US Patent & Trademark Office
Patent Public Search | Text View

United States Patent	12395722
Kind Code	B2
Date of Patent	August 19, 2025
Inventor(s)	Figueroa; Luis et al.

Removing distracting objects from digital images

Abstract

The present disclosure relates to systems, methods, and non-transitory computer-readable media that modify digital images via scene-based editing using image understanding facilitated by artificial intelligence. For instance, in one or more embodiments, the disclosed systems provide, for display within a graphical user interface of a client device, a digital image displaying a plurality of objects, the plurality of objects comprising a plurality of different types of objects. The disclosed systems generate, utilizing a segmentation neural network and without user input, an object mask for objects of the plurality of objects. The disclosed systems determine, utilizing a distractor detection neural network, a classification for the objects of the plurality of objects. The disclosed systems remove at least one object from the digital image, based on classifying the at least one object as a distracting object, by deleting the object mask for the at least one object.

Inventors: **Figueroa; Luis (San Jose, CA), Ding; Zhihong (Fremont, CA), Cohen; Scott (Sunnyvale, CA), Lin; Zhe (Fremont, CA), Liu; Qing (Santa Clara, CA)**

Applicant: **Adobe Inc.** (San Jose, CA)

Family ID: **1000008763055**

Assignee: **Adobe Inc. (San Jose, CA)**

Appl. No.: **18/058554**

Filed: **November 23, 2022**

Prior Publication Data

Document Identifier	Publication Date
US 20240171848 A1	May. 23, 2024

Publication Classification

Int. Cl.: **H04N23/63** (20230101); **G06V10/26** (20220101); **G06V10/764** (20220101); **G06V10/82** (20220101); **G06V10/94** (20220101); **H04N5/262** (20060101)

U.S. Cl.:

CPC **H04N23/632** (20230101); **G06V10/273** (20220101); **G06V10/764** (20220101); **G06V10/82** (20220101); **G06V10/945** (20220101); **H04N5/2628** (20130101);

Field of Classification Search

CPC: G06F (3/04842); G06F (3/04845); G06F (3/0481); G06T (11/60); G06V (10/273); G06V (10/82); H04N (23/611); H04N (23/617); H04N (23/631); H04N (23/632); H04N (23/633); H04N (23/80); H04N (5/2628)

References Cited

U.S. PATENT DOCUMENTS

Patent No.	Issued Date	Patentee Name	U.S. Cl.	CPC
6204840	12/2000	Petelycky et al.	N/A	N/A
6750890	12/2003	Sugimoto	N/A	N/A
8125492	12/2011	Wainwright et al.	N/A	N/A
9256621	12/2015	Lessin	N/A	N/A
9405463	12/2015	Won et al.	N/A	N/A
10192129	12/2018	Price et al.	N/A	N/A
10249044	12/2018	Cheung et al.	N/A	N/A
10460214	12/2018	Lu et al.	N/A	N/A
10515160	12/2018	Jarvis	N/A	N/A
10692221	12/2019	Pao et al.	N/A	N/A
10748324	12/2019	Shechtman et al.	N/A	N/A
10755447	12/2019	Chhaya et al.	N/A	N/A
11055566	12/2020	Pham et al.	N/A	N/A
11094083	12/2020	Eisenmann et al.	N/A	N/A
11100917	12/2020	Bui et al.	N/A	N/A
11107219	12/2020	Cohen et al.	N/A	N/A
11126890	12/2020	Lin et al.	N/A	N/A
11238362	12/2021	Jin et al.	N/A	N/A
11373390	12/2021	Zhao et al.	N/A	N/A
11462040	12/2021	Lin et al.	N/A	N/A
11538216	12/2021	Sunkavalli et al.	N/A	N/A
11568627	12/2022	Price et al.	N/A	N/A
11676390	12/2022	Tang et al.	N/A	N/A
11682105	12/2022	Duan et al.	N/A	N/A
12026845	12/2023	Pardeshi	N/A	N/A
12175619	12/2023	Joachim	N/A	N/A
12210800	12/2024	Soni et al.	N/A	N/A
12223523	12/2024	Bhole et al.	N/A	N/A
2008/0109717	12/2007	Krauter	N/A	N/A
2009/0316957	12/2008	Chen et al.	N/A	N/A
2013/0076773	12/2012	Chen et al.	N/A	N/A
2013/0167087	12/2012	Tighe et al.	N/A	N/A

2014/0081625	12/2013	Wilensky et al.	N/A	N/A
2015/0356343	12/2014	Jain et al.	N/A	N/A
2015/0365591	12/2014	Aronsson et al.	N/A	N/A
2016/0180201	12/2015	Aubert et al.	N/A	N/A
2017/0032551	12/2016	Fried	N/A	G06T 7/11
2017/0123641	12/2016	Lance	N/A	N/A
2017/0347024	12/2016	Yanagi	N/A	N/A
2017/0358092	12/2016	Bleibel et al.	N/A	N/A
2019/0042871	12/2018	Pogorelik	N/A	G06T 5/73
2019/0205643	12/2018	Liu et al.	N/A	N/A
2019/0340798	12/2018	Garg	N/A	N/A
2020/0042286	12/2019	Bui et al.	N/A	N/A
2020/0051561	12/2019	Lai	N/A	N/A
2020/0143838	12/2019	Peleg et al.	N/A	N/A
2020/0151860	12/2019	Safdarnejad et al.	N/A	N/A
2020/0160042	12/2019	Bui et al.	N/A	N/A
2020/0302656	12/2019	Kumar et al.	N/A	N/A
2020/0312298	12/2019	Bui et al.	N/A	N/A
2021/0004576	12/2020	Bui et al.	N/A	N/A
2021/0027448	12/2020	Cohen et al.	N/A	N/A
2021/0027471	12/2020	Cohen et al.	N/A	N/A
2021/0027497	12/2020	Ding et al.	N/A	N/A
2021/0067344	12/2020	Drake et al.	N/A	N/A
2021/0073267	12/2020	Chopra et al.	N/A	N/A
2021/0103607	12/2020	Fisher	N/A	N/A
2021/0142497	12/2020	Pugh et al.	N/A	N/A
2021/0263962	12/2020	Chang et al.	N/A	N/A
2021/0349674	12/2020	Britton	N/A	N/A
2021/0374905	12/2020	Duan et al.	N/A	N/A
2022/0058001	12/2021	Chang et al.	N/A	N/A
2022/0068037	12/2021	Pardeshi	N/A	N/A
2022/0129670	12/2021	Lin et al.	N/A	N/A
2022/0129682	12/2021	Tang	N/A	G06T 7/74
2022/0237799	12/2021	Price et al.	N/A	N/A
2022/0270215	12/2021	Lee	N/A	N/A
2022/0284613	12/2021	Yin et al.	N/A	N/A
2022/0337709	12/2021	Tsuruta	N/A	N/A
2022/0383037	12/2021	Pham et al.	N/A	N/A
2023/0103305	12/2022	Xu et al.	N/A	N/A
2023/0127460	12/2022	Xie et al.	N/A	N/A
2023/0147722	12/2022	Ramirez de Chanlatte et al.	N/A	N/A
2023/0252752	12/2022	Cheng et al.	N/A	N/A
2023/0326028	12/2022	Zhang et al.	N/A	N/A
2024/0135511	12/2023	Singh et al.	N/A	N/A
2024/0153047	12/2023	Smith et al.	N/A	N/A
2024/0169502	12/2023	Cohen et al.	N/A	N/A
2024/0171848	12/2023	Figueroa et al.	N/A	N/A
2024/0331322	12/2023	Smith	N/A	N/A
2024/0355107	12/2023	Liba	N/A	G06V 10/462

2024/0412458	12/2023	Jampani et al.	N/A	N/A

FOREIGN PATENT DOCUMENTS

Patent No.	Application Date	Country	CPC
3158287	12/2020	CA	N/A
1512403	12/2003	CN	N/A
110033475	12/2019	CN	N/A
2623620	12/2023	GB	N/A
2014235553	12/2013	JP	N/A
2023/027681	12/2022	WO	N/A
2023/250088	12/2022	WO	N/A

OTHER PUBLICATIONS

B. AlBahar, J. Lu, J. Yang, Z. Shu, E. Shechtman, and J. Huang in Pose with Style: Detail-Preserving Pose-Guided Image Synthesis with Conditional StyleGAN, SIGGRAPH Asia 2021 (Pose with Style). cited by applicant
G. Huang et al., Densely Connected Convolutional Networks, in IEEE Conference on Computer Vision and Pattern Recognition, 2016. cited by applicant
J. Ho, A. Jain, P Abbeel, Denoising Diffusion Probabilistic Models, arXiv:2006:11239. cited by applicant
Jaemin Cho et al., Fine-grained Image Captioning with CLIP Reward, arXiv:2205.13115, 2022. cited by applicant
Jiaming Song, et al. in Denoising diffusion implicit models. In ICLR, 2021. cited by applicant
O. Ronneberger, P. Fischer, and T. Brox, in U-net: Convolutional networks for biomedical image. Segmentation, MICCAI (3), vol. 9351 of Lecture Notes in Computer Science, p. 234-250 (2015). cited by applicant
R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, in High-Resolution Image Synthesis with Latent Diffusion Models, arXiv:2112.10752v2. cited by applicant
Bin Ding, Chengjiang Long, Ling Zhang, Chunxia Xiao. "ARGAN: Attentive Recurrent Generative Adversarial Network for Shadow Detection and Removal." 2019 IEEE/CVF International Conference on Computer Vision (ICCV), 2019, pp. 10212-10221. cited by applicant
Bin Liao, Yao Zhu, Chao Liang, Fei Luo, Chunxia Xiao: "Illumination animating and editing in a single picture using scene structure estimation." Computers & Graphics (2019). cited by applicant
Bin Liu, Kun Xu, Ralph R. Martin: "Static scene illumination estimation from videos with applications." Journal of Computer Science and Technology (2017). cited by applicant
Chao Tan, Xin Feng. "Unsupervised Shadow Removal Using Target Consistency Generative Adversarial Network." arXiv preprint; arXiv:2010.01291v2, 14 pages, 2021. cited by applicant
Daquan Liu, Chengjiang Long, Hongpan Zhang, Hanning Yu, Xinzhi Dong, Chunxia Xiao: "ARshadowGAN: Shadow generative adversarial network for augmented reality in single light scenes." CVPR (2020). In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 8139-8148, 2020. cited by applicant
Daquan Liu, Chengjiang Long, Hongpan Zhang, Hanning Yu, Xinzhi Dong, Chunxia Xiao. (2020). "ARShadowGAN." [dataset]. The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). https://github.com/ldq9526/ARShadowGAN. cited by applicant
Fangneng Zhan, Shijian Lu, Changgong Zhang, Feiying Ma, Xuansong Xie: "Adversarial Image Composition with Auxiliary Illumination." ACCV 17 Pages. (2020). cited by applicant
Florin-Alexandru Vasluianu, Andres Romero, Luc Van Gool, and Radu Timofte. "Self-Supervised Shadow Removal." arXiv preprint; arXiv:2010.11619v1. 10 pages, 2021. cited by applicant
Haitian Zheng, Zhe Lin, Jingwan Lu, Scott Cohen, Eli Shechtman, Connelly Barnes, Jianming Zhang, Ning Xu, Sohrab Amirghodsi, and Jiebo Luo. "Image Inpainting with Cascaded Modulation GAN and Object-Aware Training". arXiv preprint, arXiv:2203.11947v3, 32 pages, 2022. cited by applicant
Hieu Le and Dimitris Samaras. "Physics-based Shadow Image Decomposition for Shadow Removal."

IEEE Transactions on Pattern Analysis and Machine Intelligence. 14 Pages. 2021. cited by applicant
Hieu Le and Dimitris Samaras. “From Shadow Segmentation to Shadow Removal.” In proceedings of the European Conference on Computer Vision. 18 pages, 2020. cited by applicant
Hieu Le, Dimitris Samaras. “Shadow Removal via Shadow Image Decomposition.” International Conference on Computer Vision 2019. (ICCV). pp. 8578-8587. 2019. cited by applicant
Jiawei Liu; Qiang Wang; Huijie Fan; Jiandong Tian; Yandong Tang. “A Shadow Imaging Bilinear Model and Three-Branch Residual Network for Shadow Removal,” in IEEE Transactions on Neural Networks and Learning Systems, doi: 10.1109/TNNLS.2023.3290078. cited by applicant
Jiawei Liu; Qiang Wang; Huijie Fan; Wentao Li; Liangqiong Qu; Yandong Tang. “A Decoupled Multi-Task Network for Shadow Removal,” in IEEE Transactions on Multimedia, vol. 25, pp. 9449-9463, 2023, doi: 10.1109/TMM.2023.3252271. cited by applicant
Jifeng Wang, Xiang Li, Le Hui, Jian Yang. “Stacked Conditional Generative Adversarial Networks for Jointly Learning Shadow Detection and Shadow Removal.” 2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) pp. 10. 2018. cited by applicant
Jin Wan, Hui Yin, Zhenyao Wu, Xinyi Wu, Yanting Liu, and Song Wan. “Style-Guided Shadow Removal.” In proceedings of European Conference on Computer Vision 2022. 17 Pages. 2022. cited by applicant
Jin Wan, Hui Yin, Zhenyao Wu, Xinyi Wu, Zhihao Liu, Song Wang. “CRFormer: A Cross-Region Transformer for Shadow Removal.” arXiv preprint., arXiv:2207.01600v1. 11 Pages. 2022. cited by applicant
Kevin Karsch, Kalyan Sunkavalli, Sunil Hadap, Nathan Carr, Hailin Jin, Rafael Fonte, Michael Sittig, David Forsyth: “Automatic scene inference for 3d object compositing.” ACM Transactions on Graphics (2014). cited by applicant
Kevin Karsch, Varsha Hedau, David Forsyth, Derek Hoiem: “Rendering synthetic objects into legacy photographs.” ACM Transactions on Graphics (2011). cited by applicant
Kunpeng Niu; Yanli Liu; Enhua Wu; Guanyu Xing. “A Boundary-Aware Network for Shadow Removal.” IEEE Transactions on Multimedia (vol. 25), pp. 6782-6793, 2022. cited by applicant
Lan Fu, Changqing Zhou, Qing Guo, Felix Juefei-Xu, Hongkai Yu, Wei Feng, Yang Liu, Song Wang. “Auto-Exposure Fusion for Single-Image Shadow Removal.” Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021, pp. 10571-10580. 2021. cited by applicant
Lanqing Guo, Chong Wang, Wenhan Yang, Siyu Huang, Yufei Wang, Hanspeter Pfister, Bihan Wen. “ShadowDiffusion: When Degradation Prior Meets Diffusion Model for Shadow Removal.” In CVPR 2023 pp. 14049-14058, 2023. cited by applicant
Lanqing Guo, Siyu Huang, Ding Liu, Hao Cheng, Bihan Wen. “ShadowFormer: Global Context Helps Image Shadow Removal.” In Proceedings AAAI Conference on Artificial Intelligence, 9 pages, 2023. cited by applicant
Lei Zhu, Zijun Deng, Xiaowei Hu, Chi-Wing Fu, Xuemiao Xu, Jing Qin, and Pheng-Ann Heng. “Bidirectional Feature Pyramid Network with Recurrent Attention Residual Modules for Shadow Detection.” Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 121-136. cited by applicant
Li Niu, Wenyan Cong, Liu Liu, Yan Hong, Bo Zhang, Jing Liang, Liqing Zhang: “Making Images Real Again: A Comprehensive Survey on Deep Image Composition.” arXiv preprint arXiv:2106.14490 (2021). cited by applicant
Liangqiong Qu, Jiandong Tian, Shengfeng He, Yandong Tang, Rynson W. H. Lau. “DeshadowNet: A Multi-Context Embedding Deep Network for Shadow Removal.” 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) pp. 4067-4075. 2017. cited by applicant
Ling Zhang, Chengjiang Long, Xiaolong Zhang, Chunxia Xiao. “RIS-GAN: Explore Residual and Illumination with Generative Adversarial Networks for Shadow Removal.” The Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI-20). pp. 12829-12836. 2020. cited by applicant
Lucas Valença, Jinsong Zhang, Michaël Gharbi, Yannick Hold-Geoffroy, Jean-François Lalonde:

"Shadow Harmonization for Realistic Compositing." SIGGRAPH Asia (2023) 12 Pages. cited by applicant
Lvmin Zhang, Jinyue Jiang, Yi Ji, Chunping Liu. "SmartShadow: Artistic Shadow Drawing Tool for Line Drawings." 2021 IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 2021, pp. 5371-5380, doi: 10.1109/ICCV48922.2021.00534, 2021. cited by applicant
Mehmet Kerim Yucel, Valia Dimaridou, Bruno Manganelli, Mete Ozay, Anastasios Drosou, Albert Saa-Garriga. "LRA&LDRA: Rethinking Residual Predictions for Efficient Shadow Detection and Removal." IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 4925-4935, 2023. cited by applicant
Mrinmoy Sen, Sai Pradyumna Chermala, Nazrinbanu Nurmohammad Nagori, Venkat Peddigari, Praful Mathur, B H Pawan Prasad, Moonhwan Jeong. "SHARDS: Efficient SHAdow Removal using Dual Stage Network for High-Resolution Images." IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 1809-1817, 2023. cited by applicant
Naoto Inoue, Toshihiko Yamasaki. "Learning from Synthetic Shadows for Shadow Detection and Removal." IEEE Transactions on Circuits and Systems for Video Technology (vol. 31, Issue: 11, Nov. 2021), pp. 4187-4197, 2020. cited by applicant
Qianhao Yu, Naishan Zheng, Jie Huang, and Feng Zhao. "CNSNet: A Cleanness-Navigated-Shadow Network for Shadow Removal." European Conference on Computer Vision Workshop 2022. 18 pages. 2022. cited by applicant
Qingyang Liu, Jianting Wang, Li Niu. (2023). "Object Shadow Generation Dataset—DESOBAv2." Version 2. [dataset]. AAAI 2023. https://github.com/bcmi/Object-Shadow-Generation-Dataset-DESOBAv2. cited by applicant
Qingyuan Zheng, Zhuoru Li, Adam Bargteil. "Learning to Shadow Hand-Drawn Sketches". The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 7436-7445, 2020. cited by applicant
Quanling Meng, Shengping Zhang, Zonglin Li, Chenyang Wang, Weigang Zhang, Qingming Huang: "Automatic Shadow Generation via Exposure Fusion." IEEE Transactions on Multimedia, vol. 25, pp. 9044-9056, 2023, doi: 10.1109/TMM.2023.3244398. cited by applicant
Shengfeng He, Bing Peng; Junyu Dong; Yong Du, "Mask-ShadowNet: Toward Shadow Removal via Masked Adaptive Instance Normalization," in IEEE Signal Processing Letters, vol. 28, pp. 957-961, 2021, doi: 10.1109/LSP.2021.3074082. cited by applicant
Shuyang Zhang, Runze Liang, Miao Wang: "ShadowGAN: Shadow synthesis for virtual objects with conditional adversarial networks." Computational Visual Media (2019). vol. 5, No. 1, Mar. 2019, 105-115. cited by applicant
Subhrajyoti Dasgupta, Arindam Das, Sudip Das, Andrei Bursuc, Ujjwal Bhattacharya and Senthil Yogamani. "UnShadowNet: Illumination Critic Guided Contrastive Learning for Shadow Removal." arXiv preprint arXiv: 2203.15441v1. 13 Pages. 2022. cited by applicant
Tianrun Chen, Lanyun Zhu, Chaotao Ding, Runlong Cao, Yan Wang, Zejian Li, Lingyun Sun, Papa Mao, Ying Zang. "SAM Fails to Segment Anything?—SAM-Adapter: Adapting SAM in Underperformed Scenes: Camouflage, Shadow, Medical Image Segmentation, and More." arXiv preprint, arXiv:2304.09148v3, 13 pages, 2023. cited by applicant
Tianyanshi Liu, Yuhang Li, Youdong Ding: "Shadow Generation for Composite Image with Multi-level Feature Fusion." EITCE (2022). cited by applicant
Tianyu Wang, Xiaowei Hu, Chi-Wing Fu, and Pheng-Ann Heng. "Single-Stage Instance Shadow Detection with Bidirectional Relation Learning." 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 11 pages, 2021. cited by applicant
Tianyu Wang, Xiaowei Hu, Pheng-Ann Heng, Chi-Wing Fu. "Instance Shadow Detection with A Single-Stage Detector." Institute of Electrical and Electronics Engineers (IEEE), IEEE Transactions on Pattern Analysis and Machine Intelligence, 14 pages, 2023. cited by applicant
Tianyu Wang, Xiaowei Hu, Qiong Wang, Pheng-Ann Heng, and Chi-Wing Fu. "Instance Shadow

Detection." In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 1880-1889). 2021. cited by applicant
Xiaodong Cun, Chi-Man Pun, Cheng Shi. "Towards Ghost-Free Shadow Removal via Dual Hierarchical Aggregation Network and Shadow Matting GAN." Proceedings of the AAAI Conference on Artificial Intelligence, 34(07), 10680-10687. https://doi.org/10.1609/aaai.v34i07.6695, 2020. cited by applicant
Xiaowei Hu, Chi-Wing Fu, Lei Zhu, Jing Qin, and Pheng-Ann Heng. "Direction-aware Spatial Context Features for Shadow Detection and Removal." IEEE Transactions on Pattern Analysis and Machine Intelligence. 14 pages. 2019. cited by applicant
Xiaowei Hu, Lei Zhu, Chi-Wing Fu, Jing Qin, and Pheng-Ann Heng. "Direction-aware Spatial Context Features for Shadow Detection." 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7454-7462, 2018. cited by applicant
Xiaowei Hu, Yitong Jiang, Chi-Wing Fu, Pheng-Ann Heng, "Mask-ShadowGAN: Learning to Remove Shadows from Unpaired Data." 2019 IEEE/CVF International Conference on Computer Vision (ICCV), pp. 2472-2481, 2019. cited by applicant
Xinhao Tao, Junyan Cao, Yan Hong, Li Niu: "Shadow Generation with Decomposed Mask Prediction and Attentive Shadow Filling." AAAI (2024). cited by applicant
Xinhao Tao, Junyan Cao, Yan Hong, Li Niu. "Rendered Shadow Generation Dataset—RDSOBA." [dataset] AAAI 2024. https://github.com/bcmi/Rendered-Shadow-Generation-Dataset-RdSOBA. cited by applicant
Xuaner Cecilia Zhang, Jonathan T. Barron, Yun-Ta Tsai, Rohit Pandey, Xiuming Zhang, Ren Ng, and David E. Jacobs. "Portrait Shadow Manipulation." In SIGGRAPH, CoRR abs/2005.08925, 14 pages, 2020. cited by applicant
Yan Hong, Li Niu, Jianfu Zhang: "Shadow Generation for Composite Image in Real-world Scenes." AAAI arXiv:2104.10338v1 14 Pages. (2022). cited by applicant
Yan Hong, Li Niu, Jianfu Zhang. (2022). "Object Shadow Generation Dataset—DESOBA." [dataset]. AAAI 2022. https://github.com/bcmi/Object-Shadow-Generation-Dataset-DESOBA. cited by applicant
Yan Hong, Li Niu, Jianfu Zhang. "Shadow Generation for Composite Image in Real-World Scenes." AAAI. (2022) . arXiv:2104.10338v3. 16 Pages. cited by applicant
Yaojie Liu, Andrew Hou, Xinyu Huang, Liu Ren, Xiaoming Liu. "Blind Removal of Facial Foreign Shadows." In Proceedings of the British Machine Vision Conference 2022, 14 pages, 2022. cited by applicant
Yeying Jin, Aashish Sharma, and Robby T. Tan. "DC-ShadowNet: Single-Image Hard and Soft Shadow Removal Using Unsupervised Domain-Classifier Guided Network.". In Proceedings of the IEEE international conference on computer vision, pp. 5027-5036. cited by applicant
Yeying Jin, Wenhan Yang, Wei Ye, Yuan Yuan and Robby T. Tan. "ShadowDiffusion: Diffusion-based Shadow Removal using Classifier-driven Attention and Structure Preservation." arXiv preprint arXiv:2211.08089v3. 10 pages. 2022. cited by applicant
Mchen Sheng, Jianming Zhang, Bedrich Benes: "SSN: Soft shadow network for image compositing." CVPR (2021). In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4380-4390, 2021. cited by applicant
Mchen Sheng, Jianming Zhang, Julien Philip, Yannick Hold-Geoffroy, Xin Sun, He Zhang, Lu Ling, Bedrich Benes: "PixHt-Lab: Pixel Height Based Light Effect Generation for Image Compositing." CVPR (2023). cited by applicant
Mchen Sheng, Yifan Liu, Jianming Zhang, Wei Yin, Oztireli Cengiz, He Zhang, Lin Zhe, Shechtman Eli, Bedrich Benes: "Controllable Shadow Generation Using Pixel Height Maps." ECCV (2022). arXiv:2207.05385v2. Jul. 15, 2022. 17 Pages. cited by applicant
Yimin Xu, Mingbao Lin, Hong Yang, Fei Chao, Rongrong Ji. "Shadow-Aware Dynamic Convolution for Shadow Removal." arXiv preprint. arXiv:2205.04908v3. 10 Pages. 2022. cited by applicant
Yingqing He, Yazhou Xing, Tianjia Zhang, and Qifeng Chen. 2021. "Unsupervised Portrait Shadow Removal via Generative Priors." In Proceedings of the 29th ACM Int'l Conference on Multimedia (MM

'21), Oct. 20-24, 2021, Virtual Event, 9 pages. https://doi.org/10.1145/3474085.3475663. cited by applicant
Yurui Zhu, Jie Huang, Xueyang Fu, Feng Zhao, Qibin Sun, Zheng-Jun Zha. “Bijective Mapping Network for Shadow Removal.” 2022 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 5627-5636. 2022. cited by applicant
Yurui Zhu, Xueyang Fu, Chengzhi Cao, Xi Wang, Qibin Sun, and Zheng-Jun Zha. 2022. “Single Image Shadow Detection via Complementary Mechanism.” In Proceedings of the 30th ACM International Conference on Multimedia (MM'22), Oct. 10-14, 2022, 10 pages. https://doi.org/10.1145/3503161.3547904. cited by applicant
Yurui Zhu, Zeyu Xiao, Yanchi Fang, Xueyang Fu, Zhiwei Xiong, and Zheng-Jun Zha. (2022). Efficient Model-Driven Network for Shadow Removal. Proceedings of the AAAI Conference on Artificial Intelligence, 36(3), 3635-3643. https://doi.org/10.1609/aaai.v36i3.2027. cited by applicant
Zhenghao Xing, Tianyu Wang, Xiaowei Hu, Haoran Wu, Chi-Wing Fu, Pheng-Ann Heng. “Video Instance Shadow Detection.” arXiv preprint, arXiv:2211.12827v1, 10 pages, 2023. cited by applicant
Zhihao Liu, Hui Yin, Xinyi Wu, Zhenyao Wu, Yang Mi, Song Wang. “From Shadow Generation to Shadow Removal.” The IEEE / CVF Computer Vision and Pattern Recognition Conference (CVPR). 10 pages, 2021. cited by applicant
Zhihao Liu; Hui Yin; Yang Mi; Mengyang Pu; Song Wang. “Shadow Removal by a Lightness-Guided Network With Training on Unpaired Data,” in IEEE Transactions on Image Processing, vol. 30, pp. 1853-1865, 2021, doi: 10.1109/TIP.2020.3048677. cited by applicant
Zipei Chen, Chengjiang Long, Ling Zhang, Chunxia Xiao. “CANet: A Context-Aware Network for Shadow Removal.” International Conference on Computer Vision 2021 (ICCV2021). 13 Pages. 2021. cited by applicant
U.S. Appl. No. 18/058,630, Feb. 28, 2024, Notice of Allowance. cited by applicant
Office Action as Received in CN Application No. 202311286113.8 dated Nov. 16, 2023. cited by applicant
U.S. Appl. No. 18/311,713, Jan. 4, 2024, Office Action. cited by applicant
Xihui Liu, Zhe Lin, Jianming Zhang, Handong Zhao, Quan Tran, Xiaogang Wang, and Hongsheng Li; “Open-Edit: Open-Domain Image Manipulation with Open-Vocabulary Instructions”; arXiv:2008.01576v2[cs.CV] Apr. 21, 2021. cited by applicant
Khoi Pham, Kushal Kafle, Zhe Lin, Zhihong Ding, Scott Cohen, Quan Tran, Abhinav Shrivastava; “Learning to Predict Visual Attributes in the Wild”; CVPR 2021. cited by applicant
Redmon, et al., You Only Look Once: Unified, Real-Time Object Detection, CVPR 2016. cited by applicant
Ning Xu et al., “Deep GrabCut for Object Selection,” published Jul. 14, 2017. cited by applicant
Lu Chi, Borui Jiang, and Yadong Mu in Fast Fourier convolution, Advances in Neural Information Processing Systems, 33 (2020). cited by applicant
Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila in Analyzing and improving the image quality of StyleGAN, Proc. CVPR (2020) (hereinafter StyleGan2). cited by applicant
Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu in Semantic image synthesis with spatially-adaptive normalization, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2019). cited by applicant
S. Ren, K. He, R. Girshick, and J. Sun, Faster r-cnn: Towards real-time object detection with region proposal networks, NIPS, 2015. cited by applicant
Y. Li, W. Ouyang, B. Zhou, Y. Cui, J. Shi, and X. Wang, Factorizable net: An efficient subgraph based framework for scene graph generation, ECCV, Jun. 29, 2018. cited by applicant
F. Faghri et al., Vse++: Improving visual-semantic Embeddings with Hard Negatives, arXiv:1707.05612, 2017. cited by applicant
Liang-Chieh Chen et al., Rethinking Atrous Convolution for Semantic Image Segmentation,

arXiv:1706.05587, 2017. cited by applicant
Liang-Chieh Chen et al., Deeplab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs, arXiv:1606.00915, 2016. cited by applicant
Chen, Hsiang-Ting et al. ACM Digital Library: Data-driven adaptive history for image editing, published Feb. 27, 2016, pp. 1-9. https://dl.acm.org/doi/10.1145/2856400.2856417 (Year: 2016). cited by applicant
Combined Search and Examination Report received in application No. GB2314585.7 dated Feb. 28, 2024. cited by applicant
Combined Search and Examination Report received in application No. GB2314323.3 dated Feb. 28, 2024. cited by applicant
Combined Search and Examination Report received in application No. GB2316991.5 dated May 8, 2024. cited by applicant
Combined Search and Examination Report received in application No. GB2317196.0 dated May 21, 2024. cited by applicant
Dhamo H, Farshad A, Laina I, Navab N, Hager GD, Tombari F, Rupprecht C. "Semantic Image Manipulation Using Scene Graphs", 2020, IEEE/CVF Conf on Computer Vision and Pattern Recognition (CVPR), 20200613-20200619, Seattle, WA, pp. 5212-5221. cited by applicant
Draftback Replays a Google Document's Changes in Video Format!, Published Sep. 27, 2019, pp. 1-10, https://web.archive.org/web/20220902104215/https://terrystechtipscchs.blogspot.com/2019/09/draftback-replays-google-documents.html (Year: 2019). cited by applicant
M. Choi, "Referring Object Manipulation of Natural Images with Conditional Classifier-free guidance", Computer Vision ECCV 2022. ECCV 2022, Lecture Notes in Computer Science, vol. 13696. Springer Cham. https://doi.org/. cited by applicant
P.K. Saha et al., "Data Augmentation Technique to Expand Road Dataset Using Mask RCNN and Image Inpainting," 2021 International Conference on Intelligent Technologies (CONIT), Hubli, India, 2021, pp. 1-6, doi: 10.1109/CONIT51480.2021.9498505. cited by applicant
U.S. Appl. No. 18/058,622, Mar. 21, 2024, Office Action. cited by applicant
U.S. Appl. No. 18/311,705, Mar. 27, 2024, Office Action. cited by applicant
Xiaohang Zhan, Xingang Pan, Bo Dai, Ziwei Liu, Dahua Lin, Chen Change Loy, "Self-Supervised Scene De-occlusion", arXiv:2004.02788v1, Apr. 6, 2020, 11 pages. cited by applicant
U.S. Appl. No. 18/058,630, Nov. 5, 2023, Office Action. cited by applicant
U.S. Appl. No. 18/058,538, Aug. 19, 2024, Office Action. cited by applicant
U.S. Appl. No. 18/311,705, Sep. 13, 2024, Office Action. cited by applicant
Combined Search and Examination Report received in application GB 2403915.8 dated Aug. 9, 2024. cited by applicant
Search Report received in GB2402982.9 dated Jul. 10, 2024. cited by applicant
"10 things we love about Affinity Photo 1.7", available on Jul. 25, 2021, pp. 1-6 https://web.archive.org/web/20210725082242/https://affinityspotlight. com/article/10-things-we-love-about-affi nity-photo-17 / (Year: 2021). cited by applicant
U.S. Appl. No. 18/058,622, Oct. 8, 2024, Office Action. cited by applicant
U.S. Appl. No. 18/311,705, Dec. 6, 2024, Office Action. cited by applicant
U.S. Appl. No. 18/311,713, Nov. 21, 2024, Notice of Allowance. cited by applicant
Wang, Instance Shadow Detection (2020). cited by applicant
U.S. Appl. No. 18/058,538, Jan. 31, 2025, Notice of Allowance. cited by applicant
U.S. Appl. No. 18/404,648, Jan. 30, 2025, Notice of Allowance. cited by applicant
U.S. Appl. No. 18/058,575, Feb. 12, 2025, Office Action. cited by applicant
Combined Search and Examination Report received in application No. GB2314323.3 dated Dec. 11, 2024. cited by applicant
Combined Search and Examination Report received in application No. GB2317072.3 dated May 16, 2024. cited by applicant
Search Report received in GB2318128.2 dated Jun. 4, 2024. cited by applicant

Search Report received in GB2318199.3 dated Jun. 5, 2024. cited by applicant
Combined Search and Examination Report received in application No. GB2403106.4 dated Jul. 12, 2024. cited by applicant
U.S. Appl. No. 18/311,713, Jul. 5, 2024, Office Action. cited by applicant
U.S. Appl. No. 18/404,648, Jul. 17, 2024, Office Action. cited by applicant
U.S. Appl. No. 18/058,622, Feb. 27, 2025, Office Action. cited by applicant
U.S. Appl. No. 18/320,664, Mar. 5, 2025, Office Action. cited by applicant
U.S. Appl. No. 18/460,365, Feb. 28, 2025, Notice of Allowance. cited by applicant
Tim Chen; Nonlinear Revision Control for Images, by Tim Chen Apr. 11, 2011, pp. 1-2 https://www.youtube.com/watch?v=RBL 1cVzlQik (Year: 2011). cited by applicant
U.S. Appl. No. 18/311,705, Mar. 28, 2025, Office Action. cited by applicant
Combined Search and Examination Report received in application No. GB2417128.2 dated Mar. 13, 2025. cited by applicant
Jinjiang Wei et al., “Shadow Inpainting and Removal Using Generative Adversarial Networks with Slice Convolutions” 2019, pp. 381-392. (Year: 2019). cited by applicant
Song, Yizhi, et al. “ObjectStitch date infomation.pdf”, publishing information for NPL “U”. cited by applicant
Song, Yizhi, et al. “ObjectStitch: Object Compositing with Diffusion Model” p. 18310-18319, “Objectstitch.pdf” (Year: 2023). cited by applicant
U.S. Appl. No. 18/460,150, Jul. 10, 2025, Office Action. cited by applicant
U.S. Appl. No. 18/532,485, Jun. 4, 2025, Office Action. cited by applicant
Second Examination Report received in application No. GB2314585.7 dated May 20, 2025. cited by applicant

---

Primary Examiner: Berhan; Ahmed A

Attorney, Agent or Firm: Keller Preece PLLC

Background/Summary

BACKGROUND

(1) Recent years have seen significant advancement in hardware and software platforms for performing computer vision and image editing tasks. Indeed, systems provide a variety of image-related tasks, such as object identification, classification, segmentation, composition, style transfer, image inpainting, etc.

SUMMARY

(2) One or more embodiments described herein provide benefits and/or solve one or more problems in the art with systems, methods, and non-transitory computer-readable media that implement artificial intelligence models to facilitate flexible and efficient scene-based image editing. To illustrate, in one or more embodiments, a system utilizes one or more machine learning models to learn/identify characteristics of a digital image, anticipate potential edits to the digital image, and/or generate supplementary components that are usable in various edits. Accordingly, the system gains an understanding of the two-dimensional digital image as if it were a real scene, having distinct semantic areas reflecting real-world (e.g., three-dimensional) conditions. Further, the system enables the two-dimensional digital image to be edited so that the changes automatically and consistently reflect the corresponding real-world conditions without relying on additional user input. Thus, the system facilitates flexible and intuitive editing of digital images while efficiently reducing the user interactions typically required to make such edits.

(3) Additional features and advantages of one or more embodiments of the present disclosure are outlined in the description which follows, and in part will be obvious from the description, or may be learned by the practice of such example embodiments.

Description

BRIEF DESCRIPTION OF THE DRAWINGS

(1) This disclosure will describe one or more embodiments of the invention with additional specificity and detail by referencing the accompanying figures. The following paragraphs briefly describe those figures, in which:

(2) FIG. **1** illustrates an example environment in which a scene-based image editing system operates in accordance with one or more embodiments;

(3) FIG. **2** illustrates an overview diagram of the scene-based image editing system editing a digital image as a real scene in accordance with one or more embodiments;

(4) FIG. **3** illustrates a segmentation neural network utilized by the scene-based image editing system to generate object masks for objects in accordance with one or more embodiments;

(5) FIG. **4** illustrates utilizing a cascaded modulation inpainting neural network to generate an inpainted digital image in accordance with one or more embodiments;

(6) FIG. **5** illustrates an example architecture of a cascaded modulation inpainting neural network in accordance with one or more embodiments;

(7) FIG. **6** illustrates global modulation blocks and spatial modulation blocks implemented in a cascaded modulation inpainting neural network in accordance with one or more embodiments;

(8) FIG. **7** illustrates a diagram for generating object masks and content fills to facilitate object-aware modifications to a digital image in accordance with one or more embodiments;

(9) FIGS. **8**A-**8**D illustrate a graphical user interface implemented by the scene-based image editing system to facilitate a move operation in accordance with one or more embodiments;

(10) FIGS. **9**A-**9**C illustrate a graphical user interface implemented by the scene-based image editing system to facilitate a delete operation in accordance with one or more embodiments;

(11) FIG. **10** illustrates an image analysis graph utilized by the scene-based image editing system in generating a semantic scene graph in accordance with one or more embodiments;

(12) FIG. **11** illustrates a real-world class description graph utilized by the scene-based image editing system in generating a semantic scene graph in accordance with one or more embodiments;

(13) FIG. **12** illustrates a behavioral policy graph utilized by the scene-based image editing system in generating a semantic scene graph in accordance with one or more embodiments;

(14) FIG. **13** illustrates a semantic scene graph generated by the scene-based image editing system for a digital image in accordance with one or more embodiments;

(15) FIG. **14** illustrates a diagram for generating a semantic scene graph for a digital image utilizing template graphs in accordance with one or more embodiments;

(16) FIG. **15** illustrates another diagram for generating a semantic scene graph for a digital image in accordance with one or more embodiments;

(17) FIG. **16** illustrates an overview of a multi-attribute contrastive classification neural network in accordance with one or more embodiments;

(18) FIG. **17** illustrates an architecture of a multi-attribute contrastive classification neural network in accordance with one or more embodiments;

(19) FIG. **18** illustrates an attribute modification neural network utilized by the scene-based image editing system to modify object attributes in accordance with one or more embodiments;

(20) FIGS. **19**A-**19**C illustrate a graphical user interface implemented by the scene-based image editing system to facilitate modifying object attributes of objects portrayed in a digital image in accordance with one or more embodiments;

(21) FIGS. **20**A-**20**C illustrate another graphical user interface implemented by the scene-based

image editing system to facilitate modifying object attributes of objects portrayed in a digital image in accordance with one or more embodiments;
(22) FIGS. **21**A-**21**C illustrate yet another graphical user interface implemented by the scene-based image editing system to facilitate modifying object attributes of objects portrayed in a digital image in accordance with one or more embodiments;
(23) FIGS. **22**A-**22**D illustrate a graphical user interface implemented by the scene-based image editing system to facilitate a relationship-aware object modification in accordance with one or more embodiments;
(24) FIGS. **23**A-**23**C illustrate another graphical user interface implemented by the scene-based image editing system to facilitate a relationship-aware object modification in accordance with one or more embodiments;
(25) FIGS. **24**A-**24**C illustrate yet another graphical user interface implemented by the scene-based image editing system to facilitate a relationship-aware object modification in accordance with one or more embodiments;
(26) FIGS. **25**A-**25**D illustrate a graphical user interface implemented by the scene-based image editing system to add objects to a selection for modification based on classification relationships in accordance with one or more embodiments;
(27) FIG. **26** illustrates a neural network pipeline utilized by the scene-based image editing system to identify and remove distracting objects from a digital image in accordance with one or more embodiments;
(28) FIG. **27** illustrates an architecture of a distractor detection neural network utilized by the scene-based image editing system to identify and classify distracting objects in of a digital image in accordance with one or more embodiments;
(29) FIG. **28** illustrates an architecture of a heatmap network utilized by the scene-based image editing system as part of a distractor detection neural network in accordance with one or more embodiments;
(30) FIG. **29** illustrates an architecture of a hybrid classifier utilized by the scene-based image editing system as part of a distractor detection neural network in accordance with one or more embodiments;
(31) FIGS. **30**A-**30**C illustrate a graphical user interface implemented by the scene-based image editing system to identify and remove distracting objects from a digital image in accordance with one or more embodiments;
(32) FIGS. **31**A-**31**C illustrate another graphical user interface implemented by the scene-based image editing system to identify and remove distracting objects from a digital image in accordance with one or more embodiments;
(33) FIGS. **32**A-**32**B illustrate the scene-based image editing system utilizing smart dilation to remove an object from a digital image in accordance with one or more embodiments;
(34) FIG. **33** illustrates an overview of a shadow detection neural network in accordance with one or more embodiments;
(35) FIG. **34** illustrates an overview of an instance segmentation component of a shadow detection neural network in accordance with one or more embodiments;
(36) FIG. **35** illustrates an overview of an object awareness component of a shadow detection neural network in accordance with one or more embodiments;
(37) FIG. **36** illustrates an overview of a shadow prediction component of a shadow detection neural network in accordance with one or more embodiments;
(38) FIG. **37** illustrates an overview of the architecture of a shadow detection neural network in accordance with one or more embodiments;
(39) FIG. **38** illustrates a diagram for using the second stage of the shadow detection neural network for determining shadows associated with objects portrayed in a digital image in accordance with one or more embodiments;

(40) FIGS. **39**A-**39**C illustrate a graphical user interface implemented by the scene-based image editing system to identify and remove shadows of objects portrayed in a digital image in accordance with one or more embodiments;
(41) FIGS. **40**A-**40**C illustrate a graphical user interface implemented by the scene-based image editing system to facilitate zoom-based image editing in accordance with one or more embodiments;
(42) FIGS. **41**A-**41**C illustrate a graphical user interface implemented by the scene-based image editing system to facilitate the automatic display of semantic information in accordance with one or more embodiments;
(43) FIGS. **42**A-**42**B illustrate a graphical user interface implemented by the scene-based image editing system to provide a timed confirmation for user edits in accordance with one or more embodiments;
(44) FIGS. **43**A-**43**B illustrate another graphical user interface implemented by the scene-based image editing system to provide a timed confirmation for user edits in accordance with one or more embodiments;
(45) FIG. **44** illustrates an example schematic diagram of a scene-based image editing system in accordance with one or more embodiments;
(46) FIG. **45** illustrates a flowchart for a series of acts for implementing an object-aware modification of a digital image in accordance with one or more embodiments;
(47) FIG. **46** illustrates a flowchart for a series of acts for removing distracting objects from a digital image in accordance with one or more embodiments;
(48) FIG. **47** illustrates a flowchart for a series of acts for detecting shadows cast by objects in a digital image in accordance with one or more embodiments;
(49) FIG. **48** illustrates a flowchart for a series of acts for expanding an object mask for an object in a digital image in accordance with one or more embodiments;
(50) FIG. **49** illustrates a flowchart for a series of acts for modifying attributes of an object in a digital image in accordance with one or more embodiments;
(51) FIG. **50** illustrates a flowchart for a series of acts for modifying objects within a digital image based on relationships with other objects in accordance with one or more embodiments; and
(52) FIG. **51** illustrates a block diagram of an exemplary computing device in accordance with one or more embodiments.

