INVERSE HALFTONE COLORIZATION: MAKING HALFTONE PRINTS COLOR PHOTOS {SUPPLEMENTARY MATERIALS}

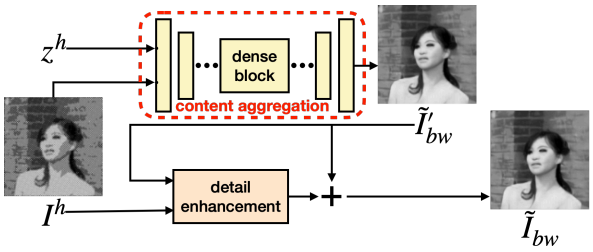
Yu-Ting Yen, Chia-Chi Cheng, Wei-Chen Chiu

Department of Computer Science, National Chiao Tung University

1. NETWORK ARCHITECTURE AND TRAINING DETAILS

We implement our model based on the Pytorch framework with NVIDIA Geforce GTX 2080 Ti GPU. All networks are trained with Adam optimizer. We now detail our network architecture and learning settings of each network as follows.

1.1. Inverse Halftone Network



The inverse halftone network IHN is composed of content aggregation and detail enhancement modules, in which its architecture is shown above. The content aggregation is a 7-layer U-net with a dense block in the bottleneck. The U-net architecture is same with the generator in [1], and the architecture of dense block is same with the one in DenseNet [2]. As for the detail enhancement module, we directly use the one proposed in PRL [3]. In our training scheme, IHN is trained for 130 epochs with learning rate 2×10^{-4}, where the training data can be easily produced by making continous-tone gray-scale images into halftone ones.

1.2. Encoder, Colorization Network, and Discriminator

The architectures of encoder E and colorization network G are the same with those proposed (i.e. encoder and generator respectively) in [1]. For the discriminator D, its architecture is almost identical to the one in [1], with simply modifying its first convolution layer to support the input size which is the concatenation of a colorization result and its corresponding color latent code. These three networks are trained for 300 epochs with learning rate 10^{-4} in our experiment.

layer	channel	kernel	stride	activation
conv1	32	3×3	2	ReLU
conv2	64	3×3	2	ReLU
conv3	64	3×3	2	ReLU
fc1	256	-	-	Softmax

Table 1: Architecture of Histogram Estimator H.

name	layer	channel	kernel	stride	activation
Input	conv	64	7×7	1	ReLU
ResBlock1	conv1	64	3×3	1	ReLU
	conv2	64	3×3	1	-
activation1	-	-	-	-	ReLU
ResBlock2	conv1	64	3×3	1	ReLU
	conv2	64	3×3	1	-
activation2	-	-	-	-	ReLU
ResBlock3	conv1	64	3×3	1	ReLU
	conv2	64	3×3	1	-
activation3	-	-	-	-	ReLU
Output	conv	3	3×3	1	-

Table 2: Architecture of Fusion Network F.

1.3. Histogram Estimator H

As mentioned in the manuscript for the histogram loss $\mathcal{L}_{hist}$, we design the histogram estimator $H(I)$ to estimate the color distribution of each color channel for a given image I. Table 1 shows its detailed architecture. The model is composed of 3 convolution layers and 1 fully-connected layer. We use the training data of Helen [4] as the training dataset to pretrain the histogram estimator for 200 epochs with the learning rate set to 10^{-4}.

1.4. Fusion Network F

Fusion network F is composed of multiple residual blocks. Table 2 shows its detailed architecture. Input and output layers are convolution layers. There are three residual blocks composed of convolution layers in the middle and each one is followed by a ReLU activation function. We train it for 300 epochs and the learning rate is 10^{-4}.

2. MODEL COMPLEXITY ANALYSIS

We analyze the complexity of each baseline by computing the number of parameters. We only consider the colorization network since all baseline should pass through the inverse halftone network. As shown in Table 3, the model sizes of our architecture and other baselines are of the same order of magnitude. It should be noted that [5] and [6] are smaller because the input image size of [5] is quarter than others, and [6] uses shallower U-Net to only generate fixed four kinds of colorization instead of arbitrary colorization with random noise.

Method	[1]	[5]	[7]	[6]	Ours
Model Size	54.793	5.728	46.086	8.671	30.496

Table 3: Model sizes of each baseline. The unit is million.

3. MORE EXAMPLES FOR ABLATION STUDY

We provide the qualitative examples in Figure 1 for the ablation study. The experimental settings are all the same as those used in the main manuscript.

