CS230

GONE Offers Nice Editing

Jing Lim*, Alexis Goh*, Travis McGuire**
Department of Computer Science Stanford University*, Department of Electrical Engineering Stanford University**
{jinglim2, gweiying, tmcgurie}@stanford.edu
https://github.com/travismcguire/gone-offers-nice-editing (*cs230-stanford* has access)

Abstract

Image segmentation and generative inpainting are both active research areas in computer vision. We construct a model that combines state-of-the-art from both domains to provide an end-to-end approach for object detection and inpainting in images. We present results combining Mask-RCNN (1) to segment people and Generative Inpainting (2) to generatively infill segments with inferred backgrounds. Specifically, our contribution is in tuning and training Generative Inpainting on modified Mask-RCNN detection outputs with the end goal of removing people using the MS COCO 2014 data set (3).

1 Introduction

End-to-end detection, removal, and inpainting of objects in images is an open and interesting research question. It has implications for image-editing, image-based rendering and computational photography (4; 5; 6). The goal is to provide high quality image-editing where specific objects can be removed and inpainted accurately with low computational costs. In this paper, we present our work toward this task on specifically people in images. Given an input image, the goal is to generate white masks to crop people, generate reasonable background content to fill the mask, and output an image with the people removed. As the task can be broken down into a two-part process, detection and inpainting, our work is structured as a two-part model. We use:

1. Mask R-CNN (1) for instance segmentation to generate masks for target objects for deletion
2. Generative Inpainting (2) to propose visually plausible image content for masked regions

Mask R-CNN is a state-of-the-art instance segmentation model proposed in 2017 by He et al (1). Generative Inpainting is an feed-forward generative network for inpainting proposed in 2018 by Yu et al (2). We combine these two state-of-the-art models and optimize them to provide an end-to-end solution for removing people from input images. Specifically, our key contributions in this paper are modifying the output of Mask R-CNN to generate better masks that remove human relics, and training Generative Inpainting on the COCO 2014 (3) data set.

2 Related work

Our student project was inspired as a modified and new application application of MIT's Deep Angel project (7). There are few papers that treat detection and deletion as an end-to-end task; one such paper by Shetty et. al trains a single GAN to detect and remove objects automatically (8). The GAN has a two-part architecture, a mask generator and an inpainter, which poses difficulties in training. We investigate if combining two state-of-the-art models trained separately can improve performance. State-of-the-art results have been demonstrated on instance segmentation with Mask R-CNN, (mAP scores of 62.3% for an IoU = 0.5, 43.4% for an IoU = 0.7 and 39.8% on COCO 2016) (1). We focus our related work section on inpainting as inpainting is the less established of the two sub tasks.

Classical inpainting largely relies on texture matching to generate plausible content for missing regions: To infill a missing region, a search for nearest-neighbor patches with similar texture is conducted, and a simple copy-paste replication is done. (9; 10)

Deep learning and in particular generative models have enabled the use of GANs for image editing (11; 12) and inpainting. Recent work formulates the task as a conditional generation based on both texture synthesis, and higher-level image features

CS230: Deep Learning, Fall 2018, Stanford University, CA. (LateX template borrowed from NIPS 2017.)

(4; 13; 14). Our chosen architecture, Generative Inpainting by Yu et. al, proposes a novel contextual attention layer which matches generated patches with known contextual patches through convolution filters to improve texture synthesis (2), achieving more novel and realistic inpainting than previous works.