DETAILED DESCRIPTION

(53) One or more embodiments described herein include a scene-based image editing system that implements scene-based image editing techniques using intelligent image understanding. Indeed, in one or more embodiments, the scene-based image editing system utilizes one or more machine learning models to process a digital image in anticipation of user interactions for modifying the digital image. For example, in some implementations, the scene-based image editing system performs operations that build a knowledge set for the digital image and/or automatically initiate workflows for certain modifications before receiving user input for those modifications. Based on the pre-processing, the scene-based image editing system facilitates user interactions with the digital image as if it were a real scene reflecting real-world conditions. For instance, the scene-based image editing system enables user interactions that target pre-processed semantic areas (e.g., objects that have been identified and/or masked via pre-processing) as distinct components for editing rather than target the individual underlying pixels. Further, the scene-based image editing system automatically modifies the digital image to consistently reflect the corresponding real-world conditions.
(54) As indicated above, in one or more embodiments, the scene-based image editing system utilizes machine learning to process a digital image in anticipation of future modifications. In particular, in some cases, the scene-based image editing system employs one or more machine learning models to perform preparatory operations that will facilitate subsequent modification. In

some embodiments, the scene-based image editing system performs the pre-processing automatically in response to receiving the digital image. For instance, in some implementations, the scene-based image editing system gathers data and/or initiates a workflow for editing the digital image before receiving user input for such edits. Thus, the scene-based image editing system allows user interactions to directly indicate intended edits to the digital image rather than the various preparatory steps often utilized for making those edits.
(55) As an example, in one or more embodiments, the scene-based image editing system pre-processes a digital image to facilitate object-aware modifications. In particular, in some embodiments, the scene-based image editing system pre-processes a digital image in anticipation of user input for manipulating one or more semantic areas of a digital image, such as user input for moving or deleting one or more objects within the digital image.
(56) To illustrate, in some instances, the scene-based image editing system utilizes a segmentation neural network to generate, for each object portrayed in a digital image, an object mask. In some cases, the scene-based image editing system utilizes a hole-filing model to generate, for each object (e.g., for each corresponding object mask), a content fill (e.g., an inpainting segment). In some implementations, the scene-based image editing system generates a completed background for the digital image by pre-filling object holes with the corresponding content fill. Accordingly, in one or more embodiments, the scene-based image editing system pre-processes the digital image in preparation for an object-aware modification, such as a move operation or a delete operation, by pre-generating object masks and/or content fills before receiving user input for such a modification.
(57) Thus, upon receiving one or more user inputs targeting an object of the digital image for an object-aware modification (e.g., a move operation or a delete operation), the scene-based image editing system leverages the corresponding pre-generated object mask and/or content fill to complete the modification. For instance, in some cases, the scene-based image editing system detects, via a graphical user interface displaying the digital image, a user interaction with an object portrayed therein (e.g., a user selection of the object). In response to the user interaction, the scene-based image editing system surfaces the corresponding object mask that was previously generated. The scene-based image editing system further detects, via the graphical user interface, a second user interaction with the object (e.g., with the surfaced object mask) for moving or deleting the object. Accordingly, the moves or deletes the object, revealing the content fill previously positioned behind the object.
(58) Additionally, in one or more embodiments, the scene-based image editing system pre-processes a digital image to generate a semantic scene graph for the digital image. In particular, in some embodiments, the scene-based image editing system generates a semantic scene graph to map out various characteristics of the digital image. For instance, in some cases, the scene-based image editing system generates a semantic scene graph that describes the objects portrayed in the digital image, the relationships or object attributes of those objects, and/or various other characteristics determined to be useable for subsequent modification of the digital image.
(59) In some cases, the scene-based image editing system utilizes one or more machine learning models to determine the characteristics of the digital image to be included in the semantic scene graph. Further, in some instances, the scene-based image editing system generates the semantic scene graph utilizing one or more predetermined or pre-generated template graphs. For instance, in some embodiments, the scene-based image editing system utilizes an image analysis graph, a real-world class description graph, and/or a behavioral policy graph in generating the semantic scene.
(60) Thus, in some cases, the scene-based image editing system uses the semantic scene graph generated for a digital image to facilitate modification of the digital image. For instance, in some embodiments, upon determining that an object has been selected for modification, the scene-based image editing system retrieves characteristics of the object from the semantic scene graph to facilitate the modification. To illustrate, in some implementations, the scene-based image editing system executes or suggests one or more additional modifications to the digital image based on the

characteristics from the semantic scene graph.
(61) As one example, in some embodiments, upon determining that an object has been selected for modification, the scene-based image editing system provides one or more object attributes of the object for display via the graphical user interface displaying the object. For instance, in some cases, the scene-based image editing system retrieves a set of object attributes for the object (e.g., size, shape, or color) from the corresponding semantic scene graph and presents the set of object attributes for display in association with the object.
(62) In some cases, the scene-based image editing system further facilitates user interactivity with the displayed set of object attributes for modifying one or more of the object attributes. For instance, in some embodiments, the scene-based image editing system enables user interactions that change the text of the displayed set of object attributes or select from a provided set of object attribute alternatives. Based on the user interactions, the scene-based image editing system modifies the digital image by modifying the one or more object attributes in accordance with the user interactions.
(63) As another example, in some implementations, the scene-based image editing system utilizes a semantic scene graph to implement relationship-aware object modifications. To illustrate, in some cases, the scene-based image editing system detects a user interaction selecting an object portrayed in a digital image for modification. The scene-based image editing system references the semantic scene graph previously generated for the digital image to identify a relationship between that object and one or more other objects portrayed in the digital image. Based on the identified relationships, the scene-based image editing system also targets the one or more related objects for the modification.
(64) For instance, in some cases, the scene-based image editing system automatically adds the one or more related objects to the user selection. In some instances, the scene-based image editing system provides a suggestion that the one or more related objects be included in the user selection and adds the one or more related objects based on an acceptance of the suggestion. Thus, in some embodiments, the scene-based image editing system modifies the one or more related objects as it modifies the user-selected object.
(65) In one or more embodiments, in addition to pre-processing a digital image to identify objects portrayed as well as their relationships and/or object attributes, the scene-based image editing system further pre-processes a digital image to aid in the removal of distracting objects. For example, in some cases, the scene-based image editing system utilizes a distractor detection neural network to classify one or more objects portrayed in a digital image as subjects of the digital image and/or classify one or more other objects portrayed in the digital image as distracting objects. In some embodiments, the scene-based image editing system provides a visual indication of the distracting objects within a display of the digital image, suggesting that these objects be removed to present a more aesthetic and cohesive visual result.
(66) Further, in some cases, the scene-based image editing system detects the shadows of distracting objects (or other selected objects) for removal along with the distracting objects. In particular, in some cases, the scene-based image editing system utilizes a shadow detection neural network to identify shadows portrayed in the digital image and associate those shadows with their corresponding objects. Accordingly, upon removal of a distracting object from a digital image, the scene-based image editing system further removes the associated shadow automatically.
(67) The scene-based image editing system provides advantages over conventional systems. Indeed, conventional image editing systems suffer from several technological shortcomings that result in inflexible and inefficient operation. To illustrate, conventional systems are typically inflexible in that they rigidly perform edits on a digital image on the pixel level. In particular, conventional systems often perform a particular edit by targeting pixels individually for the edit. Accordingly, such systems often rigidly require user interactions for editing a digital image to interact with individual pixels to indicate the areas for the edit. Additionally, many conventional systems (e.g.,

due to their pixel-based editing) require users to have a significant amount of deep, specialized knowledge in how to interact with digital images, as well as the user interface of the system itself, to select the desired pixels and execute the appropriate workflow to edit those pixels.

(68) Additionally, conventional image editing systems often fail to operate efficiently. For example, conventional systems typically require a significant amount of user interaction to modify a digital image. Indeed, in addition to user interactions for selecting individual pixels, conventional systems typically require a user to interact with multiple menus, sub-menus, and/or windows to perform the edit. For instance, many edits may require multiple editing steps using multiple different tools. Accordingly, many conventional systems require multiple interactions to select the proper tool at a given editing step, set the desired parameters for the tool, and utilize the tool to execute the editing step.

(69) The scene-based image editing system operates with improved flexibility when compared to conventional systems. In particular, the scene-based image editing system implements techniques that facilitate flexible scene-based editing. For instance, by pre-processing a digital image via machine learning, the scene-based image editing system allows a digital image to be edited as if it were a real scene, in which various elements of the scene are known and are able to be interacted with intuitively on the semantic level to perform an edit while continuously reflecting real-world conditions. Indeed, where pixels are the targeted units under many conventional systems and objects are generally treated as groups of pixels, the scene-based image editing system allows user interactions to treat whole semantic areas (e.g., objects) as distinct units. Further, where conventional systems often require deep, specialized knowledge of the tools and workflows needed to perform edits, the scene-based editing system offers a more intuitive editing experience that enables a user to focus on the end goal of the edit.

(70) Further, the scene-based image editing system operates with improved efficiency when compared to conventional systems. In particular, the scene-based image editing system implements a graphical user interface that reduces the user interactions required for editing. Indeed, by pre-processing a digital image in anticipation of edits, the scene-based image editing system reduces the user interactions that are required to perform an edit. Specifically, the scene-based image editing system performs many of the operations required for an edit without relying on user instructions to perform those operations. Thus, in many cases, the scene-based image editing system reduces the user interactions typically required under conventional systems to select pixels to target for editing and to navigate menus, sub-menus, or other windows to select a tool, select its corresponding parameters, and apply the tool to perform the edit. By implementing a graphical user interface that reduces and simplifies user interactions needed for editing a digital image, the scene-based image editing system offers improved user experiences on computing devices—such as tablets or smart phone devices—having relatively limited screen space.

(71) Additional detail regarding the scene-based image editing system will now be provided with reference to the figures. For example, FIG. **1** illustrates a schematic diagram of an exemplary system **100** in which a scene-based image editing system **106** operates. As illustrated in FIG. **1**, the system **100** includes a server(s) **102**, a network **108**, and client devices **110***a*-**110***n*.

(72) Although the system **100** of FIG. **1** is depicted as having a particular number of components, the system **100** is capable of having any number of additional or alternative components (e.g., any number of servers, client devices, or other components in communication with the scene-based image editing system **106** via the network **108**). Similarly, although FIG. **1** illustrates a particular arrangement of the server(s) **102**, the network **108**, and the client devices **110***a*-**110***n*, various additional arrangements are possible.

(73) The server(s) **102**, the network **108**, and the client devices **110***a*-**110***n* are communicatively coupled with each other either directly or indirectly (e.g., through the network **108** discussed in greater detail below in relation to FIG. **51**). Moreover, the server(s) **102** and the client devices **110***a*-**110***n* include one or more of a variety of computing devices (including one or more

computing devices as discussed in greater detail with relation to FIG. **51**).

(74) As mentioned above, the system **100** includes the server(s) **102**. In one or more embodiments, the server(s) **102** generates, stores, receives, and/or transmits data including digital images and modified digital images. In one or more embodiments, the server(s) **102** comprises a data server. In some implementations, the server(s) **102** comprises a communication server or a web-hosting server.

(75) In one or more embodiments, the image editing system **104** provides functionality by which a client device (e.g., a user of one of the client devices **110***a*-**110***n*) generates, edits, manages, and/or stores digital images. For example, in some instances, a client device sends a digital image to the image editing system **104** hosted on the server(s) **102** via the network **108**. The image editing system **104** then provides options that the client device may use to edit the digital image, store the digital image, and subsequently search for, access, and view the digital image. For instance, in some cases, the image editing system **104** provides one or more options that the client device may use to modify objects within a digital image.

(76) In one or more embodiments, the client devices **110***a*-**110***n* include computing devices that access, view, modify, store, and/or provide, for display, digital images. For example, the client devices **110***a*-**110***n* include smartphones, tablets, desktop computers, laptop computers, head-mounted-display devices, or other electronic devices. The client devices **110***a*-**110***n* include one or more applications (e.g., the client application **112**) that can access, view, modify, store, and/or provide, for display, digital images. For example, in one or more embodiments, the client application **112** includes a software application installed on the client devices **110***a*-**110***n*. Additionally, or alternatively, the client application **112** includes a web browser or other application that accesses a software application hosted on the server(s) **102** (and supported by the image editing system **104**).

(77) To provide an example implementation, in some embodiments, the scene-based image editing system **106** on the server(s) **102** supports the scene-based image editing system **106** on the client device **110***n*. For instance, in some cases, the scene-based image editing system **106** on the server(s) **102** learns parameters for a neural network(s) **114** for analyzing and/or modifying digital images. The scene-based image editing system **106** then, via the server(s) **102**, provides the neural network(s) **114** to the client device **110***n*. In other words, the client device **110***n* obtains (e.g., downloads) the neural network(s) **114** with the learned parameters from the server(s) **102**. Once downloaded, the scene-based image editing system **106** on the client device **110***n* utilizes the neural network(s) **114** to analyze and/or modify digital images independent from the server(s) **102**.

(78) In alternative implementations, the scene-based image editing system **106** includes a web hosting application that allows the client device **110***n* to interact with content and services hosted on the server(s) **102**. To illustrate, in one or more implementations, the client device **110***n* accesses a software application supported by the server(s) **102**. In response, the scene-based image editing system **106** on the server(s) **102** modifies digital images. The server(s) **102** then provides the modified digital images to the client device **110***n* for display.

(79) Indeed, the scene-based image editing system **106** is able to be implemented in whole, or in part, by the individual elements of the system **100**. Indeed, although FIG. **1** illustrates the scene-based image editing system **106** implemented with regard to the server(s) **102**, different components of the scene-based image editing system **106** are able to be implemented by a variety of devices within the system **100**. For example, one or more (or all) components of the scene-based image editing system **106** are implemented by a different computing device (e.g., one of the client devices **110***a*-**110***n*) or a separate server from the server(s) **102** hosting the image editing system **104**. Indeed, as shown in FIG. **1**, the client devices **110***a*-**110***n* include the scene-based image editing system **106**. Example components of the scene-based image editing system **106** will be described below with regard to FIG. **44**.

(80) As mentioned, in one or more embodiments, the scene-based image editing system **106**

manages a two-dimensional digital image as a real scene reflecting real-world conditions. In particular, the scene-based image editing system **106** implements a graphical use interface that facilitates the modification of a digital image as a real scene. FIG. **2** illustrates an overview diagram of the scene-based image editing system **106** managing a digital image as a real scene in accordance with one or more embodiments.

(81) As shown in FIG. **2**, the scene-based image editing system **106** provides a graphical user interface **202** for display on a client device **204**. As further shown, the scene-based image editing system **106** provides, for display within the graphical user interface **202**, a digital image **206**. In one or more embodiments, the scene-based image editing system **106** provides the digital image **206** for display after the digital image **206** is captured via a camera of the client device **204**. In some instances, the scene-based image editing system **106** receives the digital image **206** from another computing device or otherwise accesses the digital image **206** at some storage location, whether local or remote.

(82) As illustrated in FIG. **2**, the digital image **206** portrays various objects. In one or more embodiments, an object includes a distinct visual component portrayed in a digital image. In particular, in some embodiments, an object includes a distinct visual element that is identifiable separately from other visual elements portrayed in a digital image. In many instances, an object includes a group of pixels that, together, portray the distinct visual element separately from the portrayal of other pixels. An object refers to a visual representation of a subject, concept, or sub-concept in an image. In particular, an object refers to a set of pixels in an image that combine to form a visual depiction of an item, article, partial item, component, or element. In some cases, an object is identifiable via various levels of abstraction. In other words, in some instances, an object includes separate object components that are identifiable individually or as part of an aggregate. To illustrate, in some embodiments, an object includes a semantic area (e.g., the sky, the ground, water, etc.). In some embodiments, an object comprises an instance of an identifiable thing (e.g., a person, an animal, a building, a car, or a cloud, clothing, or some other accessory). In one or more embodiments, an object includes sub-objects, parts, or portions. For example, a person's face, hair, or leg can be objects that are part of another object (e.g., the person's body). In still further implementations, a shadow or a reflection comprises part of an object. As another example, a shirt is an object that can be part of another object (e.g., a person).

(83) As shown in FIG. **2**, the digital image **206** portrays a static, two-dimensional image. In particular, the digital image **206** portrays a two-dimensional projection of a scene that was captured from the perspective of a camera. Accordingly, the digital image **206** reflects the conditions (e.g., the lighting, the surrounding environment, or the physics to which the portrayed objects are subject) under which the image was captured; however, it does so statically. In other words, the conditions are not inherently maintained when changes to the digital image **206** are made. Under many conventional systems, additional user interactions are required to maintain consistency with respect to those conditions when editing a digital image.

(84) Further, the digital image **206** includes a plurality of individual pixels that collectively portray various semantic areas. For instance, the digital image **206** portrays a plurality of objects, such as the objects **208***a*-**208***c*. While the pixels of each object are contributing to the portrayal of a cohesive visual unit, they are not typically treated as such. Indeed, a pixel of a digital image is typically inherently treated as an individual unit with its own values (e.g., color values) that are modifiable separately from the values of other pixels. Accordingly, conventional systems typically require user interactions to target pixels individually for modification when making changes to a digital image.

(85) As illustrated in FIG. **2**, however, the scene-based image editing system **106** manages the digital image **206** as a real scene, consistently maintaining the conditions under which the image was captured when modifying the digital image. In particular, the scene-based image editing system **106** maintains the conditions automatically without relying on user input to reflect those conditions.

Further, the scene-based image editing system **106** manages the digital image **206** on a semantic level. In other words, the digital image **206** manages each semantic area portrayed in the digital image **206** as a cohesive unit. For instance, as shown in FIG. **2** and as will be discussed, rather than requiring a user interaction to select the underlying pixels in order to interact with a corresponding object, the scene-based image editing system **106** enables user input to target the object as a unit and the scene-based image editing system **106** automatically recognizes the pixels that are associated with that object.

(86) To illustrate, as shown in FIG. **2**, in some cases, the scene-based image editing system **106** operates on a computing device **200** (e.g., the client device **204** or a separate computing device, such as the server(s) **102** discussed above with reference to FIG. **1**) to pre-process the digital image **206**. In particular, the scene-based image editing system **106** performs one or more pre-processing operations in anticipation of future modification to the digital image. In one or more embodiments, the scene-based image editing system **106** performs these pre-processing operations automatically in response to receiving or accessing the digital image **206** before user input for making the anticipated modifications have been received. As further shown, the scene-based image editing system **106** utilizes one or more machine learning models, such as the neural network(s) **114** to perform the pre-processing operations.

(87) In one or more embodiments, the scene-based image editing system **106** pre-processes the digital image **206** by learning characteristics of the digital image **206**. For instance, in some cases, the scene-based image editing system **106** segments the digital image **206**, identifies objects, classifies objects, determines relationships and/or attributes of objects, determines lighting characteristics, and/or determines depth/perspective characteristics. In some embodiments, the scene-based image editing system **106** pre-processes the digital image **206** by generating content for use in modifying the digital image **206**. For example, in some implementations, the scene-based image editing system **106** generates an object mask for each portrayed object and/or generates a content fill for filling in the background behind each portrayed object. Background refers to what is behind an object in an image. Thus, when a first object is positioned in front of a second object, the second object forms at least part of the background for the first object. Alternatively, the background comprises the furthest element in the image (often a semantic area like the sky, ground, water, etc.). The background for an object, in or more embodiments, comprises multiple object/semantic areas. For example, the background for an object can comprise part of another object and part of the furthest element in the image. The various pre-processing operations and their use in modifying a digital image will be discussed in more detail below with reference to the subsequent figures.

(88) As shown in FIG. **2**, the scene-based image editing system **106** detects, via the graphical user interface **202**, a user interaction with the object **208***c*. In particular, the scene-based image editing system **106** detects a user interaction for selecting the object **208***c*. Indeed, in one or more embodiments, the scene-based image editing system **106** determines that the user interaction targets the object even where the user interaction only interacts with a subset of the pixels that contribute to the object **208***c* based on the pre-processing of the digital image **206**. For instance, as mentioned, the scene-based image editing system **106** pre-processes the digital image **206** via segmentation in some embodiments. As such, at the time of detecting the user interaction, the scene-based image editing system **106** has already partitioned/segmented the digital image **206** into its various semantic areas. Thus, in some instances, the scene-based image editing system **106** determines that the user interaction selects a distinct semantic area (e.g., the object **208***c*) rather than the particular underlying pixels or image layers with which the user interacted.

(89) As further shown in FIG. **2**, the scene-based image editing system **106** modifies the digital image **206** via a modification to the object **208***c*. Though FIG. **2** illustrates a deletion of the object **208***c*, various modifications are possible and will be discussed in more detail below. In some embodiments, the scene-based image editing system **106** edits the object **208***c* in response to

detecting a second user interaction for performing the modification.
(90) As illustrated, upon deleting the object **208***c* from the digital image **206**, the scene-based image editing system **106** automatically reveals background pixels that have been positioned in place of the object **208***c*. Indeed, as mentioned, in some embodiments, the scene-based image editing system **106** pre-processes the digital image **206** by generating a content fill for each portrayed foreground object. Thus, as indicated by FIG. **2**, the scene-based image editing system **106** automatically exposes the content fill **210** previously generated for the object **208***c* upon removal of the object **208***c* from the digital image **206**. In some instances, the scene-based image editing system **106** positions the content fill **210** within the digital image so that the content fill **210** is exposed rather than a hole appearing upon removal of object **208***c*.
(91) Thus, the scene-based image editing system **106** operates with improved flexibility when compared to many conventional systems. In particular, the scene-based image editing system **106** implements flexible scene-based editing techniques in which digital images are modified as real scenes that maintain real-world conditions (e.g., physics, environment, or object relationships). Indeed, in the example shown in FIG. **2**, the scene-based image editing system **106** utilizes pre-generated content fills to consistently maintain the background environment portrayed in the digital image **206** as though the digital image **206** had captured that background in its entirety. Thus, the scene-based image editing system **106** enables the portrayed objects to be moved around freely (or removed entirely) without disrupting the scene portrayed therein.
(92) Further, the scene-based image editing system **106** operates with improved efficiency. Indeed, by segmenting the digital image **206** and generating the content fill **210** in anticipation of a modification that would remove the object **208***c* from its position in the digital image **206**, the scene-based image editing system **106** reduces the user interactions that are typically required to perform those same operations under conventional systems. Thus, the scene-based image editing system **106** enables the same modifications to a digital image with less user interactions when compared to these conventional systems.
(93) As just discussed, in one or more embodiments, the scene-based image editing system **106** implements object-aware image editing on digital images. In particular, the scene-based image editing system **106** implements object-aware modifications that target objects as cohesive units that are interactable and can be modified. FIGS. **3-9**B illustrate the scene-based image editing system **106** implementing object-aware modifications in accordance with one or more embodiments.
(94) Indeed, many conventional image editing systems are inflexible and inefficient with respect to interacting with objects portrayed in a digital image. For instance, as previously mentioned, conventional systems are often rigid in that they require user interactions to target pixels individually rather than the objects that those pixels portray. Thus, such systems often require a rigid, meticulous process of selecting pixels for modification. Further, as object identification occurs via user selection, these systems typically fail to anticipate and prepare for potential edits made to those objects.
(95) Further, many conventional image editing systems require a significant amount of user interactions to modify objects portrayed in a digital image. Indeed, in addition to the pixel-selection process for identifying objects in a digital image—which can require a series of user interactions on its own—conventional systems may require workflows of significant length in which a user interacts with multiple menus, sub-menus, tool, and/or windows to perform the edit. Often, performing an edit on an object requires multiple preparatory steps before the desired edit is able to be executed, requiring additional user interactions.
(96) The scene-based image editing system **106** provides advantages over these systems. For instance, the scene-based image editing system **106** offers improved flexibility via object-aware image editing. In particular, the scene-based image editing system **106** enables object-level—rather than pixel-level or layer level—interactions, facilitating user interactions that target portrayed objects directly as cohesive units instead of their constituent pixels individually.

(97) Further, the scene-based image editing system **106** improves the efficiency of interacting with objects portrayed in a digital image. Indeed, previously mentioned, and as will be discussed further below, the scene-based image editing system **106** implements pre-processing operations for identifying and/or segmenting for portrayed objects in anticipation of modifications to those objects. Indeed, in many instances, the scene-based image editing system **106** performs these pre-processing operations without receiving user interactions for those modifications. Thus, the scene-based image editing system **106** reduces the user interactions that are required to execute a given edit on a portrayed object.
(98) In some embodiments, the scene-based image editing system **106** implements object-aware image editing by generating an object mask for each object/semantic area portrayed in a digital image. In particular, in some cases, the scene-based image editing system **106** utilizes a machine learning model, such as a segmentation neural network, to generate the object mask(s). FIG. **3** illustrates a segmentation neural network utilized by the scene-based image editing system **106** to generate object masks for objects in accordance with one or more embodiments.
(99) In one or more embodiments, an object mask includes a map of a digital image that has an indication for each pixel of whether the pixel corresponds to part of an object (or other semantic area) or not. In some implementations, the indication includes a binary indication (e.g., a "1" for pixels belonging to the object and a "0" for pixels not belonging to the object). In alternative implementations, the indication includes a probability (e.g., a number between 1 and 0) that indicates the likelihood that a pixel belongs to an object. In such implementations, the closer the value is to 1, the more likely the pixel belongs to an object and vice versa.
(100) In one or more embodiments, a machine learning model includes a computer representation that is tunable (e.g., trained) based on inputs to approximate unknown functions used for generating the corresponding outputs. In particular, in some embodiments, a machine learning model includes a computer-implemented model that utilizes algorithms to learn from, and make predictions on, known data by analyzing the known data to learn to generate outputs that reflect patterns and attributes of the known data. For instance, in some instances, a machine learning model includes, but is not limited to a neural network (e.g., a convolutional neural network, recurrent neural network or other deep learning network), a decision tree (e.g., a gradient boosted decision tree), association rule learning, inductive logic programming, support vector learning, Bayesian network, regression-based model (e.g., censored regression), principal component analysis, or a combination thereof.
(101) In one or more embodiments, a neural network includes a model of interconnected artificial neurons (e.g., organized in layers) that communicate and learn to approximate complex functions and generate outputs based on a plurality of inputs provided to the model. In some instances, a neural network includes one or more machine learning algorithms. Further, in some cases, a neural network includes an algorithm (or set of algorithms) that implements deep learning techniques that utilize a set of algorithms to model high-level abstractions in data. To illustrate, in some embodiments, a neural network includes a convolutional neural network, a recurrent neural network (e.g., a long short-term memory neural network), a generative adversarial neural network, a graph neural network, or a multi-layer perceptron. In some embodiments, a neural network includes a combination of neural networks or neural network components.
(102) In one or more embodiments, a segmentation neural network includes a computer-implemented neural network that generates object masks for objects portrayed in digital images. In particular, in some embodiments, a segmentation neural network includes a computer-implemented neural network that detects objects within digital images and generates object masks for the objects. Indeed, in some implementations, a segmentation neural network includes a neural network pipeline that analyzes a digital image, identifies one or more objects portrayed in the digital image, and generates an object mask for the one or more objects. In some cases, however, a segmentation neural network focuses on a subset of tasks for generating an object mask.

(103) As mentioned, FIG. **3** illustrates one example of a segmentation neural network that the scene-based image editing system **106** utilizes in one or more implementations to generate object masks for objects portrayed in a digital image. In particular, FIG. **3** illustrates one example of a segmentation neural network used by the scene-based image editing system **106** in some embodiments to both detect objects in a digital image and generate object masks for those objects. Indeed, FIG. **3** illustrates a detection-masking neural network **300** that comprises both an object detection machine learning model **308** (in the form of an object detection neural network) and an object segmentation machine learning model **310** (in the form of an object segmentation neural network). Specifically, the detection-masking neural network **300** is an implementation of the on-device masking system described in U.S. patent application Ser. No. 17/589,114, "DETECTING DIGITAL OBJECTS AND GENERATING OBJECT MASKS ON DEVICE," filed on Jan. 31, 2022, the entire contents of which are hereby incorporated by reference.
(104) Although FIG. **3** illustrates the scene-based image editing system **106** utilizing the detection-masking neural network **300**, in one or more implementations, the scene-based image editing system **106** utilizes different machine learning models to detect objects, generate object masks for objects, and/or extract objects from digital images. For instance, in one or more implementations, the scene-based image editing system **106** utilizes, as the segmentation neural network (or as an alternative to a segmentation neural network), one of the machine learning models or neural networks described in U.S. patent application Ser. No. 17/158,527, entitled "Segmenting Objects In Digital Images Utilizing A Multi-Object Segmentation Model Framework," filed on Jan. 26, 2021; or U.S. patent application Ser. No. 16/388,115, entitled "Robust Training of Large-Scale Object Detectors with Noisy Data," filed on Apr. 8, 2019; or U.S. patent application Ser. No. 16/518,880, entitled "Utilizing Multiple Object Segmentation Models To Automatically Select User-Requested Objects In Images," filed on Jul. 22, 2019; or U.S. patent application Ser. No. 16/817,418, entitled "Utilizing A Large-Scale Object Detector To Automatically Select Objects In Digital Images," filed on Mar. 20, 2020; or Ren, et al., *Faster r-cnn: Towards real-time object detection with region proposal networks*, NIPS, 2015; or Redmon, et al., *You Only Look Once: Unified, Real-Time Object Detection*, CVPR 2016, the contents of each of the foregoing applications and papers are hereby incorporated by reference in their entirety.
(105) Similarly, in one or more implementations, the scene-based image editing system **106** utilizes, as the segmentation neural network (or as an alternative to a segmentation neural network), one of the machine learning models or neural networks described in Ning Xu et al., "Deep GrabCut for Object Selection," published Jul. 14, 2017; or U.S. Patent Application Publication No. 2019/0130229, entitled "Deep Salient Content Neural Networks for Efficient Digital Object Segmentation," filed on Oct. 31, 2017; or U.S. patent application Ser. No. 16/035,410, entitled "Automatic Trimap Generation and Image Segmentation," filed on Jul. 13, 2018; or U.S. Pat. No. 10,192,129, entitled "Utilizing Interactive Deep Learning To Select Objects In Digital Visual Media," filed Nov. 18, 2015, each of which are incorporated herein by reference in their entirety.
(106) In one or more implementations the segmentation neural network is a panoptic segmentation neural network. In other words, the segmentation neural network creates object mask for individual instances of a given object type. Furthermore, the segmentation neural network, in one or more implementations, generates object masks for semantic regions (e.g., water, sky, sand, dirt, etc.) in addition to countable things. Indeed, in one or more implementations, the scene-based image editing system **106** utilizes, as the segmentation neural network (or as an alternative to a segmentation neural network), one of the machine learning models or neural networks described in U.S. patent application Ser. No. 17/495,618, entitled "PANOPTIC SEGMENTATION REFINEMENT NETWORK," filed on Oct. 2, 2021; or U.S. patent application Ser. No. 17/454,740, entitled "MULTI-SOURCE PANOPTIC FEATURE PYRAMID NETWORK," filed on Nov. 12, 2021, each of which are incorporated herein by reference in their entirety.
(107) Returning now to FIG. **3**, in one or more implementations, the scene-based image editing

system **106** utilizes a detection-masking neural network **300** that includes an encoder **302** (or neural network encoder) having a backbone network, detection heads **304** (or neural network decoder head), and a masking head **306** (or neural network decoder head). As shown in FIG. **3**, the encoder **302** encodes a digital image **316** and provides the encodings to the detection heads **304** and the masking head **306**. The detection heads **304** utilize the encodings to detect one or more objects portrayed in the digital image **316**. The masking head **306** generates at least one object mask for the detected objects.

(108) As just mentioned, the detection-masking neural network **300** utilizes both the object detection machine learning model **308** and the object segmentation machine learning model **310**. In one or more implementations, the object detection machine learning model **308** includes both the encoder **302** and the detection heads **304** shown in FIG. **3**. While the object segmentation machine learning model **310** includes both the encoder **302** and the masking head **306**.

(109) Furthermore, the object detection machine learning model **308** and the object segmentation machine learning model **310** are separate machine learning models for processing objects within target and/or source digital images. FIG. **3** illustrates the encoder **302**, detection heads **304**, and the masking head **306** as a single model for detecting and segmenting objects of a digital image. For efficiency purposes, in some embodiments the scene-based image editing system **106** utilizes the network illustrated in FIG. **3** as a single network. The collective network (i.e., the object detection machine learning model **308** and the object segmentation machine learning model **310**) is referred to as the detection-masking neural network **300**. The following paragraphs describe components relating to the object detection machine learning model **308** of the network (such as the detection heads **304**) and transitions to discussing components relating to the object segmentation machine learning model **310**.

(110) As just mentioned, in one or more embodiments, the scene-based image editing system **106** utilizes the object detection machine learning model **308** to detect and identify objects within the digital image **316** (e.g., a target or a source digital image). FIG. **3** illustrates one implementation of the object detection machine learning model **308** that the scene-based image editing system **106** utilizes in accordance with at least one embodiment. In particular, FIG. **3** illustrates the scene-based image editing system **106** utilizing the object detection machine learning model **308** to detect objects. In one or more embodiments, the object detection machine learning model **308** comprises a deep learning convolutional neural network (CNN). For example, in some embodiments, the object detection machine learning model **308** comprises a region-based (R-CNN).

(111) As shown in FIG. **3**, the object detection machine learning model **308** includes lower neural network layers and higher neural network layers. In general, the lower neural network layers collectively form the encoder **302** and the higher neural network layers collectively form the detection heads **304** (e.g., decoder). In one or more embodiments, the encoder **302** includes convolutional layers that encodes a digital image into feature vectors, which are outputted from the encoder **302** and provided as input to the detection heads **304**. In various implementations, the detection heads **304** comprise fully connected layers that analyze the feature vectors and output the detected objects (potentially with approximate boundaries around the objects).

(112) In particular, the encoder **302**, in one or more implementations, comprises convolutional layers that generate a feature vector in the form of a feature map. To detect objects within the digital image **316**, the object detection machine learning model **308** processes the feature map utilizing a convolutional layer in the form of a small network that is slid across small windows of the feature map. The object detection machine learning model **308** further maps each sliding window to a lower-dimensional feature. In one or more embodiments, the object detection machine learning model **308** processes this feature using two separate detection heads that are fully connected layers. In some embodiments, the first head comprises a box-regression layer that generates the detected object and an object-classification layer that generates the object label.

(113) As shown by FIG. **3**, the output from the detection heads **304** shows object labels above each

of the detected objects. For example, the detection-masking neural network **300**, in response to detecting objects, assigns an object label to each of the detected objects. In particular, in some embodiments, the detection-masking neural network **300** utilizes object labels based on classifications of the objects. To illustrate, FIG. **3** shows a label **318** for woman, a label **320** for bird, and a label **322** for man. Though not shown in FIG. **3**, the detection-masking neural network **300** further distinguishes between the woman and the surfboard held by the woman in some implementations. Additionally, the detection-masking neural network **300** optionally also generates object masks for the semantic regions shown (e.g., the sand, the sea, and the sky).

(114) As mentioned, the object detection machine learning model **308** detects the objects within the digital image **316**. In some embodiments, and as illustrated in FIG. **3**, the detection-masking neural network **300** indicates the detected objects utilizing approximate boundaries (e.g., bounding boxes **319**, **321**, and **323**). For example, each of the bounding boxes comprises an area that encompasses an object. In some embodiments, the detection-masking neural network **300** annotates the bounding boxes with the previously mentioned object labels such as the name of the detected object, the coordinates of the bounding box, and/or the dimension of the bounding box.

(115) As illustrated in FIG. **3**, the object detection machine learning model **308** detects several objects for the digital image **316**. In some instances, the detection-masking neural network **300** identifies all objects within the bounding boxes. In one or more embodiments, the bounding boxes comprise the approximate boundary area indicating the detected object. In some cases, an approximate boundary refers to an indication of an area including an object that is larger and/or less accurate than an object mask. In one or more embodiments, an approximate boundary includes at least a portion of a detected object and portions of the digital image **316** not comprising the detected object. An approximate boundary includes various shape, such as a square, rectangle, circle, oval, or other outline surrounding an object. In one or more embodiments, an approximate boundary comprises a bounding box.

(116) Upon detecting the objects in the digital image **316**, the detection-masking neural network **300** generates object masks for the detected objects. Generally, instead of utilizing coarse bounding boxes during object localization, the detection-masking neural network **300** generates segmentations masks that better define the boundaries of the object. The following paragraphs provide additional detail with respect to generating object masks for detected objects in accordance with one or more embodiments. In particular, FIG. **3** illustrates the scene-based image editing system **106** utilizing the object segmentation machine learning model **310** to generate segmented objects via object masks in accordance with some embodiments.

(117) As illustrated in FIG. **3**, the scene-based image editing system **106** processes a detected object in a bounding box utilizing the object segmentation machine learning model **310** to generate an object mask, such as an object mask **324** and an object mask **326**. In alternative embodiments, the scene-based image editing system **106** utilizes the object detection machine learning model **308** itself to generate an object mask of the detected object (e.g., segment the object for selection).

(118) In one or more implementations, prior to generating an object mask of a detected object, scene-based image editing system **106** receives user input **312** to determine objects for which to generate object masks. For example, the scene-based image editing system **106** receives input from a user indicating a selection of one of the detected objects. To illustrate, in the implementation shown, the scene-based image editing system **106** receives user input **312** of the user selecting bounding boxes **321** and **323**. In alternative implementations, the scene-based image editing system **106** generates objects masks for each object automatically (e.g., without a user request indicating an object to select).

(119) As mentioned, the scene-based image editing system **106** processes the bounding boxes of the detected objects in the digital image **316** utilizing the object segmentation machine learning model **310**. In some embodiments, the bounding box comprises the output from the object detection machine learning model **308**. For example, as illustrated in FIG. **3**, the bounding box comprises a

rectangular border about the object. Specifically, FIG. **3** shows bounding boxes **319**, **321** and **323** which surround the woman, the bird, and the man detected in the digital image **316**.

(120) In some embodiments, the scene-based image editing system **106** utilizes the object segmentation machine learning model **310** to generate the object masks for the aforementioned detected objects within the bounding boxes. For example, the object segmentation machine learning model **310** corresponds to one or more deep neural networks or models that select an object based on bounding box parameters corresponding to the object within the digital image **316**. In particular, the object segmentation machine learning model **310** generates the object mask **324** and the object mask **326** for the detected man and bird, respectively.

(121) In some embodiments, the scene-based image editing system **106** selects the object segmentation machine learning model **310** based on the object labels of the object identified by the object detection machine learning model **308**. Generally, based on identifying one or more classes of objects associated with the input bounding boxes, the scene-based image editing system **106** selects an object segmentation machine learning model tuned to generate object masks for objects of the identified one or more classes. To illustrate, in some embodiments, based on determining that the class of one or more of the identified objects comprises a human or person, the scene-based image editing system **106** utilizes a special human object mask neural network to generate an object mask, such as the object mask **324** shown in FIG. **3**.

(122) As further illustrated in FIG. **3**, the scene-based image editing system **106** receives the object mask **324** and the object mask **326** as output from the object segmentation machine learning model **310**. As previously discussed, in one or more embodiments, an object mask comprises a pixel-wise mask that corresponds to an object in a source or target digital image. In one example, an object mask includes a segmentation boundary indicating a predicted edge of one or more objects as well as pixels contained within the predicted edge.

(123) In some embodiments, the scene-based image editing system **106** also detects the objects shown in the digital image **316** via the collective network, i.e., the detection-masking neural network **300**, in the same manner outlined above. For example, in some cases, the scene-based image editing system **106**, via the detection-masking neural network **300** detects the woman, the man, and the bird within the digital image **316**. In particular, the scene-based image editing system **106**, via the detection heads **304**, utilizes the feature pyramids and feature maps to identify objects within the digital image **316** and generates object masks via the masking head **306**.

(124) Furthermore, in one or more implementations, although FIG. **3** illustrates generating object masks based on the user input **312**, the scene-based image editing system **106** generates object masks without user input **312**. In particular, the scene-based image editing system **106** generates object masks for all detected objects within the digital image **316**. To illustrate, in at least one implementation, despite not receiving the user input **312**, the scene-based image editing system **106** generates object masks for the woman, the man, and the bird.

(125) In one or more embodiments, the scene-based image editing system **106** implements object-aware image editing by generating a content fill for each object portrayed in a digital image (e.g., for each object mask corresponding to portrayed objects) utilizing a hole-filing model. In particular, in some cases, the scene-based image editing system **106** utilizes a machine learning model, such as a content-aware hole-filling machine learning model to generate the content fill(s) for each foreground object. FIGS. **4-6** illustrate a content-aware hole-filling machine learning model utilized by the scene-based image editing system **106** to generate content fills for objects in accordance with one or more embodiments.

(126) In one or more embodiments, a content fill includes a set of pixels generated to replace another set of pixels of a digital image. Indeed, in some embodiments, a content fill includes a set of replacement pixels for replacing another set of pixels. For instance, in some embodiments, a content fill includes a set of pixels generated to fill a hole (e.g., a content void) that remains after (or if) a set of pixels (e.g., a set of pixels portraying an object) has been removed from or moved

within a digital image. In some cases, a content fill corresponds to a background of a digital image. To illustrate, in some implementations, a content fill includes a set of pixels generated to blend in with a portion of a background proximate to an object that could be moved/removed. In some cases, a content fill includes an inpainting segment, such as an inpainting segment generated from other pixels (e.g., other background pixels) within the digital image. In some cases, a content fill includes other content (e.g., arbitrarily selected content or content selected by a user) to fill in a hole or replace another set of pixels.

(127) In one or more embodiments, a content-aware hole-filling machine learning model includes a computer-implemented machine learning model that generates content fill. In particular, in some embodiments, a content-aware hole-filling machine learning model includes a computer-implemented machine learning model that generates content fills for replacement regions in a digital image. For instance, in some cases, the scene-based image editing system **106** determines that an object has been moved within or removed from a digital image and utilizes a content-aware hole-filling machine learning model to generate a content fill for the hole that has been exposed as a result of the move/removal in response. As will be discussed in more detail, however, in some implementations, the scene-based image editing system **106** anticipates movement or removal of an object and utilizes a content-aware hole-filling machine learning model to pre-generate a content fill for that object. In some cases, a content-aware hole-filling machine learning model includes a neural network, such as an inpainting neural network (e.g., a neural network that generates a content fill—more specifically, an inpainting segment—using other pixels of the digital image). In other words, the scene-based image editing system **106** utilizes a content-aware hole-filling machine learning model in various implementations to provide content at a location of a digital image that does not initially portray such content (e.g., due to the location being occupied by another semantic area, such as an object).

(128) FIG. **4** illustrates the scene-based image editing system **106** utilizing a content-aware machine learning model, such as a cascaded modulation inpainting neural network **420**, to generate an inpainted digital image **408** from a digital image **402** with a replacement region **404** in accordance with one or more embodiments.

(129) Indeed, in one or more embodiments, the replacement region **404** includes an area corresponding to an object (and a hole that would be present if the object were moved or deleted). In some embodiments, the scene-based image editing system **106** identifies the replacement region **404** based on user selection of pixels (e.g., pixels portraying an object) to move, remove, cover, or replace from a digital image. To illustrate, in some cases, a client device selects an object portrayed in a digital image. Accordingly, the scene-based image editing system **106** deletes or removes the object and generates replacement pixels. In some case, the scene-based image editing system **106** identifies the replacement region **404** by generating an object mask via a segmentation neural network. For instance, the scene-based image editing system **106** utilizes a segmentation neural network (e.g., the detection-masking neural network **300** discussed above with reference to FIG. **3**) to detect objects with a digital image and generate object masks for the objects. Thus, in some implementations, the scene-based image editing system **106** generates content fill for the replacement region **404** before receiving user input to move, remove, cover, or replace the pixels initially occupying the replacement region **404**

(130) As shown, the scene-based image editing system **106** utilizes the cascaded modulation inpainting neural network **420** to generate replacement pixels for the replacement region **404**. In one or more embodiments, the cascaded modulation inpainting neural network **420** includes a generative adversarial neural network for generating replacement pixels. In some embodiments, a generative adversarial neural network (or "GAN") includes a neural network that is tuned or trained via an adversarial process to generate an output digital image (e.g., from an input digital image). In some cases, a generative adversarial neural network includes multiple constituent neural networks such as an encoder neural network and one or more decoder/generator neural networks. For

example, an encoder neural network extracts latent code from a noise vector or from a digital image. A generator neural network (or a combination of generator neural networks) generates a modified digital image by combining extracted latent code (e.g., from the encoder neural network). During training, a discriminator neural network, in competition with the generator neural network, analyzes a generated digital image to generate an authenticity prediction by determining whether the generated digital image is real (e.g., from a set of stored digital images) or fake (e.g., not from the set of stored digital images). The discriminator neural network also causes the scene-based image editing system **106** to modify parameters of the encoder neural network and/or the one or more generator neural networks to eventually generate digital images that fool the discriminator neural network into indicating that a generated digital image is a real digital image.
(131) Along these lines, a generative adversarial neural network refers to a neural network having a specific architecture or a specific purpose such as a generative inpainting neural network. For example, a generative inpainting neural network includes a generative adversarial neural network that inpaints or fills pixels of a digital image with a content fill (or generates a content fill in anticipation of inpainting or filling in pixels of the digital image). In some cases, a generative inpainting neural network inpaints a digital image by filling hole regions (indicated by object masks). Indeed, as mentioned above, in some embodiments an object mask defines a replacement region using a segmentation or a mask indicating, overlaying, covering, or outlining pixels to be removed or replaced within a digital image.
(132) Accordingly, in some embodiments, the cascaded modulation inpainting neural network **420** includes a generative inpainting neural network that utilizes a decoder having one or more cascaded modulation decoder layers. Indeed, as illustrated in FIG. **4**, the cascaded modulation inpainting neural network **420** includes a plurality of cascaded modulation decoder layers **410**, **412**, **414**, **416**. In some cases, a cascaded modulation decoder layer includes at least two connected (e.g., cascaded) modulations blocks for modulating an input signal in generating an inpainted digital image. To illustrate, in some instances, a cascaded modulation decoder layer includes a first global modulation block and a second global modulation block. Similarly, in some cases, a cascaded modulation decoder layer includes a first global modulation block (that analyzes global features and utilizes a global, spatially-invariant approach) and a second spatial modulation block (that analyzes local features utilizing a spatially-varying approach). Additional detail regarding modulation blocks will be provided below (e.g., in relation to FIGS. **5-6**).
(133) As shown, the scene-based image editing system **106** utilizes the cascaded modulation inpainting neural network **420** (and the cascaded modulation decoder layers **410**, **412**, **414**, **416**) to generate the inpainted digital image **408**. Specifically, the cascaded modulation inpainting neural network **420** generates the inpainted digital image **408** by generating a content fill for the replacement region **404**. As illustrated, the replacement region **404** is now filled with a content fill having replacement pixels that portray a photorealistic scene in place of the replacement region **404**.
(134) As mentioned above, the scene-based image editing system **106** utilizes a cascaded modulation inpainting neural network that includes cascaded modulation decoder layers to generate inpainted digital images. FIG. **5** illustrates an example architecture of a cascaded modulation inpainting neural network **502** in accordance with one or more embodiments.
(135) As illustrated, the cascaded modulation inpainting neural network **502** includes an encoder **504** and a decoder **506**. In particular, the encoder **504** includes a plurality of convolutional layers **508***a*-**508***n* at different scales/resolutions. In some cases, the scene-based image editing system **106** feeds the digital image input **510** (e.g., an encoding of the digital image) into the first convolutional layer **508***a* to generate an encoded feature vector at a higher scale (e.g., lower resolution). The second convolutional layer **508***b* processes the encoded feature vector at the higher scale (lower resolution) and generates an additional encoded feature vector (at yet another higher scale/lower resolution). The cascaded modulation inpainting neural network **502** iteratively generates these

encoded feature vectors until reaching the final/highest scale convolutional layer **508***n* and generating a final encoded feature vector representation of the digital image.
(136) As illustrated, in one or more embodiments, the cascaded modulation inpainting neural network **502** generates a global feature code from the final encoded feature vector of the encoder **504**. A global feature code includes a feature representation of the digital image from a global (e.g., high-level, high-scale, low-resolution) perspective. In particular, a global feature code includes a representation of the digital image that reflects an encoded feature vector at the highest scale/lowest resolution (or a different encoded feature vector that satisfies a threshold scale/resolution).
(137) As illustrated, in one or more embodiments, the cascaded modulation inpainting neural network **502** applies a neural network layer (e.g., a fully connected layer) to the final encoded feature vector to generate a style code **512** (e.g., a style vector). In addition, the cascaded modulation inpainting neural network **502** generates the global feature code by combining the style code **512** with a random style code **514**. In particular, the cascaded modulation inpainting neural network **502** generates the random style code **514** by utilizing a neural network layer (e.g., a multi-layer perceptron) to process an input noise vector. The neural network layer maps the input noise vector to a random style code **514**. The cascaded modulation inpainting neural network **502** combines (e.g., concatenates, adds, or multiplies) the random style code **514** with the style code **512** to generate the global feature code **516**. Although FIG. **5** illustrates a particular approach to generate the global feature code **516**, the scene-based image editing system **106** is able to utilize a variety of different approaches to generate a global feature code that represents encoded feature vectors of the encoder **504** (e.g., without the style code **512** and/or the random style code **514**).
(138) As mentioned above, in some embodiments, the cascaded modulation inpainting neural network **502** generates an image encoding utilizing the encoder **504**. An image encoding refers to an encoded representation of the digital image. Thus, in some cases, an image encoding includes one or more encoding feature vectors, a style code, and/or a global feature code.
(139) In one or more embodiments, the cascaded modulation inpainting neural network **502** utilizes a plurality of Fourier convolutional encoder layer to generate an image encoding (e.g., the encoded feature vectors, the style code **512**, and/or the global feature code **516**). For example, a Fourier convolutional encoder layer (or a fast Fourier convolution) comprises a convolutional layer that includes non-local receptive fields and cross-scale fusion within a convolutional unit. In particular, a fast Fourier convolution can include three kinds of computations in a single operation unit: a local branch that conducts small-kernel convolution, a semi-global branch that processes spectrally stacked image patches, and a global branch that manipulates image-level spectrum. These three branches complementarily address different scales. In addition, in some instances, a fast Fourier convolution includes a multi-branch aggregation process for cross-scale fusion. For example, in one or more embodiments, the cascaded modulation inpainting neural network **502** utilizes a fast Fourier convolutional layer as described by Lu Chi, Borui Jiang, and Yadong Mu in Fast Fourier convolution, Advances in Neural Information Processing Systems, 33 (2020), which is incorporated by reference herein in its entirety.
(140) Specifically, in one or more embodiments, the cascaded modulation inpainting neural network **502** utilizes Fourier convolutional encoder layers for each of the encoder convolutional layers **508***a*-**508***n*. Thus, the cascaded modulation inpainting neural network **502** utilizes different Fourier convolutional encoder layers having different scales/resolutions to generate encoded feature vectors with improved, non-local receptive field.
(141) Operation of the encoder **504** can also be described in terms of variables or equations to demonstrate functionality of the cascaded modulation inpainting neural network **502**. For instance, as mentioned, the cascaded modulation inpainting neural network **502** is an encoder-decoder network with proposed cascaded modulation blocks at its decoding stage for image inpainting. Specifically, the cascaded modulation inpainting neural network **502** starts with an encoder E that takes the partial image and the mask as inputs to produce multi-scale feature maps from input

resolution to resolution 4×4:

$F_e^{(1)}, \ldots, F_e^{(L)} = E(x \odot (1-m), m)$,

where F.sub.e.sup.(i) are the generated feature at scale 1≤i≤L (and L is the highest scale or resolution). The encoder is implemented by a set of stride-2 convolutions with residual connection.

