CHAPTER 11

Generative adversarial networks for histopathology staining

Aashutosh Ganesh[a,b] **and Koshy George**[a,c,d]
[a]PES Center for Intelligent Systems, PES University, Bangalore, India
[b]Radboud University, Nijmegen, The Netherlands
[c]Department of Electronics and Communication Engineering, PES University, Bangalore, India
[d]SRM University—AP, Guntur District, Andhra Pradesh, India

11.1 Introduction

Generative adversarial networks (GANs), a type of deep learning proposed in Ref. [1], consist of two networks, the *generator* and the *discriminator*. The former belongs to the class of generative or forward models, which depends on unsupervised learning to determine the distribution of the training data, and the latter belongs to the class of discriminative or backward models that ascertains the decision boundaries via supervised learning [2]. (Generative modeling and some applications are treated in Ref. [3]. Some recent books on GANs are Refs. [4, 5].) While generative methods model class-conditional distributions and prior probabilities, discriminative methods estimate posterior probabilities without explicitly modeling the probability distributions. Note that the discriminator is a classifier. In the context of GANs, it attempts to distinguish between real data and the data created by the generator. Albeit it is possible to arrive at several generative-discriminative pairs, what makes GANs unique is that the generator and the discriminator are pitted against each other in a two-player game seeking to find the Nash equilibrium [6].

Several discriminators have been proposed that successfully map a high-dimensional input to a class label [7–9]. This has been made possible due to the back-propagation algorithm and the use of piecewise linear activation functions with well-behaved gradients [10–12]. Dropout algorithms [13–15] have also contributed to this success. A GAN essentially is a procedure for creating generators that mitigates the difficulties faced in creating meaningful generator-discriminator models. These obstacles include approximating intractable probabilistic computations and leverage piecewise linear activation functions.

GANs are similar to variational autoencoders (VAE) [16, 17] in that both approaches are used to determine the distribution of data using unsupervised learning. Accordingly, both have two networks. While the decoder is generative, the encoder is a recognition model. Such an approach leads to an intractable distribution which then has to be

Generative Adversarial Networks for Image-to-Image Translation
https://doi.org/10.1016/B978-0-12-823519-5.00010-5

approximated by another tractable distribution and then use the method of variational inference. The different approach adopted makes GANs better than VAEs.

Suppose that $\mathcal{G}$ and $\mathcal{D}$ are the two feed-forward neural networks, respectively, representing the generative and discriminative networks. In a GAN, $\mathcal{G}$ and $\mathcal{D}$ simultaneously participate in the following two-player game:

$$\min_{\mathcal{G}} \max_{\mathcal{D}} \{\mathbb{E}_x[\log \mathcal{D}(x)] + \mathbb{E}_z[\log\{1 - \mathcal{D}(\mathcal{G}(z))\}]\} \tag{11.1}$$

An input prior $p(z)$ is first defined, and then mapped to the space $\mathcal{G}(z)$. While the discriminator maximizes the probability of assigning the correct label to training examples (real data) and minimizes the probability to samples from $\mathcal{G}$, the generator $\mathcal{G}$ is trained to maximize the probability assigned by the discriminator to those samples it generates. Thus, $\mathcal{G}$ attempts to capture the data distribution and $\mathcal{D}$ estimates the probability that a sample came from the training data rather than from $\mathcal{G}$.

The applications of GANs have been quite varied, and includes image segmentation, text-to-image synthesis, and high-resolution image generation [18–20]. (A recent survey of applications is available in Ref. [21].) In particular, GANs have been found useful in medical imaging; see Refs. [22–25]. Deep learning and GANs have helped in the automation of diagnostics of diseases such as breast cancer and gastrointestinal disease, segmentation of nuclei, image reconstruction, and image translation of X-ray image to CT scans [26–31]. This has been made possible largely due to strides in computing power, storage capacity, and image capture techniques [32].

Histology and histopathology are the careful study of microscopic tissues. Histopathology is important for diagnosis and is considered a gold standard; for example, it is required for cancer diagnosis, where the microscopic tissue is analyzed by a pathologist. A fundamental step in histopathology is staining caused by chemical reactions induced in the tissue under analysis, and results in accentuated features that help in diagnosis. The stains range from the commonly used hematoxylin and eosin (H&E) stain—devised independently by Wissowzky in 1876 and Busch in 1877 [33]—to the relatively rare Grocott-Gömöri methenamine silver (GMS) stain proposed by Gömöri in 1946 [34]. Different stains affect the tissue on a slide distinctively thereby highlighting particular features for the pathologist [35]. Whenever required, developing diverse stained histopathological slides of the same tissue sample is a parallel, laborious, and a time-consuming process. Moreover, it is subject to human error. Evidently, histology staining and histological analysis are cumbersome processes, where automation can be beneficial to diagnosticians.