3 Data set and Features

We utilized the COCO 2014 data set. To make training feasible, we sampled random subsets of COCO 2014 for our train/val/test sets.In particular, a problem observed in our data set was the relative size of people in scene; in images where people occupied the majority of the image, both Places2 and our trained weights faced difficulties generating infills as the majority of the image was cropped. We elaborate on this in Section 5. In all, we used the following train/val/test data sets (numbered):

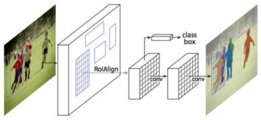
1. *Train-40k*: 40k images from COCO 2014-val used as main training set
2. *Train-2k-no-people*: 2k images sampled from 1. where images containing people were removed to investigate the performance of model trained on images without people
3. *Train-2k*: 2k images sampled from 1. to perform hyperparameter search
4. *Val-100*: 100 images from COCO 2014-test used for initial error analysis
5. *Val-100-no-large-crop*: 100 images from COCO 2014-test where images needing large crops (with people occupying > 50% of image) was removed
6. *Test-100-no-large-crop*: 100 images from COCO 2014-test sampled from the same distribution as 5.

4 Model Architecture

4.1 Image segmentation: Mask R-CNN

Mask R-CNN builds on the object detection architecture Faster R-CNN (15; 1). The loss function on Mask R-CNN is given by: $L = L_{cls} + L_{box} + L_{mask}$ where $L_{cls} + L_{box}$. L_{mask} is the average binary cross-entropy loss of the per-pixel sigmoid on the binary masks. We modified Mask R-CNN to output both templates and cropped images, experimenting with boxes crops, masks crops, and dilated masks crops of the targetted people.

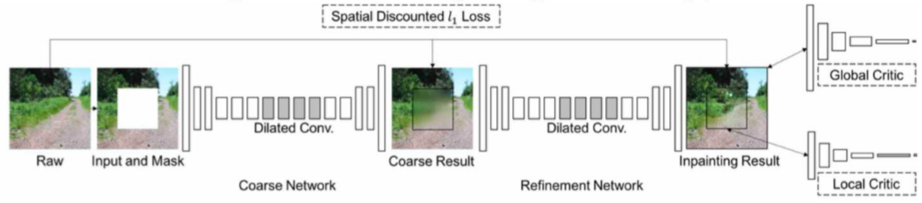
Figure 1: Mask R-CNN architecture (1)



4.2 Inpainting: Generative Inpainting

Generative Inpainting is trained by performing random crops to remove sections of training images. The first stage is a coarse reconstruction stage that uses a dilated convolution network trained on with ℓ_1 pixel-wise loss that generates rough semantic content for missing regions. The second stage is a refinement stage that is trained on two WGAN loss functions, one to examine the global coherency of generated image, and the other for the local plausibility of missing content.

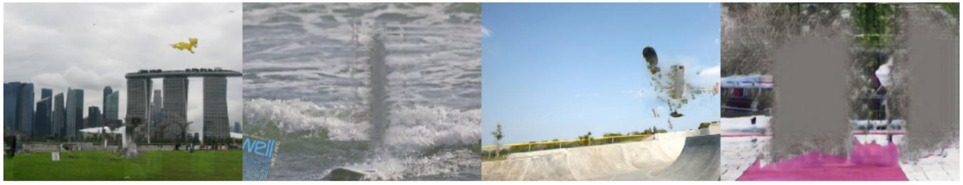
Figure 2: Generative Inpainting architecture (2)



5 Experiments

We used pretrained weights from COCO 2014 (1) on Mask R-CNN (16). These weights obtain 0.43 mAR@[0.5, 0.95] for bounding boxes, and 0.38 mAR@[0.5, 0.95] for segmentation masks.

Figure 3: (Left to right) examples of quality labels for good, okay, striped, solid



As we wanted our model to work end-to-end on images from the same distribution, we retrained Generative Inpainting on COCO 2014. Pretrained weights for Places2 (2) are used as our baseline. We discuss some performance metrics briefly:

Qualitative Metrics: Image quality was evaluated by a team of evaluators with labels defined below.