(142) After generating the highest scale feature F.sub.e.sup.(L), a fully connected layer followed by a l.sub.2 normalization products a global style code s=fc(F.sub.e.sup.(L))/∥fc(F.sub.e.sup.(L))∥.sub.2 to represent the input globally. In parallel to the encoder, an MLP-based mapping network produces a random style code w from a normalized random Gaussian noise z, simulating the stochasticity of the generation process. Moreover, the scene-based image editing system **106** joins w with s to produce the final global code g=[s; w] for decoding. As mentioned, in some embodiments, the scene-based image editing system **106** utilizes the final global code as an image encoding for the digital image.

(143) As mentioned above, in some implementations, full convolutional models suffer from slow growth of effective receptive field, especially at the early stage of the network. Accordingly, utilizing strided convolution within the encoder can generate invalid features inside the hole region, making the feature correction at decoding stage more challenging. Fast Fourier convolution (FFC) can assist early layers to achieve receptive field that covers an entire image. Conventional systems, however, have only utilized FFC at a bottleneck layer, which is computationally demanding. Moreover, the shallow bottleneck layer cannot capture global semantic features effectively. Accordingly, in one or more implementations the scene-based image editing system **106** replaces the convolutional block in the encoder with FFC for the encoder layers. FFC enables the encoder to propagate features at early stage and thus address the issue of generating invalid features inside the hole, which helps improve the results.

(144) As further shown in FIG. **5**, the cascaded modulation inpainting neural network **502** also includes the decoder **506**. As shown, the decoder **506** includes a plurality of cascaded modulation layers **520***a*-**520***n*. The cascaded modulation layers **520***a*-**520***n* process input features (e.g., input global feature maps and input local feature maps) to generate new features (e.g., new global feature maps and new local feature maps). In particular, each of the cascaded modulation layers **520***a*-**520***n* operate at a different scale/resolution. Thus, the first cascaded modulation layer **520***a* takes input features at a first resolution/scale and generates new features at a lower scale/higher resolution (e.g., via upsampling as part of one or more modulation operations). Similarly, additional cascaded modulation layers operate at further lower scales/higher resolutions until generating the inpainted digital image at an output scale/resolution (e.g., the lowest scale/highest resolution).

(145) Moreover, each of the cascaded modulation layers include multiple modulation blocks. For example, with regard to FIG. **5** the first cascaded modulation layer **520***a* includes a global modulation block and a spatial modulation block. In particular, the cascaded modulation inpainting neural network **502** performs a global modulation with regard to input features of the global modulation block. Moreover, the cascaded modulation inpainting neural network **502** performs a spatial modulation with regard to input features of the spatial modulation block. By performing both a global modulation and spatial modulation within each cascaded modulation layer, the scene-based image editing system **106** refines global positions to generate more accurate inpainted digital images.

(146) As illustrated, the cascaded modulation layers **520***a*-**520***n* are cascaded in that the global modulation block feeds into the spatial modulation block. Specifically, the cascaded modulation inpainting neural network **502** performs the spatial modulation at the spatial modulation block based on features generated at the global modulation block. To illustrate, in one or more embodiments the cascaded modulation inpainting neural network **502** utilizes the global modulation block to generate an intermediate feature. The cascaded modulation inpainting neural network **502** further utilizes a convolutional layer (e.g., a 2-layer convolutional affine parameter network) to convert the intermediate feature to a spatial tensor. The cascaded modulation inpainting neural

network **502** utilizes the spatial tensor to modulate the input features analyzed by the spatial modulation block.
(147) For example, FIG. **6** provides additional detail regarding operation of global modulation blocks and spatial modulation blocks in accordance with one or more embodiments. Specifically, FIG. **6** illustrates a global modulation block **602** and a spatial modulation block **603**. As shown in FIG. **6**, the global modulation block **602** includes a first global modulation operation **604** and a second global modulation operation **606**. Moreover, the spatial modulation block **603** includes a global modulation operation **608** and a spatial modulation operation **610**.
(148) For example, a modulation block (or modulation operation) includes a computer-implemented process for modulating (e.g., scaling or shifting) an input signal according to one or more conditions. To illustrate, modulation block includes amplifying certain features while counteracting/normalizing these amplifications to preserve operation within a generative model. Thus, for example, a modulation block (or modulation operation) includes a modulation layer, a convolutional layer, and a normalization layer in some cases. The modulation layer scales each input feature of the convolution, and the normalization removes the effect of scaling from the statistics of the convolution's output feature maps.
(149) Indeed, because a modulation layer modifies feature statistics, a modulation block (or modulation operation) often includes one or more approaches for addressing these statistical changes. For example, in some instances, a modulation block (or modulation operation) includes a computer-implemented process that utilizes batch normalization or instance normalization to normalize a feature. In some embodiments, the modulation is achieved by scaling and shifting the normalized activation according to affine parameters predicted from input conditions. Similarly, some modulation procedures replace feature normalization with a demodulation process. Thus, in one or more embodiments, a modulation block (or modulation operation) includes a modulation layer, convolutional layer, and a demodulation layer. For example, in one or more embodiments, a modulation block (or modulation operation) includes the modulation approaches described by Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila in Analyzing and improving the image quality of StyleGAN, Proc. CVPR (2020) (hereinafter StyleGan2), which is incorporated by reference herein in its entirety. In some instances, a modulation block includes one or more modulation operations.
(150) Moreover, in one or more embodiments, a global modulation block (or global modulation operation) includes a modulation block (or modulation operation) that modulates an input signal in a spatially-invariant manner. For example, in some embodiments, a global modulation block (or global modulation operation) performs a modulation according to global features of a digital image (e.g., that do not vary spatially across coordinates of a feature map or image). Thus, for example, a global modulation block includes a modulation block that modulates an input signal according to an image encoding (e.g., global feature code) generated by an encoder. In some implementations, a global modulation block includes multiple global modulation operations.
(151) In one or more embodiments, a spatial modulation block (or spatial modulation operation) includes a modulation block (or modulation operation) that modulates an input signal in a spatially-varying manner (e.g., according to a spatially-varying feature map). In particular, in some embodiments, a spatial modulation block (or spatial modulation operation) utilizes a spatial tensor, to modulate an input signal in a spatially-varying manner. Thus, in one or more embodiments a global modulation block applies a global modulation where affine parameters are uniform across spatial coordinates, and a spatial modulation block applies a spatially-varying affine transformation that varies across spatial coordinates. In some embodiments, a spatial modulation block includes both a spatial modulation operation in combination with another modulation operation (e.g., a global modulation operation and a spatial modulation operation).
(152) For instance, in some embodiments, a spatial modulation operation includes spatially-adaptive modulation as described by Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan

Zhu in Semantic image synthesis with spatially-adaptive normalization, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2019), which is incorporated by reference herein in its entirety (hereinafter Taesung). In some embodiments, the spatial modulation operation utilizes a spatial modulation operation with a different architecture than Taesung, including a modulation-convolution-demodulation pipeline.
(153) Thus, with regard to FIG. **6**, the scene-based image editing system **106** utilizes a global modulation block **602**. As shown, the global modulation block **602** includes a first global modulation operation **604** and a second global modulation operation **606**. Specifically, the first global modulation operation **604** processes a global feature map **612**. For example, the global feature map **612** includes a feature vector generated by the cascaded modulation inpainting neural network reflecting global features (e.g., high-level features or features corresponding to the whole digital image). Thus, for example, the global feature map **612** includes a feature vector reflecting global features generated from a previous global modulation block of a cascaded decoder layer. In some instances, the global feature map **612** also includes a feature vector corresponding to the encoded feature vectors generated by the encoder (e.g., at a first decoder layer the scene-based image editing system **106** utilizes an encoded feature vector, style code, global feature code, constant, noise vector, or other feature vector as input in various implementations).
(154) As shown, the first global modulation operation **604** includes a modulation layer **604***a*, an upsampling layer **604***b*, a convolutional layer **604***c*, and a normalization layer **604***d*. In particular, the scene-based image editing system **106** utilizes the modulation layer **604***a* to perform a global modulation of the global feature map **612** based on a global feature code **614** (e.g., the global feature code **516**). Specifically, the scene-based image editing system **106** applies a neural network layer (i.e., a fully connected layer) to the global feature code **614** to generate a global feature vector **616**. The scene-based image editing system **106** then modulates the global feature map **612** utilizing the global feature vector **616**.
(155) In addition, the scene-based image editing system **106** applies the upsampling layer **604***b* (e.g., to modify the resolution scale). Further, the scene-based image editing system **106** applies the convolutional layer **604***c*. In addition, the scene-based image editing system **106** applies the normalization layer **604***d* to complete the first global modulation operation **604**. As shown, the first global modulation operation **604** generates a global intermediate feature **618**. In particular, in one or more embodiments, the scene-based image editing system **106** generates the global intermediate feature **618** by combining (e.g., concatenating) the output of the first global modulation operation **604** with an encoded feature vector **620** (e.g., from a convolutional layer of the encoder having a matching scale/resolution).
(156) As illustrated, the scene-based image editing system **106** also utilizes a second global modulation operation **606**. In particular, the scene-based image editing system **106** applies the second global modulation operation **606** to the global intermediate feature **618** to generate a new global feature map **622**. Specifically, the scene-based image editing system **106** applies a global modulation layer **606***a* to the global intermediate feature **618** (e.g., conditioned on the global feature vector **616**). Moreover, the scene-based image editing system **106** applies a convolutional layer **606***b* and a normalization layer **606***c* to generate the new global feature map **622**. As shown, in some embodiments, the scene-based image editing system **106** applies a spatial bias in generating the new global feature map **622**.
(157) Furthermore, as shown in FIG. **6**, the scene-based image editing system **106** utilizes a spatial modulation block **603**. In particular, the spatial modulation block **603** includes a global modulation operation **608** and a spatial modulation operation **610**. The global modulation operation **608** processes a local feature map **624**. For example, the local feature map **624** includes a feature vector generated by the cascaded modulation inpainting neural network reflecting local features (e.g., low-level, specific, or spatially variant features). Thus, for example, the local feature map **624** includes a feature vector reflecting local features generated from a previous spatial modulation block of a

cascaded decoder layer. In some cases, the global feature map **612** also includes a feature vector corresponding to the encoded feature vectors generated by the encoder (e.g., at a first decoder layer, the scene-based image editing system **106** utilizes an encoded feature vector, style code, noise vector or other feature vector in various implementations).

(158) As shown, the scene-based image editing system **106** utilizes the global modulation operation **608** to generate a local intermediate feature **626** from the local feature map **624**. Specifically, the scene-based image editing system **106** applies a modulation layer **608***a*, an upsampling layer **608***b*, a convolutional layer **608***c*, and a normalization layer **608***d*. Moreover, in some embodiments, the scene-based image editing system **106** applies spatial bias and broadcast noise to the output of the global modulation operation **608** to generate the local intermediate feature **626**.

(159) As illustrated in FIG. **6**, the scene-based image editing system **106** utilizes the spatial modulation operation **610** to generate a new local feature map **628**. Indeed, the spatial modulation operation **610** modulates the local intermediate feature **626** based on the global intermediate feature **618**. Specifically, the scene-based image editing system **106** generates a spatial tensor **630** from the global intermediate feature **618**. For example, the scene-based image editing system **106** applies a convolutional affine parameter network to generate the spatial tensor **630**. In particular, the scene-based image editing system **106** applies a convolutional affine parameter network to generate an intermediate spatial tensor. The scene-based image editing system **106** combines the intermediate spatial tensor with the global feature vector **616** to generate the spatial tensor **630**. The scene-based image editing system **106** utilizes the spatial tensor **630** to modulate the local intermediate feature **626** (utilizing the spatial modulation layer **610***a*) and generated a modulated tensor.

(160) As shown, the scene-based image editing system **106** also applies a convolutional layer **610***b* to the modulated tensor. In particular, the convolutional layer **610***b* generates a convolved feature representation from the modulated tensor. In addition, the scene-based image editing system **106** applies a normalization layer **610***c* to convolved feature representation to generate the new local feature map **628**.

(161) Although illustrated as a normalization layer **610***c*, in one or more embodiments, the scene-based image editing system **106** applies a demodulation layer. For example, the scene-based image editing system **106** applies a modulation-convolution-demodulation pipeline (e.g., general normalization rather than instance normalization). In some cases, this approach avoids potential artifacts (e.g., water droplet artifacts) caused by instance normalization. Indeed, a demodulation/normalization layer includes a layer that scales each output feature map by a uniform demodulation/normalization value (e.g., by a uniform standard deviation instead of instance normalization that utilizes data-dependent constant normalization based on the contents of the feature maps).

(162) As shown in FIG. **6**, in some embodiments, the scene-based image editing system **106** also applies a shifting tensor **632** and broadcast noise to the output of the spatial modulation operation **610**. For example, the spatial modulation operation **610** generates a normalized/demodulated feature. The scene-based image editing system **106** also generates the shifting tensor **632** by applying the affine parameter network to the global intermediate feature **618**. The scene-based image editing system **106** combines the normalized/demodulated feature, the shifting tensor **632**, and/or the broadcast noise to generate the new local feature map **628**.

(163) In one or more embodiments, upon generating the new global feature map **622** and the new local feature map **628**, the scene-based image editing system **106** proceeds to the next cascaded modulation layer in the decoder. For example, the scene-based image editing system **106** utilizes the new global feature map **622** and the new local feature map **628** as input features to an additional cascaded modulation layer at a different scale/resolution. The scene-based image editing system **106** further utilizes the additional cascaded modulation layer to generate additional feature maps (e.g., utilizing an additional global modulation block and an additional spatial modulation block). In some cases, the scene-based image editing system **106** iteratively processes feature maps utilizing

cascaded modulation layers until coming to a final scale/resolution to generate an inpainted digital image.
(164) Although FIG. **6** illustrates the global modulation block **602** and the spatial modulation block **603**, in some embodiments, the scene-based image editing system **106** utilizes a global modulation block followed by another global modulation block. For example, the scene-based image editing system **106** replaces the spatial modulation block **603** with an additional global modulation block. In such an embodiment, the scene-based image editing system **106** replaces APN (and spatial tensor) and corresponding spatial modulation illustrated in FIG. **6** with a skip connection. For example, the scene-based image editing system **106** utilizes the global intermediate feature to perform a global modulation with regard to the local intermediate vector. Thus, in some cases, the scene-based image editing system **106** utilizes a first global modulation block and a second global modulation block.
(165) As mentioned, the decoder can also be described in terms of variables and equations to illustrate operation of the cascaded modulation inpainting neural network. For example, as discussed, the decoder stacks a sequence of cascaded modulation blocks to upsample the input feature map $F_e^{(L)}$. Each cascaded modulation block takes the global code g as input to modulate the feature according to the global representation of the partial image. Moreover, in some cases, the scene-based image editing system **106** provides mechanisms to correct local error after predicting the global structure.
(166) In particular, in some embodiments, the scene-based image editing system **106** utilizes a cascaded modulation block to address the challenge of generating coherent features both globally and locally. At a high level, the scene-based image editing system **106** follows the following approach: i) decomposition of global and local features to separate local details from the global structure, ii) a cascade of global and spatial modulation that predicts local details from global structures. In one or more implementations, the scene-based image editing system **106** utilizes spatial modulations generated from the global code for better predictions (e.g., and discards instance normalization to make the design compatible with StyleGAN2).
(167) Specifically, the cascaded modulation takes the global and local feature $F_g^{(i)}$ and $F_l^{(i)}$ from previous scale and the global code g as input and produces the new global and local features $F_g^{(i+1)}$ and $F_l^{(i+1)}$ at next scale/resolution. To produce the new global code $F_g^{(i+1)}$ from $F_g^{(i)}$, the scene-based image editing system **106** utilizes a global code modulation stage that includes a modulation-convolution-demodulation procedure, which generates an upsampled feature X.
(168) Due to the limited expressive power of the global vector g on representing 2-d visual details, and the inconsistent features inside and outside the hole, the global modulation may generate distorted features inconsistent with the context. To compensate, in some cases, the scene-based image editing system **106** utilizes a spatial modulation that generates more accurate features. Specifically, the spatial modulation takes X as the spatial code and g as the global code to modulate the input local feature $F_l^{(i)}$ in a spatially adaptive fashion.
(169) Moreover, the scene-based image editing system **106** utilizes a unique spatial modulation-demodulation mechanism to avoid potential "water droplet" artifacts caused by instance normalization in conventional systems. As shown, the spatial modulation follows a modulation-convolution-demodulation pipeline.
(170) In particular, for spatial modulation, the scene-based image editing system **106** generates a spatial tensor A_0=APN(Y) from feature X by a 2-layer convolutional affine parameter network (APN). Meanwhile, the scene-based image editing system **106** generates a global vector α=fc(g) from global node g with a fully connected layer (fc) to capture global context. The scene-based image editing system **106** generates a final spatial tensor $A=A_0+\alpha$ as the broadcast summation of A_0 and α for scaling intermediate feature Y of the block with element-wise product ⊙:
$$Y=Y\odot A$$

(171) Moreover, for convolution, the modulated tensor Y is convolved with a 3×3 learnable kernel K, resulting in:

$\hat{Y}=Y*K$

(172) For spatially-aware demodulation, the scene-based image editing system **106** applies a demodularization step to compute the normalized output {tilde over (Y)}. Specifically, the scene-based image editing system **106** assumes that the input features Y are independent random variables with unit variance and after the modulation, the expected variance of the output is not changed, i.e., custom character.sub.y∈{tilde over (Y)}[Var(y)]=1. Accordingly, this gives the demodulation computation:

{tilde over (Y)}=Ŷ⊙D,

where D=1/√{square root over (K.sup.2⊙custom character.sub.a∈A[a.sup.2])} is the demodulation coefficient. In some cases, the scene-based image editing system **106** implements the foregoing equation with standard tensor operations.

(173) In one or more implementations, the scene-based image editing system **106** also adds spatial bias and broadcast noise. For example, the scene-based image editing system **106** adds the normalized feature {tilde over (Y)} to a shifting tensor B=APN(X) produced by another affine parameter network (APN) from feature X along with the broadcast noise n to product the new local feature F.sub.l.sup.(i+1):

F.sub.l.sup.(i+1)=*{tilde over (Y)}+B+n*

(174) Thus, in one or more embodiments, to generate a content fill having replacement pixels for a digital image having a replacement region, the scene-based image editing system **106** utilizes an encoder of a content-aware hole-filling machine learning model (e.g., a cascaded modulation inpainting neural network) to generate an encoded feature map from the digital image. The scene-based image editing system **106** further utilizes a decoder of the content-aware hole-filling machine learning model to generate the content fill for the replacement region. In particular, in some embodiments, the scene-based image editing system **106** utilizes a local feature map and a global feature map from one or more decoder layers of the content-aware hole-filling machine learning model in generating the content fill for the replacement region of the digital image.

(175) As discussed above with reference to FIGS. **3-6**, in one or more embodiments, the scene-based image editing system **106** utilizes a segmentation neural network to generate object masks for objects portrayed in a digital image and a content-aware hole-filling machine learning model to generate content fills for those objects (e.g., for the object masks generated for the objects). As further mentioned, in some embodiments, the scene-based image editing system **106** generates the object mask(s) and the content fill(s) in anticipation of one or more modifications to the digital image—before receiving user input for such modifications. For example, in one or more implementations, upon opening, accessing, or displaying the digital image **706**, the scene-based image editing system **106** generates the object mask(s) and the content fill(s) automatically (e.g., without user input to do so). Thus, in some implementations the scene-based image editing system **106** facilitates object-aware modifications of digital images. FIG. **7** illustrates a diagram for generating object masks and content fills to facilitate object-aware modifications to a digital image in accordance with one or more embodiments.

(176) In one or more embodiments, an object-aware modification includes an editing operation that targets an identified object in a digital image. In particular, in some embodiments, an object-aware modification includes an editing operation that targets an object that has been previously segmented. For instance, as discussed, the scene-based image editing system **106** generates a mask for an object portrayed in a digital image before receiving user input for modifying the object in some implementations. Accordingly, upon user selection of the object (e.g., a user selection of at least some of the pixels portraying the object), the scene-based image editing system **106** determines to target modifications to the entire object rather than requiring that the user specifically designate each pixel to be edited. Thus, in some cases, an object-aware modification includes a

modification that targets an object by managing all the pixels portraying the object as part of a cohesive unit rather than individual elements. For instance, in some implementations an object-aware modification includes, but is not limited to, a move operation or a delete operation.
(177) As shown in FIG. **7**, the scene-based image editing system **106** utilizes a segmentation neural network **702** and a content-aware hole-filling machine learning model **704** to analyze/process a digital image **706**. The digital image **706** portrays a plurality of objects **708***a*-**708***d* against a background. Accordingly, in one or more embodiments, the scene-based image editing system **106** utilizes the segmentation neural network **702** to identify the objects **708***a*-**708***d* within the digital image.
(178) In one or more embodiments, the scene-based image editing system **106** utilizes the segmentation neural network **702** and the content-aware hole-filling machine learning model **704** to analyze the digital image **706** in anticipation of receiving user input for modifications of the digital image **706**. Indeed, in some instances, the scene-based image editing system **106** analyzes the digital image **706** before receiving user input for such modifications. For instance, in some embodiments, the scene-based image editing system **106** analyzes the digital image **706** automatically in response to receiving or otherwise accessing the digital image **706**. In some implementations, the scene-based image editing system **106** analyzes the digital image in response to a general user input to initiate pre-processing in anticipation of subsequent modification.
(179) As shown in FIG. **7**, the scene-based image editing system **106** utilizes the segmentation neural network **702** to generate object masks **710** for the objects **708***a*-**708***d* portrayed in the digital image **706**. In particular, in some embodiments, the scene-based image editing system **106** utilizes the segmentation neural network **702** to generate a separate object mask for each portrayed object.
(180) As further shown in FIG. **7**, the scene-based image editing system **106** utilizes the content-aware hole-filling machine learning model **704** to generate content fills **712** for the objects **708***a*-**708***d*. In particular, in some embodiments, the scene-based image editing system **106** utilizes the content-aware hole-filling machine learning model **704** to generate a separate content fill for each portrayed object. As illustrated, the scene-based image editing system **106** generates the content fills **712** using the object masks **710**. For instance, in one or more embodiments, the scene-based image editing system **106** utilizes the object masks **710** generated via the segmentation neural network **702** as indicators of replacement regions to be replaced using the content fills **712** generated by the content-aware hole-filling machine learning model **704**. In some instances, the scene-based image editing system **106** utilizes the object masks **710** to filter out the objects from the digital image **706**, which results in remaining holes in the digital image **706** to be filled by the content fills content fills **712**.
(181) As shown in FIG. **7**, the scene-based image editing system **106** utilizes the object masks **710** and the content fills **712** to generate a completed background **714**. In one or more embodiments, a completed background image includes a set of background pixels having objects replaced with content fills. In particular, a completed background includes the background of a digital image having the objects portrayed within the digital image replaced with corresponding content fills. In one or more implementations, a completed background comprises generating a content fill for each object in the image. Thus, the completed background may comprise various levels of completion when objects are in front of each other such that the background for a first object comprises part of a second object and the background of the second object comprises a semantic area or the furthest element in the image.
(182) Indeed, FIG. **7** illustrates the background **716** of the digital image **706** with holes **718***a*-**718***d* where the objects **708***a*-**708***d* were portrayed. For instance, in some cases, the scene-based image editing system **106** filters out the objects **708***a*-**708***d* using the object masks **710**, causing the holes **718***a*-**718***d* to remain. Further, the scene-based image editing system **106** utilizes the content fills **712** to fill in the holes **718***a*-**718***d*, resulting in the completed background **714**.
(183) In other implementations, the scene-based image editing system **106** utilizes the object masks

710 as indicators of replacement regions in the digital image **706**. In particular, the scene-based image editing system **106** utilizes the object masks **710** as indicators of potential replacement regions that may result from receiving user input to modify the digital image **706** via moving/removing one or more of the objects **708***a*-**708***d*. Accordingly, the scene-based image editing system **106** utilizes the content fills **712** to replace pixels indicated by the object masks **710**.

(184) Though FIG. **7** indicates generating a separate completed background, it should be understood that, in some implementations, the scene-based image editing system **106** creates the completed background **714** as part of the digital image **706**. For instance, in one or more embodiments, the scene-based image editing system **106** positions the content fills **712** behind their corresponding object (e.g., as a separate image layer) in the digital image **706**. Further, in one or more embodiments, the scene-based image editing system **106** positions the object masks **710** behind their corresponding object (e.g., as a separate layer). In some implementations, the scene-based image editing system **106** places the content fills **712** behind the object masks **710**.

(185) Further, in some implementations, the scene-based image editing system **106** generates multiple filled-in backgrounds (e.g., semi-completed backgrounds) for a digital image. For instance, in some cases, where a digital image portrays a plurality of objects, the scene-based image editing system **106** generates a filled-in background for each object from the plurality of objects. To illustrate, the scene-based image editing system **106** generates a filled-in background for an object by generating a content fill for that object while treating the other objects of the digital image as part of the background. Thus, in some instances, the content fill includes portions of other objects positioned behind the object within the digital image.

(186) Thus, in one or more embodiments, the scene-based image editing system **106** generates a combined image **718** as indicated in FIG. **7**. Indeed, the scene-based image editing system **106** generates the combined image having the digital image **706**, the object masks **710**, and the content fills **712** as separate layers. Though, FIG. **7** shows the object masks **710** on top of the objects **708***a*-**708***d* within the combined image **718**, it should be understood that the scene-based image editing system **106** places the object masks **710** as well as the content fills **712** behind the objects **708***a*-**708***d* in various implementations. Accordingly, the scene-based image editing system **106** presents the combined image **718** for display within a graphical user interface so that the object masks **710** and the content fills **712** are hidden from view until user interactions that trigger display of those components is received.

(187) Further, though FIG. **7** shows the combined image **718** as separate from the digital image **706**, it should be understood that the combined image **718** represents modifications to the digital image **706** in some implementations. In other words, in some embodiments, to generate the combined image **718** the scene-based image editing system **106** modifies the digital image **706** by adding additional layers composed of the object masks **710** and the content fills **712**.

(188) In one or more embodiments, the scene-based image editing system **106** utilizes the combined image **718** (e.g., the digital image **706**, the object masks **710**, and the content fills **712**) to facilitate various object-aware modifications with respect to the digital image **706**. In particular, the scene-based image editing system **106** utilizes the combined image **718** to implement an efficient graphical user interface that facilitates flexible object-aware modifications. FIGS. **8**A-**8**D illustrate a graphical user interface implemented by the scene-based image editing system **106** to facilitate a move operation in accordance with one or more embodiments.

(189) Indeed, as shown in FIG. **8**A, the scene-based image editing system **106** provides a graphical user interface **802** for display on a client device **804**, such as a mobile device. Further, the scene-based image editing system **106** provides a digital image **806** for display with the graphical user interface.

(190) It should be noted that the graphical user interface **802** of FIG. **8**A is minimalistic in style. In particular, the graphical user interface **802** does not include a significant number of menus, options, or other visual elements aside from the digital image **806**. Though the graphical user interface **802**

of FIG. **8**A displays no menus, options, or other visual elements aside from the digital image **806**, it should be understood that the graphical user interface **802** displays at least some menus, options, or other visual elements in various embodiments—at least when the digital image **806** is initially displayed.
(191) As further shown in FIG. **8**A, the digital image **806** portrays a plurality of objects **808***a*-**808***d*. In one or more embodiments, the scene-based image editing system **106** pre-processes the digital image **806** before receiving user input for the move operation. In particular, in some embodiments, the scene-based image editing system **106** utilizes a segmentation neural network to detect and generate masks for the plurality of objects **808***a*-**808***d* and/or utilizes a content-aware hole-filling machine learning model to generate content fills that correspond to the objects **808***a*-**808***d*. Furthermore, in one or more implementations, the scene-based image editing system **106** generates the object masks, content fills, and a combined image upon loading, accessing, or displaying the digital image **806**, and without, user input other than to open/display the digital image **806**.
(192) As shown in FIG. **8**B, the scene-based image editing system **106** detects a user interaction with the object **808***d* via the graphical user interface **802**. In particular, FIG. **8**B illustrates the scene-based image editing system **106** detecting a user interaction executed by a finger (part of a hand **810**) of a user (e.g., a touch interaction), though user interactions are executed by other instruments (e.g., stylus or pointer controlled by a mouse or track pad) in various embodiments. In one or more embodiments, the scene-based image editing system **106** determines that, based on the user interaction, the object **808***d* has been selected for modification.
(193) The scene-based image editing system **106** detects the user interaction for selecting the object **808***d* via various operations in various embodiments. For instance, in some cases, the scene-based image editing system **106** detects the selection via a single tap (or click) on the object **808***d*. In some implementations, the scene-based image editing system **106** detects the selection of the object **808***d* via a double tap (or double click) or a press and hold operation. Thus, in some instances, the scene-based image editing system **106** utilizes the second click or the hold operation to confirm the user selection of the object **808***d*.
(194) In some cases, the scene-based image editing system **106** utilizes various interactions to differentiate between a single object select or a multi-object select. For instance, in some cases, the scene-based image editing system **106** determines that a single tap is for selecting a single object and a double tap is for selecting multiple objects. To illustrate, in some cases, upon receiving a first tap on an object, the scene-based image editing system **106** selects the object. Further, upon receiving a second tap on the object, the scene-based image editing system **106** selects one or more additional objects. For instance, in some implementations, the scene-based image editing system **106** selects one or more additional object having the same or a similar classification (e.g., selecting other people portrayed in an image when the first tap interacted with a person in the image). In one or more embodiments, the scene-based image editing system **106** recognizes the second tap as an interaction for selecting multiple objects if the second tap is received within a threshold time period after receiving the first tap.
(195) In some embodiments, the scene-based image editing system **106** recognizes other user interactions for selecting multiple objects within a digital image. For instance, in some implementations, the scene-based image editing system **106** receives a dragging motion across the display of a digital image and selects all object captured within the range of the dragging motion. To illustrate, in some cases, the scene-based image editing system **106** draws a box that grows with the dragging motion and selects all objects that falls within the box. In some cases, the scene-based image editing system **106** draws a line that follows the path of the dragging motion and selects all objects intercepted by the line.
(196) In some implementations, the scene-based image editing system **106** further allows for user interactions to select distinct portions of an object. To illustrate, in some cases, upon receiving a first tap on an object, the scene-based image editing system **106** selects the object. Further, upon

receiving a second tap on the object, the scene-based image editing system **106** selects a particular portion of the object (e.g., a limb or torso of a person or a component of a vehicle). In some cases, the scene-based image editing system **106** selects the portion of the object touched by the second tap. In some cases, the scene-based image editing system **106** enters into a "sub object" mode upon receiving the second tap and utilizes additional user interactions for selecting particular portions of the object.
(197) Returning to FIG. **8**B, as shown, based on detecting the user interaction for selecting the object **808***d*, the scene-based image editing system **106** provides a visual indication **812** in association with the object **808***d*. Indeed, in one or more embodiments, the scene-based image editing system **106** detects the user interaction with a portion of the object **808***d*—e.g., with a subset of the pixels that portray the object—and determines that the user interaction targets the object **808***d* as a whole (rather than the specific pixels with which the user interacted). For instance, in some embodiments, the scene-based image editing system **106** utilizes the pre-generated object mask that corresponds to the object **808***d* to determine whether the user interaction targets the object **808***d* or some other portion of the digital image **806**. For example, in some cases, upon detecting that the user interacts with an area inside the object mask that corresponds to the object **808***d*, the scene-based image editing system **106** determines that the user interaction targets the object **808***d* as a whole. Thus, the scene-based image editing system **106** provides the visual indication **812** in association with the object **808***d* as a whole.
(198) In some cases, the scene-based image editing system **106** utilizes the visual indication **812** to indicate, via the graphical user interface **802**, that the selection of the object **808***d* has been registered. In some implementations, the scene-based image editing system **106** utilizes the visual indication **812** to represent the pre-generated object mask that corresponds to the object **808***d*. Indeed, in one or more embodiments, in response to detecting the user interaction with the object **808***d*, the scene-based image editing system **106** surfaces the corresponding object mask. For instance, in some cases, the scene-based image editing system **106** surfaces the object mask in preparation for a modification to the object **808***d* and/or to indicate that the object mask has already been generated and is available for use. In one or more embodiments, rather than using the visual indication **812** to represent the surfacing of the object mask, the scene-based image editing system **106** displays the object mask itself via the graphical user interface **802**.
(199) Additionally, as the scene-based image editing system **106** generated the object mask for the object **808***d* prior to receiving the user input to select the object **808***d*, the scene-based image editing system **106** surfaces the visual indication **812** without latency or delay associated with conventional systems. In other words, the scene-based image editing system **106** surfaces the visual indication **812** without any delay associated with generating an object mask.
(200) As further illustrated, based on detecting the user interaction for selecting the object **808***d*, the scene-based image editing system **106** provides an option menu **814** for display via the graphical user interface **802**. The option menu **814** shown in FIG. **8**B provides a plurality of options, though the option menu includes various numbers of options in various embodiments. For instance, in some implementations, the option menu **814** includes one or more curated options, such as options determined to be popular or used with the most frequency. For example, as shown in FIG. **8**B, the option menu **814** includes an option **816** to delete the object **808***d*.
(201) Thus, in one or more embodiments, the scene-based image editing system **106** provides modification options for display via the graphical user interface **802** based on the context of a user interaction. Indeed, as just discussed, the scene-based image editing system **106** provides an option menu that provides options for interacting with (e.g., modifying) a selected object. In doing so, the scene-based image editing system **106** minimizes the screen clutter that is typical under many conventional systems by withholding options or menus for display until it is determined that those options or menus would be useful in the current context in which the user is interacting with the digital image. Thus, the graphical user interface **802** used by the scene-based image editing system

106 allows for more flexible implementation on computing devices with relatively limited screen space, such as smart phones or tablet devices.

(202) As shown in FIG. **8**C, the scene-based image editing system **106** detects, via the graphical user interface **802**, an additional user interaction for moving the object **808***d* across the digital image **806** (as shown via the arrow **818**). In particular, the scene-based image editing system **106** detects the additional user interaction for moving the object **808***d* from a first position in the digital image **806** to a second position. For instance, in some cases, the scene-based image editing system **106** detects the second user interaction via a dragging motion (e.g., the user input selects the object **808***d* and moves across the graphical user interface **802** while holding onto the object **808***d*). In some implementations, after the initial selection of the object **808***d*, the scene-based image editing system **106** detects the additional user interaction as a click or tap on the second position and determines to use the second position as a new position for the object **808***d*. It should be noted that the scene-based image editing system **106** moves the object **808***d* as a whole in response to the additional user interaction.

(203) As indicated in FIG. **8**C, upon moving the object **808***d* from the first position to the second position, the scene-based image editing system **106** exposes the content fill **820** that was placed behind the object **808***d* (e.g., behind the corresponding object mask). Indeed, as previously discussed, the scene-based image editing system **106** places pre-generated content fills behind the objects (or corresponding object masks) for which the content fills were generated. Accordingly, upon removing the object **808***d* from its initial position within the digital image **806**, the scene-based image editing system **106** automatically reveals the corresponding content fill. Thus, the scene-based image editing system **106** provides a seamless experience where an object is movable without exposing any holes in the digital image itself. In other words, the scene-based image editing system **106** provides the digital image **806** for display as if it were a real scene in which the entire background is already known.

(204) Additionally, as the scene-based image editing system **106** generated the content fill **820** for the object **808***d* prior to receiving the user input to move the object **808***d*, the scene-based image editing system **106** exposes or surfaces the content fill **820** without latency or delay associated with conventional systems. In other words, the scene-based image editing system **106** exposes the content fill **820** incrementally as the object **808***d* is moved across the digital image **806** without any delay associated with generating content.

(205) As further shown in FIG. **8**D, the scene-based image editing system **106** deselects the object **808***d* upon completion of the move operation. In some embodiments, the object **808***d* maintains the selection of the object **808***d* until receiving a further user interaction to indicate deselection of the object **808***d* (e.g., a user interaction with another portion of the digital image **806**). As further indicated, upon deselecting the object **808***d*, the scene-based image editing system **106** further removes the option menu **814** that was previously presented. Thus, the scene-based image editing system **106** dynamically presents options for interacting with objects for display via the graphical user interface **802** to maintain a minimalistic style that does not overwhelm the displays of computing devices with limited screen space.

(206) FIGS. **9**A-**9**C illustrate a graphical user interface implemented by the scene-based image editing system **106** to facilitate a delete operation in accordance with one or more embodiments. Indeed, as shown in FIG. **9**A, the scene-based image editing system **106** provides a graphical user interface **902** for display on a client device **904** and provides a digital image **906** for display in the graphical user interface **902**.

(207) As further shown in FIG. **9**B, the scene-based image editing system **106** detects, via the graphical user interface **902**, a user interaction with an object **908** portrayed in the digital image **906**. In response to detecting the user interaction, the scene-based image editing system **106** surfaces the corresponding object mask, providing the visual indication **910** (or the object mask itself) for display in association with the object **908**, and provides the option menu **912** for display.

In particular, as shown, the option menu **912** includes an option **914** for deleting the object **908** that has been selected.

(208) Additionally, as shown in FIG. **9**C, the scene-based image editing system **106** removes the object **908** from the digital image **906**. For instance, in some cases, the scene-based image editing system **106** detects an additional user interaction via the graphical user interface **902** (e.g., an interaction with the option **914** for deleting the object **908**) and removes the object **908** from the digital image **906** in response. As further shown, upon removing the object **908** from the digital image **906**, the scene-based image editing system **106** automatically exposes the content fill **916** that was previously placed behind the object **908** (e.g., behind the corresponding object mask). Thus, in one or more embodiments, the scene-based image editing system **106** provides the content fill **916** for immediate display upon removal of the object **908**.

(209) While FIGS. **8**B, **8**C, and **9**B illustrate the scene-based image editing system **106** providing a menu, in or more implementations, the scene-based image editing system **106** allows for object-based editing without requiring or utilizing a menu. For example, the scene-based image editing system **106** selects an object **808***d*, **908** and surfaces a visual indication **812**, **910** in response to a first user interaction (e.g., a tap on the respective object). The scene-based image editing system **106** performs an object-based editing of the digital image in response to second user interaction without the use of a menu. For example, in response to a second user input dragging the object across the image, the scene-based image editing system **106** moves the object. Alternatively, in response to a second user input (e.g., a second tap), the scene-based image editing system **106** deletes the object.

(210) The scene-based image editing system **106** provides more flexibility for editing digital images when compared to conventional systems. In particular, the scene-based image editing system **106** facilitates object-aware modifications that enable interactions with objects rather than requiring targeting the underlying pixels. Indeed, based on a selection of some pixels that contribute to the portrayal of an object, the scene-based image editing system **106** flexibly determines that the whole object has been selected. This is in contrast to conventional systems that require a user to select an option from a menu indicating an intention to selection an object, provide a second user input indicating the object to select (e.g., a bounding box about the object or drawing of another rough boundary about the object), and another user input to generate the object mask. The scene-based image editing system **106** instead provides for selection of an object with a single user input (a tap on the object).

(211) Further, upon user interactions for implementing a modification after the prior selection, the scene-based image editing system **106** applies the modification to the entire object rather than the particular set of pixels that were selected. Thus, the scene-based image editing system **106** manages objects within digital images as objects of a real scene that are interactive and can be handled as cohesive units. Further, as discussed, the scene-based image editing system **106** offers improved flexibility with respect to deployment on smaller devices by flexibly and dynamically managing the amount of content that is displayed on a graphical user interface in addition to a digital image.

(212) Additionally, the scene-based image editing system **106** offers improved efficiency when compared to many conventional systems. Indeed, as previously discussed, conventional systems typically require execution of a workflow consisting of a sequence of user interactions to perform a modification. Where a modification is meant to target a particular object, many of these systems require several user interactions just to indicate that the object is the subject of the subsequent modification (e.g., user interactions for identifying the object and separating the object from the rest of the image) as well as user interactions for closing the loop on executed modifications (e.g., filling in the holes remaining after removing objects). The scene-based image editing system **106**, however, reduces the user interactions typically required for a modification by pre-processing a digital image before receiving user input for such a modification. Indeed, by generating object masks and content fills automatically, the scene-based image editing system **106** eliminates the

need for user interactions to perform these steps.
(213) In one or more embodiments, the scene-based image editing system **106** performs further processing of a digital image in anticipation of modifying the digital image. For instance, as previously mentioned, the scene-based image editing system **106** generates a semantic scene graph from a digital image in some implementations. Thus, in some cases, upon receiving one or more user interactions for modifying the digital image, the scene-based image editing system **106** utilizes the semantic scene graph to execute the modifications. Indeed, in many instances, the scene-based image editing system **106** generates a semantic scene graph for use in modifying a digital image before receiving user input for such modifications. FIGS. **10-15** illustrate diagrams for generating a semantic scene graph for a digital image in accordance with one or more embodiments.
(214) Indeed, many conventional systems are inflexible in that they typically wait upon user interactions before determining characteristics of a digital image. For instance, such conventional systems often wait upon a user interaction that indicates a characteristic to be determined and then performs the corresponding analysis in response to receiving the user interaction. Accordingly, these systems fail to have useful characteristics readily available for use. For example, upon receiving a user interaction for modifying a digital image, conventional systems typically must perform an analysis of the digital image to determine characteristics to change after the user interaction has been received.
(215) Further, as previously discussed, such operation results in inefficient operation as image edits often require workflows of user interactions, many of which are used in determining characteristics to be used in execution of the modification. Thus, conventional systems often require a significant number of user interactions to determine the characteristics needed for an edit.
(216) The scene-based image editing system **106** provides advantages by generating a semantic scene graph for a digital image in anticipation of modifications to the digital image. Indeed, by generating the semantic scene graph, the scene-based image editing system **106** improves flexibility over conventional systems as it makes characteristics of a digital image readily available for use in the image editing process. Further, the scene-based image editing system **106** provides improved efficiency by reducing the user interactions required in determining these characteristics. In other words, the scene-based image editing system **106** eliminates the user interactions often required under conventional systems for the preparatory steps of editing a digital image. Thus, the scene-based image editing system **106** enables user interactions to focus on the image edits more directly themselves.
(217) Additionally, by generating a semantic scene graph for a digital image, the scene-based image editing system **106** intelligently generates/obtains information the allows an image to be edited like a real-world scene. For example, the scene-based image editing system **106** generates a scene graph that indicates objects, object attributes, object relationships, etc. that allows the scene-based image editing system **106** to enable object/scene-based image editing.
(218) In one or more embodiments, a semantic scene graph includes a graph representation of a digital image. In particular, in some embodiments, a semantic scene graph includes a graph that maps out characteristics of a digital image and their associated characteristic attributes. For instance, in some implementations, a semantic scene graph includes a node graph having nodes that represent characteristics of the digital image and values associated with the node representing characteristic attributes of those characteristics. Further, in some cases, the edges between the nodes represent the relationships between the characteristics.
(219) As mentioned, in one or more implementations, the scene-based image editing system **106** utilizes one or more predetermined or pre-generated template graphs in generating a semantic scene graph for a digital image. For instance, in some cases, the scene-based image editing system **106** utilizes an image analysis graph in generating a semantic scene graph. FIG. **10** illustrates an image analysis graph **1000** utilized by the scene-based image editing system **106** in generating a semantic scene graph in accordance with one or more embodiments.

(220) In one or more embodiments, an image analysis graph includes a template graph for structuring a semantic scene graph. In particular, in some embodiments, an image analysis graph includes a template graph used by the scene-based image editing system **106** to organize the information included in a semantic scene graph. For instance, in some implementations, an image analysis graph includes a template graph that indicates how to organize the nodes of the semantic scene graph representing characteristics of a digital image. In some instances, an image analysis graph additionally or alternatively indicates the information to be represented within a semantic scene graph. For instance, in some cases, an image analysis graph indicates the characteristics, relationships, and characteristic attributes of a digital image to be represented within a semantic scene graph.

(221) Indeed, as shown in FIG. **10**, the image analysis graph **1000** includes a plurality of nodes **1004***a*-**1004***g*. In particular, the plurality of nodes **1004***a*-**1004***g* correspond to characteristics of a digital image. For instance, in some cases, the plurality of nodes **1004***a*-**1004***g* represent characteristic categories that are to be determined when analyzing a digital image. Indeed, as illustrated, the image analysis graph **1000** indicates that a semantic scene graph is to represent the objects and object groups within a digital image as well as the scene of a digital image, including the lighting source, the setting, and the particular location.

(222) As further shown in FIG. **10**, the image analysis graph **1000** includes an organization of the plurality of nodes **1004***a*-**1004***g*. In particular, the image analysis graph **1000** includes edges **1006***a*-**1006***h* arranged in a manner that organizes the plurality of nodes **1004***a*-**1004***g*. In other words, the image analysis graph **1000** illustrates the relationships among the characteristic categories included therein. For instance, the image analysis graph **1000** indicates that the object category represented by the node **1004***f* and the object group category represented by the node **1004***g* are closely related, both describing objects that portrayed in a digital image.

