

Learning to Blend Photos

Supplementary Materials

1 Implementation Details of Quality Network

We train the proposed quality networks in PyTorch framework with a single GPU. We use stochastic gradient descent (SGD) as optimizer with a fixed learning rate 0.01 and batch size of 16. The quality network converges with around 100 epochs. For the ranking losses, we set the margin term $m = 1$, and $\lambda = 1$ to combine \mathcal{L}_r and \mathcal{L}_{bce} .

2 Additional Details for Reinforcement Learning

As stated in Section 5.2 of the main paper, we extend the quality network to the policy network. In addition to the RL training, we introduce another L1 loss term to maintain the quality score output to be consistent with the pretrained quality network.

Table 1 shows the detailed operations of each action. In all our experiments, we set the hyperparameter α in Table 1 as 0.05 and β as 0.1. We adopt both dueling-DQN [9] and A2C [8] as our deep reinforcement learning algorithm. In Table 2 and 3, we present the detailed hyperparameters for training the reinforcement agent. It is worth noting that we set the discount factor of both algorithms as 0.5 since we found that the higher discount factor will lead to worse convergence of the value term. We implement the system with PyTorch framework on a single TitanX GPU.

3 Details of Global Optimization Baselines

We compare the proposed deep reinforcement search with several global optimization baselines, including Particle Swarm Optimization (PSO) [4] and Simulated Annealing (SA) [5]. We show the details of applying these two algorithms in this document.

3.1 Particle Swarm Optimization

In PSO, we initialize several particles with random ROI and photometric adjustment as the initial seeds. We describe each particle with their state variables (x,y,w,b,c), which correspond to the ROI position, ROI width, foreground brightness, foreground contrast, respectively. Note that we do not use the ROI height as a state variable since we maintain the aspect ratio of ROI to be the same as the foreground image. We then perform the following updates on each state variable:

$$v_{s,i} = \omega v_{s,i} + r_p \phi_p (s_i^p - s_i) + r_g \phi_g (s_i^g - s_i), \quad (1)$$

$$s_i = s_i + v_{s,i}, \quad (2)$$

Table 1. The detailed Actions Description. We denote the ROI before applying actions as (x, y, w, h) , and the pixel value as p with mean m .

Action	Description
Right	new ROI: $(x + \alpha w, y, w, h)$
Left	new ROI: $(x - \alpha w, y, w, h)$
Down	new ROI: $(x, y + \alpha h, w, h)$
Up	new ROI: $(x, y - \alpha h, w, h)$
Smaller	new ROI: $(x + \alpha w, y + \alpha h, w - 2\alpha w, h - 2\alpha h)$
Bigger	new ROI: $(x - \alpha w, y - \alpha h, w + 2\alpha w, h + 2\alpha h)$
Brightness+	new pixel value: $p \times (1 + \beta)$
Brightness-	new pixel value: $p \times (1 - \beta)$
Contrast+	new pixel value: $p + \beta \times (p - m)$
Contrast-	new pixel value: $p - \beta \times (p - m)$

Table 2. The detailed hyperparameters for Dueling-DQN.

Parameter	Value
Max. Episodes	200k
Max. Steps per Episode	50
Discount Factor (γ)	0.5
Memory Size	5000
Batch Size	12
Learning Rate	2.5e-4
Train Interval	Every 4 episodes
Target Model Update Frequency	500 episodes
Gradient Clip	± 40.0
ϵ -greedy Start ϵ	1.0
ϵ -greedy End ϵ	0.1
ϵ -greedy Start Episode	1k
ϵ -greedy End Episode	100k

where $i = 0, 1, 2, 3, 4$ is the variable index, $v_{s,i}$ is the velocity for updating state values, s^p is the best known state of the particle (with highest quality score), and s^g is the best known state of all particles. r_p and r_g are two random variables sampled from a uniform distribution $U(0, 1)$, while ω, ϕ_p , and ϕ_g are the update coefficients.

We optimize PSO with the evaluation cost at 100 steps for various combinations of the hyperparameters. As a result, we obtain the best result with 20 particles, 5 updates, $\omega = 0.5$, $\phi_p = 1.0$, and $\phi_g = 1.0$.

Table 3. The detailed hyperparameters for A2C.

Parameter	Value
Max. Episodes	200k
Max. Steps per Episode	50
Discount Factor (γ)	0.5
Batch Size	4
Learning Rate	7e-4
Action Repetition	1
Gradient Clip	± 40.0
Entropy Term Coefficient	0.01

3.2 Simulated Annealing

In SA, we initialize the seed with random ROI and photometric adjustment from Random-50. During the update process, we sample a neighbor state s_n of current state s , and decide whether to update to the new state based on the following acceptance probability

$$p_A = \exp((Q(s) - Q(s_n))/T), \quad (3)$$

where $Q(s)$ is the quality score of state s . Note that $p_A > 1$ when the algorithm samples a better state with a higher quality score, and thus it will move to the sampled new state with probability 1. We optimize SA with the evaluation cost at 50 steps (50 steps for Random-50). As a result, SA performs the best with T starts from 10 and linearly decreases to 1, while we sample the neighbor state from $(x \pm 50, y \pm 50, w \pm 50, b \pm 0.2, c \pm 0.2)$.

