A Benchmark and Baseline for Language Driven Image Editing (Supplementary Material)

Jing Shi[1], Ning Xu[2], Trung Bui[2], Franck Dernoncourt[2], Zheng Wen[2], and Chenliang Xu[1]

[1]University of Rochester [2]Adobe Research

[1]{j.shi,chenliang.xu}@rochester.edu

[2]{nxu,bui,dernonco}@adobe.com zhengwen@alumni.stanford.edu

In this supplementary document, we first introduce the user study process and present more editing results of our model (App. A). Then, we show more intermediate experimental results (App. B), describe the implementation of operations (App. C) and experiment implementation details (App. D). Finally, we elaborate the data collection interface (App. E) and visualize the two data collection criteria of our dataset (App. F).

A User Study and More Visualization Results

In this section, we firstly describe the user study and explain the related metrics. Then present more visualization of our editing methods.

Table 1. The comparison between a GAN-based method with our method. (This table is the same with Tab. 6 in main paper, shown here for better explaination of user evaluation.)

	L1	User Rating	User Interact
Target	-	3.60	-
Random Edit	0.1639	-	-
Pix2pixAug [6]	**0.1033**	2.68	13.5%
Our method	0.1071	**3.20**	**86.5%**

A.1 User Study Setup

We conduct two user studies to evaluate the editing quality and the feasibility for human-interactive editing on Amazon Mechanic Turk (AMT). For each user study, we random select 250 unique source images from the test set and each edit will be evaluated twice.

User Rating. *User rating* reflects the user perceptual evaluation for editing quality. We collect user rating by showing users with input image, language

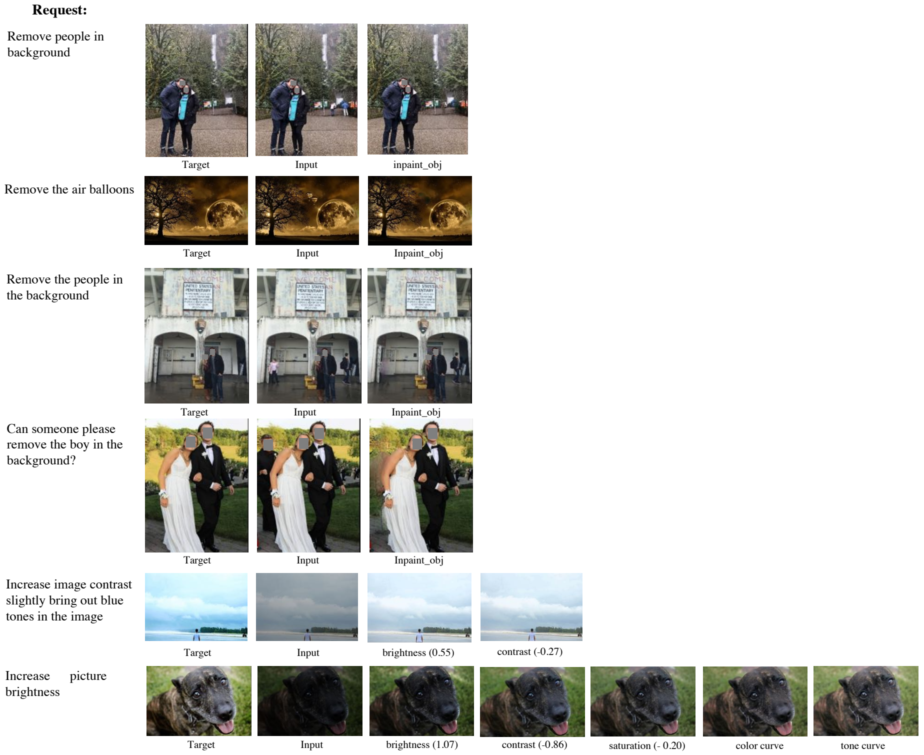


Fig. 1. Visualization of the editing results including the intermediate steps from the whole pipeline of our method.

request, and random order of target image, editing results of pix2pixAug and our method (user does not know which image corresponds to which method). We let 38 users to rate score from 1 (worst) to 5 (best) for each of the edited image, and average over all users and images.