With recent developments in deep learning, accelerated computing, and storage, histological image analysis has had some transformative changes. In Ref. [36], breast cancer classification has an accuracy of over 98.4% with a recurrent patch-based convolutional neural network (CNN). GANs have showcased its usefulness in histopathology: stain

normalization is introduced in Ref. [37], InfoGAN [38] and WGAN [39] are used for feature extraction in Ref. [40], and synthetic histopathology image generation is discussed in Ref. [41]. The process of histology staining has also shown some scope for automation, as illustrated in Refs. [42, 43], where histopathology staining is achieved through style transfer [44] or through residual GANs [45].

From a machine learning perspective, each stain results in a different feature space, and a transformative network has the ability to transform one space to another. The latter is a classic image-to-image translation problem. Thus, we can frame the problem of transforming one stained tissue to another as an image-to-image translation problem [46]. In this chapter, we consider the problem of transformation of a feature space corresponding to one stain to another posed as an image-to-image translation problem, and present a solution based on GANs.

Specifically, the use case and limits of the image-to-image translation utilizing GANs are demonstrated here using the images from the Automatic Nonrigid Histological Image Registration (the ANHIR) challenge dataset [47–52]. (ANHIR challenge was part of the IEEE International Symposium on Biomedical Imaging [ISBI] in 2019, where the call was to register tissues across different samples for large images.) Histology staining in this chapter is framed as a domain adaptation problem for each stain. The dataset consists of various tissues with different types of stains per tissue. This challenge requires the tissues to be registered for a given pair of input and target images. Registration is an important task in medical imaging as it allows diagnosticians to extract more information from one image than they typically do from single samples. Histology registration postalignment allows the viewer to see the information from multiple stains on the same sample.

However, since these samples of differently stained tissues are not readily available, there is a potential application of converting one tissue stain to another stain type. As mentioned earlier, GANs have proved effective in various image generation tasks such as segmentation and synthetic data generation. The ANHIR dataset is utilized here to demonstrate this domain adaptation problem wherein a stained histology image leads to a histology image with a different stain. In this chapter, we discuss the details of the implementation. Specifically, the preparation of the dataset and the methodology to solve an image-to-image translation problem are discussed. Moreover, the efficacy of GANs when the number of available images is relatively small is showcased and we illustrate some techniques that yield better performance. The results of this chapter are based on an implementation of the code primarily done in python, specifically TensorFlow [53]. Due to constraints on available datasets, it must be emphasized that the suggested methodology may not be completely clinically viable as yet.

This chapter is organized as follows. GANs are presented in Section 11.2. In this section, we present the vanilla GAN and other variations relevant in our context, the objective functions considered for optimization, and the image-quality metrics.

The image-to-image transformation problem is discussed in Section 11.3. Histology is outlined in Section 11.4. The networks and the dataset used in this chapter are described in Section 11.5, and the results presented and discussed in Section 11.6, followed by conclusions in Section 11.7.

11.2 Generative adversarial networks

The vanilla GAN introduced in Ref. [1] ensured that the generator-discriminator pair played a two-player min-max game seeking the Nash equilibrium (a saddle point), as described in Eq. (11.1). Both $\mathcal{D}(\mathcal{G}(z))$ and $\mathcal{D}(x)$ represent probabilities. A trained discriminator $\mathcal{D}$ is such that it maximizes the probability $\mathcal{D}(x)$ for an image x that belongs to the input distribution. From a prior distribution $p(z)$, a sample z is input to $\mathcal{G}$. The resulting output $\mathcal{G}(z)$ of an untrained generator evidently does not belong to the input distribution and hence rightly classified as a *fake* image by the trained discriminator; that is, the value of $\mathcal{D}(\mathcal{G}(z))$ is nowhere near unity. The generator $\mathcal{G}$ is trained so that the probability $\mathcal{D}(\mathcal{G}(z))$ is maximized. Essentially, $\mathcal{D}$ tries to reject these images as fake while $\mathcal{G}$ attempts to fool $\mathcal{D}$ into thinking they are real. Eventually, $\mathcal{G}$ learns sufficiently to generate samples that correspond to the distribution of the real data.