4 Agent Policy Visualization

In Figure 3-6, we show the intermediate search steps performed by the proposed DRL search. In most cases, the agent starts from a bigger ROI and quickly localizes the rough regions, and then searches locally to finetune the spatial alignment. During the process, the agent also adjusts the brightness and contrast until the facial components can be seen clearly.

5 Multimodal Blending

5.1 Diverse Blending Engine

To show that the proposed system can generate pleasing results with diverse appearance, we adopt four modules for constructing an arbitrary double exposure based blending engines, where each module can be applied independently: **BG**

Removal applies CNN-based background removal on the foreground image and color the removed regions as white. We implement the segmentation model based on Deeplab [1]. Refer to our supplementary materials for more details. **Sky Coloring** detects the sky color in background photo with scene parsing model [7] and apply the detected color on the removed regions of foreground image after BG removal. **FG Filter** applies Instagram-like color filters to the foreground image. **B&W Filter** transforms the result blending image to gray-scale. During the training process, we randomly generate a blending engine with these modules to make the system robust to different blending engines.

5.2 Diverse User Preference

To introduce diversity and user preference modeling to the quality network, we apply the multiple choice learning [3], which is recently used by [2] to model the image synthesis diversities. Similar to [2], we modify the quality network to a multiple choice quality network which outputs K scores instead of just one by increasing the number of the final classifiers. We assume that within the predefined K modes there exists a mode which could best describe the preference of the current rating, and the quality network should only be optimized concerning the chosen mode. Then the challenge becomes how to pick the right mode for the current rating. Then we modify the ranking loss in (2) in the paper as follows:

$$\mathcal{L}_r^K(S_i, S_j) = \max\{0, -y(S_i^k - S_j^k) + m\}, \quad (4)$$

where $k = \arg \max_k \{y(S_i^k - S_j^k)\}$ is the mode index that agrees with the user rating most, and $S_i = \{S_i^1, S_i^2 \dots S_i^K\}$ represents the scores of K modes generated by the multiple-choice quality network with input (F_i, B_i) . During training, we also apply the binary cross entropy loss in the multiple choice form as

$$\mathcal{L}_{bce}(S_i) = -r \log(\sigma(S_i^k)) - (1 - r) \log(1 - \sigma(S_i^k)). \quad (5)$$

In Figure 7, we show additional blending results with our method using different pretrained preference mode.

6 Quality Score Visualization

We present two visualization maps of quality score manifolds in Figure 8. For each map, we sample a pair of foreground and background photo, sliding a fixed scale ROI with different spatial positions. We generate the quality scores with the proposed quality network and visualize them with a heat map. We observe that the score manifold has many local optimums, and therefore many heuristic search methods do not perform well in our task.

7 Qualitative Comparisons and Additional Blending Results

Figure 9-11 show all 20 sets of blending results used in the user study. In addition, we present blending results using different blending engine configurations in Figure 12-14. We note that the module used for removing background is a DeepLab segmentation network [1] finetuned on the portrait dataset [6].

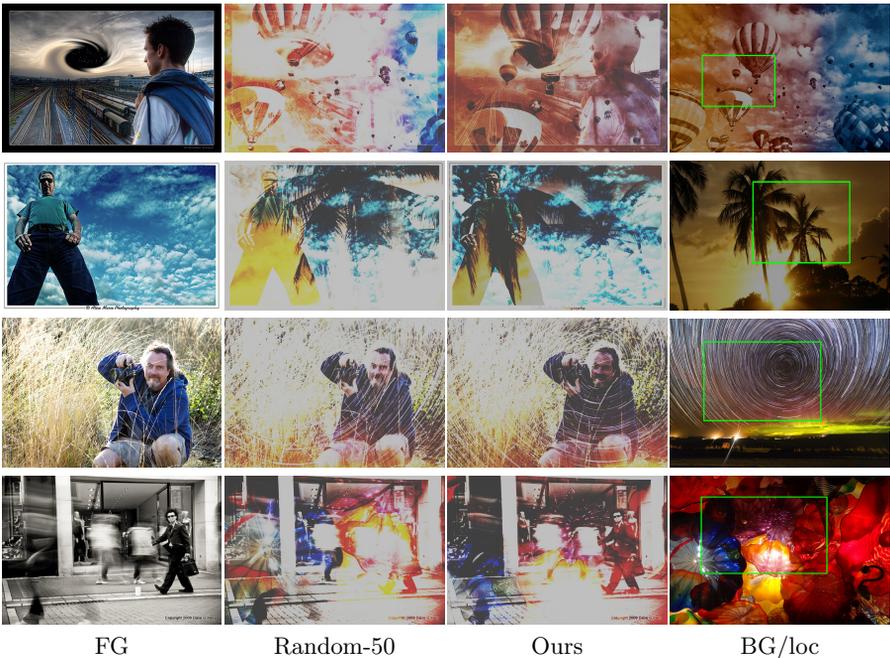


Fig. 1. Sample blending results using photos from the Imagenet where the person are not centered and with cluttered background. The green bounding box indicates the final localization of our method.