User Interact. *User Interact* indicates the feasibility of editing method for human-interactive editing. We measure such feasibility by showing sequence of editing images from our method and the editing image from Pix2pixAug for users, and let users choose the image based on which they will follow up their own edits to complete the editing request. In total, 26 users are involved in this user study. We show the percentage of each method that users would like to start with in Tab. 1. And it manifests 86.5% chance that users prefer follow up editing based on the editing generated by our methods. Among the users who choose our method, they prefer the image at 66.56% of the total length of the editing sequence, indicating that the intermediate images are more favored for human-interactive editing.

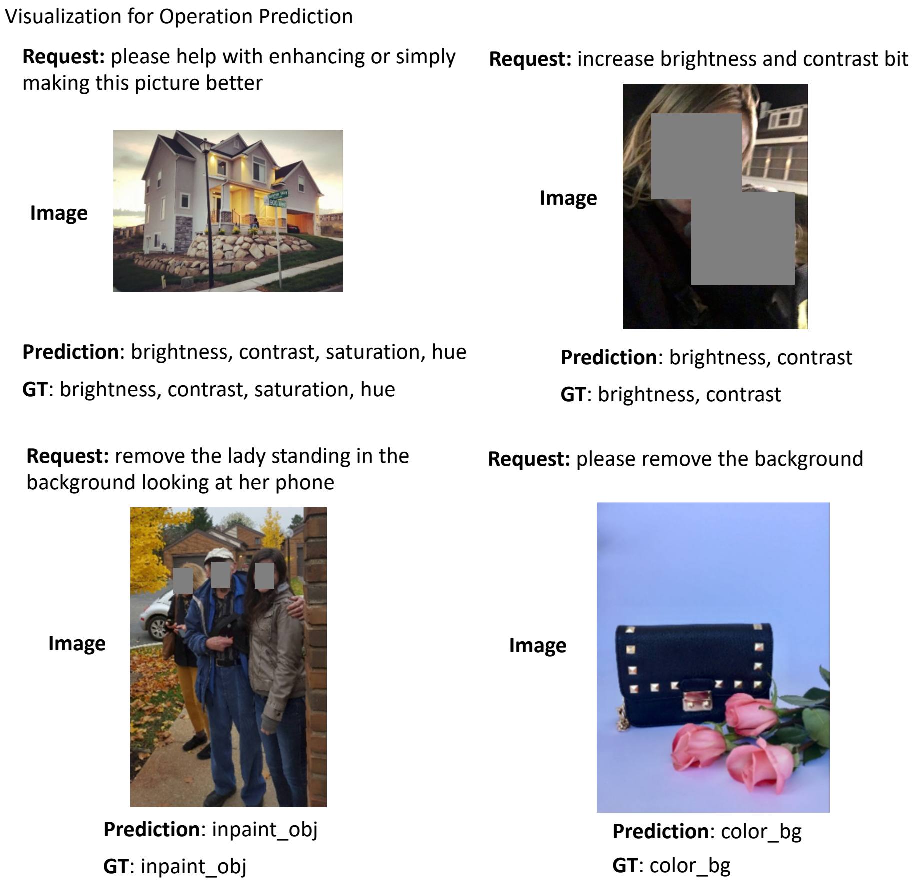
A.2 More Visual Results

Figure 1 shows more our editing results.

B More Experiment Results

B.1 Operation prediction

The visualization of our predicted operation against ground truth is shown in Fig. 2. The failure case is shown in Fig. 3, indicating the model might miss or over predict the operations if multiple operations are applied.



Visualization for Operation Prediction

Request: please help with enhancing or simply making this picture better

Image

Prediction: brightness, contrast, saturation, hue

GT: brightness, contrast, saturation, hue

Request: increase brightness and contrast bit

Image

Prediction: brightness, contrast

GT: brightness, contrast

Request: remove the lady standing in the background looking at her phone

Image

Prediction: inpaint_obj

GT: inpaint_obj

Request: please remove the background

Image

Prediction: color_bg

GT: color_bg

Fig. 2. The visualization of operation prediction. The operation prediction model takes in the request and the input image, outputs the predicted operations. The ground truth operations (GT) are listed for reference.