11.2.1 Improvements to vanilla GAN

Albeit the vanilla GAN is comparatively better than its contemporaries, a number of issues can affect its performance. Some of the issues are as follows: First, it is likely that the generator network does not improve as fast as the discriminator network causing the former to output less than ideal images. Second, the generator produces samples from a limited class; this issue is called mode collapse. Third, the networks are trained using the back-propagation algorithm and hence require the computation of gradients. The problem of unstable gradients, vanishing gradients associated with Kullback-Leibler (KL) [54] and Jensen-Shannon (JS) [55] divergences have been well reported. Hyperparameter optimization is essential to strike a balance between the generator and discriminator wherein one does not improve at a rate that the other cannot keep up. Several improvements to the architecture, batch training, and techniques to avoid mode collapse have been proposed [56]. In this chapter, we suggest the required improvements for the image-to-image translation problem.

11.2.2 Deep convolutional GANs

Introduced by Radford et al. [57], the deep convolutional GAN (DCGAN) showcased a marked improvement in generating natural images from multiple modalities of real-world data. The DCGAN adopted the following steps to improve the efficacy: strided convolutions as opposed to downsampling through max pooling [58]; batch

normalization [59] to ensure zero mean and unit variance; the use of Leaky ReLU activation function for the discriminator; and the use of inception score to evaluate the efficacy. In particular, DCGAN improved the generation of images from the ImageNet, faces and CIFAR10 datasets [56, 57, 60]. However, for the specific application in medical imaging, there is some more room for improvement.

11.2.3 Variations in optimization functions

As the implementation of GANs are susceptible to issues in the gradients and the quality of generated images, better results may be obtained by varying the performance objective. These functions are used during the training of GANs. Some possibilities are listed here:

1. The $\mathcal{L}_2$ loss function was introduced in Ref. [61] to overcome the problem of vanishing gradients especially for those fake samples sufficiently far from real data but classified correctly. Since the least squares loss function penalizes such samples, the least-squares GAN (LSGAN) attempts to generate samples closer to the real data. The suggested objective functions for the 0–1 coding scheme are as follows:

$$\mathcal{J}_{\mathcal{D}} = \mathbb{E}_x\left[(\mathcal{D}(x)-1)^2\right] + \mathbb{E}_z\left[(\mathcal{D}(\mathcal{G}(z)))^2\right] \tag{11.2}$$

$$\mathcal{J}_{\mathcal{G}} = \mathbb{E}_z\left[\mathcal{D}(\mathcal{G}(z)-1)^2\right] \tag{11.3}$$

2. In order to minimize the pixel-wise distance between the generated and target images, the mean square error (MSE) [42] is often used in the objective function. Let x be the input image, $\hat{x}$ the output image of the generator, and y the target image. Then

$$\mathcal{J}_{\mathcal{D}} = \mathbb{E}[\log(1-\mathcal{D}(\hat{x}))] + \mathbb{E}[\log(\mathcal{D}(y))] \tag{11.4}$$

$$\mathcal{J}_{\mathcal{G}} = \mathbb{E}[\log\mathcal{D}(\hat{x})] + \lambda\,\mathcal{E}_{\text{MSE}}(\hat{x}, y) \tag{11.5}$$

where λ is a regularization parameter and

$$\mathcal{E}_{\text{MSE}}(A,B) = \frac{1}{NM}\sum_{i=1}^{N}\sum_{j=1}^{M}(a_{ij}-b_{ij})^2 \tag{11.6}$$

Here, A and B are two images of dimensions $N \times M$ pixels. Let a_{ij} and b_{ij}, respectively, be the values of the pixels in the (i, j)th position in images A and B.

3. Instead of MSE, some researchers prefer to use the mean absolute error (MAE) instead [62]. This also has an effect on the pixel-wise distance between the target and output images.

$$\mathcal{J}_{\mathcal{D}} = \mathbb{E}[\log(1-\mathcal{D}(\hat{x}))] + \mathbb{E}[\log(\mathcal{D}(y))] \tag{11.7}$$

$$\mathcal{J}_{\mathcal{G}} = \mathbb{E}[\log \mathcal{D}(\hat{x})] + \lambda \mathcal{E}_{\text{MAE}}(\hat{x}, y) \tag{11.8}$$

where λ is a regularization parameter and

$$\mathcal{E}_{\text{MAE}}(A, B) = \frac{1}{NM} \sum_{i=1}^{N} \sum_{j=1}^{M} |a_{ij} - b_{ij}| \tag{11.9}$$

The other quantities are as defined earlier.

