3. Machine Art

In this chapter, we introduce some of the concepts that enable deep learning models to seemingly *create* art, an idea that may be paradoxical to some. The University of California, Berkeley, philosopher Alva Noë, for one, opined, "Art can help us frame a better picture of our human nature."[1] If this is true, how can machines create art? Or put differently, are the creations that emerge from these machines, in fact, art? Another interpretation—and one we like best—is that these creations are indeed art and that programmers are artists wielding deep learning models as brushes. We're not the only ones who view these works as bona fide artistry: generative adversarial network (GAN)-produced paintings have been snapped up to the tune of $400,000 a pop.[2]

[1] . Noë, A. (2015, October 5). What art unveils. The *New York Times*.

[2] . Cohn, G. (2018, October 25). AI art at Christie's sells for $432,500. The *New York Times*.

Over the course of this chapter, we cover the high-level concepts behind GANs, and you will see examples of the novel visual works they can produce. We will draw a link between the latent spaces associated with GANs and the word-vector spaces of Chapter 2. And we will cover a deep learning model that can be used as an automated tool for dramatically improving the quality of photos. But before we do any of that, let's grab a drink . . .

A BOOZY ALL-NIGHTER

Below Google's offices in Montreal sits a bar called *Les 3 Brasseurs*, a moniker that translates from French to "The 3 Brewers." It was at this watering hole in 2014, while a PhD student in Yoshua Bengio's renowned lab (Figure 1.10), that Ian Goodfellow conceived of an algorithm for fabricating realistic-looking images,[3] a technique that Yann LeCun (Figure 1.9) has hailed as the "most important" recent breakthrough in deep learning.[4]

[3] . Giles, M. (2018, February 21). The GANfather: The man who's given machines the gift of imagination. *MIT Technology Review*.

[4] LeCun, Y. (2016, July 28). *Quora*. `bit.ly/DLbreakthru`

Goodfellow's friends described to him a *generative model* they were working on, that is, a computational model that aims to produce something novel, be it a quote in the style of Shakespeare, a musical melody, or a work of abstract art. In their particular case, the friends were attempting to design a model that could generate photorealistic images such as portraits of human faces. For this to work well via the traditional machine learning approach (Figure 1.12), the engineers designing the model would need to not only catalog and approximate the critical individual features of faces like eyes, noses, and mouths, but also accurately estimate how these features should be arranged relative to each other. Thus far, their results had been underwhelming. The generated faces tended to be excessively blurry, or they tended to be missing essential elements like the nose or the ears.

Perhaps with his creativity heightened by a pint of beer or two, [5] Goodfellow proposed a revolutionary idea: a deep learning model in which two artificial neural networks (ANNs) act against each other competitively as adversaries. As illustrated in Figure 3.1, one of these deep ANNs would be programmed to produce forgeries while the other would be programmed to act as a detective and distinguish the fakes from real images (which would be provided separately). These adversarial deep learning networks would play off one another: As the *generator* became better at producing fakes, the *discriminator* would need to become better at identifying them, and so the generator would need to produce even more compelling counterfeits, and so on. This virtuous cycle would eventually lead to convincing novel images in the style of the real training images, be they of faces or otherwise. Best of all, Goodfellow's approach would circumnavigate the need to program features into the generative model manually. As we expounded with respect to machine vision (Chapter 1) and natural language processing (Chapter 2), deep learning would sort out the model's features automatically.

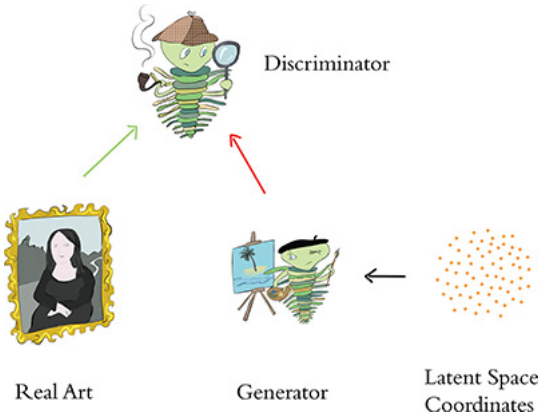

Figure 3.1 High-level schematic of a generative adversarial network (GAN). Real images, as well as forgeries produced by the generator, are provided to the discriminator, which is tasked with identifying which are the genuine articles. The orange cloud represents latent space (Figure 3.4) "guidance" that is provided to the forger. This guidance can either be random (as is generally the case during network training; see Chapter 12) or selective (during post-training exploration, as in Figure 3.3).