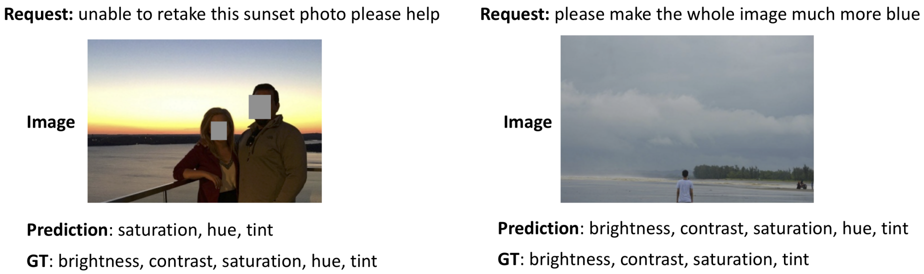

Request: unable to retake this sunset photo please help
Image
Prediction: saturation, hue, tint
GT: brightness, contrast, saturation, hue, tint
Request: please make the whole image much more blue
Image
Prediction: brightness, contrast, saturation, hue, tint
GT: brightness, contrast, saturation, tint

Fig. 3. The failure cases for operation prediction.

B.2 Operation Attentions Visualization

The attention of the operation over the request is shown in Fig. 4. The visual result shows the operation can attend to the key words indicating where the operation should be applied to.

C Operation Submodule Design

For the current task, we choose 8 operations: `brightness`, `saturation`, `contrast`, `sharpness`, `hue`, `tint`, `inpaint_obj`, `color_bg`. The operation modular network is composed of these operations in a fixed order if they are needed. With the input image I, parameter p, and output image I', the implementation of operation submodules are illustrated as follows.

C.1 Brightness and Saturation

The hue, saturation, value in the HSV space of image I is denoted as $H(I)$, $S(I)$, $V(I)$. Here p is an unbounded scalar. Let $V'(I) = \text{clip}((1+p)\cdot V(I), 0, 1)$ and $S'(I) = \text{clip}((1+p)\cdot S(I), 0, 1)$, the output image for brightness operation is

$$I' = \text{HSVtoRGB}(H(I), S(I), V'(I)), \tag{1}$$

and the output image for saturation operation is

$$I' = \text{HSVtoRGB}(H(I), S'(I), V(I)). \tag{2}$$

The HSVtoRGB is a differentiable function mapping the RGB space to HSV space, implemented via Kornia [5], and $\text{clip}(x, 0, 1)$ is a clip function to clip x within 0 to 1.

C.2 Contrast

Contrast operation is controlled by a scalar parameter p, implemented following [2]. First compute the luminance of image I as

$$\text{Lum}(I) = 0.27I_r + 0.67I_g + 0.06I_b, \tag{3}$$

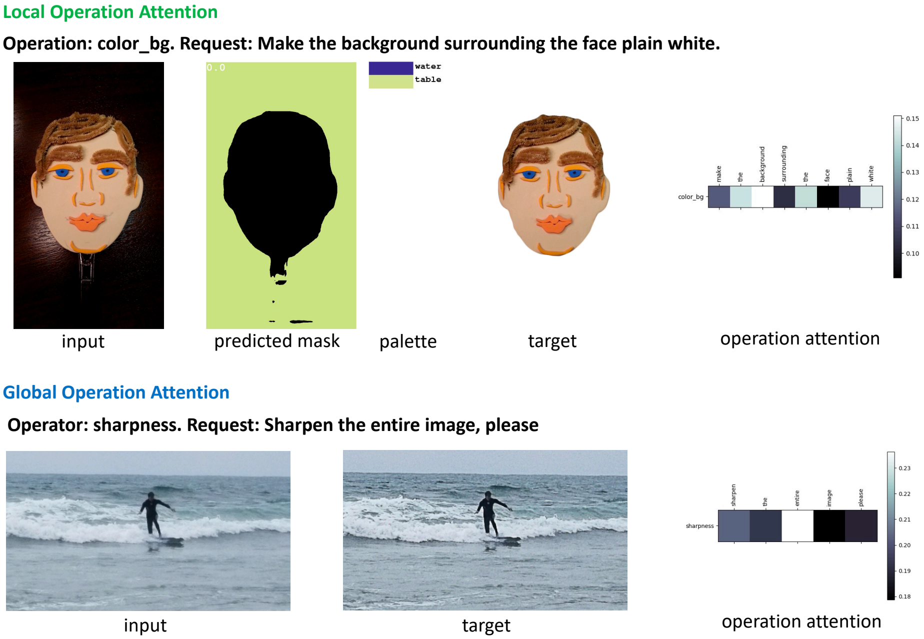


Fig. 4. Example of the operation attention over the request. The left part is local operation, also showing the predicted masks.