11.2.4 Image-quality metrics

We digress briefly to explore image-quality metrics to measure the efficacy of GANs. This is due to an inherent problem in generative modeling, and to a degree unsupervised learning. In the case of an image-to-image translation problem, a reasonable method of measuring network performance is image-quality metrics. The following image-quality metrics are used here to evaluate the performance of GANs in our context.

1. Suppose that A and B are two images of dimensions $N \times M$ pixels. Let a_{ij} and b_{ij}, respectively, be the values of the pixels in the (i, j)th position in images A and B. Then, the pixel-wise MSE metric [63] between the two images A and B is as defined earlier, and repeated here for convenience.

$$\mathcal{E}_{\text{MSE}}(A, B) = \frac{1}{NM} \sum_{i=1}^{N} \sum_{j=1}^{M} (a_{ij} - b_{ij})^2 \tag{11.10}$$

   Evidently, the smaller the value of MSE, the closer are the two images A and B with reference to this measure. However, it may be noted that this metric may not correlate well with subjective analysis of quality. In our context, images A and B, respectively, correspond to the target image and the output of GAN.

2. The peak signal-to-noise ratio (PSNR) is derived from the metric MSE, $\mathcal{E}_{\text{MSE}}(A, B)$. Similar to MSE, PSNR also does not correlate well with human quality assessment. This metric [63] is computed as follows:

$$\mathcal{E}_{\text{PSNR}}(A, B) = 10 \log \frac{255^2}{\mathcal{E}_{\text{MSE}}(A, B)} \tag{11.11}$$

   A and B are closer with respect to PSNR if the corresponding measure is large. Clearly, smaller the value of MSE, the larger the value of PSNR.

3. The structural similarity index (SSI) [63] primarily deals with three aspects of similarity—luminance, contrast, and structure. While luminance is defined as the brightness of the image, the contrast is the difference between the luminance divided by the average luminance of the image. If A and B are two images, this metric is computed as follows:

$$L(A,B) = \frac{2\mu_A\mu_B + c_1}{\mu_A^2 + \mu_B^2 + c_1} \tag{11.12}$$

$$C(A,B) = \frac{2\sigma_A\sigma_B + c_2}{\sigma_A^2 + \sigma_B^2 + c_2} \tag{11.13}$$

$$S(A,B) = \frac{\rho_{AB} + c_3}{\sigma_A\sigma_B + c_3} \tag{11.14}$$

$$\mathcal{E}_{\mathrm{SSI}}(A,B) = L^p(A,B)\, C^q(A,B)\, S^r(A,B) \tag{11.15}$$

In these equations, μ_A and μ_B are, respectively, the average values of the pixels in A and B, σ_A^2 and σ_B^2 the corresponding variances, and ρ_{AB} is the correlation coefficient between A and B. The constants c_1, c_2, and c_3 are introduced to avoid division by zero, or near-division by zero, and the constants p, q, and r represent relative importance of three components. The measure SSI is considered to be related to perception in a human visual system. Its value lie in the interval [0, 1]. Clearly, a value of unity indicates that the two images are the same.

In the context of image compression, a comparison of these metrics is available in Ref. [64] where an analytical link between PSNR and SSI has been shown for common degradations in an image including additive Gaussian noise and Gaussian blur. We note that there are full-reference, partial-reference, and nonreference methods of image comparison. In the context of medical image comparison, full-reference methods are a stronger indicator of network performance. The aforementioned metrics belong to this class of methods.

11.3 The image-to-image translational problem

The task of image-to-image translation is one of the prominent developments in deep learning applied to image processing. This task can be described as one that transforms the feature space of an image to another. Applications include transforming aerial photographs to maps, removal of background from images, and colorization of black and white images. Generative networks have been found to be rather useful for this. The goal of the network is to learn the map between the input and target images to suitably transform the former to the latter. That is, if $\mathcal{U}$ is the set of input images and $\mathcal{T}$ is the set of target images, the trained generator network should be such that $\mathcal{G}: \mathcal{U} \to \mathcal{T}$ and the distribution of images $\mathcal{G}(\mathcal{U})$ is indistinguishable from the distribution of images in $\mathcal{T}$.