[5] Jarosz, A., et al. (2012). Uncorking the muse: Alcohol intoxication facilitates creative problem solving. *Consciousness and Cognition, 21*, 487–93.

Goodfellow's friends were doubtful his imaginative approach would work. So, when he arrived home and found his girlfriend asleep, he worked late to architect his dual-ANN design. It worked the first time, and the astounding deep learning family of generative adversarial networks was born!

That same year, Goodfellow and his colleagues revealed GANs to the world at the prestigious Neural Information Processing Systems (NIPS) conference.[6] Some of their results are shown in Figure 3.2. Their GAN produced these novel images by being trained on (a) handwritten digits;[7] (b) photos of human faces;[8] and (c) and (d) photos from across ten diverse classes (e.g., planes, cars, dogs).[9] The results in (c) are markedly less crisp than in (d), because the GAN that produced the latter featured neuron layers specialized for machine vision called *convolutional* layers,[10] whereas the GAN that produced the former used a more general layer type only.[11]

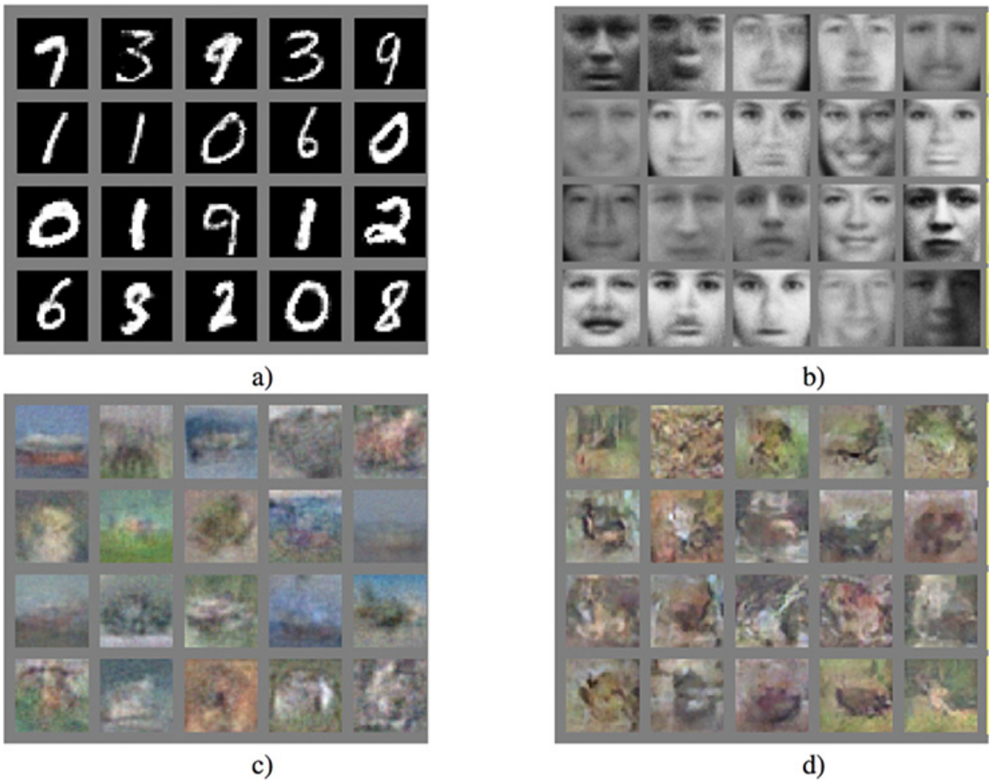

Figure 3.2 Results presented in Goodfellow and colleagues' 2014 GAN paper

[6] Goodfellow, I., et al. (2014). Generative adversarial networks. *arXiv:1406.2661*.

[7] From LeCun's classic MNIST dataset, which we use ourselves in Part II.

[8] From the Hinton (Figure 1.16) research group's Toronto Face database.

[9] The CIFAR-10 dataset, which is named after the Canadian Institute for Advanced Research that supported its creation.

[10] We detail these in Chapter 10.

[11] *Dense* layers, which are introduced in Chapter 4 and detailed in Chapter 7.