1. *good*: near perfect fill
2. *okay*: reasonable, but imperfect fill
3. *striped*: patterned but incorrect fill
4. *solid*: incorrect noise-like fill

Quantitative Metrics: As with other image generation tasks, an optimal quantitative metric is hard to establish. However, we report the gradient norm as an indicator for noise of generated patches:

- *Gradient Norm*: calculated as the absolute value of the gradient norm within the generated patch minus the gradient norm outside the generated patch with Sobel gradients

5.1 Mask R-CNN: Data set correction

From our tests, we observed that large crops were particularly difficult to fill. If large sections are removed, less context is left for the generative inference, and creating appropriate infill is a harder task. We defined a large crop as a crop greater than 50% of an image. We performed a large crop error analysis, displaying results in Table 1 below. Based on this, images with large crops were removed from the data set distribution to align with our target task. This was also justified by the assumption that people occupying $\geq$ 50% of the image are likely the main subjects of images, hence the use case of removing them is less obvious.

Table 1: Generated images by quality and crop size

	With large crop (*Val-100*)	**Without large crop (*Val-100-no-large-crop*)**
Good	13	19
Okay	12	11
Striped	23	41
Solid	52	29

5.2 Mask R-CNN: Crop strategy

Mask R-CNN was chosen on the assumption that segmentation masks would leave a greater percent of intact inference content because of its closer crop relative to boxes, allowing better inpainting. However, error analysis demonstrated a surprising trend. Images with a mask crop resulted in generally lower quality results than a box crop. It appeared human relics (hands, edges of clothes, etc) from imperfect masking remained which caused the GAN to include those edges as a feature when infilling.

To test this hypothesis, we created a dilated mask where the mask is widened to cover neighboring pixels. Infills were generated for box, mask, and dilated masked crops, shown in Figure 4. We display the error analysis results on validation data in Table 2 below. The dilated mask is generally the best and therefore utilized for future experiments.

5.3 Generative Inpainting: Learning Rate

We searched multiple values for the learning rate hyperparameter: 0.00001, 0.0001, 0.001. We trained with early stopping on *Train-2k-no-people*. Results are presented in Figure 5. As seen, modifying the learning rates caused large differences in the outputs. We theorize the learning rate 0.001 was too high and led to the discriminator learning faster than the generator, preventing the generator from learning a good mapping. We choose 0.0001, the highest learning rate able to learn textures.

Figure 4: (Left to right) generated images from original, box, mask, and dilated masks

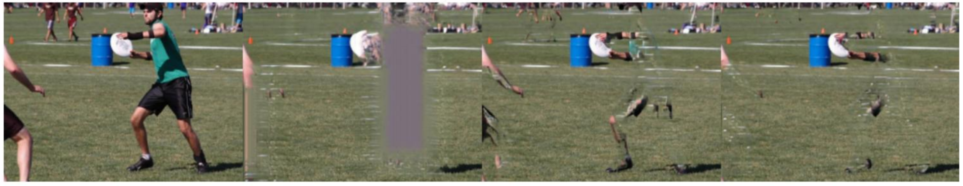
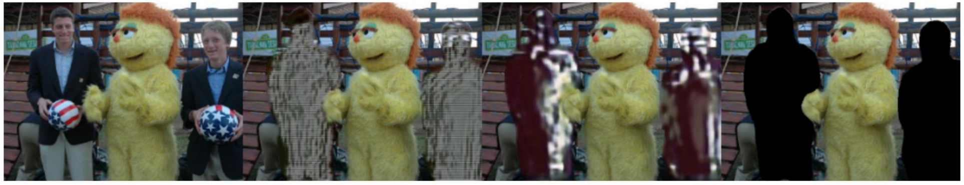


Table 2: Generated images from *Val-100-no-large-crop* by quality and type of crop

	Box	**Mask**	**Dilated Mask**
Good	19	7	29
Okay	11	18	19
Striped	38	72	46
Solid	32	3	6

Figure 5: (Left to right) original image and generated images with model learning rates of 0.00001, 0.0001, and 0.001