We observe that: the Base model almost degenerates to have no diverse colorization, where $\tilde{I}_{rec}, \tilde{I}_{ref}, \tilde{I}_{rand}$ seem to exactly the same; the Base model equipped with the reference scheme is able to produce diverse results; adding mode seeking loss $\mathcal{L}_{ms}$ and similarity loss $\mathcal{L}_{sim}$ respectively increase the diversity as well as strengthen the patch relationship within the image (e.g. the difference of face and background for smoking man in Figure 1); while including histogram loss $\mathcal{L}_{hist}$ better corrects the overall color tone. Finally, our full model with the fusion network makes the color tone of resultant images even closer to the groundtruth, removes some artifacts (e.g. the shade of the smoking man and the face contour of the smiling man), and enhances the details (e.g. the tooth of the smiling man). As shown in the Table.1 of the main manuscript, all the evaluation metrics have significant progress along with having our model designs except the diversity-score, as there is a trade-off between the realness and diversity of the generated images. For instance, mode seeking loss enlarges the diversity extremely, but it also causes the images color uneven or weird in $\tilde{I}_{rand}$. Thus, it is reasonable that the diversity-score decreases little, but FID score that evaluates overall image quality improves.

4. BILINEAR INTERPOLATION ON LATENT SPACE

We use the following process to verify that our colorization network G indeed learns the color information from z^c. We first randomly sample four z^c from $\mathcal{N}(0, I)$, and subsequently apply bilinear interpolation between them. The resultant latent codes are then used as input for the colorization network G, producing the final colorization images. The results are shown in Figure 2 (with the halftone input I_h shown in the first column). Images placed in the four corners are based on the four randomly sampled z^c, while the others are based on the interpolated latent code. We can observe that our G gradually produces images with smooth changes in color appearance along with z^c.

5. MORE QUALITATIVE RESULTS

5.1. Comparisons with exemplar-based colorization

Figure 3 shows some example results of $IHN+$ [8], which cascades our inverse halftone network with the exemplar-based colorization method from He *et al.* [8]. Basically, given a bi-level halftone image I_h, our proposed method converts I_h to $\tilde{I}$, and $IHN+$ [8] colorizes the $\tilde{I}_{bw} = IHN(I_h)$ to $\tilde{I}$. In the results produced by $IHN+$ [8], there are some unusual color patches, e.g. green color in woman's cloth of the first row and blue color in woman's neck of the third row. He *et al.* [8] has a model design which searches for similar patches across the input image and color reference in order to achieve better colorization. However, such design is also a limitation for their work: when the color reference image is not similar enough to the grayscale input image, some color appearance from the reference which are unnatural for the input image will pollute the resultant colorization. Thus, finding an appropriate reference image is a crucial issue and brings additional cost, for the exemplar-based colorization method. Our proposed method instead encodes the whole color reference image to a color latent code, therefore our results will have less strange patches.

5.2. Comparisons with automatic diverse colorization

Here we provide more examples of our qualitative results as well as the comparison with respect to other baselines built upon the automatic diverse colorization methods (i.e. Zhu *et al.* [1], Cao *et al.* [5], Deshpande *et al.* [7], and Lei *et al.* [6]) in Figure 4. The experimental settings are all the same as those in the main paper.

6. MORE EXAMPLE RESULTS

We provide more results of our proposed method: Figure 5 shows the example results produced by our method under exemplar-based colorization setting (i.e. reference scheme). Each input I_h has three different reference images I_{ref} and our model generates the corresponding outputs $\tilde{I}_{ref}$. Figure 6 shows the example results produced by our method under automatic/random colorization setting. Our model colorizes the input halftone prints I_h automatically. Each image has five results $\tilde{I}_{rand}$ based on different random noise.

7. REFERENCES

[1] Jun-Yan Zhu, Richard Zhang, Deepak Pathak, Trevor Darrell, Alexei A Efros, Oliver Wang, and Eli Shechtman, "Toward multimodal image-to-image translation," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[2] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger, "Densely connected convolutional networks," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[3] Menghan Xia and Tien-Tsin Wong, "Deep inverse halftoning via progressively residual learning," in *Asian Conference on Computer Vision (ACCV)*, 2018.

[4] Vuong Le, Jonathan Brandt, Zhe Lin, Lubomir Bourdev, and Thomas S Huang, "Interactive facial feature localization," in *European Conference on Computer Vision (ECCV)*, 2012.