ARITHMETIC ON FAKE HUMAN FACES

Following on from Goodfellow's lead, a research team led by the American machine learning engineer Alec Radford determined architectural constraints for GANs that guide considerably more realistic image creation. Some examples of portraits of fake humans that were produced by their *deep convolutional* GANs[12] are provided in Figure 3.3. In their paper, Radford and his teammates cleverly demonstrated interpolation through, and arithmetic with, the *latent space* associated with GANs. Let's start off by explaining what latent space is before moving on to latent-space interpolation and arithmetic.

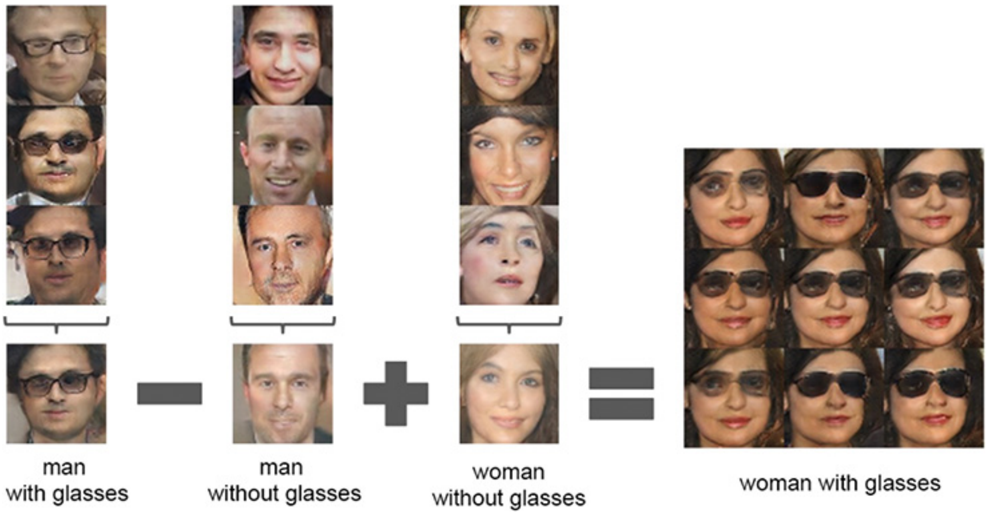

Figure 3.3 An example of latent-space arithmetic from Radford et al. (2016)

[12] Radford, A., et al. (2016). Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv:1511.06434v2*.

The latent-space cartoon in Figure 3.4 may be reminiscent of the word-vector space cartoon in Figure 2.6. As it happens, there are three major similarities between latent spaces and vector spaces. First, while the cartoon is only three-dimensional for simplicity and comprehensibility, latent spaces are *n*-dimensional spaces, usually in the order of hundreds of dimensions. The latent space of the GAN you'll later architect yourself in Chapter 12, for example, will have $n = 100$ dimensions. Second, the closer two points are in the latent space, the more similar the images that those points represent. And third, movement through the latent space in any particular direction can correspond to a gradual change in a concept being represented, such as age or gender for the case of photorealistic faces.

By picking two points far away from each other along some *n*-dimensional axis representing age, interpolating between them, and sampling points from the interpolated line, we could find what appears to be the same (fabricated) man gradually appearing to be older and older.[13] In our latent-space cartoon (Figure 3.4), we represent such an "age" axis in purple. To observe interpolation through an authentic GAN latent space, we recommend scanning through Radford and colleagues' paper for, as an example, smooth rotations of the "photo angle" of synthetic bedrooms. At the time of writing, the state of the art in GANs can be viewed at `bit.ly/InterpCeleb`. This video, produced by researchers at the graphics-card manufacturer Nvidia, provides a breathtaking interpolation through high-quality portrait "photographs" of ersatz celebrities.[14,15]