where I_r, I_g, I_b are the RGB channels of I. The enhanced luminance is

$$\mathrm{EnhancedLum}(I) = \frac{1}{2}(1 - \cos(\pi \cdot \mathrm{Lum}(I))), \tag{4}$$

and the image with enhanced contrast is

$$\mathrm{EnhancedC}(I) = I \cdot \frac{\mathrm{EnhancedLum}(I)}{\mathrm{lum}(I)}. \tag{5}$$

The output image I' is the combination of the enhanced contrast and original image

$$I' = (1 - p) \cdot I + p \cdot \mathrm{EnhancedC}(I) \tag{6}$$

C.3 Sharpness

The sharpness operation is implemented by adding to the image with its second-order spatial gradient [1], expressed as

$$I' = I + p\Delta^2 I, \tag{7}$$

where p is a scalar parameter and $(\Delta^2 \cdot)$ is the Laplace operator over the spatial domain of the image. The Laplace operator is applied to each channel of the image.

C.4 Tint and Hue

The tint and hue operation follows curve representation [2]. The curve is estimated as piece-wise linear functions with N pieces. The parameter $p = \{p_i\}_{i=0}^{M-1}$ is a vector of length M. With the input pixel $x \in [0, 1]$, the output pixel intensity is

$$f(x) = \frac{1}{Z} \sum_{i=0}^{N-1} \mathrm{clip}(Nx - i, 0, 1) p_i, \tag{8}$$

where $Z = \sum_{i=0}^{N-1} p_i$. For tint operation, $N = M = 8$, the same f will apply to each of the RGB channels of the image I. For hue operation, three different f are applied individually to each of RGB channels. Each $f(x)$ has $N = 8$, leading to $M = 3N = 24$.

C.5 Inpaint_obj and Color_bg

`inpaint_obj` indicates inpainting object and `inpaint_obj` denotes color background. The inpainting is implemented by EdgeConnect [3]. Since the `color_bg` operation plays the major role at removing the background, so we force the color to be white, so that the operation will be applied to the background mask and make the background white. `inpaint_obj` is also not resolution-independent because it is implemented using a neural network. Fortunately, these two operations do not need parameters, so we put them at the first positions in the modular network, allowing the following operation modules fully differentiable w.r.t the final output image.

D Experiment Details

The image is normalized from zero to one. Training images are resized to 128 × 128, and testing images are resized to short side 300 but capped long side 500, keeping aspect ratio unchanged. For operation prediction and operation conditioned grounding, they are optimized with Adam with initial learning rate 0.0004 and batch size 16. The learning rate is decreased by a factor of 10 every 8000 iterations after the first 8000-iteration warm-up. The word embedding size, operation embedding size and hidden size for LSTM are all 512, and all MLP and FC layers are 512-dimensional. The image feature is extracted via the Global Average Pooling (GAP) layer after the last feature map of ResNet18, with dimension 512. In operation conditioned grounding, the model configuration for MattNet variant remains the same as MattNet. The binary cross-entropy loss for local/global classification and the ranking loss of MattNet has equal balance weight. The MattNet variant does not start to train until finishing 1000 iterations of warm-up for the local/global classifier. The mask proposal feature for MattNet variant is extracted by applying GAP to the feature map masked with each mask proposal, where the feature map is the output of the semantic head in UPSNet [7], and the mask proposal is the predicted panoptic mask by UPSNet.

For operation modular network, it is optimized with Adam with learning rate 0.00001. The LSTM has two hidden layers stacked with 512 dimension each. The word embedding and operation embedding are of dimension 300 where the word embedding is initialized with Glove feature [4]. The MLP in each operation submodule is equipped with batch norm at the final FC layer to prevent over-fitting. The balance weight λ is set 1. Operation prediction, operation conditioned grounding, and operation modular network are trained in 20k iterations with best model at 12k, 14k, 16k iteration, respectively. And for integral testing, the confidence thresholds for operation prediction, local/global classification, and grounding matching score are set 0.4, 0.5, and 0.25, respectively.