8 Generalization to Other Blending Forms

In Figure 1, we show some results of applying the proposed method without foreground segmentation on some images sampled from the Imagenet with cluttered background and non-centered person. The blended photos are interesting with different styles from the ones shown in the paper, while the aforementioned general rules learned by our model still apply.

As shown in Figure 2, our model can also generate pleasing results with a different blending mode (multiply) on animal photos without retraining. Although we propose to tackle the double exposure effect in this paper, the proposed method can serve as a generic framework for automatic photo art generation.



Fig. 2. Blend results of different blending forms. We show results using “multiply” blending mode instead of “screen”. We also use animal photos instead of human to demonstrate the generalization ability of the proposed method.

Therefore, given a specific form of photo blending, the whole framework can be retrained using proper user annotations.

9 User Interface of Annotating System and User Study System

Figure 15 shows the full web interface for annotating the blending results as mentioned in Section 4.4 of the main paper. In addition, we show the user study interfaces as well as the instructions in Figure 16 and 17.

10 Editing Process of Expert

The expert uses the provided foreground and background image and applies a double exposure action script in PhotoShop. During the process, the expert is only allowed to perform the background alignment as well as photometric adjustment in order to have fair comparisons with our method.

References

1. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. In: TPAMI (2017)
2. Chen, Q., Koltun, V.: Photographic image synthesis with cascaded refinement networks. In: ICCV (2017)
3. Guzman-Rivera, A., Batra, D., Kohli, P.: Multiple choice learning: Learning to produce multiple structured outputs. In: NIPS (2012)
4. Kennedy, J.: Particle swarm optimization. In: Encyclopedia of machine learning, pp. 760–766. Springer (2011)
5. Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P.: Optimization by simulated annealing. *science* **220**(4598), 671–680 (1983)

6. Shen, X., Hertzmann, A., Jia, J., Paris, S., Price, B., Shechtman, E., Sachs, I.: Automatic portrait segmentation for image stylization. In: Computer Graphics Forum (2016)
7. Tsai, Y.H., Shen, X., Lin, Z., Sunkavalli, K., Yang, M.H.: Sky is not the limit: Semantic-aware sky replacement. In: SIGGRAPH (2016)
8. Wang, J.X., Kurth-Nelson, Z., Tirumala, D., Soyer, H., Leibo, J.Z., Munos, R., Blundell, C., Kumaran, D., Botvinick, M.: Learning to reinforcement learn. arXiv preprint arXiv:1611.05763 (2016)
9. Wang, Z., Schaul, T., Hessel, M., Van Hasselt, H., Lanctot, M., De Freitas, N.: Dueling network architectures for deep reinforcement learning. In: arXiv preprint arXiv:1511.06581 (2015)

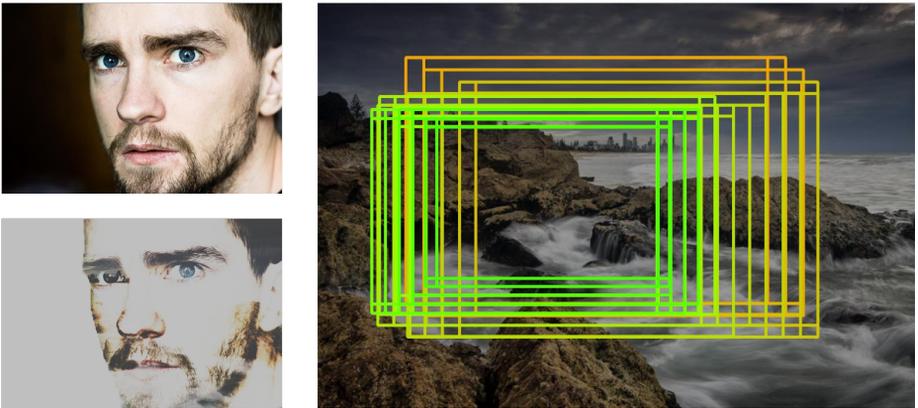


Fig. 3. Sample DRL search intermediate actions. We show the ROI actions with color encoding. The initial ROI is orange while the last ROI is green. The blending result is selected with the highest quality score throughout the actions. During the process, the brightness is increased to 1.2, and the contrast is increased to 1.4.

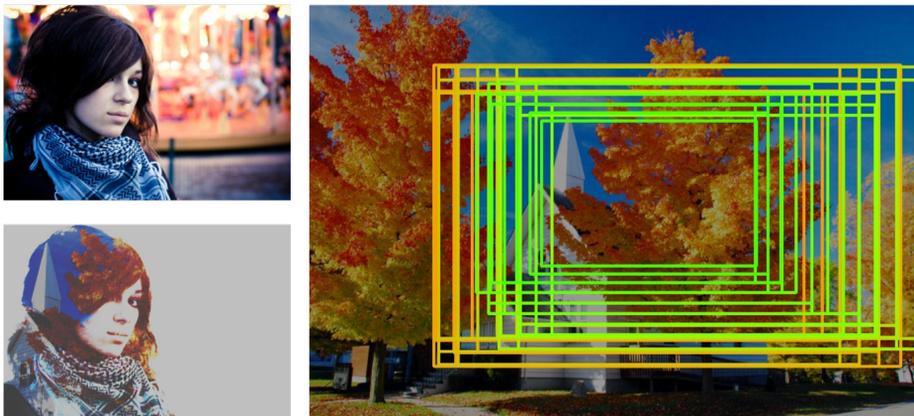


Fig. 4. Sample DRL search intermediate actions. We show the ROI actions with color encoding. The initial ROI is orange while the last ROI is green. The blending result is selected with the highest quality score throughout the actions. During the process, the brightness is increased to 1.2, and the contrast is decreased to 0.8.