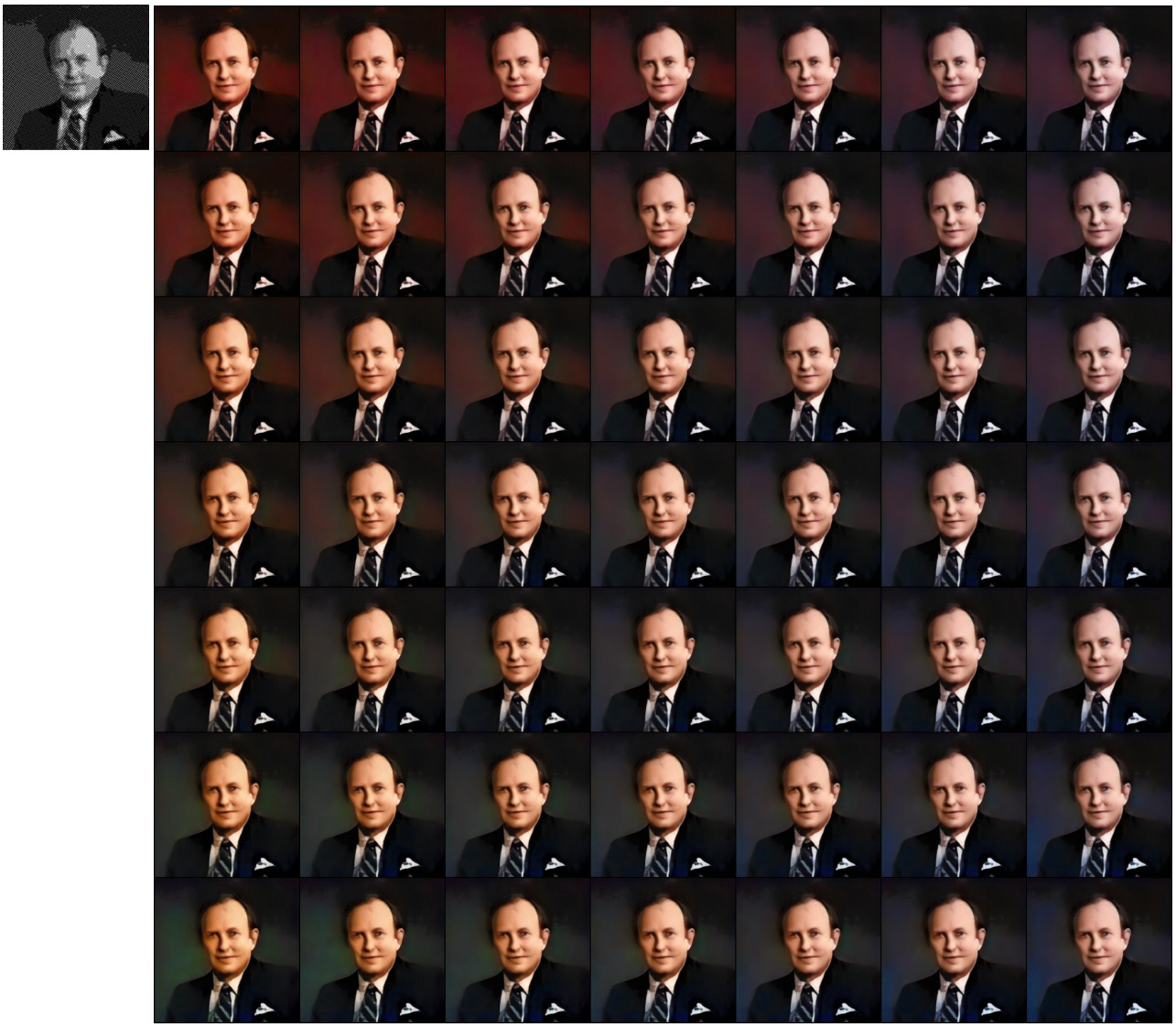
[5] Yun Cao, Zhiming Zhou, Weinan Zhang, and Yong Yu, "Unsupervised diverse colorization via generative adversarial networks," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 2017.

[6] Chenyang Lei and Qifeng Chen, "Fully automatic video colorization with self-regularization and diversity," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[7] Aditya Deshpande, Jiajun Lu, Mao-Chuang Yeh, Min Jin Chong, and David Forsyth, "Learning diverse image colorization," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[8] Mingming He, Dongdong Chen, Jing Liao, Pedro V Sander, and Lu Yuan, "Deep exemplar-based colorization," *ACM Transactions on Graphics (TOG)*, 2018.