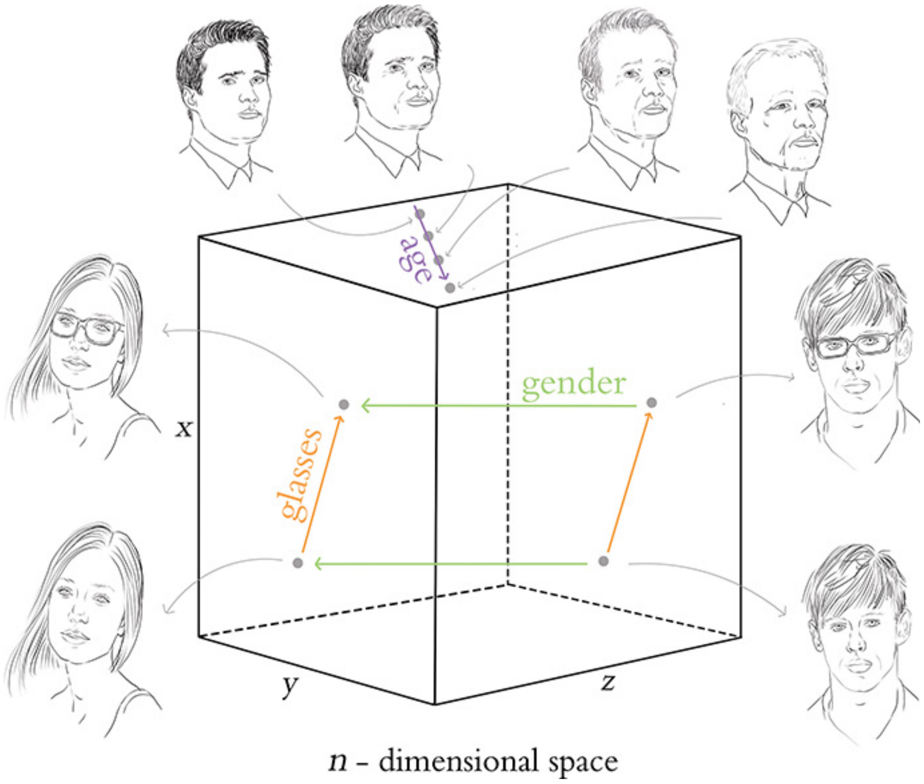

Figure 3.4 A cartoon of the latent space associated with generative adversarial networks (GANs). Moving along the purple arrow, the latent space corresponds to images of a similar-looking individual aging. The green arrow represents gender, and the orange one represents the inclusion of glasses on the face.

[13] A technical aside: As is the case with vector spaces, this “age” axis (or any other direction within latent space that represents some meaningful attribute) may be orthogonal to all of the *n* dimensions that constitute the axes of the *n*-dimensional space. We discuss this further in Chapter 11.

[14] Karras, T., et al. (2018). Progressive growing of GANs for improved quality, stability, and variation. *Proceedings of the International Conference on Learning Representations*.

[15] To try your hand at distinguishing between real and GAN-generated faces, visit whichfaceisreal.com (http://whichfaceisreal.com).

Moving a step further with what you’ve learned, you could now perform arithmetic with images sampled from a GAN’s latent space. When sampling a point within the latent space, that point can be represented by the coordinates of its location—the resulting vector is analogous to the word vectors described in Chapter 2. As with word vectors, you can perform arithmetic with these vectors and move through the latent space in a semantic way. Figure 3.3 showcases an instance of latent-space arithmetic from Radford and his coworkers. Starting with a point in their GAN’s latent space that represents a man with glasses, subtracting a point that represents a man *without* glasses, and adding a point representing a *woman* without glasses, the resulting point exists in the latent space near to images that represent women *with* glasses. Our cartoon in Figure 3.4 illustrates how the relationships between meaning in latent space are stored (again, akin to the way they are in word-vector space), thereby facilitating arithmetic on points in latent space.

STYLE TRANSFER: CONVERTING PHOTOS INTO MONET (AND VICE VERSA)

One of the more magical applications of GANs is *style transfer*. Zhu, Park, and their coworkers from the Berkeley Artificial Intelligence Research (BAIR) Lab introduced a new flavor of GAN[16] that enables stunning examples of this, as shown in Figure 3.5. Alexei Efros, one of the paper's coauthors, took photos while on holiday in France and the researchers employed their CycleGAN to transfer these photos into the style of the Impressionist painter Claude Monet, the nineteenth-century Dutch artist Vincent Van Gogh, and the Japanese Ukiyo-e genre, among others. If you navigate to `bit.ly/cycleGAN`, you'll be delighted to discover instances of the inverse (Monet paintings converted into photorealistic images), as well as:

- Summer scenes converted into wintry ones, and vice versa
- Baskets of apples converted into baskets of oranges, and vice versa
- Flat, low-quality photos converted into what appear to be ones captured by high-end (single-lens reflex) cameras
- A video of a horse running in a field converted into a zebra
- A video of a drive taken during the day converted into a nighttime one

Figure 3.5 Photos converted into the styles of well-known painters by CycleGANs

[16] Called "CycleGANs" because they retain image consistency over multiple cycles of network training. Zhu, J.-Y., et al. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. *arXiv:1703.10593*.