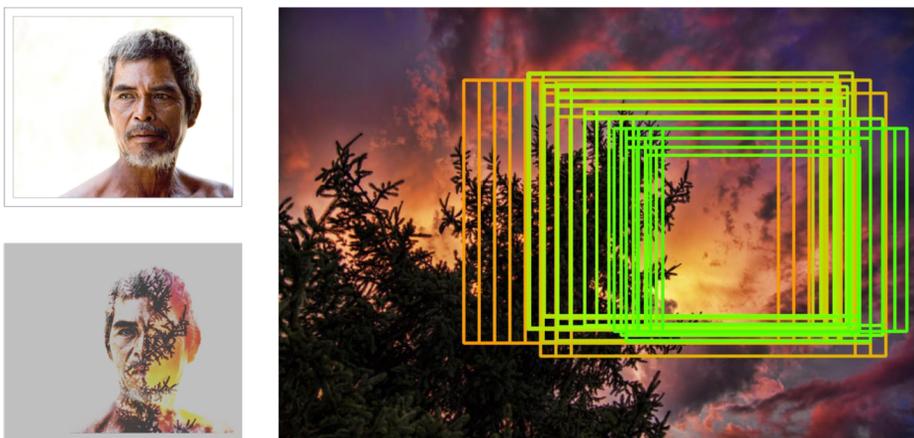


Fig. 5. Sample DRL search intermediate actions. We show the ROI actions with color encoding. The initial ROI is orange while the last ROI is green. The blending result is selected with the highest quality score throughout the actions. During the process, the brightness is unchanged, and the contrast is decreased to 0.8.

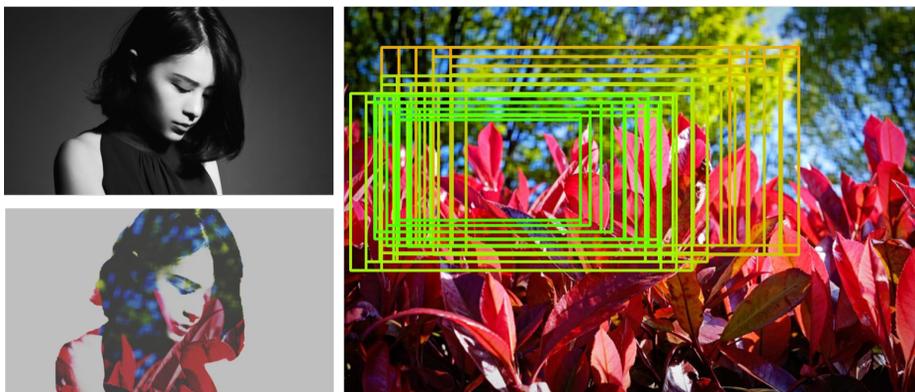


Fig. 6. Sample DRL search intermediate actions. We show the ROI actions with color encoding. The initial ROI is orange while the last ROI is green. The blending result is selected with the highest quality score throughout the actions. During the process, the brightness is decreased to 0.6, and the contrast is unchanged.