E Data Collection Interface

Firstly, Fig. 5 shows the interface enabling workers to check the editing validity according to the two criteria (1. no new things or stuffs; 2. no edit to unknown region) and select feasible editing operations. All known regions are visualized in the image segmentation. Secondly, Fig. 6 is the interface to check the previous annotated operations and also collect the operation type (local or global) and the masks where operations are applied to. Next, the language request annotation is collected through interface in Fig. 7 and Fig. 8. Figure 7 is the interface for expert workers, where they can see all the annotated operations and masks and write profession-styled requests. Meanwhile, they should also check the correctness of previously annotated operations and masks and overwrite the incorrect ones. Figure 8 is the interface for amateur workers, where they can only see a image pair and are required to write a request. For the quality control, an extra interface shown in Fig. 9 is designed to check the amateur-annotated request.

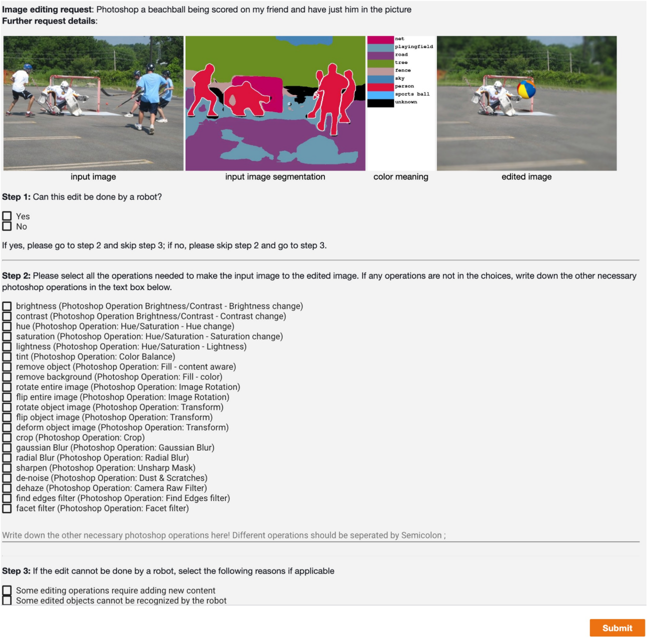


Image editing request: Photoshop a beachball being scored on my friend and have just him in the picture
Further request details:

Step 1: Can this edit be done by a robot?

□ Yes
□ No

If yes, please go to step 2 and skip step 3; if no, please skip step 2 and go to step 3.

Step 2: Please select all the operations needed to make the input image to the edited image. If any operations are not in the choices, write down the other necessary photoshop operations in the text box below.

□ brightness (Photoshop Operation Brightness/Contrast - Brightness change)
□ contrast (Photoshop Operation Brightness/Contrast - Contrast change)
□ hue (Photoshop Operation: Hue/Saturation - Hue change)
□ saturation (Photoshop Operation: Hue/Saturation - Saturation change)
□ lightness (Photoshop Operation: Hue/Saturation - Lightness)
□ tint (Photoshop Operation: Color Balance)
□ remove object (Photoshop Operation: Fill - content aware)
□ remove background (Photoshop Operation: Fill - color)
□ rotate entire image (Photoshop Operation: Image Rotation)
□ flip entire image (Photoshop Operation: Image Rotation)
□ rotate object image (Photoshop Operation: Transform)
□ flip object image (Photoshop Operation: Transform)
□ deform object image (Photoshop Operation: Transform)
□ crop (Photoshop Operation: Crop)
□ gaussian Blur (Photoshop Operation: Gaussian Blur)
□ radial Blur (Photoshop Operation: Radial Blur)
□ sharpen (Photoshop Operation: Unsharp Mask)
□ de-noise (Photoshop Operation: Dust & Scratches)
□ dehaze (Photoshop Operation: Camera Raw Filter)
□ find edges filter (Photoshop Operation: Find Edges filter)
□ facet filter (Photoshop Operation: Facet filter)

Write down the other necessary photoshop operations here! Different operations should be seperated by Semicolon ;

Step 3: If the edit cannot be done by a robot, select the following reasons if applicable

□ Some editing operations require adding new content
□ Some edited objects cannot be recognized by the robot

Fig. 5. The interface for the editing validity checking according to the two criteria and the selection of feasible operations.