5.4 Generative Inpainting: ℓ_1 pixel-wise loss

For epoch 1 through 11 of the model with *Train-40k*, we observed a considerable number of striped images (patterned, but incorrect fills) lacking use of nearby context. Based on this, we experimented with tuning the ℓ_1 pixel-wise loss hyper parameter. Higher ℓ_1, $\ell_1 \geq 2.0$, is recommended to infer more from the patch's immediate neighborhood. Lower ℓ_1, $\ell_1 \leq 1.0$, is recommended refine detail in blurry images. We trained two variations four epochs, one with $\ell_1 = 2.0$ and one with the original $\ell_1 = 1.2$ both initialized with epoch 10 of our trained COCO weights. We present results in Table 3 below. We found an unclear trade-off, as the higher ℓ_1 did lead to less striped images, however, the images that left the striped category were split among better ('okay') and worse ('solid'). For that reason, we stuck with the original $\ell_1 = 1.2$.

Table 3: Generated images from *Val-100-no-large-crop* by quality and ℓ_1

	$\ell_1 = \mathbf{2.0}$	$\ell_1 = \mathbf{1.2}$
Good	3	3
Okay	10	5
Striped	50	77
Solid	36	14

5.5 Generative Inpainting: People vs no-people

We also tested if training Generative Inpainting on data without people would result in better performance than data with people based on the task being to eliminate people and error analysis. We trained Generative Inpainting separately on both *Train-2k* and *Train-2k-no-people*, initialized with Epoch 10 weights of the model trained with *Train-40k*.

We found that the performance of the model trained on *Train-2k-no-people* was marginally better than *Train-2k* 4.

6 Results

6.1 Results with Generative Inpainting trained on COCO

We present results for our Generative Inpainting model trained on *Train-40k* data alongside the baseline below. Figure 6 shows a sample of images, full sets of which are provided in the GONE GitHub. Table 5 contains quality labels assigned during error analysis, which show our model trained on COCO steadily improving toward the baseline. The baseline was trained significantly

Table 4: Generated images from *Val-100-no-large-crop* by quality and training data

	people	**no people**
Good	3	5
Okay	12	15
Striped	56	52
Solid	26	27

longer than our model according to the Generative Inpainting paper and GitHub (2). Table 6 shows quantitative results as the gradient norm. The gradient norm is shown to decrease with training, however, the epoch 16 gradient norm is below the baseline despite having lower quality images. This shows limitations to this quantitative metric as discussed in the metrics subsection.

Figure 6: (Left to right) original image and generated image from *Test-100-no-large-crop* with original, baseline, epoch 4, epoch 8, epoch 12, and epoch 16

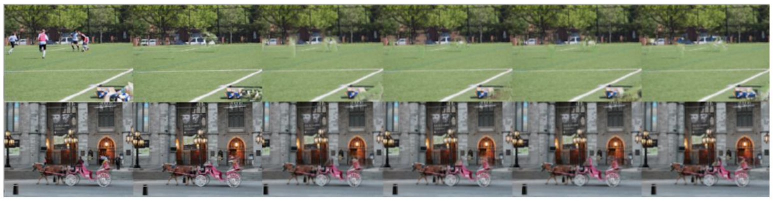


Table 5: Generated images from *Test-100-no-large-crop* by quality and COCO inpainting model epoch

	Baseline (Places)	**Epoch 4**	**Epoch 8**	**Epoch 12**	**Epoch 16**
Good	16	1	3	5	8
Okay	15	12	9	14	16
Striped	64	30	42	42	53
Solid	4	54	43	36	20

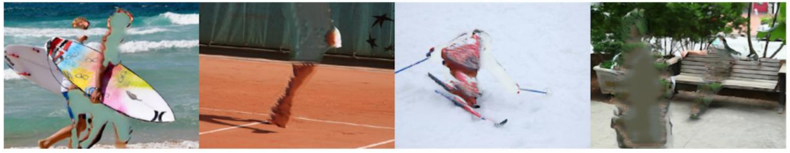
Table 6: Gradient norm on *Test-100-no-large-crop* by COCO inpainting model epoch

	Baseline (Places)	**Epoch 4**	**Epoch 8**	**Epoch 12**	**Epoch 16**
Gradient Norm	0.000183	0.000289	0.000303	0.000144	0.0000791

6.2 Test set errors

We observed several errors on test data seen in Figure 7 below. From left to right, uncropped limbs, uncropped shadows, uncropped objects used by people, and infilling a crop like a person. Some of these are limitations of the current system which are not currently target tasks (e.g. cropping shadows), but would be important to generating a perfect image.