CNNs have showcased remarkable results in several fields including medical image classification. An evident drawback of this class of networks is the requirement for large quantities of data to realize the full potential of CNNs. Moreover, some problems in medical imaging require pixel-level classification; for example, medical image

segmentation. The U-net architecture builds upon the fully connected CNN [26]. The expansive and contracting paths are somewhat symmetric leading to an U-shaped architecture. Some of the principal differences relative to typical CNNs are that the pooling layers are replaced with upsampling layers and successive convolutional layers have a large number of filters. In addition, extensive data augmentation is adopted to compensate for smaller datasets.

We note that the U-net is essentially an autoencoder (AE). In general, AEs are networks utilized to learn encodings and they are predominantly used in unsupervised learning [65]. They consist of an encoder and decoder, the former to encode the data distribution to a latent space or bottleneck and the latter to decode this latent space. AEs are typically used for principal component analysis.

Conditional GANs (CGANs) [66] are explored in Ref. [62] to deal with the image-to-image translational problem. Here, the pair of networks learn a conditional generative model [1] making them suitable to tackle such problems. In Ref. [62], the generator is a U-net architecture described earlier which allows it to encode the images into a bottleneck. The discriminator is a convolutional PatchGAN classifier, which allows the network to perform patch-level classification. This encourages the generator to learn patch-level features. In contrast to regular GANs wherein the generative models learn a mapping from the random noise vector z to an output image $y = \mathcal{G}(z)$, CGANs learn the mapping from the input image and the noise vector to the output image: $y = \mathcal{G}(x, z)$. The objective function for CGAN can therefore be described as

$$\mathcal{J}_{\mathcal{G},\mathcal{D}} = \mathbb{E}_{x,y}[\log \mathcal{D}(x, y)] + \mathbb{E}_{x,z}[\log(1 - \mathcal{D}(x, \mathcal{G}(x, z)))] \tag{11.16}$$

The optimal CGAN is then the following:

$$\arg \min_{\mathcal{G}} \max_{\mathcal{D}} \mathcal{J}_{\mathcal{G},\mathcal{D}}$$

This can be regularized as follows:

$$\arg \min_{\mathcal{G}} \max_{\mathcal{D}} \mathcal{J}_{\mathcal{G},\mathcal{D}} + \lambda \mathbb{E}_{x,y,z}[\| y - \mathcal{G}(x, z)\|_1] \tag{11.17}$$

where the $\mathcal{L}_1$ metric is preferred over the $\mathcal{L}_2$ metric to mitigate the effect of blurring [67].

The discriminator that we use is the convolutional PatchGAN classifier, originally proposed in Ref. [68]. The generator with $\mathcal{L}_1$ distance function accurately capture the low frequencies. Accordingly, the GAN discriminator needs to only enforce correctness at the higher frequencies. To accomplish this, it is sufficient to restrict attention to the structure in local image patches, leading to the terminology PatchGAN which only penalizes structure at the scale of patches. Thus, the discriminator classifies based on whether or not each $P \times P$ patch in an image is real or fake. This is run convolutionally across the image and averages the responses to yield the output image. PatchGAN is computationally efficient in that it has fewer parameters, runs faster, and can be applied to

images of arbitrary sizes. The mathematical background is that it models the image as a Markov random field assuming that pixels separated by more than the patch dimensions are statistically independent. This idea has been successfully used for both texture and style.

Thus, the end-to-end pipeline for the image-to-image translation problem uses an autoencoder (U-net) as our generator and the discriminator of PatchGAN. The latter is trained to distinguish between the real images (actual data) from the fake or synthesized images. The goal of the autoencoder is to transform from the domain space to another by utilizing a learned transformation encoded in the latent space. In contrast, the objective of the discriminator is to distinguish between real and fake images.

11.4 Histology and medical imaging

As referenced in previous sections, histology is the examination and study of microscopic structures present on tissues and histopathology is where this examination is used for diagnosis [69]. The overall goal of histopathological analysis is to understand and establish the relationship between the structures present on the tissue with the affliction that the subject of analysis has.

Histopathological analysis is primarily qualitative, where certified pathologists comb through the tissue slide images directly from the microscope or the scanned whole slide images to determine affliction. A fundamental step in histology is staining, where due to cell structures/chemical compounds present on the slide the stains react with them to accentuate the features present on the slide for diagnosis. The a priori information present on the tissue is not immediately visible, requiring a pathologist to use reagents to give a contrast. This allows them to better evaluate the tissue.