MAKE YOUR OWN SKETCHES PHOTOREALISTIC

Another GAN application out of Alexei Efros's BAIR lab, and one that you can amuse yourself with straightaway, is *pix2pix*.[17] If you make your way to `bit.ly/pix2pixDemo`, you can interactively translate images from one type to another. Using the edges2cats tool, for example, we sketched the three-eyed cat in the left-hand panel of Figure 3.6 to generate the photorealistic(-ish) mutant kitty in the right-hand panel. As it takes your fancy, you are also welcome to convert your own creative visions of felines, shoes, handbags, and building façades into photorealistic analogs within your browser. The authors of the pix2pix paper call their approach a *conditional* GAN (cGAN for short) because the generative adversarial network produces an output that is conditional on the particular input provided to it.

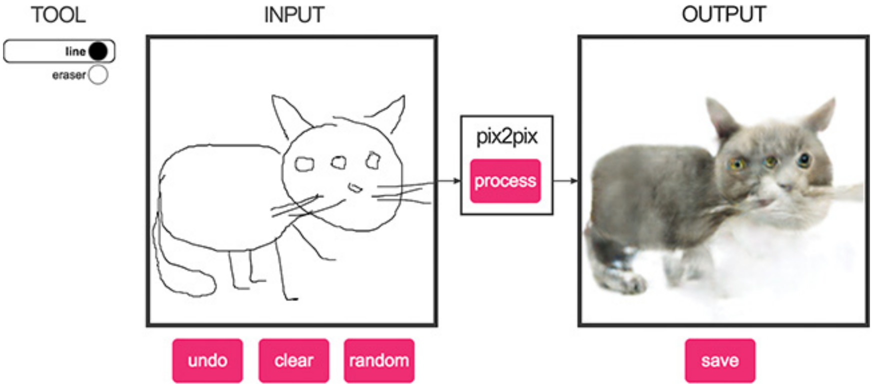

Figure 3.6 A mutant three-eyed cat (right-hand panel) synthesized via the pix2pix web application. The sketch in the left-hand panel that the GAN output was conditioned on was clearly not doodled by this book's illustrator, Aglaé, but one of its other authors (who shall remain nameless).

[17] Isola, P., et al. (2017). Image-to-image translation with conditional adversarial networks. *arXiv:1611.07004*.

CREATING PHOTOREALISTIC IMAGES FROM TEXT

To round out this chapter, we'd like you to take a *gan*der at the truly photorealistic high-resolution images in Figure 3.7. These images were generated by StackGAN,[18] an approach that *stacks* two GANs on top of each other. The first GAN in the architecture is configured to produce a rough, low-resolution image with the general shape and colors of the relevant objects in place. This is then supplied to the second GAN as its input, where the forged "photo" is refined by fixing up imperfections and adding considerable detail. The StackGAN is a cGAN like the pix2pix network in the preceding section; however, the image output is conditioned on *text* input instead of an image.

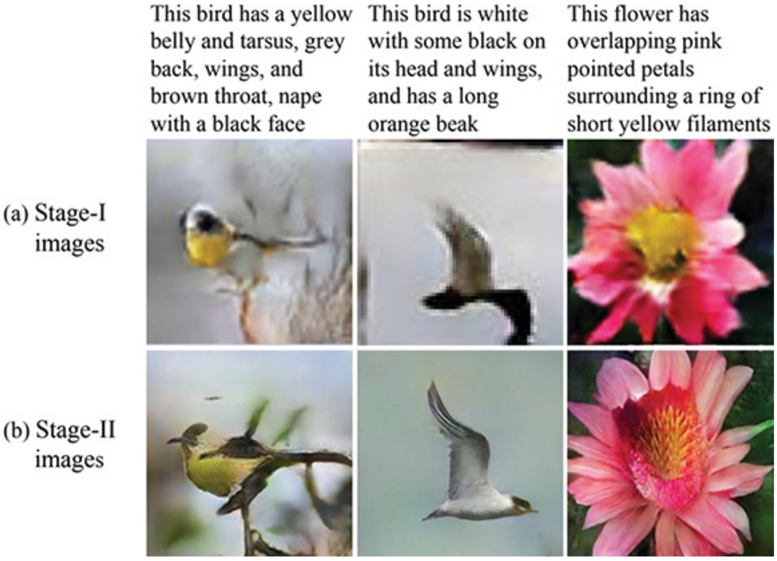
This bird has a yellow belly and tarsus, grey back, wings, and brown throat, nape with a black face
This bird is white with some black on its head and wings, and has a long orange beak
This flower has overlapping pink pointed petals surrounding a ring of short yellow filaments
(a) Stage-I images
(b) Stage-II images