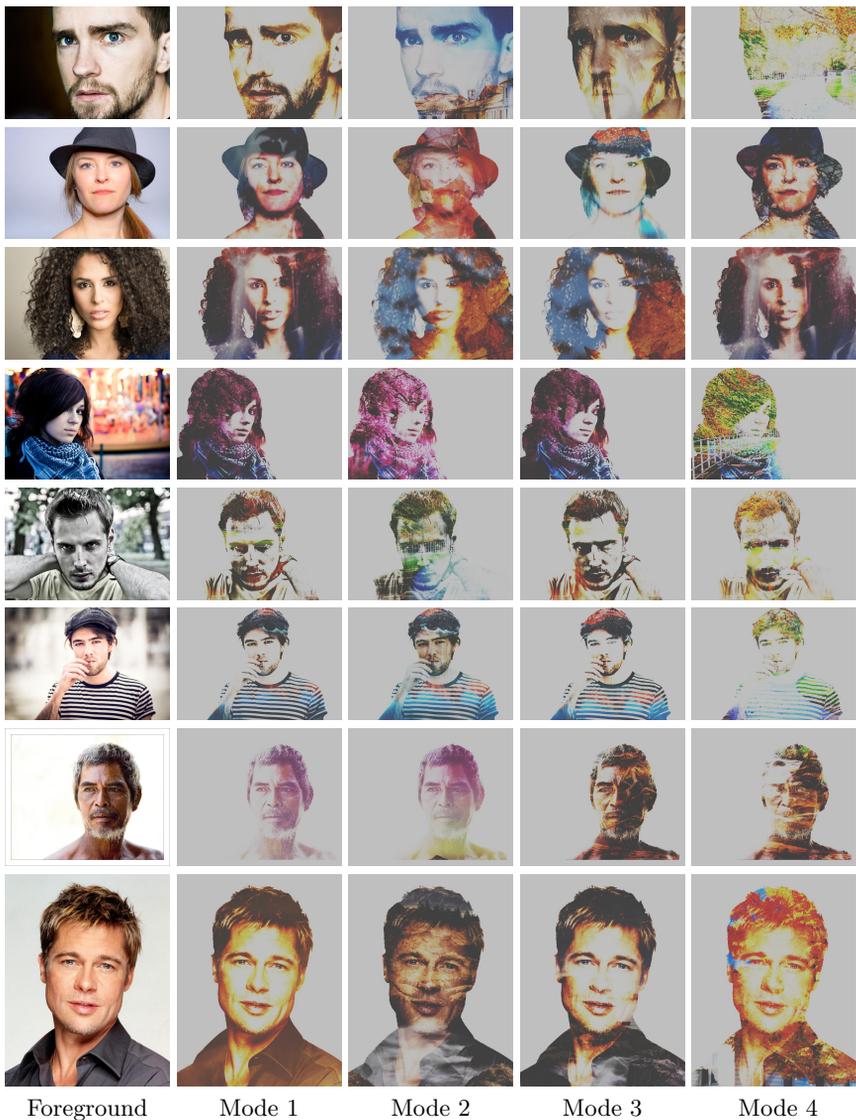


Fig. 7. Qualitative comparisons of different preference modes. For each foreground image, we pick three background photos, and for each mode we show the result with highest score among three background photos.

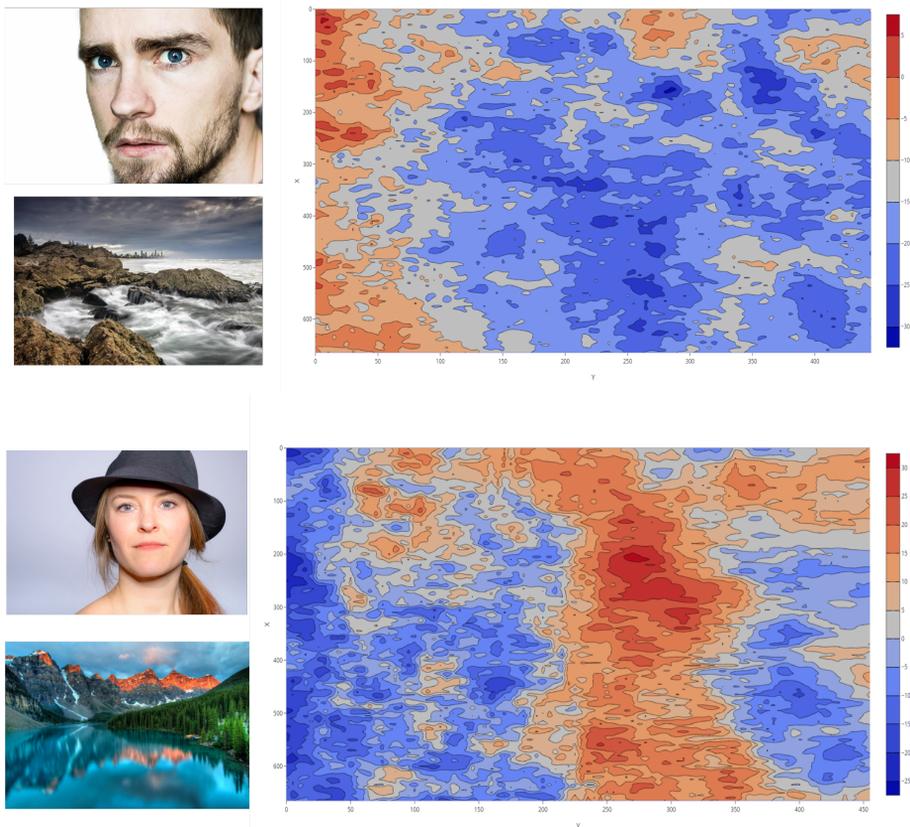


Fig. 8. Sample quality score manifolds. We show two quality score manifolds with fixed scale ROI and no photometric adjustment. We observe that the score manifold has many local optimums, and therefore many heuristic search methods do not perform well in our task.