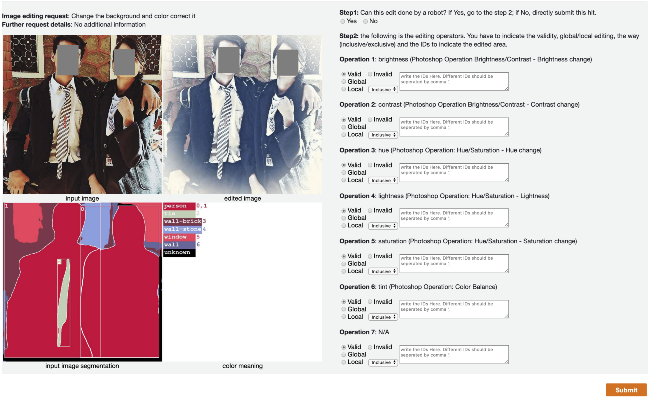


Image editing request: Change the background and color correct it
Further request details: No additional information

Step1: Can this edit done by a robot? If Yes, go to the step 2; if No, directly submit this hit.
○ Yes ○ No

Step2: the following is the editing operators. You have to indicate the validity, global/local editing, the way (inclusive/exclusive) and the IDs to indicate the edited area.

Operation 1: brightness (Photoshop Operation Brightness/Contrast - Brightness change)

● Valid ○ Invalid
○ Global
○ Local Inclusive
write the IDs Here. Different IDs should be seperated by comma ','

Operation 2: contrast (Photoshop Operation Brightness/Contrast - Contrast change)

● Valid ○ Invalid
○ Global
○ Local Inclusive
write the IDs Here. Different IDs should be seperated by comma ','

Operation 3: hue (Photoshop Operation: Hue/Saturation - Hue change)

● Valid ○ Invalid
○ Global
○ Local Inclusive
write the IDs Here. Different IDs should be seperated by comma ','

Operation 4: lightness (Photoshop Operation: Hue/Saturation - Lightness)

● Valid ○ Invalid
○ Global
○ Local Inclusive
write the IDs Here. Different IDs should be seperated by comma ','

Operation 5: saturation (Photoshop Operation: Hue/Saturation - Saturation change)

● Valid ○ Invalid
○ Global
○ Local Inclusive
write the IDs Here. Different IDs should be seperated by comma ','

Operation 6: tint (Photoshop Operation: Color Balance)

● Valid ○ Invalid
○ Global
○ Local Inclusive
write the IDs Here. Different IDs should be seperated by comma ','

Operation 7: N/A

● Valid ○ Invalid
○ Global
○ Local Inclusive
write the IDs Here. Different IDs should be seperated by comma ','

Fig. 6. The interface for annotating the masks corresponding to operations. The input image segmentation is the panoptic segmentation with each segment tagged a unique ID. Workers will first check whether each operation is valid. If valid, they further annotate whether the operation is global or local. If local, they finally should write down IDs of the segments to which the operation is applied to.

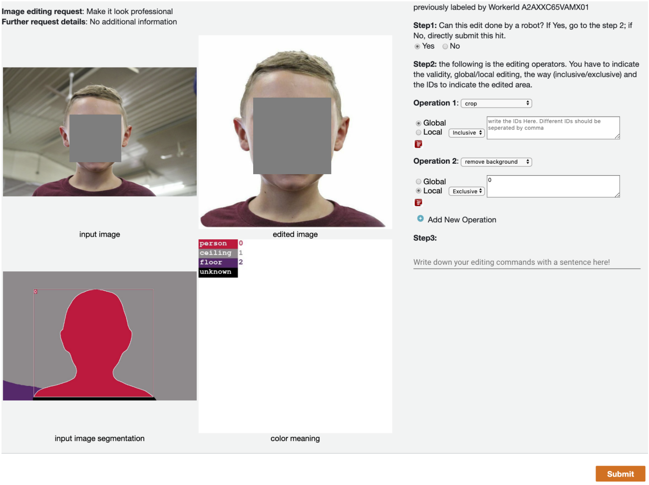


Image editing request: Make it look professional
Further request details: No additional information

previously labeled by WorkerId A2AXXC65VAMX01

Step1: Can this edit done by a robot? If Yes, go to the step 2; if No, directly submit this hit.
● Yes ○ No

Step2: the following is the editing operators. You have to indicate the validity, global/local editing, the way (inclusive/exclusive) and the IDs to indicate the edited area.

