Disentangled Image Colorization via Global Anchors: Supplementary Material

MENGHAN XIA, Tecent AI Lab, China
WENBO HU, The Chinese University of Hong Kong, China
TIEN-TSIN WONG, The Chinese University of Hong Kong, China
JUE WANG, Tecent AI Lab, China

ACM Reference Format:
Menghan Xia, Wenbo Hu, Tien-Tsin Wong, and Jue Wang. 2022. Disentangled Image Colorization via Global Anchors: Supplementary Material. *ACM Trans. Graph.* 41, 6, Article 204 (December 2022), 9 pages. https://doi.org/10.1145/3550454.3555432

1 NETWORK ARCHITECTURE

Our colorization model consists of five parts, the backbone network, color modeler, color generator, *RefineNet* and *SPixNet*. We detail their architecture parameters in Table 1.

2 ADDITIONAL COMPARATIVE RESULTS

Comparison with random hints. Thanks to the property of global coverage, our located anchors can represent the color distribution of the whole image. To study the necessity, we compare our method with a naive baseline that allocates the anchors randomly. Fig. 1 shows two examples, which obviously tells the limitation of this baseline, i.e. many parts of the image still suffer from color ambiguity and thus cause desaturated colors.

Structural coherence achieved by our color modeler. As described in the main paper, the color modeler that predicts color probability for each superpixel primitive holds a decent structural coherence, which is attributed to the shared backbone feature with the anchor-guided color generator. To verify this speculation, we construct a baseline by removing the anchor-guided color generation branch. For both methods, we sample the color of the highest probability for each primitives. Fig. 2 illustrates two examples, indicating that the color modeler of our model (b) achieves more consistent color distribution than the baseline (a).

More comparison with prior arts. In the main paper, we only qualitatively compare with the most competitive methods (i.e. UGColor [Zhang et al. 2017], Deoldify [Antic 2019], ColTran [Kumar et al. 2021]) to avoid distraction. Here we provide the full visual comparison with all the existing competitors as introduced in the paper. Firstly, we provide the results of CIColor [Zhang et al. 2016], UGColor [Zhang et al. 2017], and ChromaGAN [Vitoria et al. 2020] on the challenging examples and real-world legacy photos in Fig. 3 and Fig. 4 respectively. Additionally, we illustrate more qualitative comparison in Fig. 5 and Fig. 6. All the results of our method are generated by our default model, without any hyper-parameters tuning.

(a) Result of random hints (b) Result of our anchors (c) Ground truth

Fig. 1. Performance comparison between randomly located hints and our color anchors. The 8 color hints are marked with white boundary. Flickr ©Christopher Sullivan; Flickr ©Greg Bishop.

(a) Color modeler of the model w/o guided branch (b) Color modeler of our model (c) Ground truth

Fig. 2. Colorization through independently color sampling. Flickr ©Flickr ©Oleg Klementiev; Flickr ©Nik Morris (van Leiden).

Authors' addresses: Menghan Xia, Tecent AI Lab, Shenzhen, China, menghanxyz@gmail.com; Wenbo Hu, The Chinese University of Hong Kong, Hong Kong, China, wbhu@cse.cuhk.edu.hk; Tien-Tsin Wong, The Chinese University of Hong Kong, Hong Kong, China, ttwong@cse.cuhk.edu.hk; Jue Wang, Tecent AI Lab, Shenzhen, China, arphid@gmail.com.