Figure 7: Error sample from *Test-100-no-large-crop*



6.3 Further work

While the team was pleased with the results obtained in the short time during the CS230 course, there are certainly more challenges to be addressed. The team believes investigating methods to ensure better masking, a more thorough random hyperparameter search (including learning rate and ℓ_1), an experiment using larger no people training set (to help address the infilling people error), and more time to train the COCO model may all prove fruitful. Some of the test set errors could potentially be addressed by training Mask R-CNN to detect limbs, human shadows, and objects typically held by people. The team would particularly like to compare pros and cons of the two stage approach used here to the one stage approach of Shetty et. al (8).

7 Contributions

All members contributed equally to the project. The team is grateful for reliable guidance from their mentor Abhijeet Shenoi.

References

[1] K. He, G. Gkioxari, P. Dollár, and R. B. Girshick, "Mask R-CNN," *CoRR*, vol. abs/1703.06870, 2017.

[2] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. S. Huang, "Generative image inpainting with contextual attention," *CoRR*, vol. abs/1801.07892, 2018.

[3] T. Lin, M. Maire, S. J. Belongie, L. D. Bourdev, R. B. Girshick, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: common objects in context," *CoRR*, vol. abs/1405.0312, 2014.

[4] A. Newson, A. Almansa, M. Fradet, Y. Gousseau, and P. Pérez, "Video inpainting of complex scenes," *CoRR*, vol. abs/1503.05528, 2015.

[5] R. A. Yeh, C. Chen, T. Lim, M. Hasegawa-Johnson, and M. N. Do, "Semantic image inpainting with perceptual and contextual losses," *CoRR*, vol. abs/1607.07539, 2016.

[6] "Photoshop makes objects disappear with revamped content-aware fill." `https://www.digitaltrends.com/photography/photoshop-content-aware-fill-overhaul/`. Accessed: 2018-12-13.

[7] "Deep angel." `https://www.media.mit.edu/projects/deep-angel-ai/overview/`. Accessed: 2018-09-30.

[8] R. Shetty, M. Fritz, and B. Schiele, "Adversarial scene editing: Automatic object removal from weak supervision," *CoRR*, vol. abs/1806.01911, 2018.

[9] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman, "PatchMatch: A randomized correspondence algorithm for structural image editing," *ACM Transactions on Graphics (Proc. SIGGRAPH)*, vol. 28, Aug. 2009.

[10] J. Hays and A. A. Efros, "Scene completion using millions of photographs," *ACM Transactions on Graphics (SIGGRAPH 2007)*, vol. 26, no. 3, 2007.

[11] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo, "Stargan: Unified generative adversarial networks for multi-domain image-to-image translation," *arXiv preprint*, vol. 1711, 2017.

[12] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," *arXiv preprint*, 2017.

[13] S. Iizuka, E. Simo-Serra, and H. Ishikawa, "Globally and Locally Consistent Image Completion," *ACM Transactions on Graphics (Proc. of SIGGRAPH 2017)*, vol. 36, no. 4, pp. 107:1–107:14, 2017.

[14] D. Pathak, P. Krähenbühl, J. Donahue, T. Darrell, and A. A. Efros, "Context encoders: Feature learning by inpainting," *CoRR*, vol. abs/1604.07379, 2016.

[15] S. Ren, K. He, R. B. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," *CoRR*, vol. abs/1506.01497, 2015.

[16] W. Abdulla, "Mask r-cnn for object detection and instance segmentation on keras and tensorflow." `https://github.com/matterport/Mask_RCNN`, 2017.

[17] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. S. Huang, "Free-form image inpainting with gated convolution," *arXiv preprint arXiv:1806.03589*, 2018.