Operation 1: crop
● Global
○ Local Inclusive
write the IDs Here. Different IDs should be seperated by comma

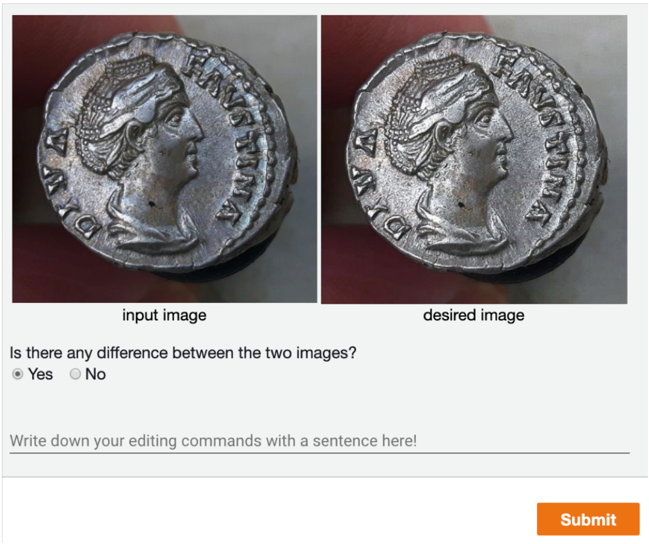
Operation 2: remove background
○ Global
● Local Exclusive
0

Add New Operation

Step3:

Write down your editing commands with a sentence here!

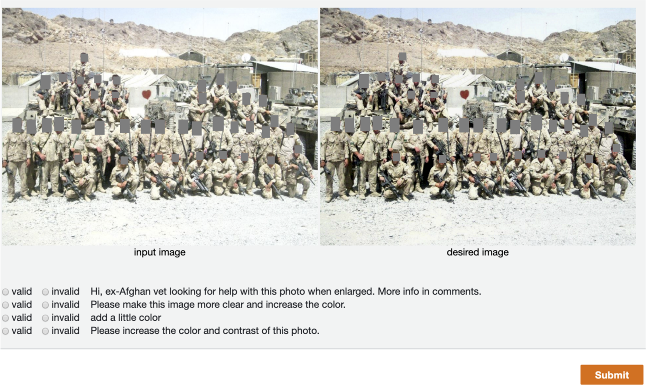
Fig. 7. The interface for expert workers to label the language request and check the previously annotated operations and masks. The worker have to check the previously annotated operations and masks and overwrite the incorrect annotations with correct ones, and also write a professional language request.



Is there any difference between the two images?
● Yes ○ No

Write down your editing commands with a sentence here!

Fig. 8. The interface for amateur workers to label the language request.



○ valid ○ invalid Hi, ex-Afghan vet looking for help with this photo when enlarged. More info in comments.
○ valid ○ invalid Please make this image more clear and increase the color.
○ valid ○ invalid add a little color
○ valid ○ invalid Please increase the color and contrast of this photo.

Fig. 9. The interface for checking the quality of amateur-annotated request, with mixture of the original crawled language request.

F Visualization of Two Data Collection Criteria

The collected image are extremely diversified with two major challenges: 1) edited images contains novel thing such as 'add a frame to the image', and are struggling to learn. 2) some edited area is hard to localize. Thus two criteria to get a simpler starting point: 1) the edited image should not have new thing added, for example, Fig. 10; 2) the edited region can be localized by our grounding model, e.g., Fig. 11 . We let Photoshop experts to filter out those images violating the two criteria, by presenting them with the image pairs, editing requests, and the image panoptic segmentations.