(223) Additionally, as shown in FIG. **10**, the image analysis graph **1000** associates characteristic attributes with one or more of the nodes **1004***a*-**1004***g* to represent characteristic attributes of the corresponding characteristic categories. For instance, as shown, the image analysis graph **1000** associates a season attribute **1008***a* and a time-of-day attribute **1008***b* with the setting category represented by the node **1004***c*. In other words, the image analysis graph **1000** indicates that the season and time of day should be determined when determining a setting of a digital image. Further, as shown, the image analysis graph **1000** associates an object mask **1010***a* and a bounding box **1010***b* with the object category represented by the node **1004***f*. Indeed, in some implementations, the scene-based image editing system **106** generates content for objects portrayed in a digital image, such as an object mask and a bounding box. Accordingly, the image analysis graph **1000** indicates that this pre-generated content is to be associated with the node representing the corresponding object within a semantic scene graph generated for the digital image.

(224) As further shown in FIG. **10**, the image analysis graph **1000** associates characteristic attributes with one or more of the edges **1006***a*-**1006***h* to represent characteristic attributes of the corresponding characteristic relationships represented by these edges **1006***a*-**1006***h*. For instance, as shown, the image analysis graph **1000** associates a characteristic attribute **1012***a* with the edge **1006***g* indicating that an object portrayed in a digital image will be a member of a particular object group. Further, the image analysis graph **1000** associates a characteristic attribute **1012***b* with the edge **1006***h* indicating that at least some objects portrayed in a digital image have relationships with one another. FIG. **10** illustrates a sample of relationships that are identified between objects in various embodiments, and additional detail regarding these relationships will be discussed in further detail below.

(225) It should be noted that the characteristic categories and characteristic attributes represented in FIG. **10** are exemplary and the image analysis graph **1000** includes a variety of characteristic categories and/or characteristic attributes not shown in various embodiments. Further, FIG. **10** illustrates a particular organization of the image analysis graph **1000**, though alternative

arrangements are used in different embodiments. Indeed, in various embodiments, the scene-based image editing system **106** accommodates a variety of characteristic categories and characteristic attributes to facilitate subsequent generation of a semantic scene graph that supports a variety of image edits. In other words, the scene-based image editing system **106** includes those characteristic categories and characteristic attributes that it determines are useful in editing a digital image.

(226) In some embodiments, the scene-based image editing system **106** utilizes a real-world class description graph in generating a semantic scene graph for a digital image. FIG. **11** illustrates a real-world class description graph **1102** utilized by the scene-based image editing system **106** in generating a semantic scene graph in accordance with one or more embodiments.

(227) In one or more embodiments, a real-world class description graph includes a template graph that describes scene components (e.g., semantic areas) that may be portrayed in a digital image. In particular, in some embodiments, a real-world class description graph includes a template graph used by the scene-based image editing system **106** to provide contextual information to a semantic scene graph regarding scene components—such as objects—potentially portrayed in a digital image. For instance, in some implementations, a real-world class description graph provides a hierarchy of object classifications and/or an anatomy (e.g., object components) of certain objects that may be portrayed in a digital image. In some instances, a real-world class description graph further includes object attributes associated with the objects represented therein. For instance, in some cases, a real-world class description graph provides object attributes assigned to a given object, such as shape, color, material from which the object is made, weight of the object, weight the object can support, and/or various other attributes determined to be useful in subsequently modifying a digital image. Indeed, as will be discussed, in some cases, the scene-based image editing system **106** utilizes a semantic scene graph for a digital image to suggest certain edits or suggest avoiding certain edits to maintain consistency of the digital image with respect to the contextual information contained in the real-world class description graph from which the semantic scene graph was built.

(228) As shown in FIG. **11**, the real-world class description graph **1102** includes a plurality of nodes **1104*a*-1104*h*** and a plurality of edges **1106*a*-1106*e*** that connect some of the nodes **1104*a*-1104*h***. In particular, in contrast to the image analysis graph **1000** of FIG. **10**, the real-world class description graph **1102** does not provide a single network of interconnected nodes. Rather, in some implementations, the real-world class description graph **1102** includes a plurality of node clusters **1108*a*-1108*c*** that are separate and distinct from one another.

(229) In one or more embodiments, each node cluster corresponds to a separate scene component (e.g., semantic area) class that may be portrayed in a digital image. Indeed, as shown in FIG. **11**, each of the node clusters **1108*a*-1108*c*** corresponds to a separate object class that may be portrayed in a digital image. As indicated above, the real-world class description graph **1102** is not limited to representing object classes and can represent other scene component classes in various embodiments.

(230) As shown in FIG. **11**, each of the node clusters **1108*a*-1108*c*** portrays a hierarchy of class descriptions (otherwise referred to as a hierarchy of object classifications) corresponding to a represented object class. In other words, each of the node clusters **1108*a*-1108*c*** portrays degrees of specificity/generality with which an object is described or labeled. Indeed, in some embodiments, the scene-based image editing system **106** applies all class descriptions/labels represented in a node cluster to describe a corresponding object portrayed in a digital image. In some implementations, however, the scene-based image editing system **106** utilizes a subset of the class descriptions/labels to describe an object.

(231) As an example, the node cluster **1108*a*** includes a node **1104*a*** representing a side table class and a node **1104*b*** representing a table class. Further, as shown in FIG. **11**, the node cluster **1108*a*** includes an edge **1106*a*** between the node **1104*a*** and the node **1104*b*** to indicate that the side table class is a subclass of the table class, thus indicating a hierarchy between these two classifications

that are applicable to a side table. In other words, the node cluster **1108***a* indicates that a side table is classifiable either as a side table and/or more generally as a table. In other words, in one or more embodiments, upon detecting a side table portrayed in a digital image, the scene-based image editing system **106** labels the side table as a side table and/or as a table based on the hierarchy represented in the real-world class description graph **1102**.

(232) The degree to which a node cluster represents a hierarchy of class descriptions varies in various embodiments. In other words, the length/height of the represented hierarchy varies in various embodiments. For instance, in some implementations, the node cluster **1108***a* further includes a node representing a furniture class, indicating that a side table is classifiable as a piece of furniture. In some cases, the node cluster **1108***a* also includes a node representing an inanimate object lass, indicating that a side table is classifiable as such. Further, in some implementations, the node cluster **1108***a* includes a node representing an entity class, indicating that a side table is classifiable as an entity. Indeed, in some implementations, the hierarchies of class descriptions represented within the real-world class description graph **1102** include a class description/label—such as an entity class—at such a high level of generality that it is commonly applicable to all objects represented within the real-world class description graph **1102**.

(233) As further shown in FIG. **11**, the node cluster **1108***a* includes an anatomy (e.g., object components) of the represented object class. In particular, the node cluster **1108***a* includes a representation of component parts for the table class of objects. For instance, as shown, the node cluster **1108***a* includes a node **1104***c* representing a table leg class. Further, the node cluster **1108***a* includes an edge **1106***b* indicating that a table leg from the table leg class is part of a table from the table class. In other words, the edge **1106***b* indicates that a table leg is a component of a table. In some cases, the node cluster **1108***a* includes additional nodes for representing other components that are part of a table, such as a tabletop, a leaf, or an apron.

(234) As shown in FIG. **11**, the node **1104***c* representing the table leg class of objects is connected to the node **1104***b* representing the table class of objects rather than the node **1104***a* representing the side table class of objects. Indeed, in some implementations, the scene-based image editing system **106** utilizes such a configuration based on determining that all tables include one or more table legs. Thus, as side tables are a subclass of tables, the configuration of the node cluster **1108***a* indicates that all side tables also include one or more table legs. In some implementations, however, the scene-based image editing system **106** additionally or alternatively connects the node **1104***c* representing the table leg class of objects to the node **1104***a* representing the side table class of objects to specify that all side tables include one or more table legs.

(235) Similarly, the node cluster **1108***a* includes object attributes **1110***a***-1110***d* associated with the node **1104***a* for the side table class and an additional object attributes **1112***a***-1112***g* associated with the node **1104***b* for the table class. Thus, the node cluster **1108***a* indicates that the object attributes **1110***a***-1110***d* are specific to the side table class while the additional object attributes **1112***a***-1112***g* are more generally associated with the table class (e.g., associated with all object classes that fall within the table class). In one or more embodiments, the object attributes **1110***a***-1110***d* and/or the additional object attributes **1112***a***-1112***g* are attributes that have been arbitrarily assigned to their respective object class (e.g., via user input or system defaults). For instance, in some cases, the scene-based image editing system **106** determines that all side tables can support one hundred pounds as suggested by FIG. **11** regardless of the materials used or the quality of the build. In some instances, however, the object attributes **1110***a***-1110***d* and/or the additional object attributes **1112***a***-1112***g* represent object attributes that are common among all objects that fall within a particular class, such as the relatively small size of side tables. In some implementations, however, the object attributes **1110***a***-1110***d* and/or the additional object attributes **1112***a***-1112***g* are indicators of object attributes that should be determined for an object of the corresponding object class. For instance, in one or more embodiments, upon identifying a side table, the scene-based image editing system **106** determines at least one of the capacity, size, weight, or supporting weight of the side table.

(236) It should be noted that there is some overlap between object attributes included in a real-world class description graph and characteristic attributes included in an image analysis graph in some embodiments. Indeed, in many implementations, object attributes are characteristic attributes that are specific towards objects (rather than attributes for the setting or scene of a digital image). Further, it should be noted that the object attributes are merely exemplary and do not necessarily reflect the object attributes that are to be associated with an object class. Indeed, in some embodiments, the object attributes that are shown and their association with particular object classes are configurable to accommodate different needs in editing a digital image.
(237) In some cases, a node cluster corresponds to one particular class of objects and presents a hierarchy of class descriptions and/or object components for that one particular class. For instance, in some implementations, the node cluster **1108***a* only corresponds to the side table class and presents a hierarchy of class descriptions and/or object components that are relevant to side tables. Thus, in some cases, upon identifying a side table within a digital image, the scene-based image editing system **106** refers to the node cluster **1108***a* for the side table class when generating a semantic scene graph but refers to a separate node cluster upon identifying another subclass of table within the digital image. In some cases, this separate node cluster includes several similarities (e.g., similar nodes and edges) with the node cluster **1108***a* as the other type of table would be included in a subclass of the table class and include one or more table legs.
(238) In some implementations, however, a node cluster corresponds to a plurality of different but related object classes and presents a common hierarchy of class descriptions and/or object components for those object classes. For instance, in some embodiments, the node cluster **1108***a* includes an additional node representing a dining table class that is connected to the node **1104***b* representing the table class via an edge indicating that dining tables are also a subclass of tables. Indeed, in some cases, the node cluster **1108***a* includes nodes representing various subclasses of a table class. Thus, in some instances, upon identifying a table from a digital image, the scene-based image editing system **106** refers to the node cluster **1108***a* when generating a semantic scene graph for the digital image regardless of the subclass to which the table belongs.
(239) As will be described, in some implementations, utilizing a common node cluster for multiple related subclasses facilitates object interactivity within a digital image. For instance, as noted, FIG. **11** illustrates multiple separate node clusters. As further mentioned however, the scene-based image editing system **106** includes a classification (e.g., an entity classification) that is common among all represented objects within the real-world class description graph **1102** in some instances. Accordingly, in some implementations, the real-world class description graph **1102** does include a single network of interconnected nodes where all node clusters corresponding to separate object classes connect at a common node, such as a node representing an entity class. Thus, in some embodiments, the real-world class description graph **1102** illustrates the relationships among all represented objects.
(240) In one or more embodiments, the scene-based image editing system **106** utilizes a behavioral policy graph in generating a semantic scene graph for a digital image. FIG. **12** illustrates a behavioral policy graph **1202** utilized by the scene-based image editing system **106** in generating a semantic scene graph in accordance with one or more embodiments.
(241) In one or more embodiments, a behavioral policy graph includes a template graph that describes the behavior of an object portrayed in a digital image based on the context in which the object is portrayed. In particular, in some embodiments, a behavioral policy graph includes a template graph that assigns behaviors to objects portrayed in a digital image based on a semantic understanding of the objects and/or their relationships to other objects portrayed in the digital image. Indeed, in one or more embodiments, a behavioral policy includes various relationships among various types of objects and designates behaviors for those relationships. In some cases, the scene-based image editing system **106** includes a behavioral policy graph as part of a semantic scene graph. In some implementations, as will be discussed further below, a behavioral policy is

separate from the semantic scene graph but provides plug-in behaviors based on the semantic understanding and relationships of objects represented in the semantic scene graph.

(242) As shown in FIG. **12**, the behavioral policy graph **1202** includes a plurality of relationship indicators **1204***a*-**1204***e* and a plurality of behavior indicators **1206***a*-**1206***e* that are associated with the relationship indicators **1204***a*-**1204***e*. In one or more embodiments, the relationship indicators **1204***a*-**1204***e* reference a relationship subject (e.g., an object in the digital image that is the subject of the relationship) and a relationship object (e.g., an object in the digital image that is the object of the relationship). For example, the relationship indicators **1204***a*-**1204***e* of FIG. **12** indicate that the relationship subject "is supported by" or "is part of" the relationship object. Further, in one or more embodiments the behavior indicators **1206***a*-**1206***e* assign a behavior to the relationship subject (e.g., indicating that the relationship subject "moves with" or "deletes with" the relationship object). In other words, the behavior indicators **1206***a*-**1206***e* provide modification instructions for the relationship subject when the relationship object is modified.

(243) FIG. **12** provides a small subset of the relationships recognized by the scene-based image editing system **106** in various embodiments. For instance, in some implementations, the relationships recognized by the scene-based image editing system **106** and incorporated into generated semantic scene graphs include, but are not limited to, relationships described as "above," "below," "behind," "in front of," "touching," "held by," "is holding," "supporting," "standing on," "worn by," "wearing," "leaning on," "looked at by," or "looking at." Indeed, as suggested by the foregoing, the scene-based image editing system **106** utilizes relationship pairs to describe the relationship between objects in both directions in some implementations. For instance, in some cases, where describing that a first object "is supported by" a second object, the scene-based image editing system **106** further describes that the second object "is supporting" the first object. Thus, in some cases, the behavioral policy graph **1202** includes these relationship pairs, and the scene-based image editing system **106** includes the information in the semantic scene graphs accordingly.

(244) As further shown, the behavioral policy graph **1202** further includes a plurality of classification indicators **1208***a*-**1208***e* associated with the relationship indicators **1204***a*-**1204***e*. In one or more embodiments, the classification indicators **1208***a*-**1208***e* indicate an object class to which the assigned behavior applies. Indeed, in one or more embodiments, the classification indicators **1208***a*-**1208***e* reference the object class of the corresponding relationship object. As shown by FIG. **12**, the classification indicators **1208***a*-**1208***e* indicate that a behavior is assigned to object classes that are a subclass of the designated object class. In other words, FIG. **12** shows that the classification indicators **1208***a*-**1208***e* reference a particular object class and indicate that the assigned behavior applies to all objects that fall within that object class (e.g., object classes that are part of a subclass that falls under that object class).

(245) The level of generality or specificity of a designated object class referenced by a classification indicator within its corresponding hierarchy of object classification varies in various embodiments. For instance, in some embodiments, a classification indicator references a lowest classification level (e.g., the most specific classification applicable) so that there are no subclasses, and the corresponding behavior applies only to those objects having that particular object lowest classification level. On the other hand, in some implementations, a classification indicator references a highest classification level (e.g., the most generic classification applicable) or some other level above the lowest classification level so that the corresponding behavior applies to objects associated with one or more of the multiple classification levels that exist within that designated classification level.

(246) To provide an illustration of how the behavioral policy graph **1202** indicates assigned behavior, the relationship indicator **1204***a* indicates a "is supported by" relationship between an object (e.g., the relationship subject) and another object (e.g., the relationship object). The behavior indicator **1206***a* indicates a "moves with" behavior that is associated with the "is supported by" relationship, and the classification indicator **1208***a* indicates that this particular behavior applies to

objects within some designated object class. Accordingly, in one or more embodiments, the behavioral policy graph **1202** shows that an object that falls within the designated object class and has a "is supported by" relationship with another object will exhibit the "moves with" behavior. In other words, if a first object of the designated object class is portrayed in a digital image being supported by a second object, and the digital image is modified to move that second object, then the scene-based image editing system **106** will automatically move the first object with the second object as part of the modification in accordance with the behavioral policy graph **1202**. In some cases, rather than moving the first object automatically, the scene-based image editing system **106** provides a suggestion to move the first object for display within the graphical user interface in use to modify the digital image.

(247) As shown by FIG. **12**, some of the relationship indicators (e.g., the relationship indicators **1204***a*-**1204***b* or the relationship indicators **1204***c*-**1204***e*) refer to the same relationship but are associated with different behaviors. Indeed, in some implementations, the behavioral policy graph **1202** assigns multiple behaviors to the same relationship. In some instances, the difference is due to the difference in the designated subclass. In particular, in some embodiments, the scene-based image editing system **106** assigns an object of one object class a particular behavior for a particular relationship but assigns an object of another object class a different behavior for the same relationship. Thus, in configuring the behavioral policy graph **1202**, the scene-based image editing system **106** manages different object classes differently in various embodiments.

(248) FIG. **13** illustrates a semantic scene graph **1302** generated by the scene-based image editing system **106** for a digital image in accordance with one or more embodiments. In particular, the semantic scene graph **1302** shown in FIG. **13** is a simplified example of a semantic scene graph and does not portray all the information included in a semantic scene graph generated by the scene-based image editing system **106** in various embodiments.

(249) As shown in FIG. **13**, the semantic scene graph **1302** is organized in accordance with the image analysis graph **1000** described above with reference to FIG. **10**. In particular, the semantic scene graph **1302** includes a single network of interconnected nodes that reference characteristics of a digital image. For instance, the semantic scene graph **1302** includes nodes **1304***a*-**1304***c* representing portrayed objects as indicated by their connection to the node **1306**. Further, the semantic scene graph **1302** includes relationship indicators **1308***a*-**1308***c* representing the relationships between the objects corresponding to the nodes **1304***a*-**1304***c*. As further shown, the semantic scene graph **1302** includes a node **1310** representing a commonality among the objects (e.g., in that the objects are all included in the digital image, or the objects indicate a subject or topic of the digital image). Additionally, as shown, the semantic scene graph **1302** includes the characteristic attributes **1314***a*-**1314***f* of the objects corresponding to the nodes **1304***a*-**1304***c*.

(250) As further shown in FIG. **13**, the semantic scene graph **1302** includes contextual information from the real-world class description graph **1102** described above with reference to FIG. **11**. In particular, the semantic scene graph **1302** includes nodes **1312***a*-**1312***c* that indicate the object class to which the objects corresponding to the nodes **1304***a*-**1304***c* belong. Though not shown in FIG. **11**, the semantic scene graph **1302** further includes the full hierarchy of object classifications for each of the object classes represented by the nodes **1312***a*-**1312***c*. In some cases, however, the nodes **1312***a*-**1312***c* each include a pointer that points to their respective hierarchy of object classifications within the real-world class description graph **1102**. Additionally, as shown in FIG. **13**, the semantic scene graph **1302** includes object attributes **1316***a*-**1316***e* of the object classes represented therein.

(251) Additionally, as shown in FIG. **13**, the semantic scene graph **1302** includes behaviors from the behavioral policy graph **1202** described above with reference to FIG. **12**. In particular, the semantic scene graph **1302** includes behavior indicators **1318***a*-**1318***b* indicating behaviors of the objects represented therein based on their associated relationships.

(252) FIG. **14** illustrates a diagram for generating a semantic scene graph for a digital image

utilizing template graphs in accordance with one or more embodiments. Indeed, as shown in FIG. **14**, the scene-based image editing system **106** analyzes a digital image **1402** utilizing one or more neural networks **1404**. In particular, in one or more embodiments, the scene-based image editing system **106** utilizes the one or more neural networks **1404** to determine various characteristics of the digital image **1402** and/or their corresponding characteristic attributes. For instance, in some cases, the scene-based image editing system **106** utilizes a segmentation neural network to identify and classify objects portrayed in a digital image (as discussed above with reference to FIG. **3**). Further, in some embodiments, the scene-based image editing system **106** utilizes neural networks to determine the relationships between objects and/or their object attributes as will be discussed in more detail below.

(253) In one or more implementations, the scene-based image editing system **106** utilizes a depth estimation neural network to estimate a depth of an object in a digital image and stores the determined depth in the semantic scene graph **1412**. For example, the scene-based image editing system **106** utilizes a depth estimation neural network as described in U.S. application Ser. No. 17/186,436, filed Feb. 26, 2021, titled "GENERATING DEPTH IMAGES UTILIZING A MACHINE-LEARNING MODEL BUILT FROM MIXED DIGITAL IMAGE SOURCES AND MULTIPLE LOSS FUNCTION SETS," which is herein incorporated by reference in its entirety. Alternatively, the scene-based image editing system **106** utilizes a depth refinement neural network as described in U.S. application Ser. No. 17/658,873, filed Apr. 12, 2022, titled "UTILIZING MACHINE LEARNING MODELS TO GENERATE REFINED DEPTH MAPS WITH SEGMENTATION MASK GUIDANCE," which is herein incorporated by reference in its entirety. The scene-based image editing system **106** then accesses the depth information (e.g., average depth for an object) for an object from the semantic scene graph **1412** when editing an object to perform a realistic scene edit. For example, when moving an object within an image, the scene-based image editing system **106** then accesses the depth information for objects in the digital image from the semantic scene graph **1412** to ensure that the object being moved is not placed in front an object with less depth.

(254) In one or more implementations, the scene-based image editing system **106** utilizes a depth estimation neural network to estimate lighting parameters for an object or scene in a digital image and stores the determined lighting parameters in the semantic scene graph **1412**. For example, the scene-based image editing system **106** utilizes a source-specific-lighting-estimation-neural network as described in U.S. application Ser. No. 16/558,975, filed Sep. 3, 2019, titled "DYNAMICALLY ESTIMATING LIGHT-SOURCE-SPECIFIC PARAMETERS FOR DIGITAL IMAGES USING A NEURAL NETWORK," which is herein incorporated by reference in its entirety. The scene-based image editing system **106** then accesses the lighting parameters for an object or scene from the semantic scene graph **1412** when editing an object to perform a realistic scene edit. For example, when moving an object within an image or inserting a new object in a digital image, the scene-based image editing system **106** accesses the lighting parameters for from the semantic scene graph **1412** to ensure that the object being moved/placed within the digital image has realistic lighting.

(255) In one or more implementations, the scene-based image editing system **106** utilizes a depth estimation neural network to estimate lighting parameters for an object or scene in a digital image and stores the determined lighting parameters in the semantic scene graph **1412**. For example, the scene-based image editing system **106** utilizes a source-specific-lighting-estimation-neural network as described in U.S. application Ser. No. 16/558,975, filed Sep. 3, 2019, titled "DYNAMICALLY ESTIMATING LIGHT-SOURCE-SPECIFIC PARAMETERS FOR DIGITAL IMAGES USING A NEURAL NETWORK," which is herein incorporated by reference in its entirety. The scene-based image editing system **106** then accesses the lighting parameters for an object or scene from the semantic scene graph **1412** when editing an object to perform a realistic scene edit. For example, when moving an object within an image or inserting a new object in a digital image, the scene-based image editing system **106** accesses the lighting parameters for from the semantic scene graph

1412 to ensure that the object being moved/placed within the digital image has realistic lighting.

(256) As further shown in FIG. **14**, the scene-based image editing system **106** utilizes the output of the one or more neural networks **1404** along with an image analysis graph **1406**, a real-world class description graph **1408**, and a behavioral policy graph **1410** to generate a semantic scene graph **1412**. In particular, the scene-based image editing system **106** generates the semantic scene graph **1412** to include a description of the digital image **1402** in accordance with the structure, characteristic attributes, hierarchies of object classifications, and behaviors provided by the image analysis graph **1406**, the real-world class description graph **1408**, and the behavioral policy graph **1410**.

(257) As previously indicated, in one or more embodiments, the image analysis graph **1406**, the real-world class description graph **1408**, and/or the behavioral policy graph **1410** are predetermined or pre-generated. In other words, the scene-based image editing system **106** pre-generates, structures, or otherwise determines the content and organization of each graph before implementation. For instance, in some cases, the scene-based image editing system **106** generates the image analysis graph **1406**, the real-world class description graph **1408**, and/or the behavioral policy graph **1410** based on user input.

(258) Further, in one or more embodiments, the image analysis graph **1406**, the real-world class description graph **1408**, and/or the behavioral policy graph **1410** are configurable. Indeed, the graphs can be re-configured, re-organized, and/or have data represented therein added or removed based on preferences or the needs of editing a digital image. For instance, in some cases, the behaviors assigned by the behavioral policy graph **1410** work in some image editing contexts but not others. Thus, when editing an image in another image editing context, the scene-based image editing system **106** implements the one or more neural networks **1404** and the image analysis graph **1406** but implements a different behavioral policy graph (e.g., one that was configured to satisfy preferences for that image editing context). Accordingly, in some embodiments, the scene-based image editing system **106** modifies the image analysis graph **1406**, the real-world class description graph **1408**, and/or the behavioral policy graph **1410** to accommodate different image editing contexts.

(259) For example, in one or more implementations, the scene-based image editing system **106** determines a context for selecting a behavioral policy graph by identifying a type of user. In particular, the scene-based image editing system **106** generates a plurality of behavioral policy graphs for various types of users. For instance, the scene-based image editing system **106** generates a first behavioral policy graph for novice or new users. The first behavioral policy graph, in one or more implementations, includes a greater number of behavior policies than a second behavioral policy graph. In particular, for newer users, the scene-based image editing system **106** utilizes a first behavioral policy graph that provides greater automation of actions and provides less control to the user. On the other hand, the scene-based image editing system **106** utilizes a second behavioral policy graph for advanced users with less behavior policies than the first behavioral policy graph. In this manner, the scene-based image editing system **106** provides the advanced user with greater control over the relationship-based actions (automatic moving/deleting/editing) of objects based on relationships. In other words, by utilizing the second behavioral policy graph for advanced users, the scene-based image editing system **106** performs less automatic editing of related objects.

(260) In one or more implementations the scene-based image editing system **106** determines a context for selecting a behavioral policy graph based on visual content of a digital image (e.g., types of objects portrayed in the digital image), the editing application being utilized, etc. Thus, the scene-based image editing system **106**, in one or more implementations, selects/utilizes a behavioral policy graph based on image content, a type of user, an editing application being utilizes, or another context.

(261) Moreover, in some embodiments, the scene-based image editing system **106** utilizes the graphs in analyzing a plurality of digital images. Indeed, in some cases, the image analysis graph

1406, the real-world class description graph **1408**, and/or the behavioral policy graph **1410** do not specifically target a particular digital image. Thus, in many cases, these graphs are universal and re-used by the scene-based image editing system **106** for multiple instances of digital image analysis.
(262) In some cases, the scene-based image editing system **106** further implements one or more mappings to map between the outputs of the one or more neural networks **1404** and the data scheme of the image analysis graph **1406**, the real-world class description graph **1408**, and/or the behavioral policy graph **1410**. As one example, the scene-based image editing system **106** utilizes various segmentation neural networks to identify and classify objects in various embodiments. Thus, depending on the segmentation neural network used, the resulting classification of a given object can be different (e.g., different wording or a different level of abstraction). Thus, in some cases, the scene-based image editing system **106** utilizes a mapping that maps the particular outputs of the segmentation neural network to the object classes represented in the real-world class description graph **1408**, allowing the real-world class description graph **1408** to be used in conjunction with multiple neural networks.
(263) FIG. **15** illustrates another diagram for generating a semantic scene graph for a digital image in accordance with one or more embodiments. In particular, FIG. **15** illustrates an example framework of the scene-based image editing system **106** generating a semantic scene graph in accordance with one or more embodiments.
(264) As shown in FIG. **15**, the scene-based image editing system **106** identifies an input image **1500**. In some cases, the scene-based image editing system **106** identifies the input image **1500** based on a request. For instances, in some cases, the request includes a request to generate a semantic scene graph for the input image **1500**. In one or more implementations the request comprises to analyze the input image comprises the scene-based image editing system **106** accessing, opening, or displaying by the input image **1500**.
(265) In one or more embodiments, the scene-based image editing system **106** generates object proposals and subgraph proposals for the input image **1500** in response to the request. For instance, in some embodiments, the scene-based image editing system **106** utilizes an object proposal network **1520** to extract a set of object proposals for the input image **1500**. To illustrate, in some cases, the scene-based image editing system **106** extracts a set of object proposals for humans detected within the input image **1500**, objects that the human(s) are wearing, objects near the human(s), buildings, plants, animals, background objects or scenery (including the sky or objects in the sky), etc.
(266) In one or more embodiments, the object proposal network **1520** comprises the detection-masking neural network **300** (specifically, the object detection machine learning model **308**) discussed above with reference to FIG. **3**. In some cases, the object proposal network **1520** includes a neural network such as a region proposal network ("RPN"), which is part of a region-based convolutional neural network, to extract the set of object proposals represented by a plurality of bounding boxes. One example RPN is disclosed in S. Ren, K. He, R. Girshick, and J. Sun, *Faster r-cnn: Towards real-time object detection with region proposal networks*, NIPS, 2015, the entire contents of which are hereby incorporated by reference. As an example, in some cases, the scene-based image editing system **106** uses the RPN to extract object proposals for significant objects (e.g., detectable objects or objects that have a threshold size/visibility) within the input image. The algorithm below represents one embodiment of a set of object proposals:
$[o_0, \ldots, 0_{N-1}]=f_{RPN}(I)$
where I is the input image, $f_{RPN}(\cdot)$ represents the RPN network, and o_i is the i-th object proposal.
(267) In some implementations, in connection with determining the object proposals, the scene-based image editing system **106** also determines coordinates of each object proposal relative to the dimensions of the input image **1500**. Specifically, in some instances, the locations of the object proposals are based on bounding boxes that contain the visible portion(s) of objects within a digital

image. To illustrate, for o.sub.i, the coordinates of the corresponding bounding box are represented by r.sub.i=[x.sub.i, y.sub.i, w.sub.i, h.sub.i], with (x.sub.i,y.sub.i) being the coordinates of the top left corner and w.sub.i and h.sub.i being the width and the height of the bounding box, respectively. Thus, the scene-based image editing system **106** determines the relative location of each significant object or entity in the input image **1500** and stores the location data with the set of object proposals.
(268) As mentioned, in some implementations, the scene-based image editing system **106** also determines subgraph proposals for the object proposals. In one or more embodiments, the subgraph proposals indicate relations involving specific object proposals in the input image **1500**. As can be appreciated, any two different objects (o.sub.i, o.sub.j) in a digital image can correspond to two possible relationships in opposite directions. As an example, a first object can be “on top of” a second object, and the second object can be “underneath” the first object. Because each pair of objects has two possible relations, the total number of possible relations for N object proposals is N(N−1). Accordingly, more object proposals result in a larger scene graph than fewer object proposals, while increasing computational cost and deteriorating inference speed of object detection in systems that attempt to determine all the possible relations in both directions for every object proposal for an input image.
(269) Subgraph proposals reduce the number of potential relations that the scene-based image editing system **106** analyze. Specifically, as mentioned previously, a subgraph proposal represents a relationship involving two or more specific object proposals. Accordingly, in some instances, the scene-based image editing system **106** determines the subgraph proposals for the input image **1500** to reduce the number of potential relations by clustering, rather than maintaining the N(N−1) number of possible relations. In one or more embodiments, the scene-based image editing system **106** uses the clustering and subgraph proposal generation process described in Y. Li, W. Ouyang, B. Zhou, Y. Cui, J. Shi, and X. Wang, *Factorizable net: An efficient subgraph based framework for scene graph generation*, ECCV, Jun. 29, 2018, the entire contents of which are hereby incorporated by reference.
(270) As an example, for a pair of object proposals, the scene-based image editing system **106** determines a subgraph based on confidence scores associated with the object proposals. To illustrate, the scene-based image editing system **106** generates each object proposal with a confidence score indicating the confidence that the object proposal is the right match for the corresponding region of the input image. The scene-based image editing system **106** further determines the subgraph proposal for a pair of object proposals based on a combined confidence score that is the product of the confidence scores of the two object proposals. The scene-based image editing system **106** further constructs the subgraph proposal as the union box of the object proposals with the combined confidence score.
(271) In some cases, the scene-based image editing system **106** also suppresses the subgraph proposals to represent a candidate relation as two objects and one subgraph. Specifically, in some embodiments, the scene-based image editing system **106** utilizes non-maximum-suppression to represent the candidate relations as custom charactero.sub.i, o.sub.j, s.sub.k.sup.i custom character, where i≠j and s.sub.k.sup.i is the k-th subgraph of all the subgraphs associated with o.sub.i, the subgraphs for o.sub.i including o.sub.j and potentially other object proposals. After suppressing the subgraph proposals, the scene-based image editing system **106** represents each object and subgraph as a feature vector, o.sub.i∈custom character.sup.D and a feature map s.sub.k.sup.i∈custom character.sup.D×K.sup.a.sup.×K.sup.a, respectively, where D and K.sub.a are dimensions.
(272) After determining object proposals and subgraph proposals for objects in the input image, the scene-based image editing system **106** retrieves and embeds relationships from an external knowledgebase **1522**. In one or more embodiments, an external knowledgebase includes a dataset of semantic relationships involving objects. In particular, in some embodiments, an external knowledgebase includes a semantic network including descriptions of relationships between objects

based on background knowledge and contextual knowledge (also referred to herein as "commonsense relationships"). In some implementations, an external knowledgebase includes a database on one or more servers that includes relationship knowledge from one or more sources including expert-created resources, crowdsourced resources, web-based sources, dictionaries, or other sources that include information about object relationships.
(273) Additionally, in one or more embodiments an embedding includes a representation of relationships involving objects as a vector. For instance, in some cases, a relationship embedding includes a vector representation of a triplet (i.e., an object label, one or more relationships, and an object entity) using extracted relationships from an external knowledgebase.
(274) Indeed, in one or more embodiments, the scene-based image editing system **106** communicates with the external knowledgebase **1522** to obtain useful object-relationship information for improving the object and subgraph proposals. Further, in one or more embodiments, the scene-based image editing system **106** refines the object proposals and subgraph proposals (represented by the box **1524**) using embedded relationships, as described in more detail below.
(275) In some embodiments, in preparation for retrieving the relationships from the external knowledgebase **1522**, the scene-based image editing system **106** performs a process of inter-refinement on the object and subgraph proposals (e.g., in preparation for refining features of the object and subgraph proposals). Specifically, the scene-based image editing system **106** uses the knowledge that each object o.sub.i is connected to a set of subgraphs S.sup.i, and each subgraph s.sub.k is associated with a set of objects O.sup.k to refine the object vector (resp. the subgraphs) by attending the associated subgraph feature maps (resp. the associated object vectors). For instance, in some cases, the inter-refinement process is represented as:
(276) $\bar{o}_i = o_i + f_{s\text{ .fwdarw. }o}\left(\text{.Math.}_{s_k^i \in S^i}\ {}_{k}^{s\text{ .fwdarw. }o}\ \text{.Math. } s_k^i\right)$

$$\bar{s}_i = s_i + f_{o\text{ .fwdarw. }s}\left(\text{.Math.}_{o_i^k \in O^k}\ {}_{i}^{o\text{ .fwdarw. }s}\ \text{.Math. } o_i^k\right)$$

where α.sub.k.sup.s.fwdarw.o (resp. α.sub.i.sup.o.fwdarw.s) is the output of a softmax layer indicating the weight for passing s.sub.k.sup.i (resp. o.sub.i.sup.k) to o.sub.i (resp. to s.sub.k), and f.sub.s.fwdarw.o and f.sub.o.fwdarw.s are non-linear mapping functions. In one or more embodiments, due to different dimensions of o.sub.i and s.sub.k, the scene-based image editing system **106** applies pooling or spatial location-based attention for s.fwdarw.o or o.fwdarw.s refinement.
(277) In some embodiments, once the inter-refinement is complete, the scene-based image editing system **106** predicts an object label from the initially refined object feature vector ō.sub.i and matches the object label with the corresponding semantic entities in the external knowledgebase **1522**. In particular, the scene-based image editing system **106** accesses the external knowledgebase **1522** to obtain the most common relationships corresponding to the object label. The scene-based image editing system **106** further selects a predetermined number of the most common relationships from the external knowledgebase **1522** and uses the retrieved relationships to refine the features of the corresponding object proposal/feature vector.
(278) In one or more embodiments, after refining the object proposals and subgraph proposals using the embedded relationships, the scene-based image editing system **106** predicts object labels **1502** and predicate labels from the refined proposals. Specifically, the scene-based image editing system **106** predicts the labels based on the refined object/subgraph features. For instance, in some cases, the scene-based image editing system **106** predicts each object label directly with the refined features of a corresponding feature vector. Additionally, the scene-based image editing system **106** predicts a predicate label (e.g., a relationship label) based on subject and object feature vectors in connection with their corresponding subgraph feature map due to subgraph features being associated with several object proposal pairs. In one or more embodiments, the inference process for predicting the labels is shown as:

P.sub.i,j˜softmax(*f*.sub.rel([*{tilde over (x)}*.sub.i.*Math.s*.sub.k*;õ*.sub.j*Ðs*.sub.k*;s*.sub.k]))
V.sub.i˜Softmax(*f*.sub.node(*õ*.sub.i))
where and f.sub.rel(·) and f.sub.node(·) denote the mapping layers for predicate and object recognition, respectively, and .Math. represents a convolution operation. Furthermore, õ.sub.i represents a refined feature vector based on the extracted relationships from the external knowledgebase.

(279) In one or more embodiments, the scene-based image editing system **106** further generates a semantic scene graph **1504** using the predicted labels. In particular, the scene-based image editing system **106** uses the object labels **1502** and predicate labels from the refined features to create a graph representation of the semantic information of the input image **1500**. In one or more embodiments, the scene-based image editing system **106** generates the scene graph as custom character=custom characterV.sub.i, P.sub.i,j, V.sub.jcustom character, i≠j where custom character is the scene graph.

(280) Thus, the scene-based image editing system **106** utilizes relative location of the objects and their labels in connection with an external knowledgebase **1522** to determine relationships between objects. The scene-based image editing system **106** utilizes the determined relationships when generating a behavioral policy graph **1410**. As an example, the scene-based image editing system **106** determines that a hand and a cell phone have an overlapping location within the digital image. Based on the relative locations and depth information, the scene-based image editing system **106** determines that a person (associated with the hand) has a relationship of “holding” the cell phone. As another example, the scene-based image editing system **106** determines that a person and a shirt have an overlapping location and overlapping depth within a digital image. Based on the relative locations and relative depth information, the scene-based image editing system **106** determines that the person has a relationship of “wearing” the shirt. On other hand, the scene-based image editing system **106** determines that a person and a shirt have an overlapping location and but the shirt has a greater average depth than an average depth of the person within a digital image. Based on the relative locations and relative depth information, the scene-based image editing system **106** determines that the person has a relationship of “in front of” with the shirt.

(281) By generating a semantic scene graph for a digital image, the scene-based image editing system **106** provides improved flexibility and efficiency. Indeed, as mentioned above, the scene-based image editing system **106** generates a semantic scene graph to provide improved flexibility as characteristics used in modifying a digital image are readily available at the time user interactions are received to execute a modification. Accordingly, the scene-based image editing system **106** reduces the user interactions typically needed under conventional systems to determine those characteristics (or generate needed content, such as bounding boxes or object masks) in preparation for executing a modification. Thus, the scene-based image editing system **106** provides a more efficient graphical user interface that requires less user interactions to modify a digital image.

(282) Additionally, by generating a semantic scene graph for a digital image, the scene-based image editing system **106** provides an ability to edit a two-dimensional image like a real-world scene. For example, based on a generated semantic scene graph for an image generated utilizing various neural networks, the scene-based image editing system **106** determines objects, their attributes (position, depth, material, color, weight, size, label, etc.). The scene-based image editing system **106** utilizes the information of the semantic scene graph to edit an image intelligently as if the image were a real-world scene.

(283) Indeed, in one or more embodiments, the scene-based image editing system **106** utilizes a semantic scene graph generated for a digital image to facilitate modification to the digital image. For instance, in one or more embodiments, the scene-based image editing system **106** facilitates modification of one or more object attributes of an object portrayed in a digital image utilizing the corresponding semantic scene graph. FIGS. **16-21**C illustrate modifying one or more object attributes of an object portrayed in a digital image in accordance with one or more embodiments.

(284) Many conventional systems are inflexible in that they often require difficult, tedious workflows to target modifications to a particular object attribute of an object portrayed in a digital image. Indeed, modifying an object attribute often requires manual manipulation of the object attribute under such systems. For example, modifying a shape of an object portrayed in a digital image often requires several user interactions to manually restructure the boundaries of an object (often at the pixel level), and modifying a size often requires tedious interactions with resizing tools to adjust the size and ensure proportionality. Thus, in addition to inflexibility, many conventional systems suffer from inefficiency in that the processes required by these systems to execute such a targeted modification typically involve a significant number of user interactions.
(285) The scene-based image editing system **106** provides advantages over conventional systems by operating with improved flexibility and efficiency. Indeed, by presenting a graphical user interface element through which user interactions are able to target object attributes of an object, the scene-based image editing system **106** offers more flexibility in the interactivity of objects portrayed in digital images. In particular, via the graphical user interface element, the scene-based image editing system **106** provides flexible selection and modification of object attributes. Accordingly, the scene-based image editing system **106** further provides improved efficiency by reducing the user interactions required to modify an object attribute. Indeed, as will be discussed below, the scene-based image editing system **106** enables user interactions to interact with a description of an object attribute in order to modify that object attribute, avoiding the difficult, tedious workflows of user interactions required under many conventional systems.
(286) As suggested, in one or more embodiments, the scene-based image editing system **106** facilitates modifying object attributes of objects portrayed in a digital image by determining the object attributes of those objects. In particular, in some cases, the scene-based image editing system **106** utilizes a machine learning model, such as an attribute classification neural network, to determine the object attributes. FIGS. **16-17** illustrates an attribute classification neural network utilized by the scene-based image editing system **106** to determine object attributes for objects in accordance with one or more embodiments. In particular, FIGS. **16-17** illustrate a multi-attribute contrastive classification neural network utilized by the scene-based image editing system **106** in one or more embodiments.
(287) In one or more embodiments, an attribute classification neural network includes a computer-implemented neural network that identifies object attributes of objects portrayed in a digital image. In particular, in some embodiments, an attribute classification neural network includes a computer-implemented neural network that analyzes objects portrayed in a digital image, identifies the object attributes of the objects, and provides labels for the corresponding object attributes in response. It should be understood that, in many cases, an attribute classification neural network more broadly identifies and classifies attributes for semantic areas portrayed in a digital image. Indeed, in some implementations, an attribute classification neural network determines attributes for semantic areas portrayed in a digital image aside from objects (e.g., the foreground or background).
(288) FIG. **16** illustrates an overview of a multi-attribute contrastive classification neural network in accordance with one or more embodiments. In particular, FIG. **16** illustrates the scene-based image editing system **106** utilizing a multi-attribute contrastive classification neural network to extract a wide variety of attribute labels (e.g., negative, positive, and unknown labels) for an object portrayed within a digital image.
(289) As shown in FIG. **16**, the scene-based image editing system **106** utilizes an embedding neural network **1604** with a digital image **1602** to generate an image-object feature map **1606** and a low-level attribute feature map **1610**. In particular, the scene-based image editing system **106** generates the image-object feature map **1606** (e.g., the image-object feature map X) by combining an object-label embedding vector **1608** with a high-level attribute feature map from the embedding neural network **1604**. For instance, the object-label embedding vector **1608** represents an embedding of an object label (e.g., “chair”).