0730-0301/2022/12-ART204 $15.00
https://doi.org/10.1145/3550454.3555432

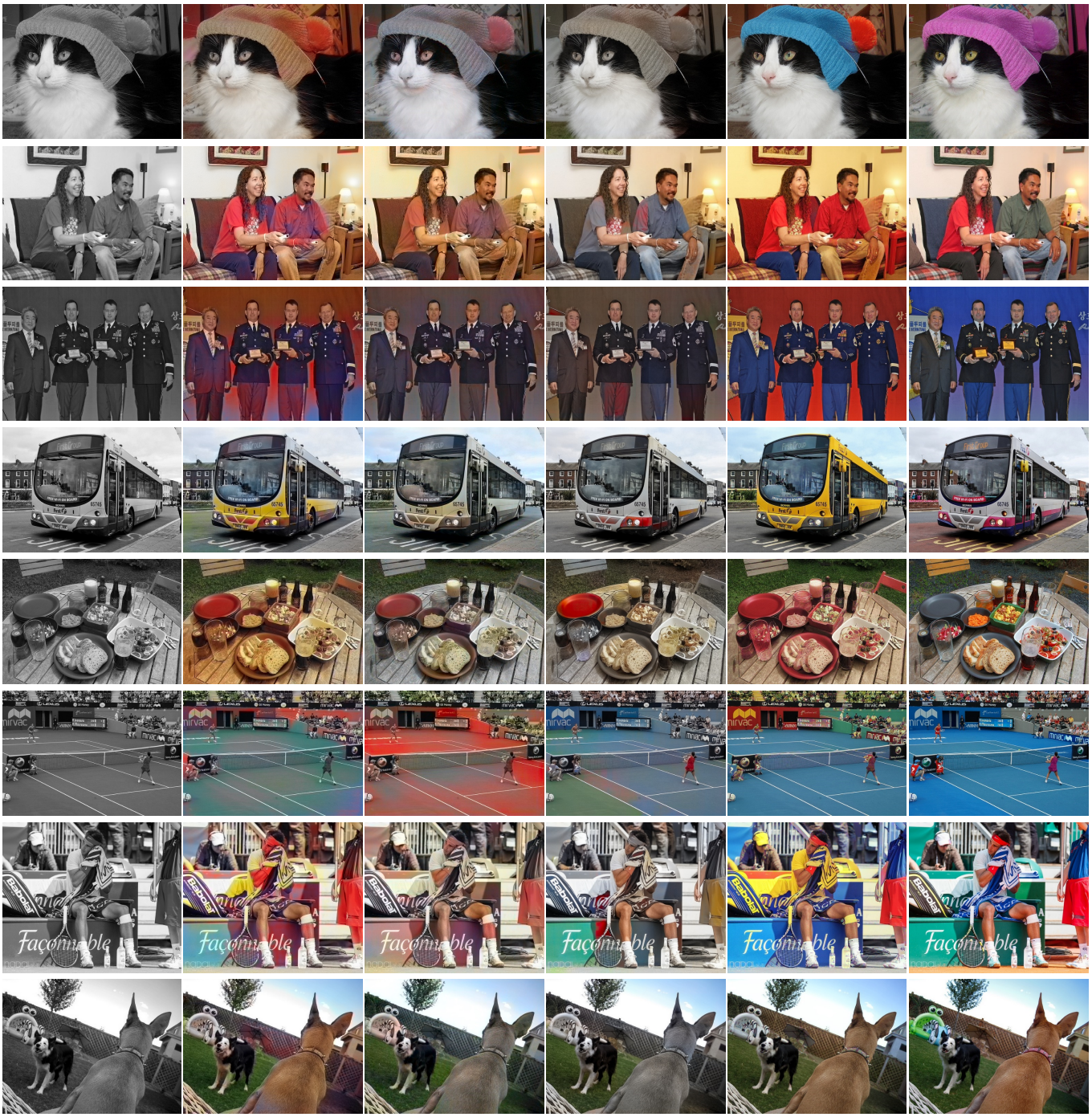
Table 1. Parameter illustration of network architectures. C(k,s,n) denotes a Convolutional layer with kernel size k, stride size s, and output channels of n. Accordingly, deC(k,s,n) denotes a transposed-convolutional layer. ResBlock(n) denotes a ResBlock with channel number of n. T(d1,d2,h) denotes a Transformer encoder layer with basic channel number of d1, forward channel number of d2, and self-attention head number of h. $\uparrow_{\times 2}$ denotes upsampling feature maps by twice (nearest neighbor). '+' denotes channel-wise addition while $\oplus$ denotes channel-wise concatenation.

