, M. Mauro, and R. Leonardi, “Multi-class semantic segmentation of faces,” in *ICIP*, 2015, pp. 827–831.
- [6] K.-c. Lee, D. Anguelov, B. Sumengen, and S. B. Gokturk, “Markov random field models for hair and face segmentation,” in *FG*, 2008, pp. 1–6.



Figure 7. Network resolution comparison (a)  $224 \times 224$  (b)  $480 \times 480$

- [7] G. B. Huang, M. Narayana, and E. Learned-Miller, “Towards unconstrained face recognition,” in *CVPRW*. IEEE, 2008, pp. 1–8.
- [8] N. Wang, H. Ai, and S. Lao, “A compositional exemplar-based model for hair segmentation,” in *ACCV*, 2010, pp. 171–184.
- [9] N. Wang, H. Ai, and F. Tang, “What are good parts for hair shape modeling?” in *CVPR*, 2012, pp. 662–669.
- [10] W. Guo and P. Aarabi, “Hair segmentation using heuristically-trained neural networks,” *TNNLS*, 2016.
- [11] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in *CVPR*, 2015, pp. 3431–3440.
- [12] M. Chai, T. Shao, H. Wu, Y. Weng, and K. Zhou, “Autohair: Fully automatic hair modeling from a single image,” *TOG*, vol. 35, no. 4, p. 116, 2016.
- [13] S. Qin, S. Kim, and R. Manduchi, “Automatic skin and hair masking using fully convolutional networks,” in *ICME*, 2017.

- [14] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, “Semantic image segmentation with deep convolutional nets and fully connected crfs,” in *ICLR*, 2015.
- [15] P. Krähenbühl and V. Koltun, “Efficient inference in fully connected crfs with gaussian edge potentials,” in *NIPS*, 2011, pp. 109–117.
- [16] N. Xu, B. Price, S. Cohen, and T. Huang, “Deep image matting,” in *CVPR*, 2017.
- [17] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv preprint arXiv:1409.1556*, 2014.
- [18] “Coreml,” <https://developer.apple.com/documentation/coreml>.
- [19] C. Rhemann, C. Rother, J. Wang, M. Gelautz, P. Kohli, and P. Rott, “A perceptually motivated online benchmark for image matting,” in *CVPR*, 2009, pp. 1826–1833.
- [20] A. Kae, K. Sohn, H. Lee, and E. Learned-Miller, “Augmenting CRFs with Boltzmann machine shape priors for image labeling,” in *CVPR*, 2013.
- [21] M. D. Zeiler, “Adadelta: an adaptive learning rate method,” *arXiv preprint arXiv:1212.5701*, 2012.
- [22] F. Chollet *et al.*, “Keras,” <https://github.com/fchollet/keras>, 2015.