(290) Furthermore, as shown in FIG. **16**, the scene-based image editing system **106** generates a localized image-object feature vector Z.sub.rel. In particular, the scene-based image editing system **106** utilizes the image-object feature map **1606** with the localizer neural network **1612** to generate the localized image-object feature vector Z.sub.rel. Specifically, the scene-based image editing system **106** combines the image-object feature map **1606** with a localized object attention feature vector **1616** (denoted G) to generate the localized image-object feature vector Z.sub.rel to reflect a segmentation prediction of the relevant object (e.g., "chair") portrayed in the digital image **1602**. As further shown in FIG. **16**, the localizer neural network **1612**, in some embodiments, is trained using ground truth object segmentation masks **1618**.
(291) Additionally, as illustrated in FIG. **16**, the scene-based image editing system **106** also generates a localized low-level attribute feature vector Z.sub.low. In particular, in reference to FIG. **16**, the scene-based image editing system **106** utilizes the localized object attention feature vector G from the localizer neural network **1612** with the low-level attribute feature map **1610** to generate the localized low-level attribute feature vector Z.sub.low.
(292) Moreover, as shown FIG. **16**, the scene-based image editing system **106** generates a multi-attention feature vector Z.sub.att. As illustrated in FIG. **16**, the scene-based image editing system **106** generates the multi-attention feature vector Z.sub.att from the image-object feature map **1606** by utilizing attention maps **1620** of the multi-attention neural network **1614**. Indeed, in one or more embodiments, the scene-based image editing system **106** utilizes the multi-attention feature vector Z.sub.att to attend to features at different spatial locations in relation to the object portrayed within the digital image **1602** while predicting attribute labels for the portrayed object.
(293) As further shown in FIG. **16**, the scene-based image editing system **106** utilizes a classifier neural network **1624** to predict the attribute labels **1626** upon generating the localized image-object feature vector Z.sub.rel, the localized low-level attribute feature vector Z.sub.low, and the multi-attention feature vector Z att (collectively shown as vectors **1622** in FIG. **16**). In particular, in one or more embodiments, the scene-based image editing system **106** utilizes the classifier neural network **1624** with a concatenation of the localized image-object feature vector Z.sub.rel, the localized low-level attribute feature vector Z.sub.low, and the multi-attention feature vector Z.sub.att to determine the attribute labels **1626** for the object (e.g., chair) portrayed within the digital image **1602**. As shown in FIG. **16**, the scene-based image editing system **106** determines positive attribute labels for the chair portrayed in the digital image **1602**, negative attribute labels that are not attributes of the chair portrayed in the digital image **1602**, and unknown attribute labels that correspond to attribute labels that the scene-based image editing system **106** could not confidently classify utilizing the classifier neural network **1624** as belonging to the chair portrayed in the digital image **1602**.
(294) In some instances, the scene-based image editing system **106** utilizes probabilities (e.g., a probability score, floating point probability) output by the classifier neural network **1624** for the particular attributes to determine whether the attributes are classified as positive, negative, and/or unknown attribute labels for the object portrayed in the digital image **1602** (e.g., the chair). For example, the scene-based image editing system **106** identifies an attribute as a positive attribute when a probability output for the particular attribute satisfies a positive attribute threshold (e.g., a positive probability, a probability that is over 0.5). Moreover, the scene-based image editing system **106** identifies an attribute as a negative attribute when a probability output for the particular attribute satisfies a negative attribute threshold (e.g., a negative probability, a probability that is below −0.5). Furthermore, in some cases, the scene-based image editing system **106** identifies an attribute as an unknown attribute when the probability output for the particular attribute does not satisfy either the positive attribute threshold or the negative attribute threshold.
(295) In some cases, a feature map includes a height, width, and dimension locations (H×W×D) which have D-dimensional feature vectors at each of the H×W image locations. Furthermore, in some embodiments, a feature vector includes a set of values representing characteristics and/or

features of content (or an object) within a digital image. Indeed, in some embodiments, a feature vector includes a set of values corresponding to latent and/or patent attributes related to a digital image. For example, in some instances, a feature vector is a multi-dimensional dataset that represents features depicted within a digital image. In one or more embodiments, a feature vector includes a set of numeric metrics learned by a machine learning algorithm.
(296) FIG. **17** illustrates an architecture of the multi-attribute contrastive classification neural network in accordance with one or more embodiments. Indeed, in one or more embodiments, the scene-based image editing system **106** utilizes the multi-attribute contrastive classification neural network, as illustrated in FIG. **17**, with the embedding neural network, the localizer neural network, the multi-attention neural network, and the classifier neural network components to determine positive and negative attribute labels (e.g., from output attribute presence probabilities) for an object portrayed in a digital image.
(297) As shown in FIG. **17**, the scene-based image editing system **106** utilizes an embedding neural network within the multi-attribute contrastive classification neural network. In particular, as illustrated in FIG. **17**, the scene-based image editing system **106** utilizes a low-level embedding layer **1704** (e.g., embedding NN **1**) (e.g., of the embedding neural network **1604** of FIG. **16**) to generate a low-level attribute feature map **1710** from a digital image **1702**. Furthermore, as shown in FIG. **17**, the scene-based image editing system **106** utilizes a high-level embedding layer **1706** (e.g., embedding NN.sub.h) (e.g., of the embedding neural network **1604** of FIG. **16**) to generate a high-level attribute feature map **1708** from the digital image **1702**.
(298) In particular, in one or more embodiments, the scene-based image editing system **106** utilizes a convolutional neural network as an embedding neural network. For example, the scene-based image editing system **106** generates a D-dimensional image feature map f.sub.img(I)∈ custom character.sup.H×W×D with a spatial size H×W extracted from a convolutional neural network-based embedding neural network. In some instance, the scene-based image editing system **106** utilizes an output of the penultimate layer of ResNet-50 as the image feature map f.sub.img(I).
(299) As shown in FIG. **17**, the scene-based image editing system **106** extracts both a high-level attribute feature map **1708** and a low-level attribute feature map **1710** utilizing a high-level embedding layer and a low-level embedding layer of an embedding neural network. By extracting both the high-level attribute feature map **1708** and the low-level attribute feature map **1710** for the digital image **1702**, the scene-based image editing system **106** addresses the heterogeneity in features between different classes of attributes. Indeed, attributes span across a wide range of semantic levels.
(300) By utilizing both low-level feature maps and high-level feature maps, the scene-based image editing system **106** accurately predicts attributes across the wide range of semantic levels. For instance, the scene-based image editing system **106** utilizes low-level feature maps to accurately predict attributes such as, but not limited to, colors (e.g., red, blue, multicolored), patterns (e.g., striped, dotted, striped), geometry (e.g., shape, size, posture), texture (e.g., rough, smooth, jagged), or material (e.g., wooden, metallic, glossy, matte) of a portrayed object. Meanwhile, in one or more embodiments, the scene-based image editing system **106** utilizes high-level feature maps to accurately predict attributes such as, but not limited to, object states (e.g., broken, dry, messy, full, old) or actions (e.g., running, sitting, flying) of a portrayed object.
(301) Furthermore, as illustrated in FIG. **17**, the scene-based image editing system **106** generates an image-object feature map **1714**. In particular, as shown in FIG. **17**, the scene-based image editing system **106** combines an object-label embedding vector **1712** (e.g., such as the object-label embedding vector **1608** of FIG. **16**) from a label corresponding to the object (e.g., “chair”) with the high-level attribute feature map **1708** to generate the image-object feature map **1714** (e.g., such as the image-object feature map **1606** of FIG. **16**). As further shown in FIG. **17**, the scene-based image editing system **106** utilizes a feature composition module (e.g., f.sub.comp) that utilizes the object-label embedding vector **1712** and the high-level attribute feature map **1708** to output the image-

object feature map **1714**.

(302) In one or more embodiments, the scene-based image editing system **106** generates the image-object feature map **1714** to provide an extra signal to the multi-attribute contrastive classification neural network to learn the relevant object for which it is predicting attributes (e.g., while also encoding the features for the object). In particular, in some embodiments, the scene-based image editing system **106** incorporates the object-label embedding vector **1712** (as an input in a feature composition module f.sub.comp to generate the image-object feature map **1714**) to improve the classification results of the multi-attribute contrastive classification neural network by having the multi-attribute contrastive classification neural network learn to avoid unfeasible object-attribute combinations (e.g., a parked dog, a talking table, a barking couch). Indeed, in some embodiments, the scene-based image editing system **106** also utilizes the object-label embedding vector **1712** (as an input in the feature composition module f.sub.comp) to have the multi-attribute contrastive classification neural network learn to associate certain object-attribute pairs together (e.g., a ball is always round). In many instances, by guiding the multi-attribute contrastive classification neural network on what object it is predicting attributes for enables the multi-attribute contrastive classification neural network to focus on particular visual aspects of the object. This, in turn, improves the quality of extracted attributes for the portrayed object.

(303) In one or more embodiments, the scene-based image editing system **106** utilizes a feature composition module (e.g., f.sub.comp) to generate the image-object feature map **1714**. In particular, the scene-based image editing system **106** implements the feature composition module (e.g., f.sub.comp) with a gating mechanism in accordance with the following:

f.sub.comp(*f*.sub.img(*I*),ϕ.sub.o)=*f*.sub.img(*I*)⊙*f*.sub.gate(ϕ.sub.o)

and

f.sub.comp(ϕ.sub.o)=σ(*W*.sub.g2.Math.ReLU(*W*.sub.g1ϕ.sub.o+*b*.sub.g1)+*b*.sub.g2)

In the first function above, the scene-based image editing system **106** utilizes a channel-wise product (⊙) of the high-level attribute feature map f.sub.img(I) and a filter f.sub.gate of the object-label embedding vector ϕ.sub.o∈custom character.sup.d to generate an image-object feature map f.sub.comp(f.sub.img(I), ϕ.sub.o)∈custom character.sup.D.

(304) In addition, in the second function above, the scene-based image editing system **106** utilizes a sigmoid function σ in the f.sub.gate(ϕ.sub.o))∈custom character.sup.D that is broadcasted to match the feature map spatial dimension as a 2-layer multilayer perceptron (MLP). Indeed, in one or more embodiments, the scene-based image editing system **106** utilizes f.sub.gate as a filter that selects attribute features that are relevant to the object of interest (e.g., as indicated by the object-label embedding vector ϕ.sub.o). In many instances, the scene-based image editing system **106** also utilizes f.sub.gate to suppress incompatible object-attribute pairs (e.g., talking table). In some embodiments, the scene-based image editing system **106** can identify object-image labels for each object portrayed within a digital image and output attributes for each portrayed object by utilizing the identified object-image labels with the multi-attribute contrastive classification neural network.

(305) Furthermore, as shown in FIG. **17**, the scene-based image editing system **106** utilizes the image-object feature map **1714** with a localizer neural network **1716** to generate a localized image-object feature vector Z.sub.rel (e.g., as also shown in FIG. **16** as localizer neural network **1612** and Z.sub.rel). In particular, as shown in FIG. **17**, the scene-based image editing system **106** generates a localized object attention feature vector **1717** (e.g., G in FIG. **16**) that reflects a segmentation prediction of the portrayed object by utilizing the image-object feature map **1714** with a convolutional layer f.sub.rel of the localizer neural network **1716**. Then, as illustrated in FIG. **17**, the scene-based image editing system **106** combines the localized object attention feature vector **1717** with the image-object feature map **1714** to generate the localized image-object feature vector Z.sub.rel. As shown in FIG. **17**, the scene-based image editing system **106** utilizes matrix multiplication **1720** between the localized object attention feature vector **1717** and the image-object feature map **1714** to generate the localized image-object feature vector Z.sub.rel.

(306) In some instances, digital images may include multiple objects (and/or a background). Accordingly, in one or more embodiments, the scene-based image editing system **106** utilizes a localizer neural network to learn an improved feature aggregation that suppresses non-relevant-object regions (e.g., regions not reflected in a segmentation prediction of the target object to isolate the target object). For example, in reference to the digital image **1702**, the scene-based image editing system **106** utilizes the localizer neural network **1716** to localize an object region such that the multi-attribute contrastive classification neural network predicts attributes for the correct object (e.g., the portrayed chair) rather than other irrelevant objects (e.g., the portrayed horse). To do this, in some embodiments, the scene-based image editing system **106** utilizes a localizer neural network that utilizes supervised learning with object segmentation masks (e.g., ground truth relevant-object masks) from a dataset of labeled images (e.g., ground truth images as described below).
(307) To illustrate, in some instances, the scene-based image editing system **106** utilizes 2-stacked convolutional layers f.sub.rel (e.g., with a keel size of 1) followed by a spatial softmax to generate a localized object attention feature vector G (e.g., a localized object region) from an image-object feature map X∈custom character.sup.H×W×D in accordance with the following:
(308) $g = f_{\mathrm{rel}}(X), g \in \mathbb{R}^{H \times W}, G_{h,w} = \frac{\exp(g_h, w)}{\text{.Math.}_{h,w} \exp(g_h, w)}, g \in \mathbb{R}^{H \times W}$
For example, the localized object attention feature vector G includes a single plane of data that is H×W (e.g., a feature map having a single dimension). In some instances, the localized object attention feature vector G includes a feature map (e.g., a localized object attention feature map) that includes one or more feature vector dimensions.
(309) Then, in one or more embodiments, the scene-based image editing system **106** utilizes the localized object attention feature vector G.sub.h,w and the image-object feature map X.sub.h,w to generate the localized image-object feature vector Z.sub.rel in accordance with the following:
(310) $Z_{\mathrm{rel}} = \underset{h,w}{\text{.Math.}} G_{h,w} X_{h,w}$
In some instances, in the above function, the scene-based image editing system **106** pools H×W D-dimensional feature vectors X.sub.h,w (from the image-object feature map) in custom character .sup.D using weights from the localized object attention feature vector G.sub.h,w into a single D-dimensional feature vector Z.sub.rel.
(311) In one or more embodiments, in reference to FIG. **17**, the scene-based image editing system **106** trains the localizer neural network **1716** to learn the localized object attention feature vector **1717** (e.g., G) utilizing direct supervision with object segmentation masks **1718** (e.g., ground truth object segmentation masks **1618** from FIG. **16**).
(312) Furthermore, as shown in FIG. **17**, the scene-based image editing system **106** utilizes the image-object feature map **1714** with a multi-attention neural network **1722** to generate a multi-attention feature vector Z.sub.att (e.g., the multi-attention neural network **1614** and Z.sub.att of FIG. **16**). In particular, as shown in FIG. **17**, the scene-based image editing system **106** utilizes a convolutional layer f.sub.att (e.g., attention layers) with the image-object feature map **1714** to extract attention maps **1724** (e.g., Attention 1 through Attention k) (e.g., attention maps **1620** of FIG. **16**). Then, as further shown in FIG. **17**, the scene-based image editing system **106** passes (e.g., via linear projection) the extracted attention maps **424** (attention 1 through attention k) through a projection layer f.sub.proj to extract one or more attention features that are utilized to generate the multi-attention feature vector Z.sub.att.
(313) In one or more embodiments, the scene-based image editing system **106** utilizes the multi-attention feature vector Z.sub.att to accurately predict attributes of a portrayed object within a digital image by providing focus to different parts of the portrayed object and/or regions surrounding the portrayed object (e.g., attending to features at different spatial locations). To illustrate, in some instances, the scene-based image editing system **106** utilizes the multi-attention feature vector Z.sub.att to extract attributes such as “barefooted” or “bald-headed” by focusing on different parts of a person (i.e., an object) that is portrayed in a digital image. Likewise, in some

embodiments, the scene-based image editing system **106** utilizes the multi-attention feature vector Z.sub.att to distinguish between different activity attributes (e.g., jumping vs crouching) that may rely on information from surrounding context of the portrayed object.
(314) In certain instances, the scene-based image editing system **106** generates an attention map per attribute portrayed for an object within a digital image. For example, the scene-based image editing system **106** utilizes an image-object feature map with one or more attention layers to generate an attention map from the image-object feature map for each known attribute. Then, the scene-based image editing system **106** utilizes the attention maps with a projection layer to generate the multi-attention feature vector Z.sub.att. In one or more embodiments, the scene-based image editing system **106** generates various numbers of attention maps for various attributes portrayed for an object within a digital image (e.g., the system can generate an attention map for each attribute or a different number of attention maps than the number of attributes).
(315) Furthermore, in one or more embodiments, the scene-based image editing system **106** utilizes a hybrid shared multi-attention approach that allows for attention hops while generating the attention maps from the image-object feature map. For example, the scene-based image editing system **106** extracts M attention maps {A.sup.(m)}.sub.m=1.sup.M from an image-object feature map X utilizing a convolutional layer f.sub.att.sup.(m) (e.g., attention layers) in accordance with the following function:
(316) $E^{(m)} = f_{\text{att}}^{(m)}(X), E^{(m)} \in \mathbb{R}^{H \times W}, m = 1,$ .Math. $, M$and
$A_{h,w}^{(m)} = \frac{\exp(E_{h,w}^{(m)})}{\text{.Math.}_{h,w} \exp(E_{h,w}^{(m)})}, A_{h,w}^{(m)} \in \mathbb{R}^{H \times W}$
In some cases, the scene-based image editing system **106** utilizes a convolutional layer f.sub.att.sup.(m) that has a similar architecture to the 2-stacked convolutional layers f.sub.rel from function (3) above. By utilizing the approach outlined in second function above, the scene-based image editing system **106** utilizes a diverse set of attention maps that correspond to a diverse range of attributes.
(317) Subsequently, in one or more embodiments, the scene-based image editing system **106** utilizes the M attention maps (e.g., A.sub.h,w.sup.(m)) to aggregate M attention feature vectors ({r.sup.(m)}.sub.m=1.sup.M) from the image-object feature map X in accordance with the following function:
(318) $r^{(m)} = \underset{h,w}{\text{.Math.}} A_{h,w}^{(m)} X_{h,w}, r^{(m)} \in \mathbb{R}^D$
Moreover, in reference to FIG. **17**, the scene-based image editing system **106** passes the M attention feature vectors ({r.sup.(m)}.sub.i=1.sup.M) through a projection layer f.sub.proj.sup.(m) to extract one or more attention feature vectors z.sup.(m) in accordance with the following function:
z.sub.att.sup.(m)=*f*.sub.proj.sup.(m)(*r*.sup.(m)),*z*.sub.att.sup.(m)∈custom character .sup.D.sup.proj
Then, in one or more embodiments, the scene-based image editing system **106** generates the multi-attention feature vector Z.sub.att by concatenating the individual attention feature vectors z.sub.att.sup.(m) in accordance with the following function:
Z.sub.att=concat([z.sub.att.sup.(1), . . . ,*z*.sub.att.sup.(M)])
(319) In some embodiments, the scene-based image editing system **106** utilizes a divergence loss with the multi-attention neural network in the M attention hops approach. In particular, the scene-based image editing system **106** utilizes a divergence loss that encourages attention maps to focus on different (or unique) regions of a digital image (from the image-object feature map). In some cases, the scene-based image editing system **106** utilizes a divergence loss that promotes diversity between attention features by minimizing a cosine similarity (e.g., custom character.sub.2-norm) between attention weight vectors (e.g., E) of attention features. For instance, the scene-based image editing system **106** determines a divergence loss custom character.sub.div in accordance with the following function:
following function:

(320) $\mathcal{L}_{\text{div}} = \text{.Math.}_{m \neq n} \frac{\text{.Math. } E^{(m)}, E^{(n)} \text{ .Math.}}{\text{.Math. } E^{(m)} \text{ .Math.}_2 \text{ .Math. } E^{(n)} \text{ .Math.}_2}$
In one or more embodiments, the scene-based image editing system **106** utilizes the divergence loss custom character.sub.div to learn parameters of the multi-attention neural network **1722** and/or the multi-attribute contrastive classification neural network (as a whole).
(321) Furthermore, as shown in FIG. **17**, the scene-based image editing system **106** also generates a localized low-level attribute feature vector Z.sub.low (e.g., Z.sub.low of FIG. **16**). Indeed, as illustrated in FIG. **17**, the scene-based image editing system **106** generates the localized low-level attribute feature vector Z.sub.low by combining the low-level attribute feature map **1710** and the localized object attention feature vector **1717**. For example, as shown in FIG. **17**, the scene-based image editing system **106** combines the low-level attribute feature map **1710** and the localized object attention feature vector **1717** utilizing matrix multiplication **1726** to generate the localized low-level attribute feature vector Z.sub.low.
(322) By generating and utilizing the localized low-level attribute feature vector Z.sub.low, in one or more embodiments, the scene-based image editing system **106** improves the accuracy of low-level features (e.g., colors, materials) that are extracted for an object portrayed in a digital image. In particular, in one or more embodiments, the scene-based image editing system **106** pools low-level features (as represented by a low-level attribute feature map from a low-level embedding layer) from a localized object attention feature vector (e.g., from a localizer neural network). Indeed, in one or more embodiments, by pooling low-level features from the localized object attention feature vector utilizing a low-level feature map, the scene-based image editing system **106** constructs a localized low-level attribute feature vector Z.sub.low.
(323) As further shown in FIG. **17**, the scene-based image editing system **106** utilizes a classifier neural network **1732** (f.sub.classifier) (e.g., the classifier neural network **1624** of FIG. **16**) with the localized image-object feature vector Z.sub.rel, the multi-attention feature vector Z.sub.att, and the localized low-level attribute feature vector Z.sub.low to determine positive attribute labels **1728** and negative attribute labels **1730** for the object (e.g., "chair") portrayed within the digital image **1702**. In some embodiments, the scene-based image editing system **106** utilizes a concatenation of the localized image-object feature vector Z.sub.rel, the multi-attention feature vector Z.sub.att, and the localized low-level attribute feature vector Z.sub.low as input in a classification layer of the classifier neural network **1732** (f.sub.classifier). Then, as shown in FIG. **17**, the classifier neural network **1732** (f.sub.classifier) generates positive attribute labels **1728** (e.g., red, bright red, clean, giant, wooden) and also generates negative attribute labels **1730** (e.g., blue, stuffed, patterned, multicolored) for the portrayed object in the digital image **1702**.
(324) In one or more embodiments, the scene-based image editing system **106** utilizes a classifier neural network that is a 2-layer MLP. In some cases, the scene-based image editing system **106** utilizes a classifier neural network that includes various amounts of hidden units and output logic values followed by sigmoid. In some embodiments, the classifier neural network is trained by the scene-based image editing system **106** to generate both positive and negative attribute labels. Although one or more embodiments described herein utilize a 2-layer MLP, in some instances, the scene-based image editing system **106** utilizes a linear layer (e.g., within the classifier neural network, for the f.sub.gate, and for the image-object feature map).
(325) Furthermore, in one or more embodiments, the scene-based image editing system **106** utilizes various combinations of the localized image-object feature vector Z.sub.rel, the multi-attention feature vector Z.sub.att, and the localized low-level attribute feature vector Z.sub.low with the classifier neural network to extract attributes for an object portrayed in a digital image. For example, in certain instances, the scene-based image editing system **106** provides the localized image-object feature vector Z.sub.rel and the multi-attention feature vector Z.sub.att to extract attributes for the portrayed object. In some instances, as shown in FIG. **17**, the scene-based image editing system **106** utilizes a concatenation of each the localized image-object feature vector

Z.sub.rel, the multi-attention feature vector Z.sub.att, and the localized low-level attribute feature vector Z.sub.low with the classifier neural network.

(326) In one or more embodiments, the scene-based image editing system **106** utilizes the classifier neural network **1732** to generate prediction scores corresponding to attribute labels as outputs. For, example, the classifier neural network **1732** can generate a prediction score for one or more attribute labels (e.g., a score of 0.04 for blue, a score of 0.9 for red, a score of 0.4 for orange). Then, in some instances, the scene-based image editing system **106** utilizes attribute labels that correspond to prediction scores that satisfy a threshold prediction score. Indeed, in one or more embodiments, the scene-based image editing system **106** selects various attribute labels (both positive and negative) by utilizing output prediction scores for attributes from a classifier neural network.

(327) Although one or more embodiments herein illustrate the scene-based image editing system **106** utilizing a particular embedding neural network, localizer neural network, multi-attention neural network, and classifier neural network, the scene-based image editing system **106** can utilize various types of neural networks for these components (e.g., CNN, FCN). In addition, although one or more embodiments herein describe the scene-based image editing system **106** combining various feature maps (and/or feature vectors) utilizing matrix multiplication, the scene-based image editing system **106**, in some embodiments, utilizes various approaches to combine feature maps (and/or feature vectors) such as, but not limited to, concatenation, multiplication, addition, and/or aggregation. For example, in some implementations, the scene-based image editing system **106** combines a localized object attention feature vector and an image-object feature map to generate the localized image-object feature vector by concatenating the localized object attention feature vector and the image-object feature map.

(328) Thus, in some cases, the scene-based image editing system **106** utilizes an attribute classification neural network (e.g., a multi-attribute contrastive classification neural network) to determine objects attributes of objects portrayed in a digital image or otherwise determined attributes of portrayed semantic areas. In some cases, the scene-based image editing system **106** adds object attributes or other attributes determined for a digital image to a semantic scene graph for the digital image. In other words, the scene-based image editing system **106** utilizes the attribute classification neural network in generating semantic scene graphs for digital images. In some implementations, however, the scene-based image editing system **106** stores the determined object attributes or other attributes in a separate storage location.

(329) Further, in one or more embodiments, the scene-based image editing system **106** facilitates modifying object attributes of objects portrayed in a digital image by modifying one or more object attributes in response to user input. In particular, in some cases, the scene-based image editing system **106** utilizes a machine learning model, such as an attribute modification neural network to modify object attributes. FIG. **18** illustrates an attribute modification neural network utilized by the scene-based image editing system **106** to modify object attributes in accordance with one or more embodiments.

(330) In one or more embodiments, an attribute modification neural network includes a computer-implemented neural network that modifies specified object attributes of an object (or specified attributes of other specified semantic areas). In particular, in some embodiments, an attribute modification neural network includes a computer-implemented neural network that receives user input targeting an object attribute and indicating a change to the object attribute and modifies the object attribute in accordance with the indicated change. In some cases, an attribute modification neural network includes a generative network.

(331) As shown in FIG. **18**, the scene-based image editing system **106** provides an object **1802** (e.g., a digital image that portrays the object **1802**) and modification input **1804***a*-**1804***b* to an object modification neural network **1806**. In particular, FIG. **18** shows the modification input **1804***a*-**1804***b* including input for the object attribute to be changed (e.g., the black color of the

object **1802**) and input for the change to occur (e.g., changing the color of the object **1802** to white).

(332) As illustrated by FIG. **18**, the object modification neural network **1806** utilizes an image encoder **1808** to generate visual feature maps **1810** from the object **1802**. Further, the object modification neural network **1806** utilizes a text encoder **1812** to generate textual features **1814*a*-1814*b*** from the modification input **1804*a*-1804*b***. In particular, as shown in FIG. **18**, the object modification neural network **1806** generates the visual feature maps **1810** and the textual features **1814*a*-1814*b*** within a joint embedding space **1816** (labeled “visual-semantic embedding space” or “VSE space”).

(333) In one or more embodiments, the object modification neural network **1806** performs text-guided visual feature manipulation to ground the modification input **1804*a*-1804*b*** on the visual feature maps **1810** and manipulate the corresponding regions of the visual feature maps **1810** with the provided textual features. For instance, as shown in FIG. **18**, the object modification neural network **1806** utilizes an operation **1818** (e.g., a vector arithmetic operation) to generate manipulated visual feature maps **1820** from the visual feature maps **1810** and the textual features **1814*a*-1814*b***.

(334) As further shown in FIG. **18**, the object modification neural network **1806** also utilizes a fixed edge extractor **1822** to extract an edge **1824** (a boundary) of the object **1802**. In other words, the object modification neural network **1806** utilizes the fixed edge extractor **1822** to extract the edge **1824** of the area to be modified.

(335) Further, as shown, the object modification neural network **1806** utilizes a decoder **1826** to generate the modified object **1828**. In particular, the decoder **1826** generates the modified object **1828** from the edge **1824** extracted from the object **1802** and the manipulated visual feature maps **1820** generated from the object **1802** and the modification input **1804*a*-1804*b***.

(336) In one or more embodiments, the scene-based image editing system **106** trains the object modification neural network **1806** to handle open-vocabulary instructions and open-domain digital images. For instance, in some cases, the scene-based image editing system **106** trains the object modification neural network **1806** utilizing a large-scale image-caption dataset to learn a universal visual-semantic embedding space. In some cases, the scene-based image editing system **106** utilizes convolutional neural networks and/or long short-term memory networks as the encoders of the object modification neural network **1806** to transform digital images and text input into the visual and textual features.

(337) The following provides a more detailed description of the text-guided visual feature manipulation. As previously mentioned, in one or more embodiments, the scene-based image editing system **106** utilizes the joint embedding space **1816** to manipulate the visual feature maps **1810** with the text instructions of the modification input **1804*a*-1804*b*** via vector arithmetic operations. When manipulating certain objects or object attributes, the object modification neural network **1806** aims to modify only specific regions while keeping other regions unchanged. Accordingly, the object modification neural network **1806** conducts vector arithmetic operations between the visual feature maps **1810** represented as V∈custom character.sup.1024×7×7 and the textual features **1814*a*-1814*b*** (e.g., represented as textual feature vectors).

(338) For instance, in some cases, the object modification neural network **1806** identifies the regions in the visual feature maps **1810** to manipulate (i.e., grounds the modification input **1804*a*-1804*b***) on the spatial feature map. In some cases, the object modification neural network **1806** provides a soft grounding for textual queries via a weighted summation of the visual feature maps **1810**. In some cases, the object modification neural network **1806** uses the textual features **1814*a*-1814*b*** (represented as t∈custom character.sup.1024×1) as weights to compute the weighted summation of the visual feature maps **1810** g=t.sup.TV. Using this approach, the object modification neural network **1806** provides a soft grounding map g∈custom character.sup.7×7, which roughly localizes corresponding regions in the visual feature maps **1810** related to the text

instructions.

(339) In one or more embodiments, the object modification neural network **1806** utilizes the grounding map as location-adaptive coefficients to control the manipulation strength at different locations. In some cases, the object modification neural network **1806** utilizes a coefficient α to control the global manipulation strength, which enables continuous transitions between source images and the manipulated ones. In one or more embodiments, the scene-based image editing system **106** denotes the visual feature vector at spatial location (i,j) (where i,j∈{0, 1, . . . 6}) in the visual feature map V∈custom character.sup.1024×7×7 as v.sup.i,j∈custom character .sup.1024.

(340) The scene-based image editing system **106** utilizes the object modification neural network **1806** to perform various types of manipulations via the vector arithmetic operations weighted by the soft grounding map and the coefficient α. For instance, in some cases, the scene-based image editing system **106** utilizes the object modification neural network **1806** to change an object attribute or a global attribute. The object modification neural network **1806** denotes the textual feature embeddings of the source concept (e.g., “black triangle”) and the target concept (e.g., “white triangle”) as t.sub.1 and t.sub.2, respectively. The object modification neural network **1806** performs the manipulation of image feature vector v.sup.i,j at location (i, j) as follows:

v.sub.m.sup.i,j=*v*.sup.i,j−*α*custom character*v*.sup.i,j,*t*.sub.1custom character*t*.sub.1+*α*custom character*v*.sup.i,j,*t*.sub.1custom character*t*.sub.2,

where i, j∈{0, 1, . . . 6} and v.sub.m.sup.i,j is the manipulated visual feature vector at location (i, j) of the 7×7 feature map.

(341) In one or more embodiments, the object modification neural network **1806** removes the source features t.sub.1 and adds the target features t.sub.2 to each visual feature vector v.sup.i,j. Additionally, custom characterv.sup.i,j, t.sub.1custom character represents the value of the soft grounding map at location (i, j), calculated as the dot product of the image feature vector and the source textual features. In other words, the value represents the projection of the visual embedding v.sup.i,j onto the direction of the textual embedding t.sub.1. In some cases, object modification neural network **1806** utilizes the value as a location-adaptive manipulation strength to control which regions in the image should be edited. Further, the object modification neural network **1806** utilizes the coefficient α as a hyper-parameter that controls the image-level manipulation strength. By smoothly increasing α, the object modification neural network **1806** achieves smooth transitions from source to target attributes.

(342) In some implementations, the scene-based image editing system **106** utilizes the object modification neural network **1806** to remove a concept (e.g., an object attribute, an object, or other visual elements) from a digital image (e.g., removing an accessory from a person). In some instances, the object modification neural network **1806** denotes the semantic embedding of the concept to be removed as t. Accordingly, the object modification neural network **1806** performs the removing operation as follows:

v.sub.m.sup.i,j=*v*.sup.i,j−*α*custom character*v*.sup.i,j,*t*custom character*t*

(343) Further, in some embodiments, the scene-based image editing system **106** utilizes the object modification neural network **1806** to modify the degree to which an object attribute (or other attribute of a semantic area) appears (e.g., making a red apple less red or increasing the brightness of a digital image). In some cases, the object modification neural network **1806** controls the strength of an attribute via the hyper-parameter α. By smoothly adjusting α, the object modification neural network **1806** gradually strengthens or weakens the degree to which an attribute appears as follows:

v.sub.m.sup.i,j=*v*.sup.i,j±*α*custom character*v*.sup.i,j,*t*custom character*t*

(344) After deriving the manipulated feature map V.sub.m∈custom character.sup.1024×7×7 the object modification neural network **1806** utilizes the decoder **1826** (an image decoder) to generate a manipulated image (e.g., the modified object **1828**). In one or more embodiments, the scene-based

image editing system **106** trains the object modification neural network **1806** as described by F. Faghri et al., *Vse++: Improving visual-semantic Embeddings with Hard Negatives*, arXiv:1707.05612, 2017, which is incorporated herein by reference in its entirety. In some cases, the decoder **1826** takes 1024×7×7 features maps as input and is composed of seven ResNet blocks with upsampling layers in between, which generates 256×256 images. Also, in some instances, the scene-based image editing system **106** utilizes a discriminator that includes a multi-scale patch-based discriminator. In some implementations, the scene-based image editing system **106** trains the decoder **1826** with GAN loss, perceptual loss, and discriminator feature matching loss. Further, in some embodiments, the fixed edge extractor **1822** includes a bi-directional cascade network.
(345) FIGS. **19**A-**19**C illustrate a graphical user interface implemented by the scene-based image editing system **106** to facilitate modifying object attributes of objects portrayed in a digital image in accordance with one or more embodiments. Indeed, though FIGS. **19**A-**19**C particularly show modifying object attributes of objects, it should be noted that the scene-based image editing system **106** similarly modifies attributes of other semantic areas (e.g., background, foreground, ground, sky, etc.) of a digital image in various embodiments.
(346) Indeed, as shown in FIG. **19**A, the scene-based image editing system **106** provides a graphical user interface **1902** for display on a client device **1904** and provides a digital image **1906** for display within the graphical user interface **1902**. As further shown, the digital image **1906** portrays an object **1908**.
(347) As further shown in FIG. **19**A, in response to detecting a user interaction with the object **1908**, the scene-based image editing system **106** provides an attribute menu **1910** for display within the graphical user interface **1902**. In some embodiments, the attribute menu **1910** provides one or more object attributes of the object **1908**. Indeed, FIG. **19**A shows that the attribute menu **1910** provides object attributes indicators **1912***a*-**1912***c*, indicating the shape, color, and material of the object **1908**, respectively. It should be noted, however, that various alternative or additional object attributes are provided in various embodiments.
(348) In one or more embodiments, the scene-based image editing system **106** retrieves the object attributes for the object attribute indicators **1912***a*-**1912***c* from a semantic scene graph generated for the digital image **1906**. Indeed, in some implementations, the scene-based image editing system **106** generates a semantic scene graph for the digital image **1906** (e.g., before detecting the user interaction with the object **1908**). In some cases, the scene-based image editing system **106** determines the object attributes for the object **1908** utilizing an attribute classification neural network and includes the determined object attributes within the semantic scene graph. In some implementations, the scene-based image editing system **106** retrieves the object attributes from a separate storage location.
(349) As shown in FIG. **19**B, the scene-based image editing system **106** detects a user interaction with the object attribute indicator **1912***c*. Indeed, in one or more embodiments, the object attribute indicators **1912***a*-**1912***c* are interactive. As shown, in response to detecting the user interaction, the scene-based image editing system **106** removes the corresponding object attribute of the object **1908** from display. Further, in response to detecting the user interaction, the scene-based image editing system **106** provides a digital keyboard **1914** for display within the graphical user interface **1902**. Thus, the scene-based image editing system **106** provides a prompt for entry of textual user input. In some cases, upon detecting the user interaction with the object attribute indicator **1912***c*, the scene-based image editing system **106** maintains the corresponding object attribute for display, allowing user interactions to remove the object attribute in confirming that the object attribute has been targeted for modification.
(350) As shown in FIG. **19**C, the scene-based image editing system **106** detects one or more user interactions with the digital keyboard **1914** displayed within the graphical user interface **1902**. In particular, the scene-based image editing system **106** receives textual user input provided via the digital keyboard **1914**. The scene-based image editing system **106** further determines that the

textual user input provides a change to the object attribute corresponding to the object attribute indicator **1912***c*. Additionally, as shown, the scene-based image editing system **106** provides the textual user input for display as part of the object attribute indicator **1912***c*.

(351) In this case, the user interactions with the graphical user interface **1902** provide instructions to change a material of the object **1908** from a first material (e.g., wood) to a second material (e.g., metal). Thus, upon receiving the textual user input regarding the second material, the scene-based image editing system **106** modifies the digital image **1906** by modifying the object attribute of the object **1908** to reflect the user-provided second material.

(352) In one or more embodiments, the scene-based image editing system **106** utilizes an attribute modification neural network to change the object attribute of the object **1908**. In particular, as described above with reference to FIG. **18**, the scene-based image editing system **106** provides the digital image **1906** as well as the modification input composed of the first material and the second material provided by the textual user input to the attribute modification neural network. Accordingly, the scene-based image editing system **106** utilizes the attribute modification neural network to provide a modified digital image portraying the object **1908** with the modified object attribute as output.

(353) FIGS. **20**A-**20**C illustrate another graphical user interface implemented by the scene-based image editing system **106** to facilitate modifying object attributes of objects portrayed in a digital image in accordance with one or more embodiments. As shown in FIG. **20**A, the scene-based image editing system **106** provides a digital image **2006** portraying an object **2008** for display within a graphical user interface **2002** of a client device **2004**. Further, upon detecting a user interaction with the object **2008**, the scene-based image editing system **106** provides an attribute menu **2010** having attribute object indicators **2012***a*-**2012***c* listing object attributes of the object **2008**.

(354) As shown in FIG. **20**B, the scene-based image editing system **106** detects an additional user interaction with the object attribute indicator **2012***a*. In response to detecting the additional user interaction, the scene-based image editing system **106** provides an alternative attribute menu **2014** for display within the graphical user interface **2002**. In one or more embodiments, the alternative attribute menu **2014** includes one or more options for changing a corresponding object attribute. Indeed, as illustrated in FIG. **20**B, the alternative attribute menu **2014** includes alternative attribute indicators **2016***a*-**2016***c* that provide object attributes that could be used in place of the current object attribute for the object **2008**.

(355) As shown in FIG. **20**C, the scene-based image editing system **106** detects a user interaction with the alternative attribute indicator **2016***b*. Accordingly, the scene-based image editing system **106** modifies the digital image **2006** by modifying the object attribute of the object **2008** in accordance with the user input with the alternative attribute indicator **2016***b*. In particular, the scene-based image editing system **106** modifies the object **2008** to reflect the alternative object attribute associated with the alternative attribute indicator **2016***b*.

(356) In one or more embodiments, the scene-based image editing system **106** utilizes a textual representation of the alternative object attribute in modifying the object **2008**. For instance, as discussed above, the scene-based image editing system **106** provides the textual representation as textual input to an attribute modification neural network and utilizes the attribute modification neural network to output a modified digital image in which the object **2008** reflects the targeted change in its object attribute.

(357) FIGS. **21**A-**21**C illustrate another graphical user interface implemented by the scene-based image editing system **106** to facilitate modifying object attributes of objects portrayed in a digital image in accordance with one or more embodiments. As shown in FIG. **21**A, the scene-based image editing system **106** provides a digital image **2106** portraying an object **2108** for display within a graphical user interface **2102** of a client device **2104**. Further, upon detecting a user interaction with the object **2108**, the scene-based image editing system **106** provides an attribute menu **2110** having attribute object indicators **2112***a*-**2012***c* listing object attributes of the object **2108**.

(358) As shown in FIG. **21**B, the scene-based image editing system **106** detects an additional user interaction with the object attribute indicator **2112***b*. In response to detecting the additional user interaction, the scene-based image editing system **106** provides a slider bar **2114** for display within the graphical user interface **2102**. In one or more embodiments, the slider bar **2114** includes a slider element **2116** that indicates a degree to which the corresponding object attribute appears in the digital image **2106** (e.g., the strength or weakness of its presence in the digital image **2106**).
(359) As shown in FIG. **21**C, the scene-based image editing system **106** detects a user interaction with the slider element **2116** of the slider bar **2114**, increasing the degree to which the corresponding object attribute appears in the digital image. Accordingly, the scene-based image editing system **106** modifies the digital image **2106** by modifying the object **2108** to reflect the increased strength in the appearance of the corresponding object attribute.
(360) In particular, in one or more embodiments, the scene-based image editing system **106** utilizes an attribute modification neural network to modify the digital image **2106** in accordance with the user interaction. Indeed, as described above with reference to FIG. **18**, the scene-based image editing system **106** is capable of modifying the strength or weakness of the appearance of an object attribute via the coefficient α. Accordingly, in one or more embodiments, the scene-based image editing system **106** adjusts the coefficient α based on the positioning of the slider element **2116** via the user interaction.
(361) By facilitating image modifications that target particular object attributes as described above, the scene-based image editing system **106** provides improved flexibility and efficiency when compared to conventional systems. Indeed, the scene-based image editing system **106** provides a flexible, intuitive approach that visually displays descriptions of an object's attributes and allows user input that interacts with those descriptions to change the attributes. Thus, rather than requiring tedious, manual manipulation of an object attribute as is typical under many conventional systems, the scene-based image editing system **106** allows user interactions to target object attributes at a high level of abstraction (e.g., without having to interact at the pixel level). Further, as scene-based image editing system **106** enables modifications to object attributes via relatively few user interactions with provided visual elements, the scene-based image editing system **106** implements a graphical user interface that provides improved efficiency.
(362) As previously mentioned, in one or more embodiments, the scene-based image editing system **106** further uses a semantic scene graph generated for a digital image to implement relationship-aware object modifications. In particular, the scene-based image editing system **106** utilizes the semantic scene graph to inform the modification behaviors of objects portrayed in a digital image based on their relationships with one or more other objects in the digital image. FIGS. **22**A-**25**D illustrate implementing relationship-aware object modifications in accordance with one or more embodiments.
(363) Indeed, many conventional systems are inflexible in that they require different objects to be interacted with separately for modification. This is often the case even where the different objects are to be modified similarly (e.g., similarly resized or moved). For instance, conventional systems often require separate workflows to be executed via user interactions to modify separate objects or, at least, to perform the preparatory steps for the modification (e.g., outlining the objects and/or separating the objects from the rest of the image). Further, conventional systems typically fail to accommodate relationships between objects in a digital image when executing a modification. Indeed, these systems may modify a first object within a digital image but fail to execute a modification on a second object in accordance with a relationship between the two objects. Accordingly, the resulting modified image can appear unnatural or aesthetically confusing as it does not properly reflect the relationship between the two objects.
(364) Accordingly, conventional systems are also often inefficient in that they require a significant number of user interactions to modify separate objects portrayed in a digital image. Indeed, as mentioned, conventional systems often require separate workflows to be performed via user

interactions to execute many of the steps needed in modifying separate objects. Thus, many of the user interactions are redundant in that a user interaction is received, processed, and responded to multiple times for the separate objects. Further, when modifying an object having a relationship with another object, conventional systems require additional user interactions to modify the other object in accordance with that relationship. Thus, these systems unnecessarily duplicate the interactions used (e.g., interactions for moving an object then moving a related object) to perform separate modifications on related objects even where the relationship is suggestive as to the modification to be performed.

(365) The scene-based image editing system **106** provides more flexibility and efficiency over conventional systems by implementing relationship-aware object modifications. Indeed, as will be discussed, the scene-based image editing system **106** provides a flexible, simplified process for selecting related objects for modification. Accordingly, the scene-based image editing system **106** flexibly allows user interactions to select and modify multiple objects portrayed in a digital image via a single workflow. Further, the scene-based image editing system **106** facilitates the intuitive modification of related objects so that the resulting modified image continues to reflect that relationship. As such, digital images modified by the scene-based image editing system **106** provide a more natural appearance when compared to conventional systems.

(366) Further, by implementing a simplified process for selecting and modifying related objects, the scene-based image editing system **106** improves efficiency. In particular, the scene-based image editing system **106** implements a graphical user interface that reduces the user interactions required for selecting and modifying multiple, related objects. Indeed, as will be discussed, the scene-based image editing system **106** processes a relatively small number of user interactions with one object to anticipate, suggest, and/or execute modifications to other objects thus eliminating the need for additional user interactions for those modifications.

(367) For instance, FIGS. **22**A-**22**D illustrate a graphical user interface implemented by the scene-based image editing system **106** to facilitate a relationship-aware object modification in accordance with one or more embodiments. Indeed, as shown in FIG. **22**A, the scene-based image editing system **106** provides, for display within a graphical user interface **2202** of a client device **2204**, a digital image **2206** that portrays objects **2208***a*-**2208***b* and object **2220**. In particular, the digital image **2206** portrays a relationship between the objects **2208***a*-**2208***b* in that the object **2208***a* is holding the object **2208***b*.

(368) In one or more embodiments, the scene-based image editing system **106** references the semantic scene graph previously generated for the digital image **2206** to identify the relationship between the objects **2208***a*-**2208***b*. Indeed, as previously discussed, in some cases, the scene-based image editing system **106** includes relationships among the objects of a digital image in the semantic scene graph generated for the digital image. For instance, in one or more embodiments, the scene-based image editing system **106** utilizes a machine learning model, such as one of the models (e.g., the clustering and subgraph proposal generation model) discussed above with reference to FIG. **15**, to determine the relationships between objects. Accordingly, the scene-based image editing system **106** includes the determined relationships within the representation of the digital image in the semantic scene graph. Further, the scene-based image editing system **106** determines the relationship between the objects **2208***a*-**2208***b* for inclusion in the semantic scene graph before receiving user interactions to modify either one of the objects **2208***a*-**2208***b*.

(369) Indeed, FIG. **22**A illustrates a semantic scene graph component **2210** from a semantic scene graph of the digital image **2206**. In particular, the semantic scene graph component **2210** includes a node **2212***a* representing the object **2208***a* and a node **2212***b* representing the object **2208***b*. Further, the semantic scene graph component **2210** includes relationship indicators **2214***a*-**2214***b* associated with the nodes **2212***a*-**2212***b*. The relationship indicators **2214***a*-**2214***b* indicate the relationship between the objects **2208***a*-**2208***b* in that the object **2208***a* is holding the object **2208***b*, and the object **2208***b* is conversely being held by the object **2208***a*.