Module	Input → Output	Layer Operation
Backbone	$\mathbf{L}(H,W,1) \rightarrow \mathbf{F}_b^1(H,W,64)$	C(3,1,64) → LReLU → C(3,1,64) → LReLU → BN
	$\mathbf{F}_b^1(H,W,64) \rightarrow \mathbf{F}_b^2(\frac{H}{2},\frac{W}{2},128)$	C(3,2,128) → LReLU → C(3,1,128) → LReLU → C(3,1,128) → LReLU → BN
	$\mathbf{F}_b^2(\frac{H}{2},\frac{W}{2},128) \rightarrow \mathbf{F}_b^3(\frac{H}{4},\frac{W}{4},256)$	C(3,2,256) → LReLU → C(3,1,256) → LReLU → C(3,1,256) → LReLU → BN
	$\mathbf{F}_b^3(\frac{H}{4},\frac{W}{4},256) \rightarrow \mathbf{F}_b^4(\frac{H}{8},\frac{W}{8},512)$	C(3,2,512) → LReLU → C(3,1,512) → LReLU → C(3,1,512) → LReLU → BN
	$\mathbf{F}_b^4(\frac{H}{8},\frac{W}{8},512) \rightarrow \mathbf{F}_b^5(\frac{H}{8},\frac{W}{8},512)$	C(3,2,512) → LReLU → C(3,1,512) → LReLU → C(3,1,512) → LReLU → BN
	$\mathbf{F}_b^5(\frac{H}{8},\frac{W}{8},512) \rightarrow \mathbf{F}_b^6(\frac{H}{8},\frac{W}{8},512)$	C(3,2,512) → LReLU → C(3,1,512) → LReLU → C(3,1,512) → LReLU → BN
	$\mathbf{F}_b^6(\frac{H}{8},\frac{W}{8},512) \rightarrow \mathbf{F}_b^7(\frac{H}{8},\frac{W}{8},512)$	C(3,2,512) → LReLU → C(3,1,512) → LReLU → C(3,1,512) → LReLU → BN
	$\mathbf{F}_b^7(\frac{H}{8},\frac{W}{8},512) \rightarrow \mathbf{F}_b^8(\frac{H}{4},\frac{W}{4},256)$	$\uparrow_{\times 2}$ → C(3,1,256) → +C(3,1,256)[$\mathbf{F}_b^3$] → ReLU → C(3,1,256) → ReLU → C(3,1,256) → ReLU → BN
	$\mathbf{F}_b^8(\frac{H}{4},\frac{W}{4},256) \rightarrow \mathbf{F}_b^9(\frac{H}{2},\frac{W}{2},128)$	$\uparrow_{\times 2}$ → C(3,1,256) → ReLU → C(3,1,256) → ReLU → BN
	$\mathbf{F}_b^9(\frac{H}{2},\frac{W}{2},128) \rightarrow \mathbf{F}(H,W,64)$	$\uparrow_{\times 2}$ → C(3,1,256) → ReLU → C(3,1,256) → ReLU
Color modeler	$\mathbf{F}(H,W,64) \rightarrow \mathbf{F}_S(\frac{H}{16},\frac{W}{16},64)$	*SP-Pooling*(**A**)
	$\mathbf{F}_S(\frac{H}{16},\frac{W}{16},64) \rightarrow \mathbf{F}_S^m(\frac{H}{16},\frac{W}{16},64)$	T(64,256,8) → T(64,256,8) → T(64,256,8) → T(64,256,8) → T(64,256,8) → T(64,256,8)
	$\mathbf{F}_S^m(\frac{H}{16},\frac{W}{16},64) \rightarrow \mathbf{P}_S(\frac{H}{16},\frac{W}{16},313)$	C(1,1,313) → Softmax
Color generator	$\mathbf{F}(H,W,64) \rightarrow \mathbf{F}_S(\frac{H}{16},\frac{W}{16},64)$	*SP-Pooling*(**A**)
	$\mathbf{F}_S(\frac{H}{16},\frac{W}{16},64) \rightarrow \mathbf{F}'_S(\frac{H}{16},\frac{W}{16},64)$	T(64,256,8) → T(64,256,8) → T(64,256,8) → T(64,256,8) → T(64,256,8) → T(64,256,8)
	$\mathbf{F}'_S(\frac{H}{16},\frac{W}{16},64) \rightarrow \mathbf{P}'_S(\frac{H}{16},\frac{W}{16},313)$	C(1,1,313) → Softmax