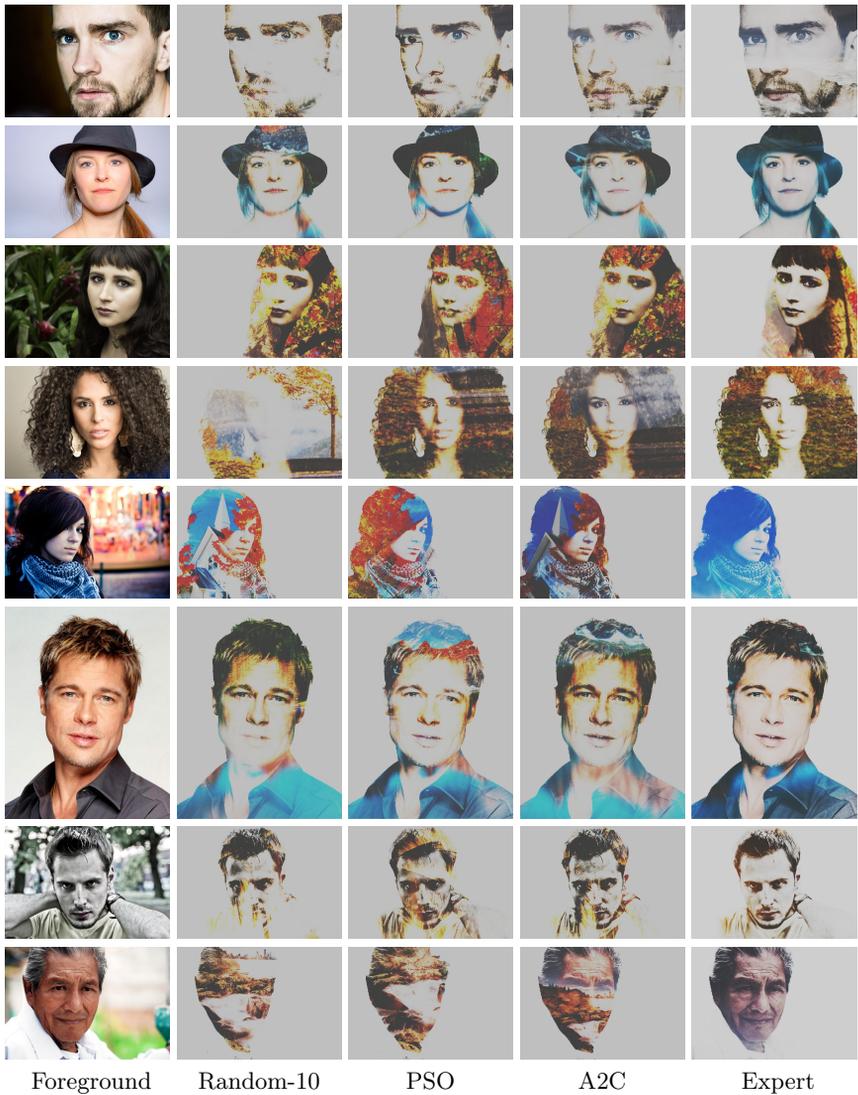


Fig. 9. Comparisons of different baseline methods.



Fig. 10. Comparisons of different baseline methods.

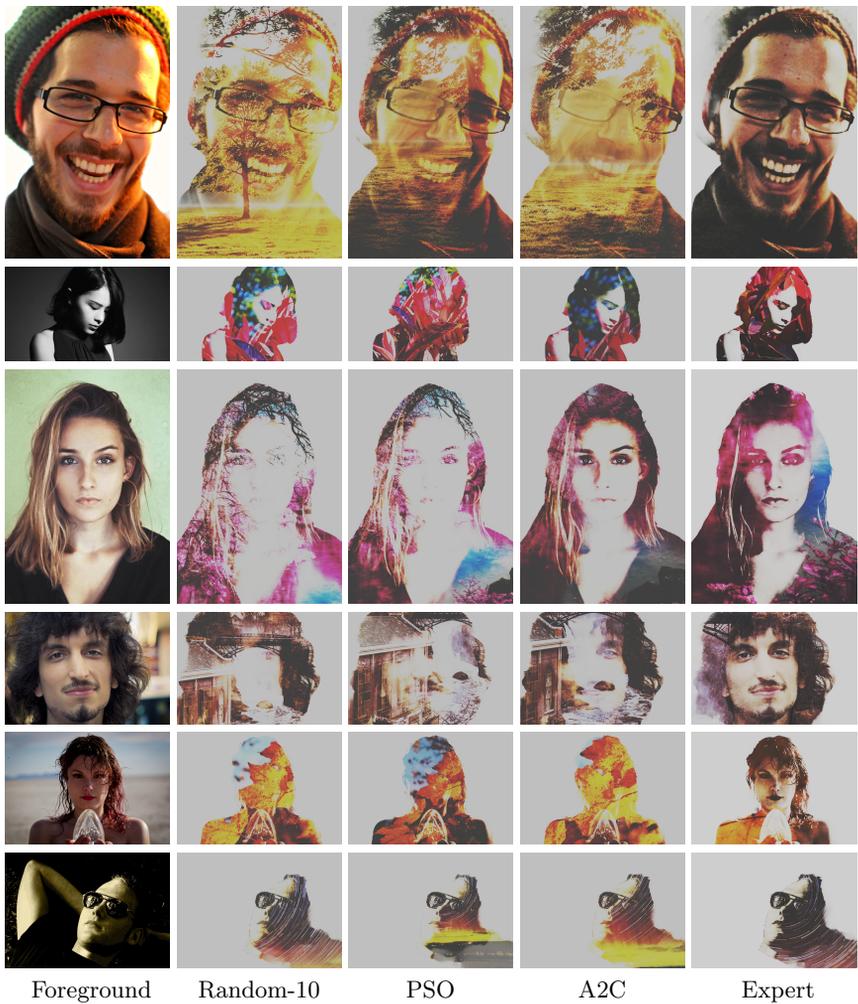


Fig. 11. Comparisons of different baseline methods.