Different stains are utilized depending on the region of interest, where different tissue sections react differently. The entire process behind histopathology, from analysis to diagnosis, is necessary but time consuming. Analysis of histology primarily revolves around structures present on the extracted sample. Important precursory steps before analysis are the biopsy of the tissue and fixing of the tissue dyeing/staining; the latter step is relevant in our discussions. The dyes reveal cellular structures and counterstains are used for contrast. The most common stain used is hematoxylin and eosin (H&E), since it is relatively quick to stain and stains a large number of cells well. However, in some cases, it does not provide the contrast required. For that different types of stains are used depending on the afflictions. Histopathology additionally examines the extent of the affliction, called disease grading. It is very useful in distinguishing between different subtypes of diseases, especially in cancers. The scope for automation is evident; companies such as Leica have introduced an automatic stainer to circumvent the process through a physical batch stainer [70, 71].

The scope for automation is quite high for histopathology image analysis as well. With techniques such as whole slide scanning, it has become easier to devise algorithms, which are able to comb through the image for diagnostic purposes. While most of the past medical image analysis has centered on cytology, histopathology serves as the gold standard for diagnostics, especially for cancers. While providing us with a rich feature space, there are challenges in automation of analysis. For this chapter, we will primarily be examining histopathology stains as different feature spaces.

11.4.1 Histology as different feature spaces

Since stains are reagents that interact with cells present on the tissue, they produce different features in that they accentuate structures based on the underlying chemical reactions. H&E clearly dye structures such as cytoplasm, nuclei, organelles, and extracellular components. It aids in the diagnosis of diseases based on the organization of the tissue. H&E chemicals are basic and acidic, respectively, and they operate on the cell nucleus and the cytoplasm cell walls, respectively. Specialized stains have been developed that deal with sections not normally dyed by H&E. For example, Masson's trichrome stains connective tissue where the basic structures are stained blue. Alcian Blue stains heavy proteins such as mucin blue.

With the established fact that each stain provides different information to the diagnosticians, each of them can be considered to be a different feature space. Special stains are sometimes used in conjunction with routine stains to extract more information from the tissue. However, there is a scope for error in fixing and staining that sample. Additionally, these special stains require expensive chemicals. Therefore, there is a use-case for generating the newly stained tissue, through image processing and machine learning-based methods.

In summary, the stains generate unique features, and these unique features are required for diagnosis. Moreover, conventional staining is prone to human error especially during redyeing the tissue, and is time consuming. Further, the process is expensive. Thus, an automated system to generate a reasonable approximation of the tissue has the potential to save resources. Accordingly, only when required, pathologists need to resort to conventional methods.

11.5 Network architecture and dataset

The methodology adopted in this chapter to use GANs for histology staining is described here. The principal issue is the unavailability of a large dataset of unstained tissues. Therefore, our goal is virtual staining of a histology slide, which transforms a slide from one feature space into another. As mentioned earlier, virtual staining has been explored through style transfer in Ref. [43] and GANs in Ref. [42]. The fundamental difference between Ref. [42] and this study is the choice of viewing tissue samples. While Ref. [42]

utilizes autofluorescent images to generate bright-field images, we attempt here to learn a map between two bright-field images. Additionally, we examine the transformation between two types of stains rather than stain a sample from their unstained equivalent image. We again emphasize that unstained images are usually not available.

We use the ANHIR dataset to showcase the efficacy of GANs for histology staining. Specifically, we demonstrate that an image of a tissue stained with one chemical is transformed to an image of the same tissue stained with a different chemical.

11.5.1 ANHIR dataset

The automatic nonrigid histological image registration (ANHIR) dataset is designed for image registration for large-scale images. Registration is required to align multiple tissue sections into one representation in 3D space, where each individual image is stacked upon each other. This allows a pathologist to extract more information from the tissue samples from multiple features and biomarkers. The characteristics of the dataset are depicted in Table 11.1. This dataset has a variety of tissues from different organs and the tissues are stained with a diverse set of histological stains. We note that the ANHIR dataset provides more stains than indicated in Table 11.1. (For example, images of an H&E-stained and IHC-stained lung lesion tissue are both available. In addition, the dataset provides images with CD10, CD31, and Ki67 stains, which are not considered here.) Moreover, the dataset also provides different levels of magnification for each tissue ranging from 10× to 40×. We opt for higher magnification in order to obtain a larger image dataset to achieve our objective.