Fig. 1: Example results for ablation study, where we sequentially include different model designs into the Base model. The first column is the input halftone images I_h and the second column is the corresponding groundtruth images I together with one color references I_{ref}. The other columns (from left to right) are results produced by: base model without reference scheme (Base w/o ref), base model with reference scheme (Base w/ ref), adding mode seeking loss $\mathcal{L}_{ms}$, adding similarity loss $\mathcal{L}_{sim}$, adding histogram loss $\mathcal{L}_{hist}$, and applying fusion network (i.e. our full model). For each group of examples, the first row is the images produced by reconstruction (i.e. using I as color reference), the second row is the images produced by color reference I_{ref}, while the other rows are based on random colorization.



I_h Bilinear Interpolation Results

Fig. 2: Examples of performing bilinear interpolation on the color latent code. The leftmost column is input image I_h. The rest columns are the results produced by random colorization, with bilinearly interpolated color latent codes. Please check the description in Section 4 of this supplementary material for more details.

I_h I_{ref} He *et al.* Ours I_h I_{ref} He *et al.* Ours

Fig. 3: Qualitative examples on exemplar-based colorization. The first column is input halftone image I_h. The second column is the reference image I_{ref}. Results in He *et al.* [8] (the third column) are produced from $\tilde{I}_{bw} = IHN(I_h)$ to $\tilde{I}$, while our proposed method (the rightmost) are from I_h to $\tilde{I}$.