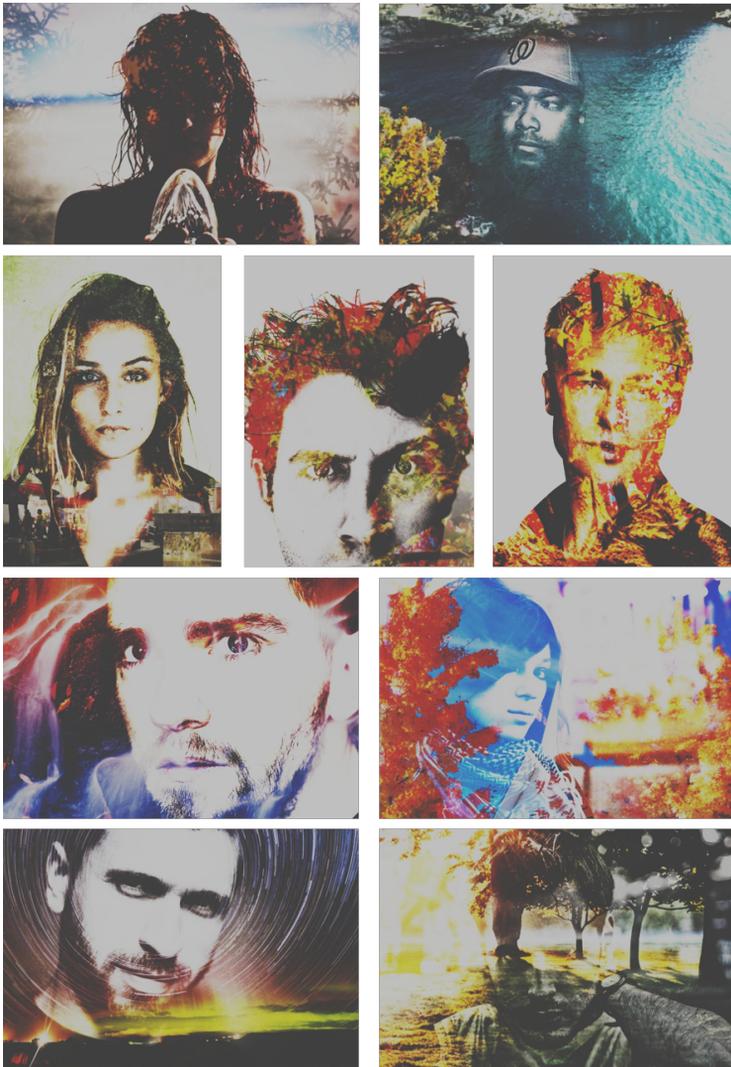


Fig. 12. Selected blending results with no pre- or post-processing.



Fig. 13. Selected blending results with **BG Removal** and **B&W Filter**.

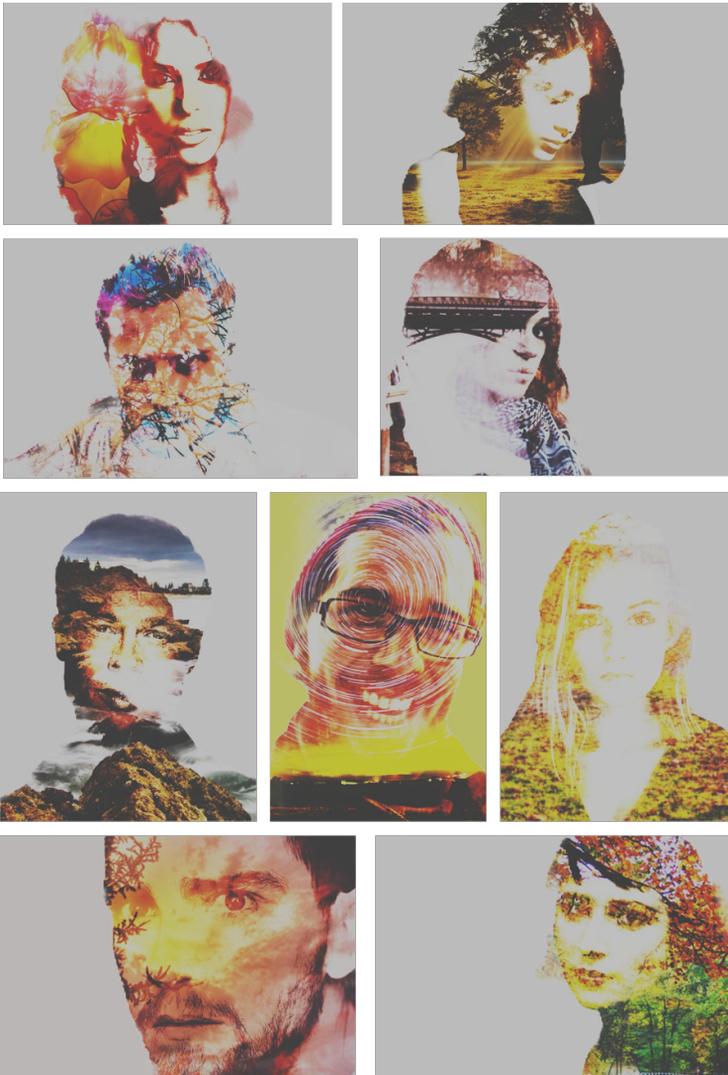


Fig. 14. Selected blending results with **BG Removal**, **Sky Coloring**, and **FG Filter**, where we apply a “Toast” filter on the foreground image.