	$\mathbf{F}'_S(\frac{H}{16},\frac{W}{16},64) \rightarrow \mathbf{F}'(H,W,64)$	*SP-Diffusing*(**A**)
RefineNet	$[\mathbf{L} \oplus \mathbf{F}'](H,W,65) \rightarrow \mathbf{F}_r^1(H,W,64)$	C(3,1,64) → ReLU → C(3,1,64) → ReLU → BN
	$\mathbf{F}_r^1(H,W,64) \rightarrow \mathbf{F}_r^2(\frac{H}{2},\frac{W}{2},128)$	C(3,2,128) → ReLU → C(3,1,128) → ReLU → BN
	$\mathbf{F}_r^2(\frac{H}{2},\frac{W}{2},128) \rightarrow \mathbf{F}_r^3(\frac{H}{4},\frac{W}{4},256)$	C(3,2,256) → ReLU → C(3,1,256) → ReLU → BN
	$\mathbf{F}_r^3(\frac{H}{4},\frac{W}{4},256) \rightarrow \mathbf{F}_r^4(\frac{H}{4},\frac{W}{4},256)$	ResBlock(256) → ResBlock(256) → ResBlock(256)
	$\mathbf{F}_r^4(\frac{H}{4},\frac{W}{4},256) \rightarrow \mathbf{F}_r^5(\frac{H}{2},\frac{W}{2},128)$	C(3,1,128) → $\uparrow_{\times 2}$ → C(3,1,128)[$\oplus\mathbf{F}_r^2$] → ReLU → C(3,1,128) → ReLU → BN
	$\mathbf{F}_r^5(\frac{H}{2},\frac{W}{2},128) \rightarrow \mathbf{F}_r^6(H,W,64)$	C(3,1,64) → $\uparrow_{\times 2}$ → C(3,1,64)[$\oplus\mathbf{F}_r^1$] → ReLU → C(3,1,64) → ReLU → BN
	$\mathbf{F}_r^6(H,W,64) \rightarrow \mathbf{C}(H,W,2)$	C(3,1,2)
SPixNet	$\mathbf{L}(H,W,1) \rightarrow \mathbf{F}_p^1(H,W,16)$	C(3,1,16) → BN → LReLU → C(3,1,16) → BN → LReLU
	$\mathbf{F}_p^1(H,W,16) \rightarrow \mathbf{F}_p^2(\frac{H}{2},\frac{W}{2},32)$	C(3,2,32) → BN → LReLU → C(3,1,32) → BN → LReLU
	$\mathbf{F}_p^2(\frac{H}{2},\frac{W}{2},32) \rightarrow \mathbf{F}_p^3(\frac{H}{4},\frac{W}{4},64)$	C(3,2,64) → BN → LReLU → C(3,1,64) → BN → LReLU
	$\mathbf{F}_p^3(\frac{H}{4},\frac{W}{4},64) \rightarrow \mathbf{F}_p^4(\frac{H}{8},\frac{W}{8},128)$	C(3,2,128) → BN → LReLU → C(3,1,128) → BN → LReLU
	$\mathbf{F}_p^4(\frac{H}{8},\frac{W}{8},128) \rightarrow \mathbf{F}_p^5(\frac{H}{16},\frac{W}{16},256)$	C(3,2,256) → BN → LReLU → C(3,1,256) → BN → LReLU
	$\mathbf{F}_p^5(\frac{H}{16},\frac{W}{16},256) \rightarrow \mathbf{F}_p^6(\frac{H}{8},\frac{W}{8},128)$	deC(3,2,128) → LReLU → C(3,1,128)[$\oplus\mathbf{F}_p^4$] → BN → LReLU
	$\mathbf{F}_p^6(\frac{H}{8},\frac{W}{8},128) \rightarrow \mathbf{F}_p^7(\frac{H}{4},\frac{W}{4},64)$	deC(3,2,64) → LReLU → C(3,1,64)[$\oplus\mathbf{F}_p^3$] → BN → LReLU
	$\mathbf{F}_p^7(\frac{H}{4},\frac{W}{4},64) \rightarrow \mathbf{F}_p^8(\frac{H}{2},\frac{W}{2},32)$	deC(3,2,32) → LReLU → C(3,1,32)[$\oplus\mathbf{F}_p^2$] → BN → LReLU
	$\mathbf{F}_p^8(\frac{H}{2},\frac{W}{2},32) \rightarrow \mathbf{F}_p^9(H,W,16)$	deC(3,2,16) → LReLU → C(3,1,16)[$\oplus\mathbf{F}_p^1$] → BN → LReLU
	$\mathbf{F}_p^9(H,W,16) \rightarrow \mathbf{A}(H,W,9)$	C(3,1,9) → Softmax