(370) As further shown, the semantic scene graph component **2210** includes behavior indicators **2216***a*-**2216***b* associated with the relationship indicator **2214***b*. The behavior indicators **2216***a*-**2216***b* assign a behavior to the object **2208***b* based on its relationship with the object **2208***a*. For instance, the behavior indicator **2216***a* indicates that, because the object **2208***b* is held by the object **2208***a*, the object **2208***b* moves with the object **2208***a*. In other words, the behavior indicator **2216***a* instructs the scene-based image editing system **106** to move the object **2208***b* (or at least suggest that the object **2208***b* be moved) when moving the object **2208***a*. In one or more embodiments, the scene-based image editing system **106** includes the behavior indicators **2216***a*-**2216***b* within the semantic scene graph based on the behavioral policy graph used in generating the semantic scene graph. Indeed, in some cases, the behaviors assigned to a “held by” relationship (or other relationships) vary based on the behavioral policy graph used. Thus, in one or more embodiments, the scene-based image editing system **106** refers to a previously generated semantic scene graph to identify relationships between objects and the behaviors assigned based on those relationships.
(371) It should be noted that the semantic scene graph component **2210** indicates that the behaviors of the behavior indicators **2216***a*-**2216***b* are assigned to the object **2208***b* but not the object **2208***a*. Indeed, in one or more, the scene-based image editing system **106** assigns behavior to an object based on its role in the relationship. For instance, while it may be appropriate to move a held object when the holding object is moved, the scene-based image editing system **106** determines that the holding object does not have to move when the held object is moved in some embodiments. Accordingly, in some implementations, the scene-based image editing system **106** assigns different behaviors to different objects in the same relationship.
(372) As shown in FIG. **22**B, the scene-based image editing system **106** determines a user interaction selecting the object **2208***a*. For instance, the scene-based image editing system **106** determines that user interaction targets the object **2208***a* for modification. As further shown, the scene-based image editing system **106** provides a visual indication **2218** for display to indicate the selection of the object **2208***a*.
(373) As illustrated by FIG. **22**C, in response to detecting the user interaction selecting the object **2208***a*, the scene-based image editing system **106** automatically selects the object **2208***b*. For instance, in one or more embodiments, upon detecting the user interaction selecting the object **2208***a*, the scene-based image editing system **106** refers to the semantic scene graph generated for the digital image **2206** (e.g., the semantic scene graph component **2210** that corresponds to the object **2208***a*). Based on the information represented in the semantic scene graph, the scene-based image editing system **106** determines that there is another object in the digital image **2206** that has a relationship with the object **2208***a*. Indeed, the scene-based image editing system **106** determines that the object **2208***a* is holding the object **2208***b*. Conversely, the scene-based image editing system **106** determines that the object **2208***b* is held by the object **2208***a*.
(374) Because the objects **2208***a*-**2208***b* have a relationship, the scene-based image editing system **106** adds the object **2208***b* to the selection. As shown in FIG. **22**C, the scene-based image editing system **106** modifies the visual indication **2218** of the selection to indicate that the object **2208***b* has been added to the selection. Though FIG. **22**C illustrates an automatic selection of the object **2208***b*, in some cases, the scene-based image editing system **106** selects the object **2208***b* based on a behavior assigned to the object **2208***b* within the semantic scene graph in accordance with its relationship with the object **2208***a*. Indeed, in some cases, the scene-based image editing system **106** specifies when a relationship between objects leads to the automatic selection of one object upon the user selection of another object (e.g., via a “selects with” behavior). As shown in FIG. **22**C, however, the scene-based image editing system **106** automatically selects the object **2208***b* by default in some instances.
(375) In one or more embodiments, the scene-based image editing system **106** surfaces object masks for the object **2208***a* and the object **2208***b* based on their inclusion within the selection. Indeed, the scene-based image editing system **106** surfaces pre-generated object masks for the

objects **2208***a*-**2208***b* in anticipation of a modification to the objects **2208***a*-**2208***b*. In some cases, the scene-based image editing system **106** retrieves the pre-generated object masks from the semantic scene graph for the digital image **2206** or retrieves a storage location for the pre-generated object masks. In either case, the object masks are readily available at the time the objects **2208***a*-**2208***b* are included in the selection and before modification input has been received.
(376) As further shown in FIG. **22**C, the scene-based image editing system **106** provides an option menu **2222** for display within the graphical user interface **2202**. In one or more embodiments, the scene-based image editing system **106** determines that at least one of the modification options from the option menu **222** would apply to both of the objects **2208***a*-**2208***b* if selected. In particular, the scene-based image editing system **106** determines that, based on behavior assigned to the object **2208***b*, a modification selected for the object **2208***a* would also apply to the object **2208***b*.
(377) Indeed, in one or more embodiments, in addition to determining the relationship between the objects **2208***a*-**2208***b*, the scene-based image editing system **106** references the semantic scene graph for the digital image **2206** to determine the behaviors that have been assigned based on that relationship. In particular, the scene-based image editing system **106** references the behavior indicators associated with the relationship between the objects **2208***a*-**2208***b* (e.g., the behavior indicators **2216***a*-**2216***b*) to determine which behaviors are assigned to the objects **2208***a*-**2208***b* based on their relationship. Thus, by determining the behaviors assigned to the object **2208***b*, the scene-based image editing system **106** determines how to respond to potential edits.
(378) For instance, as shown in FIG. **22**D, the scene-based image editing system **106** deletes the objects **2208***a*-**2208***b* together. For instance, in some cases, the scene-based image editing system **106** deletes the objects **2208***a*-**2208***b* in response to detecting a selection of the option **2224** presented within the option menu **2222**. Accordingly, while the object **2208***a* was targeted for deletion via user interactions, the scene-based image editing system **106** includes the object **2208***b* in the deletion operation based on the behavior assigned to the object **2208***b* via the semantic scene graph (i.e., the "deletes with" behavior). Thus, in some embodiments, the scene-based image editing system **106** implements relationship-aware object modifications by deleting objects based on their relationships to other objects.
(379) As previously suggested, in some implementations, the scene-based image editing system **106** only adds an object to a selection if its assigned behavior specifies that it should be selected with another object. At least, in some cases, the scene-based image editing system **106** only adds the object before receiving any modification input if its assigned behavior specifies that it should be selected with another object. Indeed, in some instances, only a subset of potential edits to a first object are applicable to a second object based on the behaviors assigned to that second object. Thus, including the second object in the selection of the first object before receiving modification input risks violating the rules set forth by the behavioral policy graph via the semantic scene graph if there is not a behavior providing for automatic selection. To avoid this risk, in some implementations, the scene-based image editing system **106** waits until modification input has been received before determining whether to add the second object to the selection. In one or more embodiments, however—as suggested by FIGS. **22**A-**22**D—the scene-based image editing system **106** automatically adds the second object upon detecting a selection of the first object. In such embodiments, the scene-based image editing system **106** deselects the second object upon determining that a modification to the first object does not apply to the second object based on the behaviors assigned to the second object.
(380) As further shown in FIG. **22**D, the object **2220** remains in the digital image **2206**. Indeed, the scene-based image editing system **106** did not add the object **2220** to the selection in response to the user interaction with the object **2208***a*, nor did it delete the object **2220** along with the objects **2208***a*-**2208***b*. For instance, upon referencing the semantic scene graph for the digital image **2206**, the scene-based image editing system **106** determines that there is not a relationship between the object **2220** and either of the objects **2208***a*-**2208***b* (at least, there is not a relationship that applies in

this scenario). Thus, the scene-based image editing system **106** enables user interactions to modify related objects together while preventing unrelated objects from being modified without more targeted user interactions.

(381) Additionally, as shown in FIG. **22**D, the scene-based image editing system **106** reveals content fill **2226** within the digital image **2206** upon removing the objects **2208***a*-**2208***b*. In particular, upon deleting the objects **2208***a*-**2208***b*, the scene-based image editing system **106** exposes a content fill previously generated for the object **2208***a* as well as a content fill previously generated for the object **2208***b*. Thus, the scene-based image editing system **106** facilitates seamless modification of the digital image **2206** as if it were a real scene.

(382) FIGS. **23**A-**23**C illustrate another graphical user interface implemented by the scene-based image editing system **106** to facilitate a relationship-aware object modification in accordance with one or more embodiments. Indeed, as shown in FIG. **23**A, the scene-based image editing system **106** provides, for display within a graphical user interface **2302** of a client device **2304**, a digital image **2306** that portrays objects **2308***a*-**2308***b* and object **2320**. In particular, the digital image **2306** portrays a relationship between the objects **2308***a*-**2308***b* in that the object **2308***a* is holding the object **2308***b*.

(383) As further shown in FIG. **23**A, the scene-based image editing system **106** detects a user interaction selecting the object **2308***a*. In response to detecting the user interaction, the scene-based image editing system **106** provides a suggestion that the object **2308***b* be added to the selection. In particular, the scene-based image editing system **106** provides a text box **2310** asking if the user wants the object **2308***b* to be added to the selection and provides an option **2312** for agreeing to add the object **2308***b* and an option **2314** for declining to add the object **2308***b*.

(384) In one or more embodiments, the scene-based image editing system **106** provides the suggestion for adding the object **2308***b* to the selection based on determining the relation ship between the objects **2308***a*-**2308***b* via the semantic scene graph generated for the digital image **2306**. In some cases, the scene-based image editing system **106** further provides the suggestion for adding the object **2308***b* based on the behaviors assigned to the object **2308***b* based on that relationship.

(385) As suggested by FIG. **23**A, the scene-based image editing system **106** does not suggest adding the object **2320** to the selection. Indeed, in one or more embodiments, based on referencing the semantic scene graph, the scene-based image editing system **106** determines that there is no relationship between the object **2320** and either of the objects **2308***a*-**2308***b* (at least, that there is not a relevant relationship). Accordingly, the scene-based image editing system **106** determines to omit the object **2320** from the suggestion.

(386) As shown in FIG. **23**B, the scene-based image editing system **106** adds the object **2308***b* to the selection. In particular, in response to receiving a user interaction with the option **2312** for agreeing to add the object **2308***b*, the scene-based image editing system **106** adds the object **2308***b* to the selection. As shown in FIG. **23**B, the scene-based image editing system **106** modifies a visual indication **2316** of the selection to indicate that the object **2308***b* has been added to the selection along with the object **2308***a*.

(387) As illustrated in FIG. **23**C, the scene-based image editing system **106** modifies the digital image **2306** by moving the object **2308***a* within the digital image **2306** in response to detecting one or more additional user interactions. Further, the scene-based image editing system **106** moves the object **2308***b* along with the object **2308***a* based on the inclusion of the object **2308***b* in the selection. Accordingly, the scene-based image editing system **106** implements a relationship-aware object modification by moving objects based on their relationship to other objects.

(388) FIGS. **24**A-**24**C illustrate yet another graphical user interface implemented by the scene-based image editing system **106** to facilitate a relationship-aware object modification in accordance with one or more embodiments. Indeed, as shown in FIG. **24**A, the scene-based image editing system **106** provides, for display within a graphical user interface **2402** of a client device **2404**, a

digital image **2406** that portrays objects **2408***a*-**2408***b* and an object **2420**. In particular, the digital image **2406** portrays a relationship between the objects **2408***a*-**2408***b* in that the object **2408***a* is holding the object **2408***b*.

(389) As shown in FIG. **24**A, the scene-based image editing system **106** detects a user interaction with the object **2408***a*. In response to detecting the user interaction, the scene-based image editing system **106** provides an option menu **2410** for display within the graphical user interface **2402**. As illustrated, the option menu **2410** includes an option **2412** for deleting the object **2408***a*.

(390) As shown in FIG. **24**B, the scene-based image editing system **106** detects an additional user interaction with the option **2412** for deleting the object **2408***a*. In response to detecting the additional user interaction, the scene-based image editing system **106** provides, for display, a suggestion for adding the object **2408***b* to the selection via a text box **2414** asking if the user wants the object **2408***b* to be added to the selection, an option **2416** for agreeing to add the object **2408***b*, and an option **2418** for declining to add the object **2308***b*.

(391) Indeed, as mentioned above, in one or more embodiments, the scene-based image editing system **106** waits upon receiving input to modify a first object before suggesting adding a second object (or automatically adding the second object). Accordingly, the scene-based image editing system **106** determines whether a relationship between the objects and the pending modification indicate that the second object should be added before including the second object in the selection.

(392) To illustrate, in one or more embodiments, upon detecting the additional user interaction with the option **2412**, the scene-based image editing system **106** references the semantic scene graph for the digital image **2406**. Upon referencing the semantic scene graph, the scene-based image editing system **106** determines that the object **2408***a* has a relationship with the object **2408***b*. Further, the scene-based image editing system **106** determines that the behaviors assigned to the object **2408***b* based on that relationship indicate that the object **2408***b* should be deleted with the object **2408***a*. Accordingly, upon receiving the additional user interaction for deleting the object **2408***a*, the scene-based image editing system **106** determines that the object **2408***b* should also be deleted and then provides the suggestion to add the object **2408***b* (or automatically adds the object **2408***b*) to the selection.

(393) As shown in FIG. **24**C, the scene-based image editing system **106** deletes the object **2408***a* and the object **2408***b* from the digital image **2406** together. In particular, in response to detecting a user interaction with the option **2416** for adding the object **2408***b* to the selection, the scene-based image editing system **106** adds the object **2408***b* and executes the delete operation. In one or more embodiments, upon detecting a user interaction with the option **2418** to decline adding the object **2408***b*, the scene-based image editing system **106** omits the object **2408***b* from the selection and only deletes the object **2408***a*.

(394) Though the above specifically discusses moving objects or deleting objects based on their relationships with other objects, it should be noted that the scene-based image editing system **106** implements various other types of relationship-aware object modifications in various embodiments. For example, in some cases, the scene-based image editing system **106** implements relationship-aware object modifications via resizing modifications, recoloring or retexturing modifications, or compositions. Further, as previously suggested, the behavioral policy graph utilized by the scene-based image editing system **106** is configurable in some embodiments. Thus, in some implementations, the relationship-aware object modifications implemented by the scene-based image editing system **106** change based on user preferences.

(395) In one or more embodiments, in addition to modifying objects based on relationships as described within a behavioral policy graph that is incorporated into a semantic scene graph, the scene-based image editing system **106** modifies objects based on classification relationships. In particular, in some embodiments, the scene-based image editing system **106** modifies objects based on relationships as described by a real-world class description graph that is incorporated into a semantic scene graph. Indeed, as previously discussed, a real-world class description graph

provides a hierarchy of object classifications for objects that may be portrayed in a digital image. Accordingly, in some implementations, the scene-based image editing system **106** modifies objects within digital images based on their relationship with other objects via their respective hierarchy of object classifications. For instance, in one or more embodiments, the scene-based image editing system **106** adds objects to a selection for modification based on their relationships with other objects via their respective hierarchy of object classifications. FIGS. **25**A-**25**D illustrate a graphical user interface implemented by the scene-based image editing system **106** to add objects to a selection for modification based on classification relationships in accordance with one or more embodiments.

(396) In particular, FIG. **25**A illustrates the scene-based image editing system **106** providing, for display in a graphical user interface **2502** of a client device **2504**, a digital image **2506** portraying a plurality of objects **2508***a*-**2508***g*. In particular, as shown the objects **2508***a*-**2508***g* include various items, such as shoes, pairs of glasses, and a coat.

(397) FIG. **25**A further illustrates semantic scene graph components **2510***a*-**2510***c* from a semantic scene graph of the digital image **2506**. Indeed, the semantic scene graph components **2510***a*-**2510***c* include portions of a semantic scene graph providing a hierarchy of object classifications for each of the objects **2508***a*-**2508***g*. Alternatively, in some cases, the semantic scene graph components **2510***a*-**2510***c* represent portions of the real-world class description graph used in making the semantic scene graph.

(398) As shown in FIG. **25**A, the semantic scene graph component **2510***a* includes a node **2512** representing a clothing class, a node **2514** representing an accessory class, and a node **2516** representing a shoe class. As further shown, the accessory class is a subclass of the clothing class, and the shoe class is a subclass of the accessory class. Similarly, the semantic scene graph component **2510***b* includes a node **2518** representing the clothing class, a node **2520** representing the accessory class, and a node **2522** representing a glasses class, which is a subclass of the accessory class. Further, the semantic scene graph component **2510***c* includes a node **2524** representing the clothing class and a node **2526** representing a coat class, which is another subclass of the clothing class. Thus, the semantic scene graph components **2510***a*-**2510***c* provide various classifications that apply to each of the objects **2508***a*-**2508***g*. In particular, the semantic scene graph component **2510***a* provides a hierarchy of object classifications associated with the shoes presented in the digital image **2506**, the semantic scene graph component **2510***b* provides a hierarchy of object classifications associated with the pairs of glasses, and the semantic scene graph component **2510***c* provides a hierarchy of object classifications associated with the coat.

(399) As shown in FIG. **25**B, the scene-based image editing system **106** detects a user interaction selecting the object **2508***e*. Further, the scene-based image editing system **106** detects a user interaction selecting the object **2508***b*. As further shown, in response to detecting the selection of the object **2508***b* and the object **2508***e*, the scene-based image editing system **106** provides a text box **2528** suggesting all shoes in the digital image **2506** be added to the selection.

(400) To illustrate, in some embodiments, in response to detecting the selection of the object **2508***b* and the object **2508***e*, the scene-based image editing system **106** references the semantic scene graph generated for the digital image **2506** (e.g., the semantic scene graph components that are associated with the object **2508***b* and the object **2508***e*). Based on referencing the semantic scene graph, the scene-based image editing system **106** determines that the object **2508***b* and the object **2508***e* are both part of the shoe class. Thus, the scene-based image editing system **106** determines that there is a classification relationship between the object **2508***b* and the object **2508***e* via the shoe class. In one or more embodiments, based on determining that the object **2508***b* and the object **2508***e* are both part of the shoe class, the scene-based image editing system **106** determines that the user interactions providing the selections are targeting all shoes within the digital image **2506**. Thus, the scene-based image editing system **106** provides the text box **2528** suggesting adding the other shoes to the selection. In one or more embodiments, upon receiving a user interaction

accepting the suggestion, the scene-based image editing system **106** adds the other shoes to the selection.
(401) Similarly, as shown in FIG. **25**C, the scene-based image editing system **106** detects a user interaction selecting the object **2508***c* and another user interaction selecting the object **2508***b*. In response to detecting the user interactions, the scene-based image editing system **106** references the semantic scene graph generated for the digital image **2506**. Based on referencing the semantic scene graph, the scene-based image editing system **106** determines that the object **2508***b* is part of the shoe class, which is a subclass of the accessory class. In other words, the scene-based image editing system **106** determines that the object **2508***b* is part of the accessory class. Likewise, the scene-based image editing system **106** determines that the object **2508***c* is part of the glasses class, which is a subclass of the accessory class. Thus, the scene-based image editing system **106** determines that there is a classification relationship between the object **2508***b* and the object **2508***c* via the accessory class. As shown in FIG. **25**C, based on determining that the object **2508***b* and the object **2508***c* are both part of the accessory class, the scene-based image editing system **106** provides a text box **2530** suggesting adding all other accessories portrayed in the digital image **2506** (e.g., the other shoes and pairs of glasses) to the selection.
(402) Further, as shown in FIG. **25**D, the scene-based image editing system **106** detects a user interaction selecting the object **2508***a* and another user interaction selecting the object **2508***b*. In response to detecting the user interactions, the scene-based image editing system **106** references the semantic scene graph generated for the digital image **2506**. Based on referencing the semantic scene graph, the scene-based image editing system **106** determines that the object **2508***b* is part of the shoe class, which is a subclass of the accessory class that is a subclass of the clothing class. Similarly, the scene-based image editing system **106** determines that the object **2508***a* is part of the coat class, which is also a subclass of the clothing class. Thus, the scene-based image editing system **106** determines that there is a classification relationship between the object **2508***b* and the object **2508***a* via the clothing class. As shown in FIG. **25**D, based on determining that the object **2508***b* and the object **2508***a* are both part of the clothing class, the scene-based image editing system **106** provides a text box **2532** suggesting adding all other clothing items portrayed in the digital image **2506** to the selection.
(403) Thus, in one or more embodiments, the scene-based image editing system **106** anticipates the objects that are targeted user interactions and facilitates quicker selection of those objects based on their classification relationships. In some embodiments, upon selection of multiple objects via provided suggestions, the scene-based image editing system **106** modifies the selected objects in response to additional user interactions. Indeed, the scene-based image editing system **106** modifies the selected objects together. Thus, the scene-based image editing system **106** implements a graphical user interface that provides a more flexible and efficient approach to selecting and modifying multiple related objects using reduced user interactions.
(404) Indeed, as previously mentioned, the scene-based image editing system **106** provides improved flexibility and efficiency when compared to conventional systems. For instance, by selecting (e.g., automatically or via suggestion) objects based on the selection of related objects, the scene-based image editing system **106** provides a flexible method of targeting multiple objects for modification. Indeed, the scene-based image editing system **106** flexibly identifies the related objects and includes them with the selection. Accordingly, the scene-based image editing system **106** implements a graphical user interface that reduces user interactions typically required under conventional system for selecting and modifying multiple objects.
(405) In one or more embodiments, the scene-based image editing system **106** further pre-processes a digital image to aid in the removal of distracting objects. In particular, the scene-based image editing system **106** utilizes machine learning to identify objects in a digital image, classify one or more of the objects as distracting objects, and facilitate the removal of the distracting objects to provide a resulting image that is more visually cohesive and aesthetically pleasing. Further, in some

cases, the scene-based image editing system **106** utilizes machine learning to facilitate the removal of shadows associated with distracting objects. FIGS. **26**-**39**C illustrate diagrams of the scene-based image editing system **106** identifying and removing distracting objects and their shadows from digital images in accordance with one or more embodiments.

(406) Many conventional systems are inflexible in the methods they use for removing distracting human in that they strip control away from users. For instance, conventional systems often remove humans they have classified as distracting automatically. Thus, when a digital image is received, such systems fail to provide the opportunity for user interactions to provide input regarding the removal process. For example, these systems fail to allow user interactions to remove human from the set of humans identified for removal.

(407) Additionally, conventional systems typically fail to flexibly remove all types of distracting objects. For instance, many conventional systems fail to flexibly remove shadows cast by distracting objects and non-human objects. Indeed, while some existing systems identify and remove distracting humans from a digital image, these systems often fail to identify shadows cast by humans or other objects within the digital image. Accordingly, the resulting digital image will still include the influence of a distracting human as its shadow remains despite the distracting human itself being removed. This further causes these conventional systems to require additional user interactions to identify and remove these shadows.

(408) The scene-based image editing system **106** addresses these issues by providing more user control in the removal process while reducing the interactions typically required to delete an object from a digital image. Indeed, as will be explained below, the scene-based image editing system **106** presents identified distracting objects for display as a set of objects selected for removal. The scene-based image editing system **106** further enables user interactions to add objects to this set, remove objects from the set, and/or determine when the selected objects are deleted. Thus, the scene-based image editing system **106** employs a flexible workflow for removing distracting objects based on machine learning and user interactions.

(409) Further, the scene-based image editing system **106** flexibly identifies and removes shadows associated with distracting objects within a digital image. By removing shadows associated with distracting objects, the scene-based image editing system **106** provides a better image result in that distracting objects and additional aspects of their influence within a digital image are removed. This allows for reduced user interaction when compared to conventional systems as the scene-based image editing system **106** does not require additional user interactions to identify and remove shadows.

(410) FIG. **26** illustrates a neural network pipeline utilized by the scene-based image editing system **106** to identify and remove distracting objects from a digital image in accordance with one or more embodiments. Indeed, as shown in FIG. **26**, the scene-based image editing system **106** receives a digital image **2602** that portrays a plurality of objects. As illustrated, the scene-based image editing system **106** provides the digital image **2602** to a pipeline of neural networks comprising a segmentation neural network **2604**, a distractor detection neural network **2606**, a shadow detection neural network **2608**, and an inpainting neural network **2610**.

(411) In one or more embodiments, the scene-based image editing system **106** utilizes, as the segmentation neural network **2604**, one of the segmentation neural networks discussed above (e.g., the detection-masking neural network **300** discussed with reference to FIG. **3**). In some embodiments, the scene-based image editing system **106** utilizes, as the inpainting neural network **2610**, one of the content-aware machine learning models discussed above (e.g., the cascaded modulation inpainting neural network **420** discussed with reference to FIG. **4**). The distractor detection neural network **2606** and the shadow detection neural network **2608** will be discussed in more detail below.

(412) As shown in FIG. **26**, the scene-based image editing system **106** utilizes the pipeline of neural networks to generate a modified digital image **2612** from the digital image **2602**. In particular, the

scene-based image editing system **106** utilizes the pipeline of neural networks to identify and remove distracting objects from the digital image **2602**. In particular, the scene-based image editing system **106** generates an object mask for the objects in the digital image utilizing the segmentation neural network **2604**. The scene-based image editing system **106** determines a classification for the objects of the plurality of objects utilizing the distractor detection neural network **2606**. More specifically, the scene-based image editing system **106** assigns each object a classification of main subject object or distracting object. The scene-based image editing system **106** removes distracting objects from the digital image utilizing the object masks. Further, the scene-based image editing system **106** utilizes inpainting neural network **2610** to generate content fill for the portions of the digital image **2602** from which the distracting objects were removed to generate the modified digital image **2612**. As shown, the scene-based image editing system **106** deletes a plurality of different types of distracting objects (multiple men and a pole). Indeed, the scene-based image editing system **106** is robust enough to identify non-human objects as distracting (e.g., the pole behind the girl).

(413) In one or more embodiments, the scene-based image editing system **106** utilizes a subset of the neural networks shown in FIG. **26** to generate a modified digital image. For instance, in some cases, the scene-based image editing system **106** utilizes the segmentation neural network **2604**, the distractor detection neural network **2606**, and the content fill **210** to generate a modified digital image from a digital image. Further, in some cases, the scene-based image editing system **106** utilizes a different ordering of the neural networks than what is shown.

(414) FIG. **27** illustrates an architecture of a distractor detection neural network **2700** utilized by the scene-based image editing system **106** to identify and classify distracting objects in of a digital image in accordance with one or more embodiments. As shown in FIG. **27**, the distractor detection neural network **2700** includes a heatmap network **2702** and a distractor classifier **2704**.

(415) As illustrated, the heatmap network **2702** operates on an input image **2706** to generate heatmaps **2708**. For instance, in some cases, the heatmap network **2702** generates a main subject heatmap representing possible main subject objects and a distractor heatmap representing possible distracting objects. In one or more embodiments, a heatmap (also referred to as a class activation map) includes a prediction made by a convolutional neural network that indicates a probability value, on a scale of zero to one, that a specific pixel of an image belongs to a particular class from a set of classes. As opposed to object detection, the goal of a heatmap network is to classify individual pixels as being part of the same region in some instances. In some cases, a region includes an area of a digital image where all pixels are of the same color or brightness.

(416) In at least one implementation, the scene-based image editing system **106** trains the heatmap network **2702** on whole images, including digital images where there are no distracting objects and digital images that portray main subject objects and distracting objects.

(417) In one or more embodiments, the heatmap network **2702** identifies features in a digital image that contribute to a conclusion that that a given region is more likely to be a distracting object or more likely to be a main subject object, such as body posture and orientation. For instance, in some cases, the heatmap network **2702** determines that objects with slouching postures as opposed to standing at attention postures are likely distracting objects and also that objects facing away from the camera are likely to be distracting objects. In some cases, the heatmap network **2702** considers other features, such as size, intensity, color, etc.

(418) In some embodiments, the heatmap network **2702** classifies regions of the input image **2706** as being a main subject or a distractor and outputs the heatmaps **2708** based on the classifications. For example, in some embodiments, the heatmap network **2702** represents any pixel determined to be part of a main subject object as white within the main subject heatmap and represents any pixel determined to not be part of a main subject object as black (or vice versa). Likewise, in some cases, the heatmap network **2702** represents any pixel determined to be part of a distracting object as white within the distractor heatmap while representing any pixel determined to not be part of a

distracting object as black (or vice versa).
(419) In some implementations, the heatmap network **2702** further generates a background heatmap representing a possible background as part of the heatmaps **2708**. For instance, in some cases, the heatmap network **2702** determines that the background includes areas that are not part of a main subject object or a distracting object. In some cases, the heatmap network **2702** represents any pixel determined to be part of the background as white within the background heatmap while representing any pixel determined to not be part of the background as black (or vice versa).
(420) In one or more embodiments, the distractor detection neural network **2700** utilizes the heatmaps **2708** output by the heatmap network **2702** as a prior to the distractor classifier **2704** to indicate a probability that a specific region of the input image **2706** contains a distracting object or a main subject object.
(421) In one or more embodiments, the distractor detection neural network **2700** utilizes the distractor classifier **2704** to consider the global information included in the heatmaps **2708** and the local information included in one or more individual objects **2710**. To illustrate, in some embodiments, the distractor classifier **2704** generates a score for the classification of an object. If an object in a digital image appears to be a main subject object based on the local information, but the heatmaps **2708** indicate with a high probability that the object is a distracting object, the distractor classifier **2704** concludes that the object is indeed a distracting object in some cases. On the other hand, if the heatmaps **2708** point toward the object being a main subject object, the distractor classifier **2704** determines that the object has been confirmed as a main subject object.
(422) As shown in FIG. **27**, the distractor classifier **2704** includes a crop generator **2712** and a hybrid classifier **2714**. In one or more embodiments, the distractor classifier **2704** receives one or more individual objects **2710** that have been identified from the input image **2706**. In some cases, the one or more individual objects **2710** are identified via user annotation or some object detection network (e.g., the object detection machine learning model **308** discussed above with reference to FIG. **3**).
(423) As illustrated by FIG. **27**, the distractor classifier **2704** utilizes the crop generator **2712** to generate cropped images **2716** by cropping the input image **2706** based on the locations of the one or more individual objects **2710**. For instance, where there are three object detections in the input image **2706**, the crop generator **2712** generates three cropped images—one for each detected object. In one or more embodiments, the crop generator **2712** generates a cropped image by removing all pixels of the input image **2706** outside the location of the corresponding inferred bounding region.
(424) As further shown, the distractor classifier **2704** also utilizes the crop generator **2712** to generate cropped heatmaps **2718** by cropping the heatmaps **2708** with respect to each detected object. For instance, in one or more embodiments, the crop generator **2712** generates—from each of the main subject heatmap, the distractor heatmap, and the background heatmap—one cropped heatmap for each of the detected objects based on a region within the heatmaps corresponding to the location of the detected objects.
(425) In one or more embodiments, for each of the one or more individual objects **2710**, the distractor classifier **2704** utilizes the hybrid classifier **2714** to operate on a corresponding cropped image (e.g., its features) and corresponding cropped heatmaps (e.g., their features) to determine whether the object is a main subject object or a distracting object. To illustrate, in some embodiments, for a detected object, the hybrid classifier **2714** performs an operation on the cropped image associated with the detected object and the cropped heatmaps associated with the detected object (e.g., the cropped heatmaps derived from the heatmaps **2708** based on a location of the detected object) to determine whether the detected object is a main subject object or a distracting object. In one or more embodiments, the distractor classifier **2704** combines the features of the cropped image for a detected object with the features of the corresponding cropped heatmaps (e.g., via concatenation or appending the features) and provides the combination to the hybrid classifier **2714**. As shown in FIG. **27**, the hybrid classifier **2714** generates, from its corresponding cropped

image and cropped heatmaps, a binary decision **2720** including a label for a detected object as a main subject object or a distracting object.

(426) FIG. **28** illustrates an architecture of a heatmap network **2800** utilized by the scene-based image editing system **106** as part of a distractor detection neural network in accordance with one or more embodiments. As shown in FIG. **28**, the heatmap network **2800** includes a convolutional neural network **2802** as its encoder. In one or more embodiments, the convolutional neural network **2802** includes a deep residual network. As further shown in FIG. **28**, the heatmap network **2800** includes a heatmap head **2804** as its decoder.

(427) FIG. **29** illustrates an architecture of a hybrid classifier **2900** utilized by the scene-based image editing system **106** as part of a distractor detection neural network in accordance with one or more embodiments. As shown in FIG. **29**, the hybrid classifier **2900** includes a convolutional neural network **2902**. In one or more embodiments, the hybrid classifier **2900** utilizes the convolutional neural network **2902** as an encoder.

(428) To illustrate, in one or more embodiments, the scene-based image editing system **106** provides the features of a cropped image **2904** to the convolutional neural network **2902**. Further, the scene-based image editing system **106** provides features of the cropped heatmaps **2906** corresponding to the object of the cropped image **2904** to an internal layer **2910** of the hybrid classifier **2900**. In particular, as shown, in some cases, the scene-based image editing system **106** concatenates the features of the cropped heatmaps **2906** with the output of a prior internal layer (via the concatenation operation **2908**) and provides the resulting feature map to the internal layer **2910** of the hybrid classifier **2900**. In some embodiments, the feature map includes 2048+N channels, where N corresponds to the channels of the output of the heatmap network and 2048 corresponds to the channels of the output of the prior internal layer (though **2048** is an example).

(429) As shown in FIG. **29**, the hybrid classifier **2900** performs a convolution on the output of the internal layer **2910** to reduce the channel depth. Further, the hybrid classifier **2900** performs another convolution on the output of the subsequent internal layer **2914** to further reduce the channel depth. In some cases, the hybrid classifier **2900** applies a pooling to the output of the final internal layer **2916** before the binary classification head **2912**. For instance, in some cases, the hybrid classifier **2900** averages the values of the final internal layer output to generate an average value. In some cases, where the average value is above the threshold, the hybrid classifier **2900** classifies the corresponding object as a distracting object and outputs a corresponding binary value; otherwise, the hybrid classifier **2900** classifies the corresponding object as a main subject object and outputs the corresponding binary value (or vice versa). Thus, the hybrid classifier **2900** provides an output **2918** containing a label for the corresponding object.

(430) FIGS. **30**A-**30**C illustrate a graphical user interface implemented by the scene-based image editing system **106** to identify and remove distracting objects from a digital image in accordance with one or more embodiments. For instance, as shown in FIG. **30**A, the scene-based image editing system **106** provides a digital image **3006** for display within a graphical user interface **3002** of a client device **3004**. As further shown, the digital image **3006** portrays an object **3008** and a plurality of additional objects **3010***a*-**3010***d*.

(431) Additionally, as shown in FIG. **30**A, the scene-based image editing system **106** provides a progress indicator **3012** for display within the graphical user interface **3002**. In some cases, the scene-based image editing system **106** provides the progress indicator **3012** to indicate that the digital image **3006** is being analyzed for distracting objects. For instance, in some embodiments, the scene-based image editing system **106** provides the progress indicator **3012** while utilizing a distractor detection neural network to identify and classify distracting objects within the digital image **3006**. In one or more embodiments, the scene-based image editing system **106** automatically implements the distractor detection neural network upon receiving the digital image **3006** and before receiving user input for modifying one or more of the objects **3010***a*-**3010***d*. In some implementations, however, the scene-based image editing system **106** waits upon receiving user

input before analyzing the digital image **3006** for distracting objects.

(432) As shown in FIG. **30**B, the scene-based image editing system **106** provides visual indicators **3014***a*-**3014***d* for display within the graphical user interface **3002** upon completing the analysis. In particular, the scene-based image editing system **106** provides the visual indicators **3014***a*-**3014***d* to indicate that the objects **3010***a*-**3010***d* have been classified as distracting objects.

(433) In one or more embodiments, the scene-based image editing system **106** further provides the visual indicators **3014***a*-**3014***d* to indicate that the objects **3010***a*-**3010***d* have been selected for deletion. In some instances, the scene-based image editing system **106** also surfaces the pre-generated object masks for the objects **3010***a*-**3010***d* in preparation of deleting the objects. Indeed, as has been discussed, the scene-based image editing system **106** pre-generates object masks and content fills for the objects of a digital image (e.g., utilizing the segmentation neural network **2604** and the inpainting neural network **2610** referenced above). Accordingly, the scene-based image editing system **106** has the object masks and content fills readily available for modifying the objects **3010***a*-**3010***d*.

(434) In one or more embodiments, the scene-based image editing system **106** enables user interactions to add to or remove from the selection of the objects for deletion. For instance, in some embodiments, upon detecting a user interaction with the object **3010***a*, the scene-based image editing system **106** determines to omit the object **3010***a* from the deletion operation. Further, the scene-based image editing system **106** removes the visual indication **3014***a* from the display of the graphical user interface **3002**. On the other hand, in some implementations, the scene-based image editing system **106** detects a user interaction with the object **3008** and determines to include the object **3008** in the deletion operation in response. Further, in some cases, the scene-based image editing system **106** provides a visual indication for the object **3008** for display and/or surfaces a pre-generated object mask for the object **3008** in preparation for the deletion.

(435) As further shown in FIG. **30**B, the scene-based image editing system **106** provides a removal option **3016** for display within the graphical user interface **3002**. In one or more embodiments, in response to detecting a user interaction with the removal option **3016**, the scene-based image editing system **106** removes the objects that have been selected for deletion (e.g., the objects **3010***a*-**3010***d* that had been classified as distracting objects). Indeed, as shown in FIG. **30**C, the scene-based image editing system **106** removes the objects **3010***a*-**3010***d* from the digital image **3006**. Further, as shown in **30**C, upon removing the objects **3010***a*-**3010***d*, the scene-based image editing system **106** reveals content fills **3018***a*-**3018***d* that were previously generated.

(436) By enabling user interactions to control which objects are included in the deletion operation and to further choose when the selected objects are removed, the scene-based image editing system **106** provides more flexibility. Indeed, while conventional systems typically delete distracting objects automatically without user input, the scene-based image editing system **106** allows for the deletion of distracting objects in accordance with user preferences expressed via the user interactions. Thus, the scene-based image editing system **106** flexibly allow for control of the removal process via the user interactions.

(437) In addition to removing distracting objects identified via a distractor detection neural network, the scene-based image editing system **106** provides other features for removing unwanted portions of a digital image in various embodiments. For instance, in some cases, the scene-based image editing system **106** provides a tool whereby user interactions can target arbitrary portions of a digital image for deletion. FIGS. **31**A-**31**C illustrate a graphical user interface implemented by the scene-based image editing system **106** to identify and remove distracting objects from a digital image in accordance with one or more embodiments.

(438) In particular, FIG. **31**A illustrates a digital image **3106** displayed on a graphical user interface **3102** of a client device **3104**. The digital image **3106** corresponds to the digital image **3006** of FIG. **30**C after distracting objects identified by a distractor detection neural network have been removed. Accordingly, in some cases, the objects remaining in the digital image **3106** represent those objects

that were not identified and removed as distracting objects. For instance, in some cases, the collection of objects **3110** near the horizon of the digital image **3106** include objects that were not identified as distracting objects by the distractor detection neural network.
(439) As further shown in FIG. **31**A, the scene-based image editing system **106** provides a brush tool option **3108** for display within the graphical user interface **3102**. FIG. **31**B illustrates that, upon detecting a user interaction with the brush tool option **3108**, the scene-based image editing system **106** enables one or more user interactions to use a brush tool to select arbitrary portions of the digital image **3106** (e.g., portions not identified by the distractor detection neural network) for removal. For instance, as illustrated, the scene-based image editing system **106** receives one or more user interactions with the graphical user interface **3102** that target a portion of the digital image **3106** that portrayed the collection of objects **3110**.
(440) As indicated by FIG. **31**B, via the brush tool, the scene-based image editing system **106** enables free-form user input in some cases. In particular, FIG. **31**B shows the scene-based image editing system **106** providing a visual indication **3112** representing the portion of the digital image **3106** selected via the brush tool (e.g., the specific pixels targeted). Indeed, rather than receiving user interactions with previously identified objects or other pre-segmented semantic areas, the scene-based image editing system **106** uses the brush tool to enable arbitrary selection of various portions of the digital image **3106**. Accordingly, the scene-based image editing system **106** utilizes the brush tool to provide additional flexibility whereby user interactions is able to designate undesirable areas of a digital image that may not be identified by machine learning.
(441) As further shown in FIG. **31**B, the scene-based image editing system **106** provides a remove option **3114** for display within the graphical user interface **3102**. As illustrated in FIG. **31**C, in response to detecting a user interaction with the remove option **3114**, the scene-based image editing system **106** removes the selected portion of the digital image **3106**. Further, as shown, the scene-based image editing system **106** fills in the selected portion with a content fill **3116**. In one or more embodiments, where the portion removed from the digital image **3106** does not include objects for which content fill was previously selected (or otherwise includes extra pixels not included in previously generated content fill), the scene-based image editing system **106** generates the content fill **3116** after removing the portion of the digital image **3106** selected via the brush tool. In particular, the scene-based image editing system **106** utilizes a content-aware hole-filling machine learning model to generate the content fill **3116** after the selected portion is removed.
(442) In one or more embodiments, the scene-based image editing system **106** further implements smart dilation when removing objects, such as distracting objects, from digital images. For instance, in some cases, the scene-based image editing system **106** utilizes smart dilation to remove objects that touch, overlap, or are proximate to other objects portrayed in a digital image. FIG. **32**A illustrates the scene-based image editing system **106** utilizes smart dilation to remove an object from a digital image in accordance with one or more embodiments.
(443) Often, conventional systems remove objects from digital images utilizing tight masks (e.g., a mask that tightly adheres to the border of the corresponding object). In many cases, however, a digital image includes color bleeding or artifacts around the border of an object. For instance, there exist some image formats (JPEG) that are particularly susceptible to having format-related artifacts around object borders. Using tight masks when these issues are present causes undesirable effects in the resulting image. For example, inpainting models are typically sensitive to these image blemishes, creating large artifacts when operating directly on the segmentation output. Thus, the resulting modified images inaccurately capture the user intent in removing an object by creating additional image noise.
(444) Thus, the scene-based image editing system **106** dilates (e.g., expands) the object mask of an object to avoid associated artifacts when removing the object. Dilating objects masks, however, presents the risk of removing portions of other objects portrayed in the digital image. For instance, where a first object to be removed overlaps, touches, or is proximate to a second object, a dilated

mask for the first object will often extend into the space occupied by the second object. Thus, when removing the first object using the dilated object mask, significant portions of the second object are often removed and the resulting hole is filled in (generally improperly), causing undesirable effects in the resulting image. Accordingly, the scene-based image editing system **106** utilizes smart dilation to avoid significantly extending the object mask of an object to be removed into areas of the digital image occupied by other objects.

(445) As shown in FIG. **32**A, the scene-based image editing system **106** determines to remove an object **3202** portrayed in a digital image **3204**. For instance, in some cases, the scene-based image editing system **106** determines (e.g., via a distractor detection neural network) that the object **3202** is a distracting object. In some implementations, the scene-based image editing system **106** receives a user selection of the object **3202** for removal. The digital image **3204** also portrays the objects **3206***a*-**3206***b*. As shown, the object **3202** selected for removal overlaps with the object **3206***b* in the digital image **3204**.

(446) As further illustrated in FIG. **32**A, the scene-based image editing system **106** generates an object mask **3208** for the object **3202** to be removed and a combined object mask **3210** for the objects **3206***a*-**3206***b*. For instance, in some embodiments, the scene-based image editing system **106** generates the object mask **3208** and the combined object mask **3210** from the digital image **3204** utilizing a segmentation neural network. In one or more embodiments, the scene-based image editing system **106** generates the combined object mask **3210** by generating an object mask for each of the objects **3206***a*-**3206***b* and determining the union between the separate object masks.

(447) Additionally, as shown in FIG. **32**A, the scene-based image editing system **106** performs an act **3212** of expanding the object mask **3208** for the object **3202** to be removed. In particular, the scene-based image editing system **106** expands the representation of the object **3202** within the object mask **3208**. In other words, the scene-based image editing system **106** adds pixels to the border of the representation of the object within the object mask **3208**. The amount of expansion varies in various embodiments and, in some implementations, is configurable to accommodate user preferences. For example, in one or more implementations, the scene-based image editing system **106** expands the object mask by extending the object mask outward ten, fifteen, twenty, twenty-five, or thirty pixels.

(448) After expanding the object mask **3208**, the scene-based image editing system **106** performs an act **3214** of detecting overlap between the expanded object mask for the object **3202** and the object masks of the other detected objects **3206***a*-**3206***b* (i.e., the combined object mask **3210**). In particular, the scene-based image editing system **106** determines where pixels corresponding to the expanded representation of the object **3202** within the expanded object mask overlap pixels corresponding to the objects **3206***a*-**3206***b* within the combined object mask **3210**. In some cases, the scene-based image editing system **106** determines the union between the expanded object mask and the combined object mask **3210** and determines the overlap using the resulting union. The scene-based image editing system **106** further performs an act **3216** of removing the overlapping portion from the expanded object mask for the object **3202**. In other words, the scene-based image editing system **106** removes pixels from the representation of the object **3202** within the expanded object mask that overlaps with the pixels corresponding to the object **3206***a* and/or the object **3206***b* within the combined object mask **3210**.

(449) Thus, as shown in FIG. **32**A, the scene-based image editing system **106** generates a smartly dilated object mask **3218** (e.g., an expanded object mask) for the object **3202** to be removed. In particular, the scene-based image editing system **106** generates the smartly dilated object mask **3218** by expanding the object mask **3208** in areas that don't overlap with either one of the objects **3206***a*-**3206***b* and avoiding expansion in areas that do overlap with at least one of the objects **3206***a*-**3206***b*. At least, in some implementations, the scene-based image editing system **106** reduces the expansion in areas that do overlap. For instance, in some cases, the smartly dilated object mask **3218** still includes expansion in overlapping areas but the expansion is significantly less when

compared to areas where there is no overlap. In other words, the scene-based image editing system **106** expands using less pixels in areas where there is overlap. For example, in one or more implementations, the scene-based image editing system **106** expands or dilates an object mask five, ten, fifteen, or twenty times as far into areas where there is no overlap compared to areas where there are overlaps.

(450) To describe it differently, in one or more embodiments, the scene-based image editing system **106** generates the smartly dilated object mask **3218** (e.g., an expanded object mask) by expanding the object mask **3208** for the object **3202** into areas not occupied by the object masks for the objects **3206***a*-**3206***b* (e.g., areas not occupied by the objects **3206***a*-**3206***b* themselves). For instance, in some cases, the scene-based image editing system **106** expands the object mask **3208** into portions of the digital image **3204** that abut the object mask **3208**. In some cases, the scene-based image editing system **106** expands the object mask **3208** into the abutting portions by a set number of pixels. In some implementations, the scene-based image editing system **106** utilizes a different number of pixels for expanding the object mask **3208** into different abutting portions (e.g., based on detecting a region of overlap between the object mask **3208** and other object masks).

(451) To illustrate, in one or more embodiments, the scene-based image editing system **106** expands the object mask **3208** into the foreground and the background of the digital image **3204**. In particular, the scene-based image editing system **106** determines foreground by combining the object masks of objects not to be deleted. The scene-based image editing system **106** expands the object mask **3208** into the abutting foreground and background. In some implementations, the scene-based image editing system **106** expands the object mask **3208** into the foreground by a first amount and expands the object mask **3208** into the background by a second amount that differs from the first amount (e.g., the second amount is greater than the first amount). For example, in one or more implementations the scene-based image editing system **106** expands the object mask by twenty pixels into background areas and two pixels into foreground areas (into abutting object masks, such as the combined object mask **3210**).

(452) In one or more embodiments, the scene-based image editing system **106** determines the first amount to use for the expanding the object mask **3208** into the foreground by expanding the object mask **3208** into the foreground by the second amount—the same amount used to expand the object mask **3208** into the background. In other words, the scene-based image editing system **106** expands the object mask **3208** as a whole into the foreground and background by the same amount (e.g., using the same number of pixels). The scene-based image editing system **106** further determines a region of overlap between the expanded object mask and the object masks corresponding to the other objects **3206***a*-**3206***b* (e.g., the combined object mask **3210**). In one or more embodiments, the region of overlap exists in the foreground of the digital image **3204** abutting the object mask **3208**. Accordingly, the scene-based image editing system **106** reduces the expansion of the object mask **3208** into the foreground so that the expansion corresponds to the second amount. Indeed, in some instances, the scene-based image editing system **106** removes the region of overlap from the expanded object mask for the object **3202** (e.g., removes the overlapping pixels). In some cases, scene-based image editing system **106** removes a portion of the region of overlap rather than the entire region of overlap, causing a reduced overlap between the expanded object mask for the object **3202** and the object masks corresponding to the objects **3206***a*-**3206***b*.