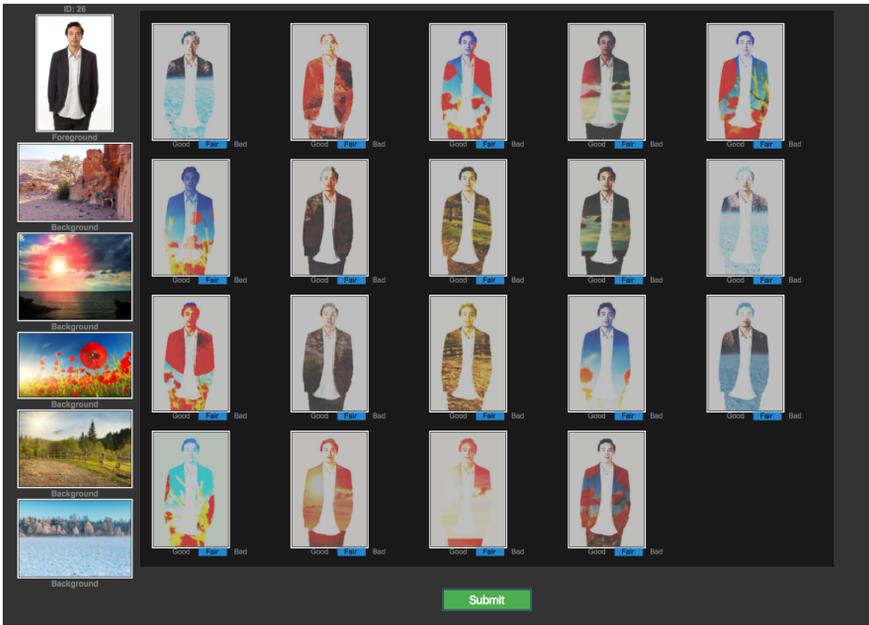


Fig. 15. User labeling system. Users are asked to give a rating between Good, Fair, or Bad to each blending photo. The blending photos are generated with random parameters.

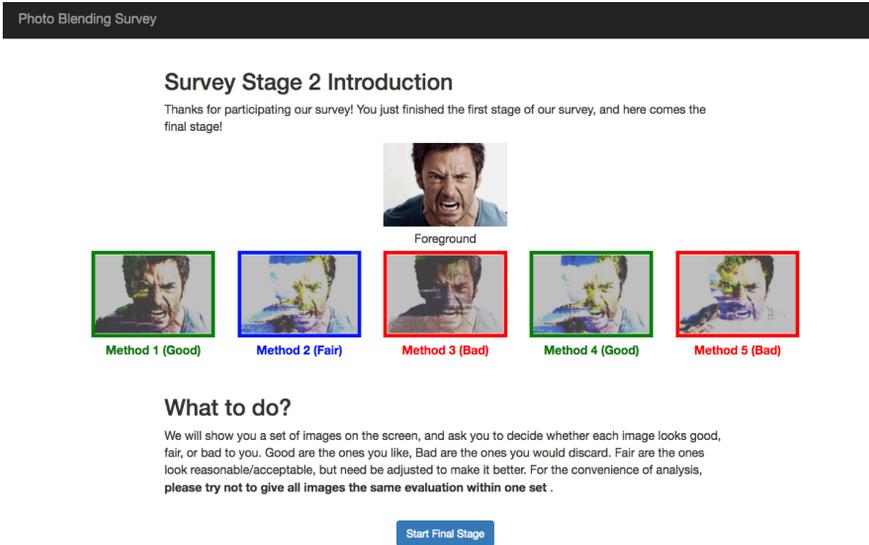


Fig. 16. Introduction for the user study.

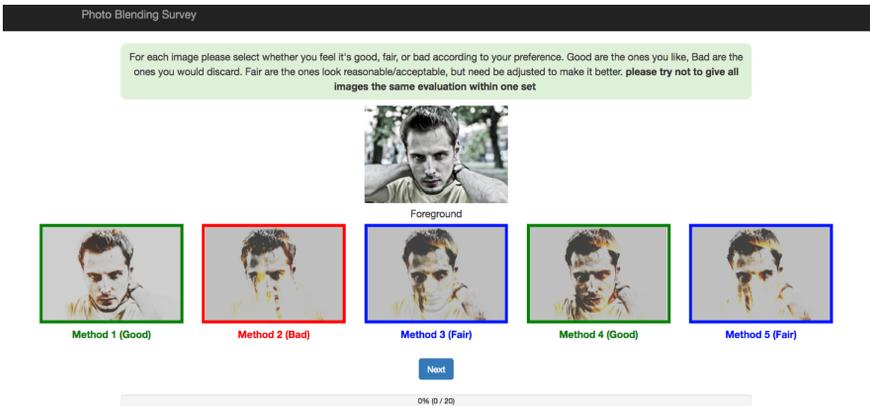


Fig. 17. User interface for the user study. User can click on each image to switch the ratings between “Good”, “Fair”, and “Bad”.