Recently, GPColor [Wu et al. 2021] is proposed to utilize GAN inversion to generate a semantically related reference image for example-based colorization. Although it is an example-based colorization inherently, it could be used as a fully automatic colorization method in practice. In Fig. 7, we provide the qualitative comparison on several examples (of ImageNet) provided by the authors. Thanks to the automatically generated color reference image, GPColor can generate colorful results in most cases. However, it inherits the poor generalization from the image synthesis GAN, namely the GAN trained on ImageNet can not properly synthesize image contents beyond this dataset. Remarkably, our method even holds superiority to this example-based colorization in most cases.

3 SAMPLES USED IN USER STUDY

In user study, we randomly select 30 samples from the validation set of COCO-Stuff, 3 of which are fully grayscale and thus get excluded in the study. So, there are 27 samples are used in this user study. The ground-truth samples are illustrated in Fig. 8 for reference. It



(a) Input (b) CIColor (c) ChromaGAN (d) InstColor (e) Ours (f) Ground truth

Fig. 3. Comparative showcase of challenging cases. Here our competitors include CIColor, ChromaGAN, and InstColor.

includes images of diverse categories, such as the relatively easy natural scenes and challenging man-made objects.

REFERENCES

Jason Antic. 2019. DeOldify: A open-source project for colorizing old images (and video).

Manoj Kumar, Dirk Weissenborn, and Nal Kalchbrenner. 2021. Colorization Transformer. In *International Conference on Learning Representations (ICLR).*

Patricia Vitoria, Lara Raad, and Coloma Ballester. 2020. ChromaGAN: Adversarial Picture Colorization with Semantic Class Distribution. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*.

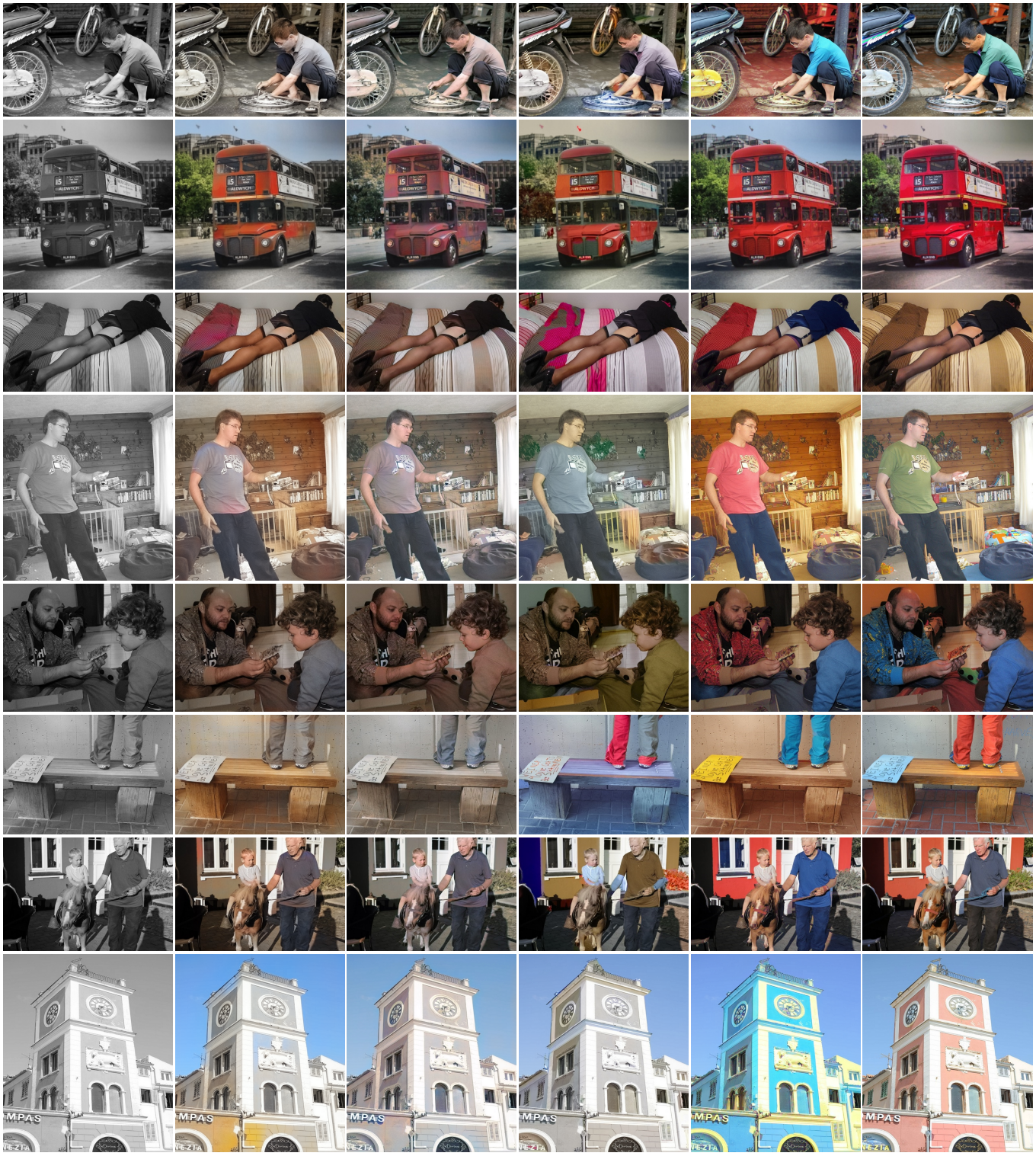
Yanze Wu, Xintao Wang, Yu Li, Honglun Zhang, Xun Zhao, and Ying Shan. 2021. Towards Vivid and Diverse Image Colorization with Generative Color Prior. In *IEEE International Conference on Computer Vision (ICCV)*.