(453) In one or more embodiments, as removing the object **3202** includes removing foreground and background abutting the smartly dilated object mask **3218** (e.g., the expanded object mask) generated for the object **3202**, the scene-based image editing system **106** inpaints a hole remaining after the removal. In particular, the scene-based image editing system **106** inpaints a hole with foreground pixels and background pixels. Indeed, in one or more embodiments, the scene-based image editing system **106** utilizes an inpainting neural network to generate foreground pixels and background pixels for the resulting hole and utilizes the generated pixels to inpaint the hole, resulting in a modified digital image (e.g., an inpainted digital image) where the object **3202** has

been removed and the corresponding portion of the digital image **3204** has been filled in.
(454) For example, FIG. **32**B illustrates the advantages provided by intelligently dilating object masks prior to performing inpainting. In particular, FIG. **32**B illustrates that when the smartly dilated object mask **3218** (e.g., the expanded object mask) is provided to an inpainting neural network (e.g., the cascaded modulation inpainting neural network **420**) as an area to fill, the inpainting neural network generates a modified digital image **3220** with the area corresponding to the smartly dilated object mask **3218** filled with pixel generated by the inpainting neural network. As shown, the modified digital image **3220** includes no artifacts in the inpainted area corresponding to the smartly dilated object mask **3218**. Indeed, the modified digital image **3220** provides a realistic appearing image.
(455) In contrast, FIG. **32**B illustrates that when the object mask **3208** (e.g., the non-expanded object mask) is provided to an inpainting neural network (e.g., the cascaded modulation inpainting neural network **420**) as an area to fill, the inpainting neural network generates a modified digital image **3222** with the area corresponding to the smartly dilated object mask **3218** filled with pixel generated by the inpainting neural network. As shown, the modified digital image **3222** includes artifacts in the inpainted area corresponding to the object mask **3208**. In particular, artifacts are along the back of the girl and event in the generated water.
(456) By generating smartly dilated object masks, the scene-based image editing system **106** provides improved image results when removing objects. Indeed, the scene-based image editing system **106** leverages expansion to remove artifacts, color bleeding, or other undesirable errors in a digital image but avoids removing significant portions of other objects that are remain in the digital image. Thus, the scene-based image editing system **106** is able to fill in holes left by removed objects without enhancing present errors where possible without needlessly replacing portions of other objects that remain.
(457) As previously mentioned, in one or more embodiments, the scene-based image editing system **106** further utilizes a shadow detection neural network to detect shadows associated with distracting objects portrayed within a digital image. FIGS. **33-38** illustrate diagrams of a shadow detection neural network utilized by the scene-based image editing system **106** to detect shadows associated with objects in accordance with one or more embodiments.
(458) In particular, FIG. **33** illustrates an overview of a shadow detection neural network **3300** in accordance with one or more embodiments. Indeed, as shown in FIG. **33**, the shadow detection neural network **3300** analyzes an input image **3302** via a first stage **3304** and a second stage **3310**. In particular, the first stage **3304** includes an instance segmentation component **3306** and an object awareness component **3308**. Further, the second stage **3310** includes a shadow prediction component **3312**. In one or more embodiments, the instance segmentation component **3306** includes the segmentation neural network **2604** of the neural network pipeline discussed above with reference to FIG. **26**.
(459) As shown in FIG. **33**, after analyzing the input image **3302**, the shadow detection neural network **3300** identifies objects **3314***a*-**3314***c* and shadows **3316***a*-**3316***c* portrayed therein. Further, the shadow detection neural network **3300** associates the objects **3314***a*-**3314***c* with their respective shadows. For instance, the shadow detection neural network **3300** associates the object **3314***a* with the shadow **3316***a* and likewise for the other objects and shadows. Thus, the shadow detection neural network **3300** facilitates inclusion of a shadow when its associated object is selected for deletion, movement, or some other modification.
(460) FIG. **34** illustrates an overview of an instance segmentation component **3400** of a shadow detection neural network in accordance with one or more embodiments. As shown in FIG. **34**, the instance segmentation component **3400** implements an instance segmentation model **3402**. In one or more embodiments, the instance segmentation model **3402** includes the detection-masking neural network **300** discussed above with reference to FIG. **3**. As shown in FIG. **34**, the instance segmentation component **3400** utilizes the instance segmentation model **3402** to analyze an input

image **3404** and identify objects **3406***a*-**3406***c* portrayed therein based on the analysis. For instance, in some cases, the scene-based image editing system **106** outputs object masks and/or bounding boxes for the objects **3406***a*-**3406***c*.

(461) FIG. **35** illustrates an overview of an object awareness component **3500** of a shadow detection neural network in accordance with one or more embodiments. In particular, FIG. **35** illustrates input image instances **3502***a*-**3502***c* corresponding to each object detected within the digital image via the prior instance segmentation component. In particular, each input image instance corresponds to a different detected object and corresponds to an object mask and/or a bounding box generated for that digital image. For instance, the input image instance **3502***a* corresponds to the object **3504***a*, the input image instance **3502***b* corresponds to the object **3504***b*, and the input image instance **3502***c* corresponds to the object **3504***c*. Thus, the input image instances **3502***a*-**3502***c* illustrate the separate object detections provided by the instance segmentation component of the shadow detection neural network.

(462) In some embodiments, for each detected object, the scene-based image editing system **106** generates input for the second stage of the shadow detection neural network (i.e., the shadow prediction component). FIG. **35** illustrates the object awareness component **3500** generating input **3506** for the object **3504***a*. Indeed, as shown in FIG. **35**, the object awareness component **3500** generates the input **3506** using the input image **3508**, the object mask **3510** corresponding to the object **3504***a* (referred to as the object-aware channel) and a combined object mask **3512** corresponding to the objects **3504***b*-**3504***c* (referred to as the object-discriminative channel). For instance, in some implementations, the object awareness component **3500** combines (e.g., concatenates) the input image **3508**, the object mask **3510**, and the combined object mask **3512**. The object awareness component **3500** similarly generates second stage input for the other objects **3504***b*-**3504***c* as well (e.g., utilizing their respective object mask and combined object mask representing the other objects along with the input image **3508**).

(463) In one or more embodiments, the scene-based image editing system **106** (e.g., via the object awareness component **3500** or some other component of the shadow detection neural network) generates the combined object mask **3512** using the union of separate object masks generated for the object **3504***b* and the object **3504***c*. In some instances, the object awareness component **3500** does not utilize the object-discriminative channel (e.g., the combined object mask **3512**). Rather, the object awareness component **3500** generates the input **3506** using the input image **3508** and the object mask **3510**. In some embodiments, however, using the object-discriminative channel provides better shadow prediction in the second stage of the shadow detection neural network.

(464) FIG. **36** illustrates an overview of a shadow prediction component **3600** of a shadow detection neural network in accordance with one or more embodiments. As shown in FIG. **36**, the shadow detection neural network provides, to the shadow prediction component **3600**, input compiled by an object awareness component consisting of an input image **3602**, an object mask **3604** for an object of interest, and a combined object mask **3606** for the other detected objects. The shadow prediction component **3600** utilizes a shadow segmentation model **3608** to generate a first shadow prediction **3610** for the object of interest and a second shadow prediction **3612** for the other detected objects. In one or more embodiments, the first shadow prediction **3610** and/or the second shadow prediction **3612** include shadow masks (e.g., where a shadow mask includes an object mask for a shadow) for the corresponding shadows. In other words, the shadow prediction component **3600** utilizes the shadow segmentation model **3608** to generate the first shadow prediction **3610** by generating a shadow mask for the shadow predicted for the object of interest. Likewise, the shadow prediction component **3600** utilizes the shadow segmentation model **3608** to generate the second shadow prediction **3612** by generating a combined shadow mask for the shadows predicted for the other detected objects.

(465) Based on the outputs of the shadow segmentation model **3608**, the shadow prediction component **3600** provides an object-shadow pair prediction **3614** for the object of interest. In other

words, the shadow prediction component **3600** associates the object of interest with its shadow cast within the input image **3602**. In one or more embodiments, the shadow prediction component **3600** similarly generates an object-shadow pair prediction for all other objects portrayed in the input image **3602**. Thus, the shadow prediction component **3600** identifies shadows portrayed in a digital image and associates each shadow with its corresponding object.

(466) In one or more embodiments, the shadow segmentation model **3608** utilized by the shadow prediction component **3600** includes a segmentation neural network. For instance, in some cases, the shadow segmentation model **3608** includes the detection-masking neural network **300** discussed above with reference to FIG. **3**. As another example, in some implementations, the shadow segmentation model **3608** includes the DeepLabv3 semantic segmentation model described by Liang-Chieh Chen et al., *Rethinking Atrous Convolution for Semantic Image Segmentation*, arXiv:1706.05587, 2017, or the DeepLab semantic segmentation model described by Liang-Chieh Chen et al., *Deeplab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs*, arXiv:1606.00915, 2016, both of which are incorporated herein by reference in their entirety.

(467) FIG. **37** illustrates an overview of the architecture of a shadow detection neural network **3700** in accordance with one or more embodiments. In particular, FIG. **37** illustrates the shadow detection neural network **3700** consisting of the instance segmentation component **3400** discussed with reference to FIG. **34**, the object awareness component **3500** discussed with reference to FIG. **35**, and the shadow prediction component **3600** discussed with reference to FIG. **36**. Further, FIG. **37** illustrates the shadow detection neural network **3700** generating object masks, shadow masks, and predictions with respect to each object portrayed in the input image **3702**. Thus, the shadow detection neural network **3700** outputs a final prediction **3704** that associates each object portrayed in a digital image with its shadow. Accordingly, as shown in FIG. **37**, the shadow detection neural network **3700** provides an end-to-end neural network framework that receives a digital image and outputs an association between objects and shadows depicted therein.

(468) In some implementations, the shadow detection neural network **3700** determines that an object portrayed within a digital image does not have an associated shadow. Indeed, in some cases, upon analyzing the digital image utilizing its various components, the shadow detection neural network **3700** determines that there is not a shadow portrayed within the digital image that is associated with the object. In some cases, the scene-based image editing system **106** provides feedback indicating the lack of a shadow. For example, in some cases, upon determining that there are no shadows portrayed within a digital image (or that there is not a shadow associated with a particular object), the scene-based image editing system **106** provides a message for display or other feedback indicating the lack of shadows. In some instances, the scene-based image editing system **106** does not provide explicit feedback but does not auto-select or provide a suggestion to include a shadow within a selection of an object as discussed below with reference to FIGS. **39**A-**39**C.

(469) In some implementations, the scene-based image editing system **106** utilizes the second stage of the shadow detection neural network to determine shadows associated with objects portrayed in a digital image when the objects masks of the objects have already been generated. Indeed, FIG. **38** illustrates a diagram for using the second stage of the shadow detection neural network for determining shadows associated with objects portrayed in a digital image in accordance with one or more embodiments.

(470) As shown in FIG. **38**, the scene-based image editing system **106** provides an input image **3804** to the second stage of a shadow detection neural network (i.e., a shadow prediction model **3802**). Further, the scene-based image editing system **106** provides an object mask **3806** to the second stage. The scene-based image editing system **106** utilizes the second stage of the shadow detection neural network to generate a shadow mask **3808** for the shadow of the object portrayed in the input image **3804**, resulting in the association between the object and the shadow cast by the

object within the input image **3804** (e.g., as illustrated in the visualization **3810**).

(471) By providing direct access to the second stage of the shadow detection neural network, the scene-based image editing system **106** provides flexibility in the shadow detection process. Indeed, in some cases, an object mask will already have been created for an object portrayed in a digital image. For instance, in some cases, the scene-based image editing system **106** implements a separate segmentation neural network to generate an object mask for a digital image as part of a separate workflow. Accordingly, the object mask for the object already exists, and the scene-based image editing system **106** leverages the previous work in determining the shadow for the object. Thus, the scene-based image editing system **106** further provides efficiency as it avoids duplicating work by accessing the shadow prediction model of the shadow detection neural network directly.

(472) FIGS. **39**A-**39**C illustrate a graphical user interface implemented by the scene-based image editing system **106** to identify and remove shadows of objects portrayed in a digital image in accordance with one or more embodiments. Indeed, as shown in FIG. **39**A, the scene-based image editing system **106** provides, for display within a graphical user interface **3902** of a client device, a digital image **3906** portraying an object **3908**. As further shown, the object **3908** casts a shadow **3910** within the digital image **3906**.

(473) In one or more embodiments, upon receiving the digital image **3906**, the scene-based image editing system **106** utilizes a shadow detection neural network to analyze the digital image **3906**. In particular, the scene-based image editing system **106** utilizes the shadow detection neural network to identify the object **3908**, identify the shadow **3910** cast by the object **3908**, and further associate the shadow **3910** with the object **3908**. As previously mentioned, in some implementations, the scene-based image editing system **106** further utilizes the shadow detection neural network to generate object masks for the object **3908** and the shadow **3910**.

(474) As previously discussed with reference to FIG. **26**, in one or more embodiments, the scene-based image editing system **106** identifies shadows cast by objects within a digital image as part of a neural network pipeline for identifying distracting objects within the digital image. For instance, in some cases, the scene-based image editing system **106** utilizes a segmentation neural network to identify objects for a digital image, a distractor detection neural network to classify one or more of the objects as distracting objects, a shadow detection neural network to identify shadows and associate the shadows with their corresponding objects, and an inpainting neural network to generate content fills to replace objects (and their shadows) that are removed. In some cases, the scene-based image editing system **106** implements the neural network pipeline automatically in response to receiving a digital image.

(475) Indeed, as shown in FIG. **39**B, the scene-based image editing system **106** provides, for display within the graphical user interface **3902**, a visual indication **3912** indicating a selection of the object **3908** for removal. As further shown, the scene-based image editing system **106** provides, for display, a visual indication **3914** indicating a selection of the shadow **3910** for removal. As suggested, in some cases, the scene-based image editing system **106** selects the object **3908** and the shadow **3910** for deletion automatically (e.g., upon determining the object **3908** is a distracting object). In some implementations, however, the scene-based image editing system **106** selects the object **3908** and/or the shadow **3910** in response to receiving one or more user interactions.

(476) For instance, in some cases, the scene-based image editing system **106** receives a user selection of the object **3908** and automatically adds the shadow **3910** to the selection. In some implementations, the scene-based image editing system **106** receives a user selection of the object **3908** and provides a suggestion for display in the graphical user interface **3902**, suggesting that the shadow **3910** be added to the selection. In response to receiving an additional user interaction, the scene-based image editing system **106** adds the shadow **3910**.

(477) As further shown in FIG. **39**B, the scene-based image editing system **106** provides a remove option **3916** for display within the graphical user interface **3902**. As indicated by FIG. **39**C, upon receiving a selection of the remove option **3916**, the scene-based image editing system **106** removes

the object **3908** and the shadow **3910** from the digital image. As further shown, the scene-based image editing system **106** replaces the object **3908** with a content fill **3918** and replaces the shadow **3910** with a content fill **3920**. In other words, the scene-based image editing system **106** reveals the content fill **3918** and the content fill **3920** upon removing the object **3908** and the shadow **3910**, respectively.

(478) Though FIGS. **39**A-**39**C illustrates implementing shadow detection with respect to a delete operation, it should be noted that the scene-based image editing system **106** implements shadow detection for other operations (e.g., a move operation) in various embodiments. Further, thought FIGS. **39**A-**39**C are discussed with respect to removing distracting objects from a digital image, the scene-based image editing system **106** implements shadow detection in the context of other features described herein. For instance, in some cases, the scene-based image editing system **106** implements shadow detection with respect to object-aware modifications where user interactions target objects directly. Thus, the scene-based image editing system **106** provides further advantages to object-aware modifications by segmenting objects and their shadows and generating corresponding content fills before receiving user interactions to modify the objects to allow for seamless interaction with digital images as if they were real scenes.

(479) By identifying shadows cast by objects within digital images, the scene-based image editing system **106** provides improved flexibility when compared to conventional systems. Indeed, the scene-based image editing system **106** flexibly identifies objects within a digital image along with other aspects of those objects portrayed in the digital image (e.g., their shadows). Thus, the scene-based image editing system **106** provides a better image result when removing or moving objects as it accommodates these other aspects. This further leads to reduced user interaction with a graphical user interface as the scene-based image editing system **106** does not require user interactions for targeting the shadows of objects for movement or removal (e.g., user interactions to identify shadow pixels and/or tie the shadow pixels to the object).

(480) In some implementations, the scene-based image editing system **106** implements one or more additional features to facilitate the modification of a digital image. In some embodiments, these features provide additional user-interface-based efficiency in that they reduce the amount of user interactions with a user interface typically required to perform some action in the context of image editing. In some instances, these features further aid in the deployment of the scene-based image editing system **106** on computing devices with limited screen space as they efficiently use the space available to aid in image modification without crowding the display with unnecessary visual elements.

(481) FIGS. **40**A-**40**C illustrate a graphical user interface implemented by the scene-based image editing system **106** to facilitate zoom-based image editing in accordance with one or more embodiments. In particular, in some implementations, the scene-based image editing system **106** facilitates focused interaction a digital image. For instance, in some cases, the scene-based image editing system **106** determines that a workflow of one or more modifications are to be implemented within a particular region portrayed in a digital image or that a user otherwise desires to view a region of a digital image in more detail. Accordingly, the scene-based image editing system **106** implements a graphical user interface that zooms into a user-designated region of a digital image for a more detailed display of that region. The scene-based image editing system **106** then facilitates user interactions with the zoomed-in region.

(482) Indeed, as shown in FIG. **40**A, the scene-based image editing system **106** provides a digital image **4006** for display within a graphical user interface **4002** of a client device **4004**. As indicated by FIG. **40**A, the scene-based image editing system **106** provides the digital image **4006** for display in a zoomed-out view. For instance, in some cases, the scene-based image editing system **106** provides the digital image **4006** for display in a default magnification (e.g., at one hundred percent magnification) so that the entirety of the digital image **4006** is displayed. In one or more embodiments, the scene-based image editing system **106** provides the digital image **4006** for

display at the zoomed-out view upon opening the image file or otherwise accessing the digital image **4006**.
(483) As shown in FIG. **40**B, the scene-based image editing system **106** receives one or more user interactions via the graphical user interface **4002**, designating a focus area **4010** for interacting with the digital image **4006**. For instance, as illustrated, the scene-based image editing system **106** receives a user interaction indicating a boundary **4008** for the focus area **4010**. In some cases, the scene-based image editing system **106** draws the boundary **4008** within the graphical user interface **4002** to provide a visualization and facilitate completion of the boundary **4008** (e.g., by informing the user where the boundary **4008** is located). The scene-based image editing system **106** utilizes other interactions to determine the focus area **4010** in other embodiments. For instance, in some cases, the scene-based image editing system **106** receives a user interaction (e.g., a tap or double tap) at a location within the digital image **4006**. In response, the scene-based image editing system **106** determines the boundary **4008** for the focus area **4010** is located at a particular radius of pixels from the location of the user interaction.
(484) As shown in FIG. **40**C, the scene-based image editing system **106** provides a zoomed-in view of the focus area **4010** in response to the user interaction designating the focus area **4010**. In one or more embodiments, the scene-based image editing system **106** provides an animation that involves zooming into the focus area **4010** or otherwise changing the display from the zoomed-out (e.g., default) view to the zoomed-in view. Thus, the scene-based image editing system **106** provides a more focused display of the area designated by the user interaction.
(485) As further shown in FIG. **40**C, the scene-based image editing system **106** also displays regions (e.g., the portion region **4012**) of the digital image **4006** that are outside of the focus area **4010** that was designated by the user interaction. For instance, in some cases, the shape of the display area of the client device **4004** and the shape of the focus area **4010** are different; thus, the scene-based image editing system **106** displays, in the zoomed-in view, at least a portion of the digital image **4006** that exists outside the focus area **4010**. As illustrated, however, the scene-based image editing system **106** blurs (or otherwise distorts or differentiates) the displayed regions of the digital image **4006** outside the focus area **4010**. For instance, in some cases, the scene-based image editing system **106** applies a filter to the regions of the digital image **4006** outside the focus area **4010** to blur or otherwise distort those pixels. Accordingly, while regions outside of the focus area **4010** are still displayed, the scene-based image editing system **106** is still able to provide a targeted view of the focus area **4010** in which the user can interaction.
(486) Indeed, in some implementations, the scene-based image editing system **106** facilitates user interactions within the focus area **4010**. For example, the scene-based image editing system **106** facilitates user interactions with objects and/or other semantic areas displayed within the focus area **4010**. In particular, in some cases, the scene-based image editing system **106** enables modification of objects and/or other semantic areas displayed within the focus area **4010** while preventing modification of objects and/or other semantic areas outside the focus area **4010** (e.g., within the blurred regions displayed).
(487) Thus, the scene-based image editing system **106** implements a graphical user interface that requires relatively few user interactions to focus on a particular region of a digital image interact with the particular region within a zoomed-in display.
(488) FIGS. **41**A-**41**C illustrate a graphical user interface implemented by the scene-based image editing system **106** to facilitate the automatic display of semantic information in accordance with one or more embodiments. Indeed, as previously mentioned, the scene-based image editing system **106** pre-processes a digital image to determine various types of semantic information for a digital image, such as objects portrayed therein, shadows of those objects, and attributes of those objects. Accordingly, in many instances, the scene-based image editing system **106** has useful data regarding a digital image before receiving user interactions for modifying the digital image. In some implementations, the scene-based image editing system **106** utilizes this collected data to

provide aid in determining how a digital image is to be modified. For example, in some cases, the scene-based image editing system **106** utilizes the previously collected semantic information to provide modification suggestions to a user.

(489) For instance, FIG. **41**A illustrates the scene-based image editing system **106** providing a digital image **4106** for display within a graphical user interface **4102** of a client device **4104**. In particular, FIG. **41**A illustrates the digital image **4106** displayed unembellished (e.g., without any additional visual elements). For example, in some cases, FIG. **41**A represents an initial display of the digital image **4106** (e.g., the display provided upon first opening the image file or otherwise accessing the digital image **4106**).

(490) FIG. **41**B illustrates the scene-based image editing system **106** highlighting an object **4108** portrayed in the digital image as part of a modification suggestion. In particular, in some cases, the scene-based image editing system **106** highlights the object **4108** to suggest modifying the object **4108**. In one or more embodiments, the scene-based image editing system **106** highlights the object **4108** as part of the modification suggestion upon determining that there has been a lack of user interactivity with respect to the digital image **4106** for a threshold amount of time.

(491) To illustrate, in one or more embodiments, the scene-based image editing system **106** determines that it has not detected a user interaction with the digital image **4106** for a predetermined threshold of time. In some cases, the scene-based image editing system **106** determines that this lack of user interactivity is due to confusion or hesitation on the part of the user. For instance, the scene-based image editing system **106** determines that the user does not know where to start in modifying the digital image **4106** or, if the user has already initiated one or more actions, does not know what to do next. In response, the scene-based image editing system **106** provides a suggestion for a potential modification that could be performed on the digital image **4106** and provides a visual indication of this suggestion (e.g., by highlighting the object **4108**).

(492) In one or more embodiments, the scene-based image editing system **106** determines which suggestion to provide based on a history of actions performed by the user. For instance, in some cases, the scene-based image editing system **106** determines that the user typically modifies objects (or particular objects, such as people) within a digital image. In some implementations, the scene-based image editing system **106** determines the suggestion to provide based on the popularity of various modifications among a plurality of other users. In some cases, the scene-based image editing system **106** establishes a hierarchy of semantic areas and provides a suggestion in accordance with this hierarchy.

(493) Thus, in some cases, the scene-based image editing system **106** highlights the object **4108** as shown in FIG. **41**B to aid in the modification of the digital image **4106** when it appears that the modification process has slowed down or failed to begin. Further, the scene-based image editing system **106** leverages the semantic information previously determined for the digital image **4106** (e.g., via semantic segmentation). In one or more embodiments, the scene-based image editing system **106** surfaces the object mask for the object **4108** along with providing the visual indication of the suggestion. Additionally, as shown, the scene-based image editing system **106** also provides a text box **4110** along with the highlighting to explicitly suggest modification of the object **4108**. In some embodiments, the scene-based image editing system **106** provides the highlighting or the text box alone.

(494) In some implementations, the scene-based image editing system **106** pairs the suggestion to a particular action that could be taken. Further, in some instances, the scene-based image editing system **106** performs an action (e.g., without receiving a user interaction to perform such an action) and requests that the action be accepted or rejected. For example, FIG. **41**C illustrates the scene-based image editing system **106** removing the background from the digital image **4106**. The scene-based image editing system **106** further provides a text box **4112** requesting a user interaction to confirm or reject the modification. Thus, in some cases, the scene-based image editing system **106** provides a preview of a potential modification for display as part of a suggestion for modifying the

digital image **4106**.

(495) FIGS. **42**A-**42**B illustrate a graphical user interface implemented by the scene-based image editing system **106** to provide a timed confirmation for user edits in accordance with one or more embodiments. Indeed, in one or more embodiments, the scene-based image editing system **106** operates to provide confidence in modifications made to a digital image. Accordingly, in some implementations, the scene-based image editing system **106** provides indications of modifications that have been made and/or interactive elements for undoing those modifications. In some cases, the scene-based image editing system **106** configures the interactive elements for undoing modifications to be timed so that a lack of interaction automatically confirms the modification. Thus, the scene-based image editing system **106** implements communication during the modification process to ensure certain modifications are intentional and/or desired.

(496) For instance, FIG. **42**A illustrates the scene-based image editing system **106** providing a digital image **4206** for display within a graphical user interface **4202** of a client device **4204**. As shown, the scene-based image editing system **106** is modifying (e.g., moving) an object **4208** portrayed in the digital image **4206** in response to a user interaction. As further shown, the scene-based image editing system **106** provides a visual element **4210** for display that indicates the modification being performed. Thus, in some implementations, the scene-based image editing system **106** recognizes the particular modification being performed and provides communication during the modification process to indicate the effect of the user interaction.

(497) As shown in FIG. **42**B, after the modification of the object **4208** is complete, the scene-based image editing system **106** provides a selectable option **4212** for display for undoing the modification. For instance, in some implementations, upon selection of the selectable option **4212**, the scene-based image editing system **106** moves the object **4208** back to its position before the user interaction for moving the object **4208** (or otherwise returns the digital image **4206** to its state before the modification). For example, in some cases, the scene-based image editing system **106** temporarily stores the position of the object **4208** before the modification and returns the object **4208** to the stored position upon selection of the selectable option **4212**.

(498) As mentioned, in some implementations, the scene-based image editing system **106** configures the selectable option **4212** to be timed. In other words, the scene-based image editing system **106** establishes a threshold time period associated with the selectable option **4212**. Upon determining that there have been no user interactions with the selectable option **4212** within the threshold time period, the scene-based image editing system **106** determines that the modification is confirmed. In some instances, the scene-based image editing system **106** removes the selectable option **4212** from display to indicate that the time for undoing the modification has expired. Indeed, while the scene-based image editing system **106** will respond to subsequent user interactions for modifying the digital image **4206**, including those attempting to return the digital image **4206** to its former state, the scene-based image editing system **106** removes the selectable option for automatically doing so.

(499) FIGS. **43**A-**43**B illustrate another graphical user interface implemented by the scene-based image editing system **106** to provide a timed confirmation for user edits in accordance with one or more embodiments. As shown in FIG. **43**A, the scene-based image editing system **106** provides a digital image **4306** for display within a graphical user interface **4302** of a client device **4304**. Further, the scene-based image editing system **106** provides a selectable option **4308** for modifying the digital image **4306**. FIG. **43**A illustrates a selectable option for modifying the brightness of the digital image **4306**, though the scene-based image editing system **106** provides a one or more selectable options for various potential modifications in various embodiments.

(500) As further illustrated, the scene-based image editing system **106** detects a user interaction with the selectable option **4308**. As shown in FIG. **43**B, in response to detecting the user interaction, the scene-based image editing system **106** modifies the digital image **4306** (e.g., by adjusting the brightness). As indicated by FIG. **43**B, the scene-based image editing system **106**

automatically modifies the digital image **4306** without direction from further user interaction. In other words, in some implementations, the user interaction with the selectable option **4308** indicates that the brightness of the digital image **4306** is to be adjusted but does not indicate how to adjust it. Accordingly, the scene-based image editing system **106** automatically determines the brightness to be implemented and adjusts the digital image **4306** to achieve that brightness.

(501) In one or more embodiments, the scene-based image editing system **106** automatically modifies the digital image **4306** (e.g., adjusts the brightness) using user preferences or user history. For instance, in some cases, the scene-based image editing system **106** tracks the settings typically used by the client device **4304** for a particular modification and implements those settings in response to selection of the selectable option **4308**. In some cases, the scene-based image editing system **106** utilizes a model, such as a machine learning model, to perform the modification. To illustrate, in some embodiments, the scene-based image editing system **106** trains a neural network to learn how to modify a given digital image to optimize its appearance based on characteristics of the digital image. Accordingly, the scene-based image editing system **106** implements a trained neural network with learned parameters upon selection of the selectable option **4308**.

(502) As further shown in FIG. **43**B, the scene-based image editing system **106** provides a selectable option **4310** for undoing the modification. As illustrated, the selectable option **4310** includes a visual indication **4312** of a time to undo the modification. Indeed, in some cases, the scene-based image editing system **106** dynamically updates the visual indication **4312** to indicate how much time is left before the modification is automatically confirmed. Thus, the scene-based image editing system **106** notifies the user of the time limit associated with the selectable option **4310** for undoing the modification.

(503) Turning to FIG. **44**, additional detail will now be provided regarding various components and capabilities of the scene-based image editing system **106**. In particular, FIG. **44** shows the scene-based image editing system **106** implemented by the computing device **4400** (e.g., the server(s) **102** and/or one of the client devices **110***a*-**110***n*). Additionally, the scene-based image editing system **106** is also part of the image editing system **104**. As shown, in one or more embodiments, the scene-based image editing system **106** includes, but is not limited to, a neural network application manager **4402**, a semantic scene graph generator **4404**, an image modification engine **4406**, a user interface manager **4408**, and data storage **4410** (which includes neural networks **4412**, an image analysis graph **4414**, a real-world class description graph **4416**, and a behavioral policy graph **4418**).

(504) As just mentioned, and as illustrated in FIG. **44**, the scene-based image editing system **106** includes the neural network application manager **4402**. In one or more embodiments, the neural network application manager **4402** implements one or more neural networks used for editing a digital image, such as a segmentation neural network, an inpainting neural network, a shadow detection neural network, an attribute classification neural network, or various other machine learning models used in editing a digital image. In some cases, the neural network application manager **4402** implements the one or more neural network automatically without user input. For instance, in some cases, the neural network application manager **4402** utilizes the one or more neural networks to pre-process a digital image before receiving user input to edit the digital image. Accordingly, in some instances, the neural network application manager **4402** implements the one or more neural networks in anticipation of modifying the digital image.

(505) Additionally, as shown in FIG. **44**, the scene-based image editing system **106** includes the semantic scene graph generator **4404**. In one or more embodiments, the semantic scene graph generator **4404** generates a semantic scene graph for a digital image. For instance, in some cases, the scene-based image editing system **106** utilizes information about a digital image gathered via one or more neural networks (e.g., as implemented by the neural network application manager **4402**) and generates a semantic scene graph for the digital image. In some cases, the semantic scene graph generator **4404** generates a semantic scene graph for a digital image automatically without

user input (e.g., in anticipation of modifying the digital image). In one or more embodiments, the semantic scene graph generator **4404** generates a semantic scene graph for a digital image using an image analysis graph, a real-world class description graph, and/or a behavioral policy graph.
(506) As shown in FIG. **44**, the scene-based image editing system **106** also includes the image modification engine **4406**. In one or more embodiments, the image modification engine **4406** modifies a digital image. For instance, in some cases, the image modification engine **4406** modifies a digital image by modifying one or more objects portrayed in the digital image. For instance, in some cases, the image modification engine **4406** deletes an object from a digital image or moves an object within the digital image. In some implementations, the image modification engine **4406** modifies one or more attributes of an object. In some embodiments, the image modification engine **4406** modifies an object in a digital image based on a relationship between the object and another object in the digital image.
(507) Further, as shown in FIG. **44**, the scene-based image editing system **106** includes the user interface manager **4408**. In one or more embodiments, the user interface manager **4408** manages the graphical user interface of a client device. For instance, in some cases, the user interface manager **4408** detects and interprets user interactions with the graphical user interface (e.g., detecting selections of objects portrayed in a digital image). In some embodiments, the user interface manager **4408** also provides visual elements for display within the graphical user interface, such as visual indications of object selections, interactive windows that display attributes of objects, and/or user interactions for modifying an object.
(508) Additionally, as shown in FIG. **44**, the scene-based image editing system **106** includes data storage **4410**. In particular, data storage **4410** includes neural networks **4412**, an image analysis graph **4414**, a real-world class description graph **4416**, and a behavioral policy graph **4418**.
(509) Each of the components **4402-4418** of the scene-based image editing system **106** optionally include software, hardware, or both. For example, the components **4402-4418** include one or more instructions stored on a computer-readable storage medium and executable by processors of one or more computing devices, such as a client device or server device. When executed by the one or more processors, the computer-executable instructions of the scene-based image editing system **106** cause the computing device(s) to perform the methods described herein. Alternatively, the components **4402-4418** include hardware, such as a special-purpose processing device to perform a certain function or group of functions. Alternatively, the components **4402-4418** of the scene-based image editing system **106** include a combination of computer-executable instructions and hardware.
(510) Furthermore, the components **4402-4418** of the scene-based image editing system **106** may, for example, be implemented as one or more operating systems, as one or more stand-alone applications, as one or more modules of an application, as one or more plug-ins, as one or more library functions or functions that may be called by other applications, and/or as a cloud-computing model. Thus, the components **4402-4418** of the scene-based image editing system **106** may be implemented as a stand-alone application, such as a desktop or mobile application. Furthermore, the components **4402-4418** of the scene-based image editing system **106** may be implemented as one or more web-based applications hosted on a remote server. Alternatively, or additionally, the components **4402-4418** of the scene-based image editing system **106** may be implemented in a suite of mobile device applications or "apps." For example, in one or more embodiments, the scene-based image editing system **106** comprises or operates in connection with digital software applications such as ADOBE® PHOTOSHOP® or ADOBE® ILLUSTRATOR®. The foregoing are either registered trademarks or trademarks of Adobe Inc. in the United States and/or other countries.
(511) FIGS. **1-44**, the corresponding text, and the examples provide a number of different methods, systems, devices, and non-transitory computer-readable media of the scene-based image editing system **106**. In addition to the foregoing, one or more embodiments can also be described in terms of flowcharts comprising acts for accomplishing the particular result, as shown in FIGS. **45-50**.

FIGS. **45-50** may be performed with more or fewer acts. Further, the acts may be performed in different orders. Additionally, the acts described herein may be repeated or performed in parallel with one another or in parallel with different instances of the same or similar acts.

(512) FIG. **45** illustrates a flowchart for a series of acts **4500** for implementing an object-aware modification of a digital image in accordance with one or more embodiments. FIG. **45** illustrates acts according to one embodiment, alternative embodiments may omit, add to, reorder, and/or modify any of the acts shown in FIG. **45**. In some implementations, the acts of FIG. **45** are performed as part of a method. For example, in some embodiments, the acts of FIG. **45** are performed as part of a computer-implemented method. Alternatively, a non-transitory computer-readable medium can store instructions thereon that, when executed by at least one processor, cause the at least one processor to perform operations comprising the acts of FIG. **45**. In some embodiments, a system performs the acts of FIG. **45**. For example, in one or more embodiments, a system includes at least one memory device comprising a segmentation neural network and a content-aware fill machine learning model. The system further includes at least one processor configured to cause the system to perform the acts of FIG. **45**.

(513) The series of acts **4500** includes an act **4502** for generating object masks for a plurality of objects for a digital image. For instance, in one or more embodiments, the act **4502** involves pre-processing a digital image to enable object-based editing of the digital image by generating, utilizing a segmentation neural network, an object mask for each object of a plurality of objects of the digital image. In one or more embodiments, generating the object mask for each object of the plurality of objects of the digital image comprises generating the object mask for each object upon receiving the digital image and before detecting a user interaction with the digital image.

(514) The series of acts **4500** also includes an act **4504** for detecting a first user interaction with an object of the digital image. For example, in one or more embodiments, the act **4504** involves, after pre-processing the digital image, detecting a first user interaction with an object in the digital image displayed via a graphical user interface.

(515) The series of acts **4500** further includes an act **4506** for surfacing the object mask for the object in response to the first user interaction. To illustrate, in one or more embodiments, the act **4506** involves, after pre-processing the digital image, surfacing, via the graphical user interface, the object mask for the object in response to the first user interaction. In one or more embodiments, surfacing, via the graphical user interface, the object mask for the object in response to the first user interaction comprises providing, for display within the graphical user interface, a visual indication of the object mask in association with the object.

(516) Additionally, the series of acts **4500** includes an act **4508** for performing an object-aware modification of the digital image using the object mask for the object. For instance, in some implementations, the act **4508** involves, after pre-processing the digital image, performing an object-aware modification of the digital image in response to a second user interaction with the object mask for the object.

(517) As shown in FIG. **45**, the act **4508** includes a sub-act **4510** for moving the object within the digital image. The act **4508** further includes a sub-act **4512** for deleting the object from the digital image. Indeed, in one or more embodiments, performing the object-aware modification of the digital image in response to the second user interaction with the object mask for the object comprises deleting the object from the digital image or moving the object within the digital image using the object mask for the object.

(518) Speaking more generally, in some embodiments, performing the object-aware modification of the digital image comprises removing the object from an initial position within the digital image. In some cases, the scene-based image editing system **106** further generates a content fill for the object mask for the object utilizing a content-aware fill machine learning model; and exposes, in response to removing the object from the initial position, the content fill for the object mask within the digital image. For example, in some implementations, the scene-based image editing system **106**

places, before detecting the first user interaction with the object, the content fill behind the object mask at the initial position of the object within the digital image. In one or more embodiments, generating the content fill utilizing the content-aware fill machine learning model comprises generating the content fill utilizing an inpainting neural network.

(519) To provide an illustration, in one or more embodiments, the scene-based image editing system **106** generates, utilizing a segmentation neural network, an object mask for each object of a plurality of objects of a digital image; generates a content fill for each object mask utilizing a content-aware fill machine learning model; generates a completed background for the digital image by placing each content fill behind the object mask for which each content fill was generated; detects, after generation of the completed background, a user input to move or delete an object in the digital image; and exposes the content fill behind the object mask for the object upon moving or deleting the object.

(520) In some cases, generating the completed background comprises generating the completed background before detecting the user input to move or delete the object. Additionally, in some instances, exposing the content fill behind the object mask upon moving or deleting the object comprises exposing a portion of the completed background associated with the content fill upon moving or deleting the object. In some cases, the scene-based image editing system **106** further provides the digital image for display within a graphical user interface in response to receiving the digital image; and maintains, before detecting the user input to move or delete the object, the digital image for display within the graphical user interface, the digital image having the content fill for each object mask covered by a corresponding object.

(521) In one or more embodiments, the scene-based image editing system **106** exposes the content fill behind the object mask for the object upon moving or deleting the object by exposing the content fill behind the object mask for the object upon moving the object from a first position to a second position within the digital image; and maintains the completed background for the digital image after moving the object by maintaining background pixels associated with the second position to which the object is moved. In some cases, maintaining the background pixels associated with the second position to which the object is moved comprises maintaining an additional content fill associated with the second position to which the object is moved, the second position corresponding to a position previously occupied by another object of the digital image. Further, in some embodiments, the scene-based image editing system **106** detects an additional user input to move or delete the object; and exposes the background pixels associated with the second position of the object upon moving or deleting the object in response to the additional user input.

(522) In some implementations, the scene-based image editing system **106** further provides the digital image for display within a graphical user interface; detects a user selection of the object via the graphical user interface; and provides, via the graphical user interface in response to the user selection, a visual indication of the object mask for the object.

(523) To provide another illustration, in one or more embodiments, the scene-based image editing system **106** receives a digital image portraying a plurality of objects; generates, in response to receiving the digital image and utilizing the segmentation neural network, an object mask for each object of the plurality of objects; generates, utilizing the content-aware fill machine learning model, a completed background for the digital image that includes generated background pixels behind each object; detects user input to move or delete an object of the plurality of objects; and modifies the digital image by moving or deleting the object in accordance with the user input; and exposing a set of background pixels behind the object from the completed background.

(524) In some embodiments, the scene-based image editing system **106** generates, utilizing the hole-filling model, the completed background for the digital image by generating, for at least one object of the digital image, a set of background pixels for behind the at least one object based on additional background pixels associated with one or more other areas of the digital image. Further, in some cases, the scene-based image editing system **106** detects an additional user input for

moving or deleting an additional object from the plurality of objects; and modifies the digital image by moving or deleting the additional object in accordance with the additional user input and exposing an additional set of background pixels behind the additional object from the completed background. In some instances, the scene-based image editing system **106** detects the user input for moving or deleting the object from the plurality of objects by detecting a plurality of user input for deleting the plurality of objects of the digital image; and modify the digital image by moving or deleting the object in accordance with the user input by modifies the digital image by deleting the plurality of objects to reveal the completed background generated using the content-aware fill machine learning model.
(525) In still further embodiments, generating the content fill for the object mask of the object comprises completing at least part of a second object positioned behind the object and then exposing the content fill behind the object mask comprises exposing the completed second object. Furthermore, the method comprises detecting an additional user input to move or delete the second object and exposing an additional content fill behind the second object upon moving or deleting the second object.
(526) FIG. **46** illustrates a flowchart for a series of acts **4600** for removing distracting objects from a digital image in accordance with one or more embodiments. FIG. **46** illustrates acts according to one embodiment, alternative embodiments may omit, add to, reorder, and/or modify any of the acts shown in FIG. **46**. In some implementations, the acts of FIG. **46** are performed as part of a method. For example, in some embodiments, the acts of FIG. **46** are performed as part of a computer-implemented method. Alternatively, a non-transitory computer-readable medium can store instructions thereon that, when executed by at least one processor, cause the at least one processor to perform operations comprising the acts of FIG. **46**. In some embodiments, a system performs the acts of FIG. **46**. For example, in one or more embodiments, a system includes at least one memory device comprising a distractor detection neural network and an object detection machine learning model. The system further includes at least one processor configured to cause the system to perform the acts of FIG. **46**.
(527) The series of acts **4600** includes an act **4602** for providing, for display, a digital image displaying different types of objects. For instance, in one or more embodiments, the act **4602** involves providing, for display within a graphical user interface of a client device, a digital image displaying a plurality of objects, the plurality of objects comprising a plurality of different types of objects. In one or more embodiments, generating, utilizing the segmentation neural network, the object mask for the objects of the plurality of objects comprises generating, utilizing the segmentation neural network, one or more object masks for one or more non-human objects from the plurality of objects.
(528) The series of acts **4600** also includes an act **4604** for generating object masks for the objects without user input. For example, in some embodiments, the act **4604** involves generating, utilizing a segmentation neural network and without user input, an object mask for objects of the plurality of objects.
(529) Additionally, the series of acts **4600** includes an act **4606** for classifying the objects as a main subject object or a distracting object. To illustrate, in one or more embodiments, the act **4606** involves determining, utilizing a distractor detection neural network, a classification for the objects of the plurality of objects, wherein the classification for an object comprises a main subject object or a distracting object.
(530) Further, the series of acts **4600** includes an act **4608** for removing at least one object from the digital image based on classifying the object as a distracting object. For instance, in some implementations, the act **4608** involves removing at least one object from the digital image, based on classifying the at least one object as a distracting object.
(531) As shown in FIG. **46**, the act **4608** includes a sub-act **4810** for deleting the object mask for the at least one object. Indeed, in one or more embodiments, the scene-based image editing system

106 removes the at least one object from the digital image, based on classifying the at least one object as a distracting object, by deleting the object mask for the at least one object. As further shown, the act **4608** further includes a sub-act **4812** for exposing content fill generated for the at least one object. Indeed, in one or more embodiments, the scene-based image editing system **106** generates, utilizing a content-aware fill machine learning model and without user input, a content fill for the at least one object; and places the content fill behind the object mask for the at least one object within the digital image so that removing the at least one object from the digital image exposes the content fill.
(532) In one or more embodiments, the scene-based image editing system **106** provides, for display via the graphical user interface of the client device, at least one visual indication for the at least one object indicating that the at least one object is a distracting object. In some instances, the scene-based image editing system **106** further receives, via the graphical user interface, a user interaction within an additional object portrayed in the digital image, indicating that the additional object includes an additional distracting object; and provides, for display via the graphical user interface, an additional visual indication for the additional object indicating that the additional object includes an additional distracting object. In some instances, the scene-based image editing system **106** removes, based on the user interaction indicating that the additional object includes an additional distracting object, the additional object from the digital image by deleting an additional object mask for the additional object.
(533) In some instances, the scene-based image editing system **106** facilitates the removal of arbitrary portions of a digital image selected by the user. For instance, in some cases, the scene-based image editing system **106** receives, via the graphical user interface of the client device, a user selection of a portion of the digital image including pixels unassociated with the objects of the plurality of objects; and modifies the digital image by removing the portion of the digital image selected by the user selection. In some embodiments, the scene-based image editing system **106** further generates, utilizing a content-aware fill machine learning model, a content fill for the portion of the digital image selected by the user selection; and modifies, in response to removing the portion of the digital image selected by the user selection, the digital image by replacing the portion of the digital image with the content fill.
(534) In one or more embodiments, the scene-based image editing system **106** also determines, utilizing a shadow detection neural network and without user input, a shadow associated with the at least one object within the digital image; and removes the shadow from the digital image along with the at least one object.
(535) To provide an illustration, in one or more embodiments, the scene-based image editing system **106** determines, utilizing a distractor detection neural network, one or more non-human distracting objects in a digital image; provides, for display within a graphical user interface of a client device, a visual indication of the one or more non-human distracting objects with a selectable option for removing the one or more non-human distracting objects; detects, via the graphical user interface, a user interaction with the selectable option for removing the one or more non-human distracting objects from the digital image; and modifies, in response to detecting the user interaction, the digital image by removing the one or more non-human distracting objects.
(536) In one or more embodiments, determining the one or more non-human distracting objects comprises determining a set of non-human distracting objects portrayed in the digital image; and providing the visual indication of the one or more non-human distracting objects comprises providing, for each non-human distracting object from the set of non-human distracting objects, a corresponding visual indication. In some instances, the scene-based image editing system **106** detects, via the graphical user interface of the client device, a user interaction with a non-human distracting object from the set of non-human distracting objects. The scene-based image editing system **106** removes, from display via the graphical user interface, the corresponding visual indication for the non-human distracting object in response to detecting the user interaction with the

non-human distracting object. Accordingly, in some embodiments, modifying the digital image by removing the one or more non-human distracting objects comprises modifying the digital image by removing at least one non-human distracting object from the set of non-human distracting objects while maintaining the non-human distracting object based on detecting the user interaction with the non-human distracting object.

(537) Further, in some cases, the scene-based image editing system **106** detects, via the graphical user interface, a user selection of an additional non-human object portrayed in the digital image; and adds the additional non-human object to the set of non-human distracting objects in response to detecting the user selection of the additional non-human object. Accordingly, in some embodiments, modifying the digital image by removing the one or more non-human distracting objects comprises modifying the digital image by removing the set of non-human distracting objects including the additional non-human object selected via the user selection.

(538) In some implementations, the scene-based image editing system **106** determines, utilizing the distractor detection neural network, at least one non-human main subject object in the digital image. As such, in some instances, modifying the digital image by removing the one or more non-human distracting objects comprises maintaining the at least one non-human main subject object within the digital image while removing the one or more non-human distracting objects.

(539) To provide another illustration, in one or more embodiments, the scene-based image editing system **106** determines, utilizing the object detection machine learning model, a plurality of objects portrayed in a digital image, the plurality of objects comprising at least one human object and at least one non-human object; determines, utilizing the distractor detection neural network, classifications for the plurality of objects by classifying a subset of objects from the plurality of objects as distracting objects; provides, for display within a graphical user interface of a client device, visual indications that indicate the subset of objects have been classified as distracting object; receives, via the graphical user interface, a user interaction modifying the subset of objects by adding an object to the subset of objects or removing at least one object from the subset of objects; and modifies the digital image by deleting the modified subset of objects from the digital image.

(540) In some embodiments, the scene-based image editing system **106** determines the classifications for the plurality of objects by classifying an additional subset of object from the plurality of objects as main subject objects. In some cases, the scene-based image editing system **106** provides the visual indications that indicate the subset of objects have been classified as distracting objects by highlighting the subset of objects or generating borders around the subset of objects within the digital image. Further, in some implementations, the scene-based image editing system **106** further determines, utilizing a shadow detection neural network, shadows for the subset of objects classified as distracting objects; provides, for display within the graphical user interface of a client device, additional visual indications for the shadows for the subset of objects; and modifies the digital image by deleting a modified subset of shadows corresponding to the modified subset of objects.