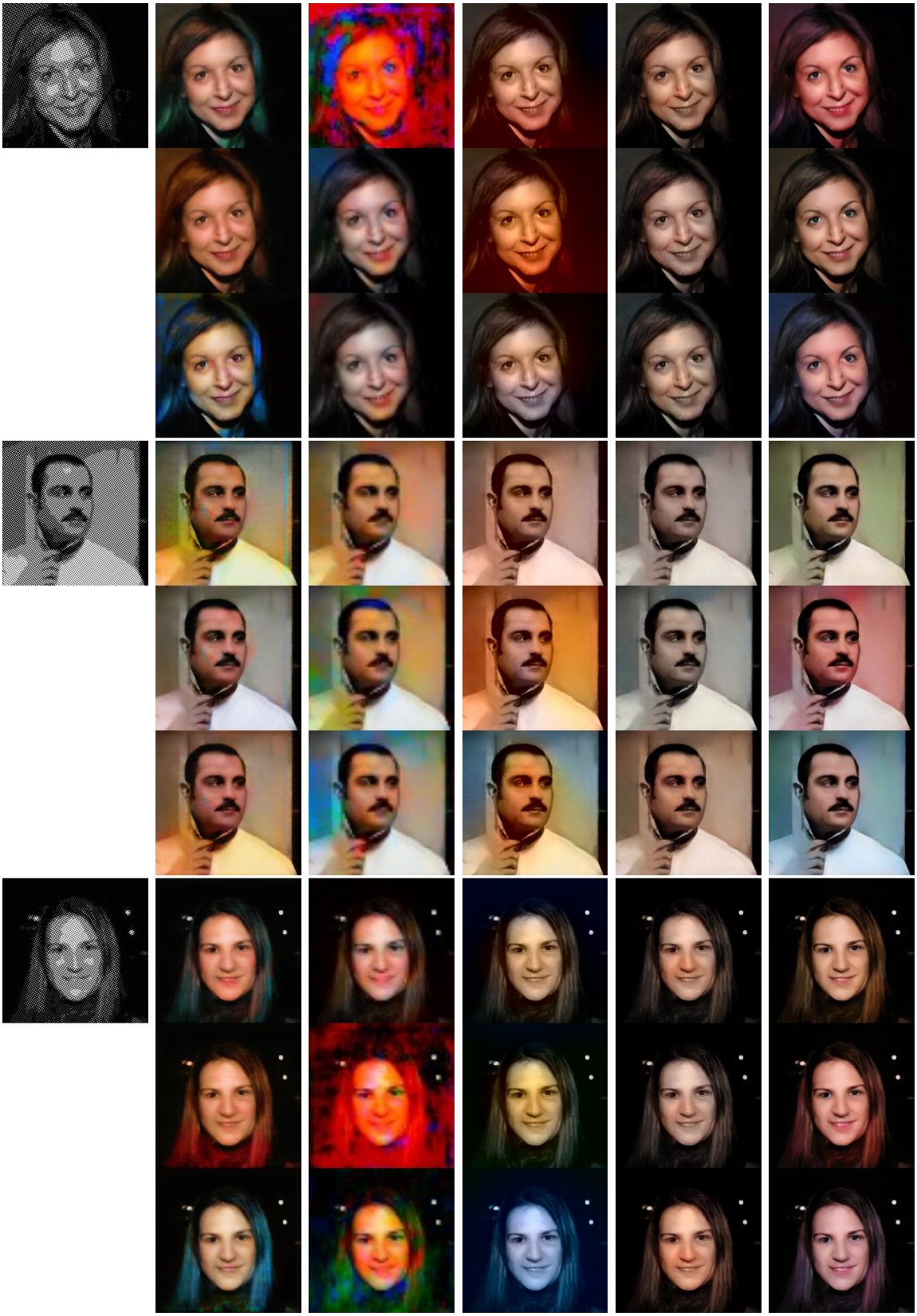


Fig. 4: Qualitative examples on diverse colorization. The first column are input halftone images I_h. Starting from the second column, we apply different colorization methods on $\tilde{I}_{bw}$ produced by $IHN(I_h)$, including Zhu *et al.* [1], Cao *et al.* [5], Deshpande *et al.* [7], and Lei *et al.* [6]. Shown in the rightmost column, the results produced by our full model not only have sufficient diversity but also look more realistic.

Fig. 5: More example results for our proposed method under exemplar-based colorization setting (i.e. reference scheme). Given input halftone image I_h (the leftmost column) and different reference images I_{ref} (the third, fifth and seventh column), our proposed method generates corresponding colorful results $\tilde{I}_{ref}$ (the fourth, sixth and eighth column respectively). The second column is the original color image I of I_h.

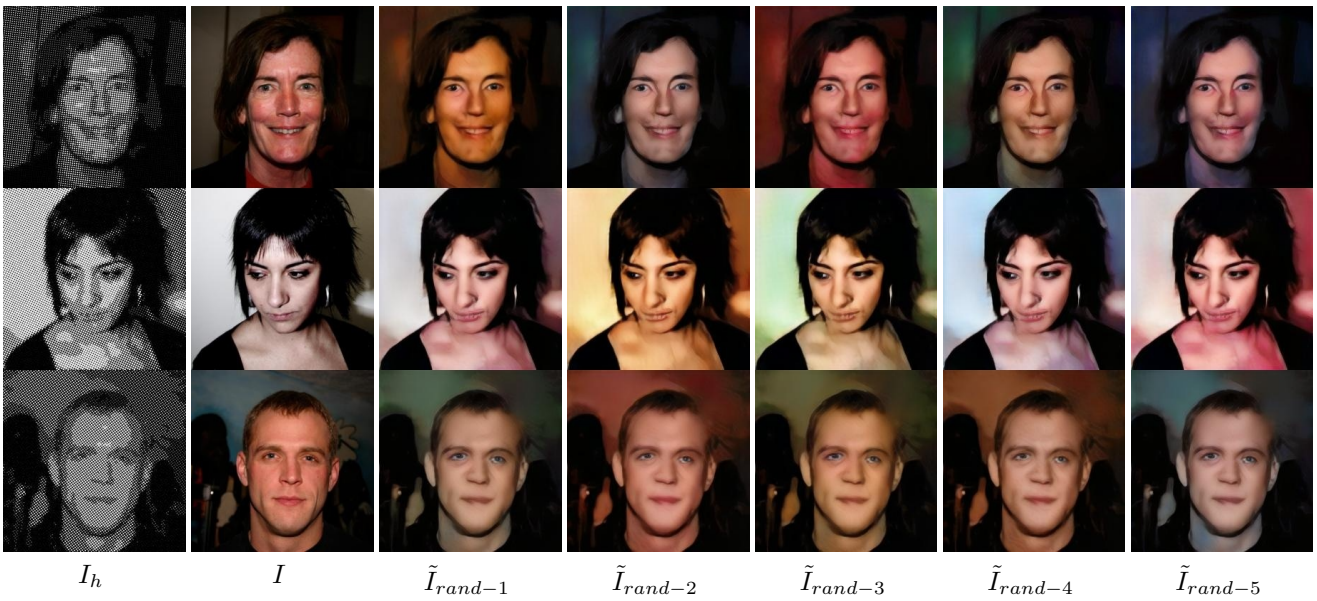


Fig. 6: More example results for our proposed method under the random/automatic colorization setting. The leftmost column is the input images I_h, and the second one is the original color image I of I_h. The rest columns are results $\tilde{I}_{rand}$ of automatic colorization based on different random noise.