Figure 3.7 Photorealistic high-resolution images output by StackGAN, which involves two GANs stacked upon each other

18. Zhang, H., et al. (2017). StackGAN: Text to photo-realistic image synthesis with stacked generative adversarial networks. *arXiv:1612.03242v2*.

IMAGE PROCESSING USING DEEP LEARNING

Since the advent of digital camera technology, image processing (both on-device and postprocessing) has become a staple in most (if not all) photographers' workflows. This ranges from simple on-device processing, such as increasing saturation and sharpness immediately after capture, to complex editing of raw image files in software applications like Adobe Photoshop and Lightroom.

Machine learning has been used extensively in on-device processing, where the camera manufacturer would like the image that the consumer sees to be vibrant and pleasing to the eye with minimal user effort. Some examples of this are:

- Early face-recognition algorithms in point-and-shoot cameras, which optimize the exposure and focus for faces or even selectively fire the shutter when they recognize that the subject is smiling (as in Figure 1.13)

- Scene-detection algorithms that adjust the exposure settings to capture the whiteness of snow or activate the flash for nighttime photos

In the postprocessing arena a variety of automatic tools exists, although generally photographers who are taking the time to postprocess images are investing considerable time and domain-specific knowledge into color and exposure correction, denoising, sharpening, tone mapping, and touching up (to name just a few of the corrections that may be applied).

Historically, these corrections have been difficult to execute programmatically, because, for example, denoising might need to be applied selectively to different images and even different parts of the same image. This is exactly the kind of intelligent application that deep learning is poised to excel at.

In a 2018 paper from Chen Chen and his collaborators at Intel Labs,[19] deep learning was applied to the enhancement of images that were taken in near total darkness, with astonishing results (Figure 3.8). In a phrase, their deep learning model involves convolutional layers organized into the innovative *U-Net*[20] architecture (which we break down for you in Chapter 10). The authors generated a custom dataset for training this model: the See-in-the-Dark dataset consists of 5,094 raw images of very dark scenes using a short-exposure image[21] with a corresponding long-exposure image (using a tripod for stability) of the same scene. The exposure times on the long-exposure images were 100 to 300 times those of the short-exposure training images, with actual exposure times in the range of 10 to 30 seconds. As demonstrated in Figure 3.8, the deep-learning-based image-processing pipeline of U-Net (right panel) far outperforms the results of the traditional pipeline (center panel). There are, however, limitations as yet:

- The model is not fast enough to perform this correction in real time (and certainly not on-device); however, runtime optimization could help here.
- A dedicated network must be trained for different camera-models and sensors, whereas a more generalized and camera-model-agnostic approach would be favorable.
- While the results far exceed the capabilities of traditional pipelines, there are still some artifacts present in the enhanced photos that could stand to be improved.
- The dataset is limited to selected static scenes and needs to be expanded to other subjects (most notably, humans).

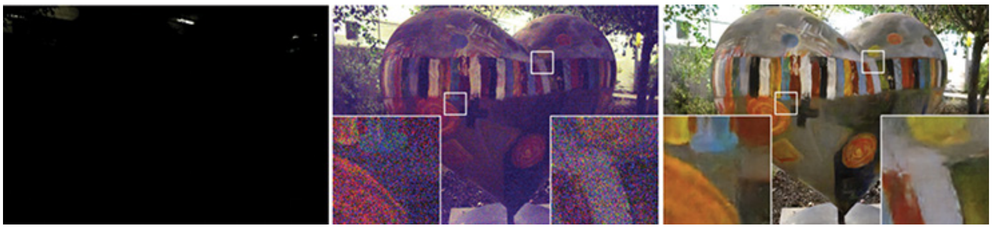

Figure 3.8 A sample image (left) processed using a traditional pipeline (center) and the deep learning pipeline by Chen et al. (right)

[19] Chen, C., et al. (2018) Learning to see in the dark. *arXiv:1805.01934*.