Richard Zhang, Phillip Isola, and Alexei A. Efros. 2016. Colorful Image Colorization. In *European Conference on Computer Vision (ECCV)*.

Richard Zhang, Jun-Yan Zhu, Phillip Isola, Xinyang Geng, Angela S. Lin, Tianhe Yu, and Alexei A. Efros. 2017. Real-time user-guided image colorization with learned deep priors. *ACM Trans. Graph. (TOG)* 36, 4 (2017), 119:1–119:11.

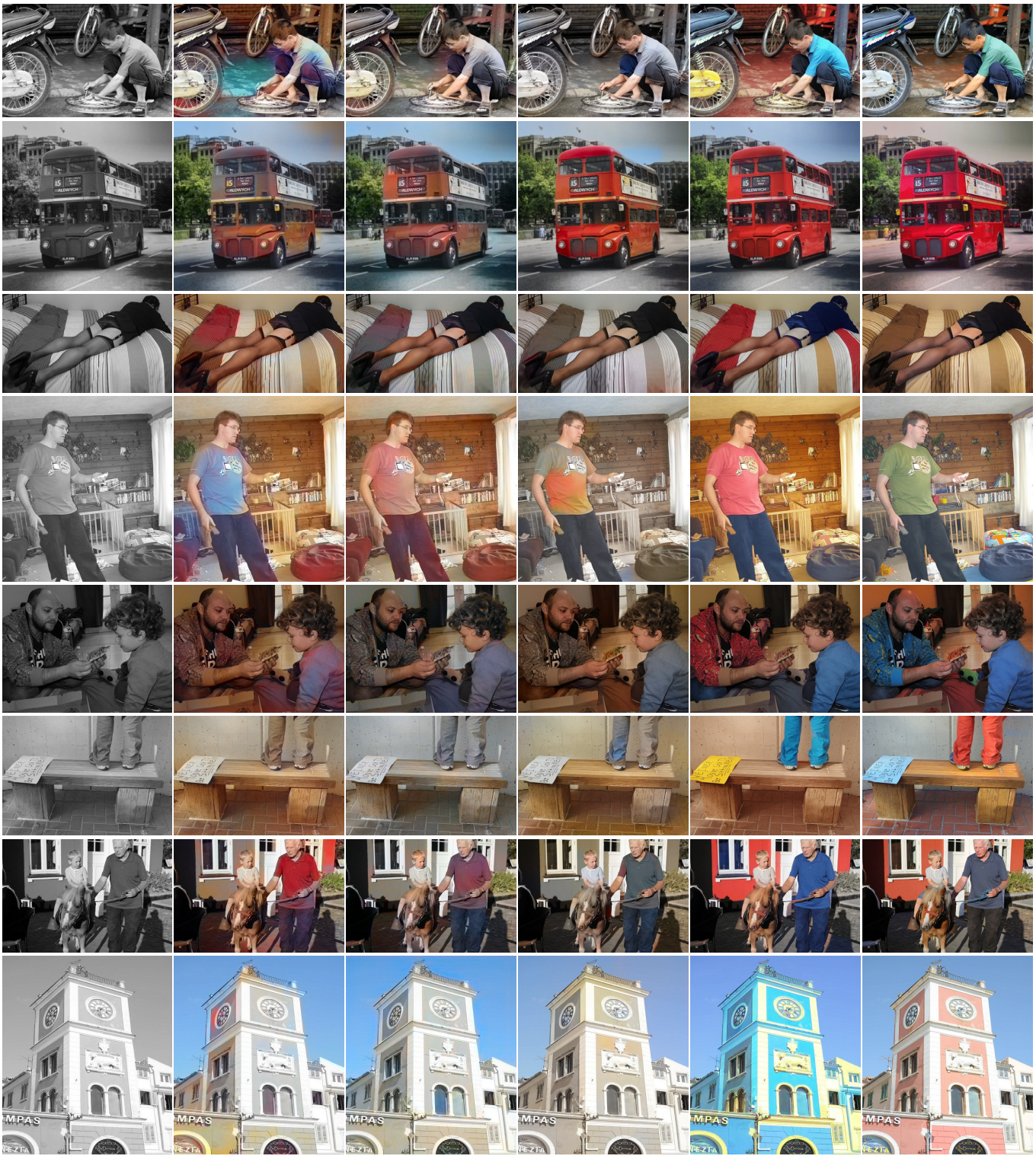
(a) Input (b) CIColor (c) ChromaGAN (d) InstColor (e) Ours

Fig. 4. Comparative showcase of colorizing historical photos (no ground truth available). Here our competitors include CIColor, ChromaGAN, and InstColor. Images are from the KeystoneDepth dataset (Public Domain).