(541) FIG. **47** illustrates a flowchart for a series of acts **4700** for detecting shadows cast by objects in a digital image in accordance with one or more embodiments. FIG. **47** illustrates acts according to one embodiment, alternative embodiments may omit, add to, reorder, and/or modify any of the acts shown in FIG. **47**. In some implementations, the acts of FIG. **47** are performed as part of a method. For example, in some embodiments, the acts of FIG. **47** are performed as part of a computer-implemented method. Alternatively, a non-transitory computer-readable medium can store instructions thereon that, when executed by at least one processor, cause the at least one processor to perform operations comprising the acts of FIG. **47**. In some embodiments, a system performs the acts of FIG. **47**. For example, in one or more embodiments, a system includes at least one memory device comprising a shadow detection neural network. The system further includes at least one processor configured to cause the system to perform the acts of FIG. **47**.

(542) The series of acts **4700** includes an act **4702** for receiving a digital image. For instance, in some cases, the act **4702** involves receiving a digital image from a client device.
(543) The series of acts **4700** also includes an act **4704** for detecting an object portrayed in the digital image. For example, in some embodiments, the act **4704** involves detecting, utilizing a shadow detection neural network, an object portrayed in the digital image.
(544) As shown in FIG. **47**, the act **4704** includes a sub-act **4706** for generating an object mask for the object. For example, in some instances, the sub-act **4706** involves generating, utilizing the shadow detection neural network, an object mask for the object.
(545) The series of acts **4700** further includes an act **4708** for detecting a shadow portrayed in the digital image. To illustrate, in some implementations, the act **4708** involves detecting, utilizing the shadow detection neural network, a shadow portrayed in the digital image.
(546) As shown in FIG. **47**, the act **4708** includes a sub-act **4710** for generating a shadow mask for the shadow. For instance, in some cases, the sub-act **4710** involves generating, utilizing the shadow detection neural network, a shadow mask for the shadow.
(547) Further, the series of acts **4700** includes an act **4712** for generating an object-shadow pair prediction that associates the object with the shadow. For example, in one or more embodiments, the act **4712** involves generating, utilizing the shadow detection neural network, an object-shadow pair prediction that associates the shadow with the object.
(548) As illustrated by FIG. **47**, the act **4712** includes a sub-act **4714** for generating the object-shadow pair prediction using the object mask and the shadow mask. To instance, in some cases, the sub-act **4714** involves generating, utilizing the shadow detection neural network, the object-shadow pair prediction that associates the shadow with the object by generating, utilizing the shadow detection neural network, the object-shadow pair prediction based on the object mask and the shadow mask.
(549) In one or more embodiments, the scene-based image editing system **106** further provides the digital image for display within a graphical user interface of the client device; receives, via the graphical user interface, a selection of the object portrayed in the digital image; and in response to receiving the selection of the object: provides, for display within the graphical user interface, a visual indication of the selection of the object; and provides, for display within the graphical user interface, an additional visual indication indicating that the shadow is included in the selection based on the object-shadow pair prediction that associates the shadow with the object. In some cases, the scene-based image editing system **106**, in response to receiving the selection of the object, provides, for display within the graphical user interface, a suggestion to add the shadow to the selection. Accordingly, in some instances, providing the additional visual indication indicating the shadow is included in the selection comprises providing the additional visual indication in response to receiving an additional user interaction for including the shadow in the selection.
(550) In some implementations, the scene-based image editing system **106** receives one or more user interactions for modifying the digital image by modifying the object portrayed in the digital image; and modifies the digital image by modifying the object and the shadow in accordance with the one or more user interactions based on the object-shadow pair prediction.
(551) Additionally, in some embodiments, detecting, utilizing the shadow detection neural network, the object portrayed in the digital image comprises detecting, utilizing the shadow detection neural network, a plurality of objects portrayed in the digital image; detecting, utilizing the shadow detection neural network, the shadow portrayed in the digital image comprises detecting, utilizing the shadow detection neural network, a plurality of shadows portrayed in the digital image; and generating, utilizing the shadow detection neural network, the object-shadow pair prediction that associates the shadow with the object, the object-shadow pair prediction that associates each object from the plurality of objects with a shadow from the plurality of shadows cast by the object within the digital image.
(552) To provide an illustration, in one or more embodiments, the scene-based image editing

system **106** generates, utilizing a shadow detection neural network, object masks for a plurality of objects portrayed in a digital image; generates, utilizing the shadow detection neural network, shadow masks for a plurality of shadows portrayed in the digital image; determines, utilizing the shadow detection neural network, an association between each shadow from the plurality of shadows and each object from the plurality of objects using the object masks and the shadow masks; and provides, to a client device, an object-shadow pair prediction that provides object-shadow pairs using the association between each shadow and each object.
(553) In some embodiments, generating, utilizing the shadow detection neural network, the object masks for the plurality of objects portrayed in the digital image comprises generating, via a first stage of the shadow detection neural network and for each object of the plurality of objects, an object mask corresponding to the object and a combined object mask corresponding to other objects of the plurality of objects. Further, in some cases, the scene-based image editing system **106** generates, for each object of the plurality of objects, a second stage input by combining the object masks corresponding to the object, the combined object mask corresponding to the other objects, and the digital image. Accordingly, in some instances, generating, utilizing the shadow detection neural network, the shadow masks for the plurality of shadows portrayed in the digital image comprises generating, via a second stage of the shadow detection neural network, the shadow masks for the plurality of shadows using the second stage input for each object.
(554) In one or more embodiments, generating, utilizing the shadow detection neural network, the shadow masks for the plurality of shadows portrayed in the digital image comprises generating, utilizing the shadow detection neural network and for each shadow portrayed in the digital image, a shadow mask corresponding to the shadow and a combined shadow mask corresponding to other shadows of the plurality of shadows. Also, in some embodiments, determining, utilizing the shadow detection neural network, the association between each shadow and each object using the object masks and the shadow masks comprises determining the association between each shadow and each object utilizing the shadow mask and the combined shadow mask generated for each shadow. Further, in some instances, providing, to the client device, the object-shadow pair prediction that provides the object-shadow pairs using the association between each shadow and each object comprises providing, for display within a graphical user interface of the client device, a visual indication indicating the association for at least one object-shadow pair.
(555) In one or more embodiments, the scene-based image editing system **106** further receives one or more user interactions to move an object of the plurality of objects within the digital image; and modifies, in response to receiving the one or more user interactions, the digital image by moving the object and a shadow associated with the object within the digital image based on the object-shadow pair prediction. Further, in some cases, the scene-based image editing system **106** receives one or more user interactions to delete an object of the plurality of objects from the digital image; and modifies, in response to receiving the one or more user interactions, the digital image by removing the object and a shadow associated with the object from the digital image based on the object-shadow pair prediction. In some instances, the scene-based image editing system **106** generates, before receiving the one or more user interactions to delete the object, a first content fill for the object and a second content fill for the shadow associated with the object utilizing a content-aware fill machine learning model; and provides the first content fill and the second content fill within the digital image so that removal of the object exposes the first content fill and removal of the shadow exposes the second content fill.
(556) To provide another illustration, in one or more embodiments, the scene-based image editing system **106** generates, utilizing an instance segmentation model of the shadow detection neural network, object masks for a plurality of objects portrayed in a digital image; determines input to a shadow segmentation model of the shadow detection neural network by combining the object masks for the plurality of objects and the digital image; generates, utilizing the shadow segmentation model and the input, shadow masks for shadows associated with the plurality of

objects within the digital image; and provides, for display on a graphical user interface of a client device, a visual indication associating an object from the plurality of objects and a shadow from the shadows utilizing the shadow masks.
(557) In one or more embodiments, the scene-based image editing system **106** generates the object masks for the plurality of objects by generating an object mask corresponding to each object of the plurality of objects; generates combined object masks for the plurality of objects, each combined object mask corresponding to two or more objects from the plurality of objects; and determines the input to the shadow segmentation model by combining the combined object masks with the object masks and the digital image. In some cases, combining the combined object masks with the object masks and the digital image comprises, for each object of the plurality of objects, concatenating an object mask corresponding to the object, a combined object mask corresponding to other objects of the plurality of objects, and the digital image.
(558) Further, in some cases, the scene-based image editing system **106** generates the shadow masks for the shadows by generating a shadow mask corresponding to each shadow of the shadows; generates, utilizing the shadow segmentation model and the input, combined shadow masks for the shadows, each combined shadow mask corresponding to two or more shadows from the shadows; and determines associations between the plurality of objects and the shadows utilizing the shadow masks and the combined shadow masks.
(559) FIG. **48** illustrates a flowchart for a series of acts **4800** for expanding an object mask for an object in a digital image in accordance with one or more embodiments. FIG. **48** illustrates acts according to one embodiment, alternative embodiments may omit, add to, reorder, and/or modify any of the acts shown in FIG. **48**. In some implementations, the acts of FIG. **48** are performed as part of a method. For example, in some embodiments, the acts of FIG. **48** are performed as part of a computer-implemented method. Alternatively, a non-transitory computer-readable medium can store instructions thereon that, when executed by at least one processor, cause the at least one processor to perform operations comprising the acts of FIG. **48**. In some embodiments, a system performs the acts of FIG. **48**. For example, in one or more embodiments, a system includes at least one memory device comprising a segmentation neural network. The system further includes at least one processor configured to cause the system to perform the acts of FIG. **48**.
(560) The series of acts **4800** includes an act **4802** for providing, for display, a digital image displaying a plurality of objects. For example, in some cases, the act **4802** involves providing, for display within a graphical user interface of a client device, a digital image displaying a plurality of objects.
(561) The series of acts **4800** also includes an act **4804** for generating object masks for objects of the plurality of objects. For instance, in some embodiments, the act **4804** involves generating, utilizing a segmentation neural network and without user input, object masks for objects of the plurality of objects.
(562) Additionally, the series of acts **4800** includes an act **4806** for identifying an object to remove from the digital image. To illustration, in some implementations, the act **4806** involves identifying an object to delete from a digital image.
(563) Further, the series of acts **4800** includes an act **4808** for determining portions of the digital image abutting the object mask for the object. As shown in FIG. **48**, the act **4808** includes a sub-act **4810** for determining foreground abutting the object mask. For instance, in some cases, the sub-act **4810** involves determining foreground abutting an object mask for the object to remove. Also, as shown in FIG. **48**, the act **4808** includes a sub-act **4812** for determining background abutting the object mask. For example, in some implementations, the sub-act **4812** involves determining background abutting the object mask for the object to remove.
(564) The series of acts **4800** further includes an act **4814** for generating an expanded object mask from the object mask. As shown in FIG. **48**, the act **4814** includes a sub-act **4816** for expanding the object mask into the foreground by a first amount, and a sub-act **4818** for expanding the object

mask into the background by a second amount Indeed, in one or more embodiments, the scene-based image editing system **106** generates an expanded object mask by: expanding the object mask into the foreground abutting the object mask by a first amount; and expanding the object mask into the background abutting the object mask by a second amount that differs from the first amount.
(565) In one or more embodiments, expanding the object mask into the background abutting the object mask by the second amount that differs from the first amount comprises expanding the object mask into the background abutting the object mask by the second amount that is greater than the first amount. In some implementations, expanding the object mask into the foreground abutting the object mask by the first amount comprises: expanding the object mask into the foreground abutting the object mask by the second amount; and reducing an expansion of the object mask into the foreground so that the expansion corresponds to the first amount. In some cases, the scene-based image editing system **106** determines that expanding the object mask into the foreground abutting the object mask by the second amount overlaps with at least one object mask of at least one other object portrayed in the digital image. Accordingly, in some instances, reducing the expansion of the object mask into the foreground so that the expansion corresponds to the first amount comprises removing overlapping pixels from the expanded object mask.
(566) Additionally, the series of acts **4800** includes an act **4820** for deleting the expanded object mask. For instance, in some cases, the act **4820** involves deleting the object from the digital image by deleting the expanded object mask.
(567) The series of acts **4800** further includes an act **4822** for inpainting a hole corresponding to the deleted expanded object mask. To illustrate, in one or more embodiments, the act **4822** involves inpainting a hole corresponding to the deleted expanded object mask utilizing an inpainting neural network.
(568) In one or more embodiments, inpainting the hole corresponding to the deleted expanded object mask utilizing the inpainting neural network comprises: generating, before deleting the expanded object mask and without user input, a content fill for the expanded object mask utilizing the inpainting neural network; placing, before deleting the expanded object mask and without user input, the content fill behind the expanded object mask; and exposing the content fill upon deleting the expanded object mask. In some embodiments, generating the content fill for the expanded object mask utilizing the inpainting neural network comprises generating, utilizing the inpainting neural network, foreground pixels for the foreground abutting the object mask corresponding to the first amount. Additionally, generating the content fill for the expanded object mask utilizing the inpainting neural network comprises generating, utilizing the inpainting neural network, background pixels for the background butting the object mask corresponding to the second amount.
(569) In one or more embodiments, the scene-based image editing system **106** further identifies an additional object to remove from the digital image; expands an additional object mask for the additional object by expanding around the object mask by the second amount; and modifies the digital image by deleting the expanded additional object mask.
(570) To provide an illustration, in one or more embodiments, the scene-based image editing system **106** provides, for display within a graphical user interface of a client device, a digital image displaying a plurality of objects; generates, utilizing a segmentation neural network and without user input, object masks for objects of the plurality of objects; identifies an object to remove from the digital image; generates an expanded object mask by expanding the object mask into areas not occupied by other object masks; deletes the expanded object mask; and inpaints a hole corresponding to the deleted expanded object mask utilizing an inpainting neural network.
(571) In some embodiments, expanding the object mask into the areas not occupied by the other object masks comprises generating a combined object mask corresponding to other objects from the plurality of objects. Such embodiments further comprise expanding the object mask into areas not occupied by the combined object mask corresponding to the other objects. In some cases, generating, utilizing the segmentation neural network, the object masks for the objects of the

plurality of objects comprises generating, utilizing the segmentation neural network, the object mask for the object and the other object masks for the other objects; and generating the combined object mask corresponding to the other objects comprises generating the combined object mask based on a union of the other object masks for the other objects. Further, in some instances, expanding the object mask into the areas not occupied by the other object masks comprises: expanding the object mask of the object into portions of the digital image that abut the object mask by a set amount of pixels; detecting a region of overlap between the expanded object mask for the object and the combined object mask corresponding to the other objects; and removing the region of overlap from the expanded object mask for the object.

(572) In one or more embodiments, identifying the object to remove from the digital image comprises detecting a user selection of the object via the graphical user interface of the client device. Additionally, in some embodiments, identifying the object to remove from the digital image comprises classifying the object as a distracting object utilizing a distractor detection neural network. Further, in some instances, inpainting the hole corresponding to the deleted expanded object mask utilizing the inpainting neural network comprises generating, utilizing the inpainting neural network, a content fill for the expanded object mask for the object, the content fill covering the areas not covered by the other object masks.

(573) In some implementations, the scene-based image editing system **106** further detects, utilizing a shadow detection neural network, a shadow for the object within the digital image; and deletes the shadow from the digital image along with the expanded object mask.

(574) To provide another illustration, in one or more embodiments, the scene-based image editing system **106** receives, from a client device, a digital image portraying a plurality of objects having a first object overlapping a second object; generate, utilizing the segmentation neural network and without user input, object masks for the plurality of objects; generate an expanded object mask for the first object by: expanding an object mask by an expansion amount; detecting a region of overlap between the expanded object mask for the first object and an object mask for the second object; and removing the region of overlap from the expanded object mask for the first object; and modifies the digital image by deleting the first object from the digital image using the expanded object mask for the first object.

(575) In some embodiments, removing the region of overlap from the expanded object mask for the first object comprises removing a portion of the expanded object mask corresponding to a subset of the region of overlap. In some cases, the scene-based image editing system **106** further generates, utilizing an inpainting neural network and without user input, a content fill for the expanded object mask for the object; and inpaints a hole that remains in the digital image after deleting the expanded object mask utilizing the content fill generated via the inpainting neural network. In some instances, the scene-based image editing system **106** further identifies the first object for deletion by receiving a user selection to delete the first object or classifying the first object as a distracting object. Further, in some embodiments, detecting the region of overlap between the expanded object mask for the first object and the object mask for the second object comprises detecting the region of overlap between the expanded object mask for the first object and a combined object mask that corresponds to the second object and at least one third object portrayed in the digital image.

(576) FIG. **49** illustrates a flowchart for a series of acts **4900** for modifying attributes of an object in a digital image in accordance with one or more embodiments. FIG. **49** illustrates acts according to one embodiment, alternative embodiments may omit, add to, reorder, and/or modify any of the acts shown in FIG. **49**. In some implementations, the acts of FIG. **49** are performed as part of a method. For example, in some embodiments, the acts of FIG. **49** are performed as part of a computer-implemented method. Alternatively, a non-transitory computer-readable medium can store instructions thereon that, when executed by at least one processor, cause the at least one processor to perform operations comprising the acts of FIG. **49**. In some embodiments, a system performs the acts of FIG. **49**. For example, in one or more embodiments, a system includes at least one memory

device comprising an attribute classification neural network. The system further includes at least one processor configured to cause the system to perform the acts of FIG. **49**.

(577) The series of acts **4900** includes an act **4902** for detecting a selection of an object portrayed in a digital image. For example, in one or more embodiments, the act **4902** involves detecting a selection of an object portrayed in a digital image displayed within a graphical user interface of a client device.

(578) The series of acts **4900** also includes an act **4904** for providing an interactive window display one or more attributes of the object. For instance, in some cases, the act **4904** involves providing, for display within the graphical user interface in response to detecting the selection of the object, an interactive window displaying one or more attributes of the object. In one or more embodiments, the scene-based image editing system **106** determines the one or more attributes of the object using an attribute classification neural network.

(579) In some embodiments, providing the interactive window displaying the one or more attributes of the object comprises providing the interactive window displaying at least one of a color of the object, a shape of the object, or a material of the object.

(580) Additionally, the series of acts **4900** includes an act **4906** for receiving a user interaction to change an attribute. To illustrate, in one or more embodiments, the act **4906** involves receiving, via the interactive window, a user interaction to change an attribute from the one or more attributes.

(581) As shown in FIG. **49**, the act **4906** includes a sub-act **4908** for receiving a user interaction changing a textual description of the attribute. Indeed, in one or more embodiments, providing the interactive window displaying the one or more attributes of the object comprises providing, within the interactive window, textual descriptions of the one or more attributes; and receiving the user interaction to change the attribute comprises receiving one or more user interactions to change a textual description of the attribute. In some embodiments, receiving the one or more user interactions to change the textual description of the attribute comprises: receiving, via the interactive window, a first user interaction selecting the attribute from the one or more attributes; providing, for display within the graphical user interface in response to receiving the first user interaction, a digital keyboard for textual input; and receiving, via the digital keyboard, one or more additional user interactions entering text to change the textual description of the attribute.

(582) Additionally, as shown in FIG. **49**, the act **4906** includes a sub-act **4910** for receiving a user interaction selecting an alternative for the attribute. For instance, in one or more embodiments, receiving the user interaction to change the attribute comprises: receiving, via the interactive window, a first user interaction selecting the attribute from the one or more attributes; providing, for display within the graphical user interface in response to receiving the first user interaction, an alternative attribute menu comprising one or more alternatives for the attribute; and receiving a second user interaction selecting an alternative from the one or more alternatives.

(583) Further, as shown in FIG. **49**, the act **4906** includes a sub-act **4912** for receiving a user interaction modifying a strength of the attribute. To illustrate, in one or more embodiments, receiving the user interaction to change the attribute comprises: receiving, via the interactive window, a first user interaction selecting the attribute from the one or more attributes; providing, for display within the graphical user interface in response to receiving the first user interaction, a slider bar having an interactive slider element that indicates a strength of appearance of the attribute of the object within the digital image; and receiving a second user interaction with the interactive slider element to modify the strength of appearance of the attribute of the object within the digital image.

(584) Further, the series of acts **4900** includes an act **4914** for modifying the digital image by changing the attribute. For example, in some implementations, the act **4914** involves modifying the digital image by changing the attribute of the object in accordance with the user interaction. In one or more embodiments, modifying the digital image by changing the attribute of the object in accordance with the user interaction comprises modifying the digital image by modifying the

attribute of the object utilizing a neural network in accordance with the user interaction.
(585) To provide an illustration, in one or more embodiments, the scene-based image editing system **106** detects a selection of an object portrayed in a digital image displayed within a graphical user interface of a client device; queries a semantic scene graph for the digital image to identify one or more attributes for the object; provides, for display within the graphical user interface in response to detecting the selection of the object, an interactive window displaying the one or more attributes of the object; receives, via the interactive window, a user interaction to change an attribute of the one or more attributes; and modifies, utilizing a neural network, the digital image by changing the attribute of the object in accordance with the user interaction.
(586) In one or more embodiments, the scene-based image editing system **106** generates, in response to receiving the digital image and without user input, the semantic scene graph for the digital image by utilizing a classification neural network to determine the one or more attributes for the object. In some embodiments, querying the semantic scene graph for the digital image to identify the one or more attributes for the object comprises: locating, within the semantic scene graph, a node representing the object portrayed in the digital image; and determining, at least one attribute associated with the node representing the object within the semantic scene graph.
(587) In some implementations, the scene-based image editing system **106** generates, utilizing a segmentation neural network and without user input, an object mask for the object portrayed in the digital image. Accordingly, in some cases, detecting the selection of the object portrayed in the digital image comprising detecting the selection of the object based on the object mask. In some instances, receiving, via the interactive window, the user interaction to change the attribute of the one or more attributes comprises: receiving, via the interactive window, a first user interaction selecting the attribute from the one or more attributes; and receiving, via a digital keyboard displayed within the graphical user interface, one or more additional user interactions entering text to change a textual description of the attribute.
(588) In some instances, providing, for display within the graphical user interface, the interactive window displaying the one or more attributes of the object comprises providing, within the interactive window and in association with the attribute, a slider bar having an interactive slider element that indicates a strength of appearance of the attribute of the object within the digital image; and receiving, via the interactive window, the user interaction to change the attribute of the one or more attributes comprises receiving the user interaction with the interactive slider element to modify the strength of appearance of the attribute of the object within the digital image. Additionally, in some cases, providing, for display within the graphical user interface, the interactive window displaying the one or more attributes of the object comprises providing, within the interactive window and in association with the attribute, an alternative attribute menu comprising one or more alternatives for the attribute; and receiving, via the interactive window, the user interaction to change the attribute of the one or more attributes comprises receiving the user interaction selecting an alternative from the one or more alternatives. Further, in some embodiments, modifying the digital image by changing the attribute of the object in accordance with the user interaction comprises modifying the digital image by changing the attribute of the object within the graphical user interface to provide an updated representation of the object within the digital image.
(589) To provide another illustration, in one or more embodiments, the scene-based image editing system **106** determines, utilizing the attribute classification neural network and without user input, attributes for one or more objects portrayed in a digital image; detects, via a graphical user interface of a client device displaying the digital image, a user selection of an object from the one or more objects; provides, for display within the graphical user interface in response to the user selection, an interactive window indicating a set of attributes of the object determined via the attribute classification neural network; receives, via the interactive window, one or more user interactions to change at least one attribute from the set of attributes for the object; and provides, in the digital

image displayed within the graphical user interface, a visual representation of a change to the at least one attribute in accordance with the one or more user interactions.
(590) In one or more embodiments, the scene-based image editing system **106** receives the one or more user interactions to change the at least one attribute by receiving a plurality of user interactions to change a plurality of attributes; and provides the visual representation to the change of the at least one attribute by providing a modified digital image for display within the graphical user interface, the modified digital image having the object with the plurality of attributes changed in accordance with the plurality of user interactions. In some embodiments, the scene-based image editing system **106** receives, via the interactive window, the one or more user interactions to change the at least one attribute by receiving the one or more user interactions with at least one of an alternatives attribute menu, a slider bar, or a textual description provided for display in association with the at least one attribute. Further, in some instances, the scene-based image editing system **106** provides the interactive window indicating a set of attributes of the object by providing the interactive window indicating a color of the object and displaying a slider bar having an interactive slider element indicating a color intensity of the color of the object within the digital image; and receives, via the interactive window, the one or more user interactions to change the at least one attribute by receiving at least one user interaction with the interactive slider element to modify a color intensity of the color of the object within the digital image.
(591) FIG. **50** illustrates a flowchart for a series of acts **5000** for modifying objects within a digital image based on relationships with other objects in accordance with one or more embodiments. FIG. **50** illustrates acts according to one embodiment, alternative embodiments may omit, add to, reorder, and/or modify any of the acts shown in FIG. **50**. In some implementations, the acts of FIG. **50** are performed as part of a method. For example, in some embodiments, the acts of FIG. **50** are performed as part of a computer-implemented method. Alternatively, a non-transitory computer-readable medium can store instructions thereon that, when executed by at least one processor, cause the at least one processor to perform operations comprising the acts of FIG. **50**. In some embodiments, a system performs the acts of FIG. **50**. For example, in one or more embodiments, a system includes at least one memory device comprising a semantic scene graph for a digital image portraying a plurality of objects. The system further includes at least one processor configured to cause the system to perform the acts of FIG. **50**.
(592) The series of acts **5000** includes an act **5002** for detecting a user selection of an object from a digital image. For example, in one or more embodiments, the act **5002** involves detecting, via a graphical user interface of a client device, a user selection of an object portrayed within a digital image.
(593) The series of acts **5000** also includes an act **5004** for determining a relationship between the object and an additional object from the digital image. For instance, in some embodiments, the act **5004** involves determining, in response to detecting the user selection of the object, a relationship between the object and an additional object portrayed within the digital image. In one or more embodiments, determining the relationship between the object and the additional object comprises determining that the additional object is supported by the object within the digital image. In some embodiments, determining the relationship between the object and the additional object comprises determining a first relationship of the object with respect to the additional object and determining a second relationship of the additional object with respect to the object.
(594) Additionally, the series of acts **5000** includes an act **5006** for receiving user interactions to modify the object. To illustration, in some cases, the act **5006** involves receiving one or more user interactions for modifying the object.
(595) Further, the series of acts **5000** includes an act **5008** for modifying the object and the additional object based on the relationship. For example, in some instances, the act **5008** involves modifying the digital image in response to the one or more user interactions by modifying the object and the additional object based on the relationship between the object and the additional

object. As shown in FIG. **50**, the act **5008** includes a sub-act **5010** for moving the additional object with the object. Indeed, in one or more embodiments, modifying the object comprises moving the object within the digital image; and modifying the additional object comprises moving the additional object with the object within the digital image based on the relationship between the object and the additional object. As further shown, the act **5008** includes a sub-act **5012** for deleting the additional object with the object. Indeed, in one or more embodiments, modifying the object comprises deleting the object from the digital image; and modifying the additional object comprises deleting the additional object with the object from the digital image based on the relationship between the object and the additional object.

(596) In one or more embodiments, the scene-based image editing system **106** adds the additional object to the user selection without receiving user input for adding the additional object to the user selection based on the relationship between the object and the additional object. Accordingly, in some cases, modifying the object and the additional object based on the relationship comprises modifying the additional object based on adding the additional object to the user selection. In one or more embodiments, the scene-based image editing system **106** provides, for display within the graphical user interface, a visual indication showing the user selection of the object portrayed in the digital image; and provides, for display within the graphical user interface, an additional visual indication showing that the additional object is added to the user selection based on the relationship between the object and the additional object.

(597) In some embodiments, the scene-based image editing system **106** provides, for display within the graphical user interface, a prompt that suggests adding the additional object to the user selection based on the relationship between the object and the additional object; and adds, in response to receiving a user interaction with the prompt, the additional object to the user selection. Accordingly, in some instances, modifying the object and the additional object based on the relationship comprises modifying the additional object based on adding the additional object to the user selection.

(598) In one or more embodiments, the scene-based image editing system **106** further receives one or more user interactions for modifying the additional object; and modifies the digital image in response to the one or more user interactions by modifying the additional object without modifying the object based on the relationship between the object and the additional object.

(599) To provide an illustration, in one or more embodiments, the scene-based image editing system **106** detects a user selection of an object portrayed in a digital image displayed within a graphical user interface of a client device; queries a semantic scene graph for the digital image to identify objects having a relationship to the object; identifies an additional object having a first type of relationship to the object based on the semantic scene graph; detects a user input to move or delete the object in the digital image; moves or deletes the object; and moves or deletes the additional object based on the first type of relationship to the object.

(600) In one or more embodiments, querying the semantic scene graph to identify object having the relationship to the object comprises querying the semantic scene graph to identify nodes representing other objects portrayed in the digital image that are connected to a node representing the object within the semantic scene graph. In one or more embodiments, the scene-based image editing system **106** queries the semantic scene graph for the digital image to identify one or more behaviors of the additional object based on having the first type of relationship to the object. Accordingly, in some instances, moving or deleting the additional object based on the first type of relationship to the object comprises moving or deleting the additional object based on the one or more behaviors of the additional object.

(601) In some embodiments, the scene-based image editing system **106** further provides, for display within the graphical user interface and in response to receiving the user input to move or delete the object, a prompt that suggests adding the additional object to the user selection based on the first type of relationship to the object; and adds, in response to receiving a user interaction with the

prompt, the additional object to the user selection. As such, in some instances, moving or deleting the additional object based on the first type of relationship comprises moving or deleting the additional object based on adding the additional object to the user selection.
(602) In one or more embodiments, the scene-based image editing system **106** identifies a second additional object having a second type of relationship to the object based on the semantic scene graph; and determines to maintain a current of the second additional object within the digital image while moving or deleting the additional object based on the second additional object having the second type or relationship to the object. In some cases, identifying the second additional object having the second type of relationship to the object comprises determining that the second additional object supports the object or is holding the object within the digital image.
(603) In some instances, the scene-based image editing system **106** identifies, based on the semantic scene graph, one or more objects portrayed in the digital image that are unassociated with the object; and determines to maintain a current state of the one or more objects within the digital image while moving or deleting the object and the additional object based on the one or more objects not having a relationship with the object.
(604) To provide another illustration, in one or more embodiments, the scene-based image editing system **106** detects, via a graphical user interface of a client device, a user selection of an object from the plurality of objects portrayed in the digital image; queries a semantic scene graph to identify one or more behaviors of an additional object having a relationship with the object based on relationship indicators and behavior indicators of the semantic scene graph; detects one or more user interactions for modifying object; provides, for display within the graphical user interface and in response to detecting the one or more user interactions, a prompt that suggests adding the additional object to the user selection based on the one or more behaviors of the additional object having the relationship with the object; adds, in response to receiving a user interaction with the prompt, the additional object to the user selection; and modifies the digital image by modifying the additional object with the object based on adding the additional object to the user selection.
(605) In one or more embodiments, modifying the additional object with the object based on adding the additional object to the user selection comprises moving the additional object a distance and a direction that the object is moved based on adding the additional object to the user selection. In some cases, modifying the additional object with the object comprises: deleting the additional object and the object from the digital image utilizing object masks generated for the additional object and the object via the one or more neural network; and exposing, upon deleting the additional object and the object, content fills generated for the additional object and the object via the one or more neural networks. Additionally, in some instances, the scene-based image editing system **106**, upon receiving the digital image and without user input, generates a plurality of object masks and a plurality of content fills for the plurality of objects utilizing one or more neural networks.
(606) Embodiments of the present disclosure may comprise or utilize a special purpose or general-purpose computer including computer hardware, such as, for example, one or more processors and system memory, as discussed in greater detail below. Embodiments within the scope of the present disclosure also include physical and other computer-readable media for carrying or storing computer-executable instructions and/or data structures. In particular, one or more of the processes described herein may be implemented at least in part as instructions embodied in a non-transitory computer-readable medium and executable by one or more computing devices (e.g., any of the media content access devices described herein). In general, a processor (e.g., a microprocessor) receives instructions, from a non-transitory computer-readable medium, (e.g., a memory), and executes those instructions, thereby performing one or more processes, including one or more of the processes described herein.
(607) Computer-readable media can be any available media that can be accessed by a general purpose or special purpose computer system. Computer-readable media that store computer-

executable instructions are non-transitory computer-readable storage media (devices). Computer-readable media that carry computer-executable instructions are transmission media. Thus, by way of example, and not limitation, embodiments of the disclosure can comprise at least two distinctly different kinds of computer-readable media: non-transitory computer-readable storage media (devices) and transmission media.

(608) Non-transitory computer-readable storage media (devices) includes RAM, ROM, EEPROM, CD-ROM, solid state drives ("SSDs") (e.g., based on RAM), Flash memory, phase-change memory ("PCM"), other types of memory, other optical disk storage, magnetic disk storage or other magnetic storage devices, or any other medium which can be used to store desired program code means in the form of computer-executable instructions or data structures and which can be accessed by a general purpose or special purpose computer.

(609) A "network" is defined as one or more data links that enable the transport of electronic data between computer systems and/or modules and/or other electronic devices. When information is transferred or provided over a network or another communications connection (either hardwired, wireless, or a combination of hardwired or wireless) to a computer, the computer properly views the connection as a transmission medium. Transmissions media can include a network and/or data links which can be used to carry desired program code means in the form of computer-executable instructions or data structures and which can be accessed by a general purpose or special purpose computer. Combinations of the above should also be included within the scope of computer-readable media.

(610) Further, upon reaching various computer system components, program code means in the form of computer-executable instructions or data structures can be transferred automatically from transmission media to non-transitory computer-readable storage media (devices) (or vice versa). For example, computer-executable instructions or data structures received over a network or data link can be buffered in RAM within a network interface module (e.g., a "NIC"), and then eventually transferred to computer system RAM and/or to less volatile computer storage media (devices) at a computer system. Thus, it should be understood that non-transitory computer-readable storage media (devices) can be included in computer system components that also (or even primarily) utilize transmission media.

(611) Computer-executable instructions comprise, for example, instructions and data which, when executed by a processor, cause a general-purpose computer, special purpose computer, or special purpose processing device to perform a certain function or group of functions. In some embodiments, computer-executable instructions are executed on a general-purpose computer to turn the general-purpose computer into a special purpose computer implementing elements of the disclosure. The computer executable instructions may be, for example, binaries, intermediate format instructions such as assembly language, or even source code. Although the subject matter has been described in language specific to structural features and/or methodological acts, it is to be understood that the subject matter defined in the appended claims is not necessarily limited to the described features or acts described above. Rather, the described features and acts are disclosed as example forms of implementing the claims.

(612) Those skilled in the art will appreciate that the disclosure may be practiced in network computing environments with many types of computer system configurations, including, personal computers, desktop computers, laptop computers, message processors, hand-held devices, multiprocessor systems, microprocessor-based or programmable consumer electronics, network PCs, minicomputers, mainframe computers, mobile telephones, PDAs, tablets, pagers, routers, switches, and the like. The disclosure may also be practiced in distributed system environments where local and remote computer systems, which are linked (either by hardwired data links, wireless data links, or by a combination of hardwired and wireless data links) through a network, both perform tasks. In a distributed system environment, program modules may be located in both local and remote memory storage devices.

(613) Embodiments of the present disclosure can also be implemented in cloud computing environments. In this description, "cloud computing" is defined as a model for enabling on-demand network access to a shared pool of configurable computing resources. For example, cloud computing can be employed in the marketplace to offer ubiquitous and convenient on-demand access to the shared pool of configurable computing resources. The shared pool of configurable computing resources can be rapidly provisioned via virtualization and released with low management effort or service provider interaction, and then scaled accordingly.
(614) A cloud-computing model can be composed of various characteristics such as, for example, on-demand self-service, broad network access, resource pooling, rapid elasticity, measured service, and so forth. A cloud-computing model can also expose various service models, such as, for example, Software as a Service ("SaaS"), Platform as a Service ("PaaS"), and Infrastructure as a Service ("IaaS"). A cloud-computing model can also be deployed using different deployment models such as private cloud, community cloud, public cloud, hybrid cloud, and so forth. In this description and in the claims, a "cloud-computing environment" is an environment in which cloud computing is employed.
(615) FIG. **51** illustrates a block diagram of an example computing device **5100** that may be configured to perform one or more of the processes described above. One will appreciate that one or more computing devices, such as the computing device **5100** may represent the computing devices described above (e.g., the server(s) **102** and/or the client devices **110***a*-**110***n*). In one or more embodiments, the computing device **5100** may be a mobile device (e.g., a mobile telephone, a smartphone, a PDA, a tablet, a laptop, a camera, a tracker, a watch, a wearable device). In some embodiments, the computing device **5100** may be a non-mobile device (e.g., a desktop computer or another type of client device). Further, the computing device **5100** may be a server device that includes cloud-based processing and storage capabilities.
(616) As shown in FIG. **51**, the computing device **5100** can include one or more processor(s) **5102**, memory **5104**, a storage device **5106**, input/output interfaces **5108** (or "I/O interfaces **5108**"), and a communication interface **5110**, which may be communicatively coupled by way of a communication infrastructure (e.g., bus **5112**). While the computing device **5100** is shown in FIG. **51**, the components illustrated in FIG. **51** are not intended to be limiting. Additional or alternative components may be used in other embodiments. Furthermore, in certain embodiments, the computing device **5100** includes fewer components than those shown in FIG. **51**. Components of the computing device **5100** shown in FIG. **51** will now be described in additional detail.
(617) In particular embodiments, the processor(s) **5102** includes hardware for executing instructions, such as those making up a computer program. As an example, and not by way of limitation, to execute instructions, the processor(s) **5102** may retrieve (or fetch) the instructions from an internal register, an internal cache, memory **5104**, or a storage device **5106** and decode and execute them.
(618) The computing device **5100** includes memory **5104**, which is coupled to the processor(s) **5102**. The memory **5104** may be used for storing data, metadata, and programs for execution by the processor(s). The memory **5104** may include one or more of volatile and non-volatile memories, such as Random-Access Memory ("RAM"), Read-Only Memory ("ROM"), a solid-state disk ("SSD"), Flash, Phase Change Memory ("PCM"), or other types of data storage. The memory **5104** may be internal or distributed memory.
(619) The computing device **5100** includes a storage device **5106** including storage for storing data or instructions. As an example, and not by way of limitation, the storage device **5106** can include a non-transitory storage medium described above. The storage device **5106** may include a hard disk drive (HDD), flash memory, a Universal Serial Bus (USB) drive or a combination these or other storage devices.
(620) As shown, the computing device **5100** includes one or more I/O interfaces **5108**, which are provided to allow a user to provide input to (such as user strokes), receive output from, and

otherwise transfer data to and from the computing device **5100**. These I/O interfaces **5108** may include a mouse, keypad or a keyboard, a touch screen, camera, optical scanner, network interface, modem, other known I/O devices or a combination of such I/O interfaces **5108**. The touch screen may be activated with a stylus or a finger.

(621) The I/O interfaces **5108** may include one or more devices for presenting output to a user, including, but not limited to, a graphics engine, a display (e.g., a display screen), one or more output drivers (e.g., display drivers), one or more audio speakers, and one or more audio drivers. In certain embodiments, I/O interfaces **5108** are configured to provide graphical data to a display for presentation to a user. The graphical data may be representative of one or more graphical user interfaces and/or any other graphical content as may serve a particular implementation.

(622) The computing device **5100** can further include a communication interface **5110**. The communication interface **5110** can include hardware, software, or both. The communication interface **5110** provides one or more interfaces for communication (such as, for example, packet-based communication) between the computing device and one or more other computing devices or one or more networks. As an example, and not by way of limitation, communication interface **5110** may include a network interface controller (NIC) or network adapter for communicating with an Ethernet or other wire-based network or a wireless NIC (WNIC) or wireless adapter for communicating with a wireless network, such as a WI-FI. The computing device **5100** can further include a bus **5112**. The bus **5112** can include hardware, software, or both that connects components of computing device **5100** to each other.

(623) In the foregoing specification, the invention has been described with reference to specific example embodiments thereof. Various embodiments and aspects of the invention(s) are described with reference to details discussed herein, and the accompanying drawings illustrate the various embodiments. The description above and drawings are illustrative of the invention and are not to be construed as limiting the invention. Numerous specific details are described to provide a thorough understanding of various embodiments of the present invention.

(624) The present invention may be embodied in other specific forms without departing from its spirit or essential characteristics. The described embodiments are to be considered in all respects only as illustrative and not restrictive. For example, the methods described herein may be performed with less or more steps/acts or the steps/acts may be performed in differing orders. Additionally, the steps/acts described herein may be repeated or performed in parallel to one another or in parallel to different instances of the same or similar steps/acts. The scope of the invention is, therefore, indicated by the appended claims rather than by the foregoing description. All changes that come within the meaning and range of equivalency of the claims are to be embraced within their scope.

Claims

1. A computer-implemented method comprising: providing, for display within a graphical user interface of a client device, a digital image displaying a plurality of objects, the plurality of objects comprising a plurality of different types of objects; generating, utilizing a segmentation neural network and without user input, an object mask for objects of the plurality of objects; determining, utilizing a distractor detection neural network, a classification for the objects of the plurality of objects, wherein the classification for an object comprises a main subject object or a distracting object; removing at least one object from the digital image, based on classifying the at least one object as the distracting object, by deleting the object mask for the at least one object; determining, utilizing a shadow detection neural network and without user input, a shadow associated with the at least one object within the digital image; and removing the shadow from the digital image along with the at least one object.

2. The computer-implemented method of claim 1, wherein generating, utilizing the segmentation

neural network, the object mask for the objects of the plurality of objects comprises generating, utilizing the segmentation neural network, one or more object masks for one or more non-human objects from the plurality of objects.

3. The computer-implemented method of claim 1, further comprising: generating, utilizing a content-aware fill machine learning model and without user input, a content fill for the at least one object; and placing the content fill behind the object mask for the at least one object within the digital image so that removing the at least one object from the digital image exposes the content fill.

4. The computer-implemented method of claim 1, further comprising providing, for display via the graphical user interface of the client device, a selectable removal option for removing the at least one object based on classifying the at least one object as the distracting object.

5. The computer-implemented method of claim 1, further comprising: receiving, via the graphical user interface, a user interaction within an additional object portrayed in the digital image, indicating that the additional object includes an additional distracting object; and generating, for display within the digital image, an additional visual indication for the additional object indicating that the additional object includes an additional distracting object.

6. The computer-implemented method of claim 5, further comprising removing, based on the user interaction indicating that the additional object includes an additional distracting object, the additional object from the digital image by deleting an additional object mask for the additional object.

7. The computer-implemented method of claim 1, further comprising: receiving, via the graphical user interface of the client device, a user selection of a portion of the digital image including pixels unassociated with the objects of the plurality of objects; and modifying the digital image by removing the portion of the digital image selected by the user selection.

8. The computer-implemented method of claim 7, further comprising: generating, utilizing a content-aware fill machine learning model, a content fill for the portion of the digital image selected by the user selection; and modifying, in response to removing the portion of the digital image selected by the user selection, the digital image by replacing the portion of the digital image with the content fill.

9. The computer-implemented method of claim 1, wherein determining the shadow using the shadow detection neural network comprises using the shadow detection neural network to generate a shadow mask for the shadow.

10. A non-transitory computer-readable medium storing instructions thereon that, when executed by at least one processor, cause the at least one processor to perform operations comprising: determining, utilizing a distractor detection neural network, one or more non-human distracting objects in a digital image; providing, for display within a graphical user interface of a client device, a visual indication of the one or more non-human distracting objects with a selectable option for removing the one or more non-human distracting objects; detecting, via the graphical user interface, a user interaction with the selectable option for removing the one or more non-human distracting objects from the digital image; and modifying, in response to detecting the user interaction, the digital image by removing the one or more non-human distracting objects.

11. The non-transitory computer-readable medium of claim 10, wherein: determining the one or more non-human distracting objects comprises determining a set of non-human distracting objects portrayed in the digital image; and providing the visual indication of the one or more non-human distracting objects comprises providing, for each non-human distracting object from the set of non-human distracting objects, a corresponding visual indication.

12. The non-transitory computer-readable medium of claim 11, further comprising instructions that, when executed by the at least one processor, cause the at least one processor to perform operations comprising: detecting, via the graphical user interface of the client device, a user interaction with a non-human distracting object from the set of non-human distracting objects; and removing, from

display via the graphical user interface, the corresponding visual indication for the non-human distracting object in response to detecting the user interaction with the non-human distracting object.

13. The non-transitory computer-readable medium of claim 12, wherein modifying the digital image by removing the one or more non-human distracting objects comprises modifying the digital image by removing at least one non-human distracting object from the set of non-human distracting objects while maintaining the non-human distracting object based on detecting the user interaction with the non-human distracting object.

14. The non-transitory computer-readable medium of claim 11, further comprising instructions that, when executed by the at least one processor, cause the at least one processor to perform operations comprising: detecting, via the graphical user interface, a user selection of an additional non-human object portrayed in the digital image; and adding the additional non-human object to the set of non-human distracting objects in response to detecting the user selection of the additional non-human object.

15. The non-transitory computer-readable medium of claim 14, wherein modifying the digital image by removing the one or more non-human distracting objects comprises modifying the digital image by removing the set of non-human distracting objects including the additional non-human object selected via the user selection.

16. The non-transitory computer-readable medium of claim 10, further comprising instructions that, when executed by the at least one processor, cause the at least one processor to perform operations comprising, determining, utilizing the distractor detection neural network, at least one non-human main subject object in the digital image, wherein modifying the digital image by removing the one or more non-human distracting objects comprises maintaining the at least one non-human main subject object within the digital image while removing the one or more non-human distracting objects.

17. A system comprising: at least one memory device comprising a distractor detection neural network and an object detection machine learning model; and at least one processor configured to cause the system to: determine, utilizing the object detection machine learning model, a plurality of objects portrayed in a digital image, the plurality of objects comprising at least one human object and at least one non-human object; determine, utilizing the distractor detection neural network, classifications for the plurality of objects by classifying a subset of objects from the plurality of objects as distracting objects; provide, for display within a graphical user interface of a client device, visual indications that indicate the subset of objects have been classified as distracting object; receive, via the graphical user interface, a user interaction modifying the subset of objects by adding an object to the subset of objects or removing at least one object from the subset of objects; and modify the digital image by deleting the modified subset of objects from the digital image.

18. The system of claim 17, wherein the at least one processor is further configured to cause the system to determine the classifications for the plurality of objects by classifying an additional subset of object from the plurality of objects as main subject objects.

19. The system of claim 17, wherein the at least one processor is configured to cause the system to provide the visual indications that indicate the subset of objects have been classified as distracting objects by highlighting the subset of objects or generating borders around the subset of objects within the digital image.

20. The system of claim 17, wherein the at least one processor is further configured to cause the system to: determine, utilizing a shadow detection neural network, shadows for the subset of objects classified as distracting objects; provide, for display within the graphical user interface of a client device, additional visual indications for the shadows for the subset of objects; and modify the digital image by deleting a modified subset of shadows corresponding to the modified subset of objects.