[20] Ronneberger et al. (2015) U-Net: Convolutional networks for biomedical image segmentation. *arXiv: 1505.04597*.

[21] That is, a short enough exposure time to enable practical handheld capture without motion blur but that renders images too dark to be useful.

Limitations aside, this work nevertheless provides a beguiling peek into how deep learning can adaptively correct images in photograph postprocessing pipelines with a level of sophistication not before seen from machines.

SUMMARY

In this chapter, we introduced GANs and conveyed that this deep learning approach encodes exceptionally sophisticated representations within their latent spaces. These rich visual representations enable GANs to create novel images with particular, granular artistic styles. The outputs of GANs aren't purely aesthetic; they can be practical, too. They can, as examples, simulate data for training autonomous vehicles, hurry the pace of prototyping in the fields of fashion and architecture, and substantially augment the capacities of creative humans.[22]

[22] Carter, S., and Nielsen, M. (2017, December 4). Using artificial intelligence to augment human intelligence. Distill. `distill.pub/2017/aia`

In Chapter 12, after we get all of the prerequisite deep learning theory out of the way, you'll construct a GAN yourself to imitate sketches from the Quick, Draw! dataset (introduced at the end of Chapter 1). Take a gander at Figure 3.9 for a preview of what you'll be able to do.

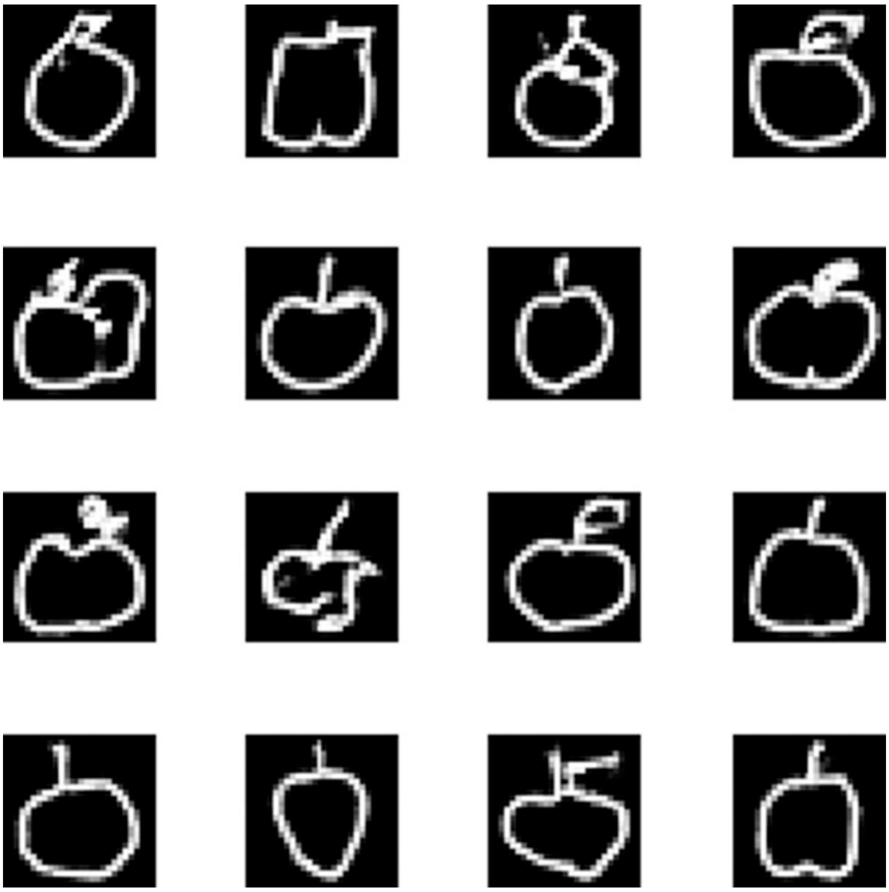

Figure 3.9 Novel "hand drawings" of apples produced by the GAN architecture we develop together in Chapter 12. Using this approach, you can produce machine-drawn "sketches" from across any of the hundreds of categories involved in the Quick, Draw! game.