References

1. Gonzales, R.C., Woods, R.E.: Digital image processing (2002) 6
2. Hu, Y., He, H., Xu, C., Wang, B., Lin, S.: Exposure: A white-box photo post-processing framework. ACM Transactions on Graphics (TOG) **37**(2), 26 (2018) 4, 6
3. Nazeri, K., Ng, E., Joseph, T., Qureshi, F., Ebrahimi, M.: Edgeconnect: Generative image inpainting with adversarial edge learning. arXiv preprint arXiv:1901.00212 (2019) 6
4. Pennington, J., Socher, R., Manning, C.: Glove: Global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). pp. 1532–1543 (2014) 7
5. Riba, E., Mishkin, D., Ponsa, D., Rublee, E., Bradski, G.: Kornia: an open source differentiable computer vision library for pytorch (2020), https://arxiv.org/pdf/1910.02190.pdf 4
6. Wang, H., Williams, J.D., Kang, S.: Learning to globally edit images with textual description. arXiv preprint arXiv:1810.05786 (2018) 1
7. Xiong, Y., Liao, R., Zhao, H., Hu, R., Bai, M., Yumer, E., Urtasun, R.: Upsnet: A unified panoptic segmentation network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 8818–8826 (2019) 7

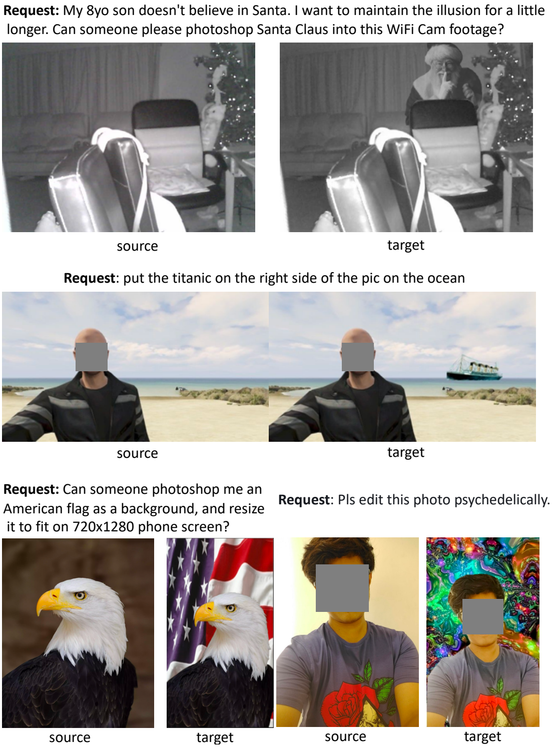

Request: My 8yo son doesn't believe in Santa. I want to maintain the illusion for a little longer. Can someone please photoshop Santa Claus into this WiFi Cam footage?
source
target
Request: put the titanic on the right side of the pic on the ocean
source
target
Request: Can someone photoshop me an American flag as a background, and resize it to fit on 720x1280 phone screen?
Request: Pls edit this photo psychedelically.
source
target
source
target

Fig. 10. The crawled data triplets (source image, target image and request) where novel things or staffs are added. The first two rows contain novel objects, and the last row contains new backgrounds

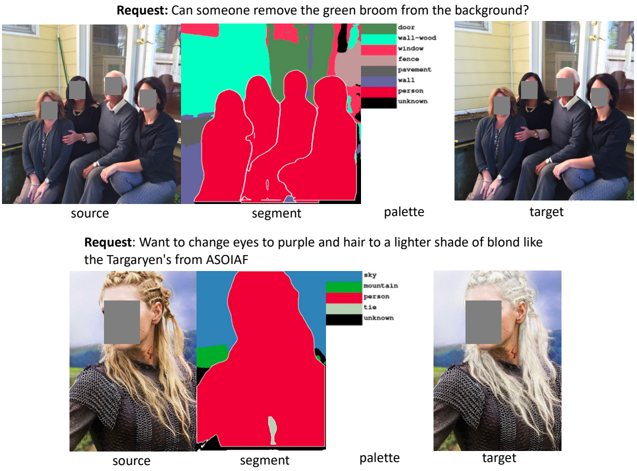

Request: Can someone remove the green broom from the background?
door
wall-wood
window
fence
pavement
wall
person
unknown
source
segment
palette
target
Request: Want to change eyes to purple and hair to a lighter shade of blond like the Targaryen's from ASOIAF
sky
mountain
person
tie
unknown
source
segment
palette
target

Fig. 11. The crawled data samples where objects cannot be localized by our grounding model. The segmentation is obtained from UPSNet and the palette indicates the categories for each color.