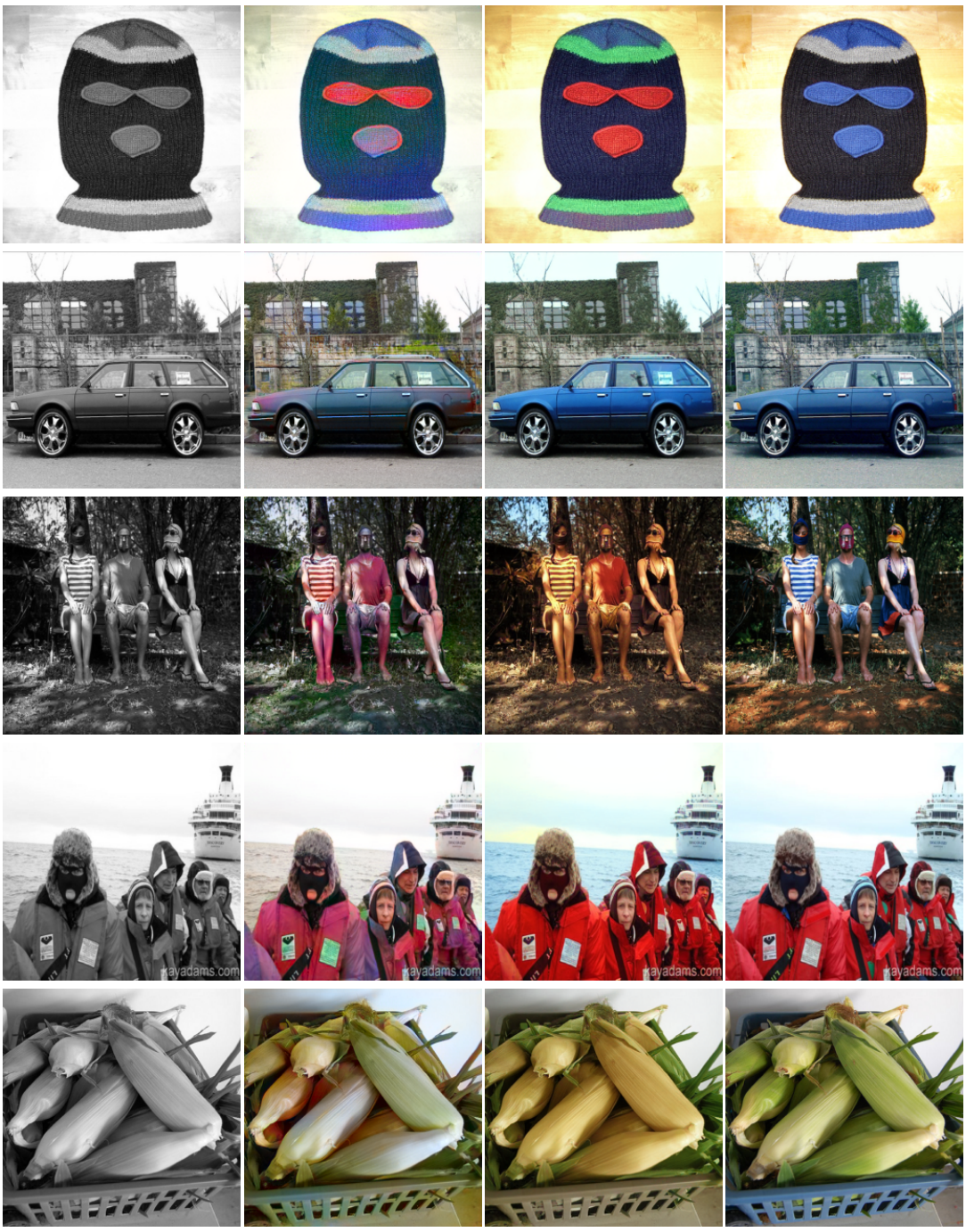
(a) Input (b) UGColor (c) Deoldify (d) ColTran (e) Ours (f) Ground truth

Fig. 5. Additional showcase. Here are comparison between UGColor, Deoldify, ColTran, and ours.



(a) Input (b) CIColor (c) ChromaGAN (d) InstColor (e) Ours (f) Ground truth

Fig. 6. Additional showcase. Here are comparison between CIColor, ChromaGAN, InstColor, and ours.



(a) Input (b) GPColor (c) Ours (d) Ground truth

Fig. 7. Comparative results from the example-based colorization GPColor and our automatic colorization.

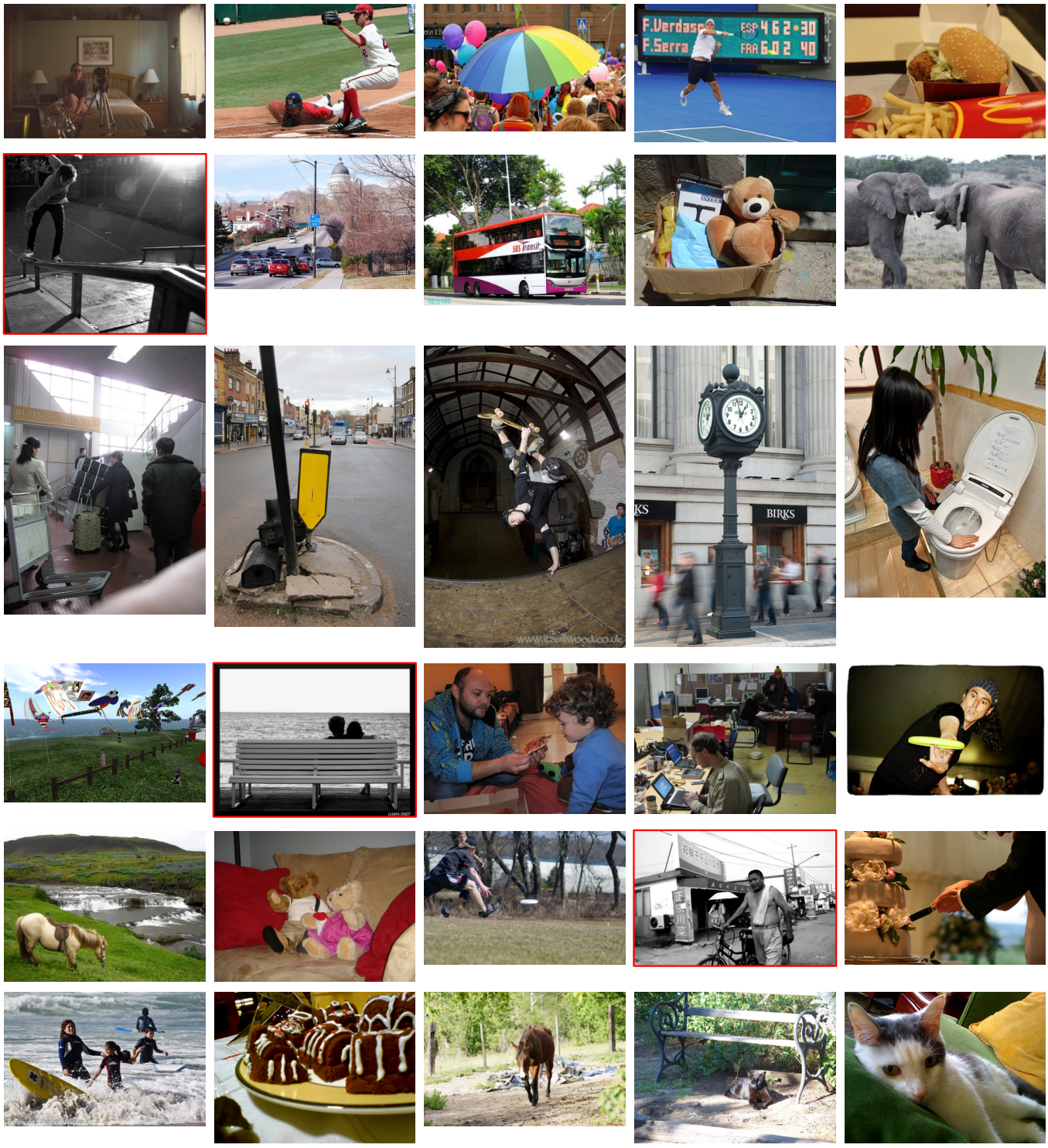


Fig. 8. Samples used in the user study. Here are the ground truth randomly sampled from the validation set of COCO-Stuff. The three grayscale images, as marked with red boundary, are not used in the study.