A brief explanation of the stains and their effects are as follows:

- CD31 is a type of immunohistochemical (IHC) stain, which mediates cell-to-cell adhesion. These are typically used in tonsils, skin, liver, and kidneys.
- Estrogen receptor (ER) antibody stains are proteins activated by the estrogen hormone. It is extensively used in breast carcinoma.

Table 11.1 The ANHIR dataset.

Tissue	Stains	Magnitude	Average size
Lung lesion	H&E, IHC	40×	18k×15k
Lung lobe	H&E, CD31	10×	11k×6k
Mammary gland	H&E, ER, PR	40×	12k×4k
Mice-kidney	PAS, SMA, CD31	20×	37k×30k
Colon adenocarcinoma	H&E, IHC, CD	10×	60k×50k
Gastric adenocarcinoma	H&E, IHC	40×	60k×75k
Breast tissue	H&E, IHC, PR	40×	65k×60k
Human kidney	H&E, PAS, MAS	40×	18k×55k

Table 11.2 Subset of ANHIR dataset.

Tissue	Stains	Scaling	Number of images
Lung lesion	H&E, CD31	50×	1250
Kidney	H&E, MAS	25×	368
Lung lobe	H&E, CD31	25×	151

- Progestrone receptor (PR) antibody stains are used for breast cancer detection and are typically used in conjunction with ERs.
- Periodic acid-Schiff (PAS) stains are used for detecting the presence of carbohydrates in tissues such as connective tissues, mucus, etc. These are used in diagnostics in diseases such as glycogen storage disease.
- Masson's trichome (MAS) stain is a three-color staining procedure. It produces red for keratin and muscle fibers, blue and green for collagen, light red or pink for cytoplasms, and dark brown to black for cell nuclei. This stain is typically used in cardiac, kidney, muscular, and hepatic pathologies.
- Smooth muscle actin (SMA) is a special type of stain, which is typically used in specialized cancer diagnostics.

Further details about these stains can be found in Refs. [72, 73]. As indicated in the table, the available magnifications and the average size of the images vary. For the purpose of this chapter, we choose a subset of the ANHIR dataset as indicated in Table 11.2. The number of available images are also indicated.

11.5.2 Dataset preparation

The tissues and stains considered here are listed in Table 11.2. The whole slide scan of the histological image is first divided into smaller sections of size 256×256. The pixel values in each image are scaled to lie within the range $[-1, 1]$. This increases the numerical stability during training. (Moreover, it has been the experience that using a floating point arithmetic is better.)

11.5.3 Network architectures

The architectures for the generator and discriminator have been arrived at after several experiments. The networks are based on U-net for the generator network and the PatchGAN architecture for the discriminator [62]. This architecture maps closely with CGANs. The generator architecture of the network includes convolutional layers with skip connections. The number of filters is 64, 128, 256, 512, and 1024. The activation functions for all but the final layer is the ReLU [74], and for the final layer the sigmoid activation function $\tanh v$. This network for the discriminator features batch normalization and leaky ReLU activation function. The other details of the architectures are provided in "Appendix" section.

We consider here the performance objectives mean absolute error (MAE) and the root mean square error (RMSE). The networks were trained using the Adam optimizer [75], at a learning rate of 10^{-5} and 10^{-4} for the discriminator and the generator networks, respectively. Batch sizes of 5, 10, and 15 were used at different trials. Shuffling was utilized, and finally the images were trained for 201 epochs for the number of input images as indicated in Table 11.2.

The approach to evaluate the efficacy of the proposed GAN should include both qualitative (visual inspection) and quantitative image metrics. Both are required as the latter alone may not suffice in that good performance with respect to quantitative metrics may not indicate clinical viability. We emphasize here that the output images must eventually be validated by a diagnostician.

11.6 Results and discussions

As indicated earlier, the subset of ANHIR dataset considered here is listed in Table 11.2. Sample input and target images for the three tissues are shown in Fig. 11.1A–F. Here, an image of a lung lesion tissue stained with H&E is shown in Fig. 11.1A. This is the input image to the proposed GAN. The target output image is shown in Fig. 11.1B, which is the image of the same tissue but stained with CD31. Likewise, images of a kidney tissue stained with H&E and MAS stains are, respectively, shown in Fig. 11.1C and D, and images of a lung lobe tissue stained with H&E and CD31 stains are, respectively, shown in Fig. 11.1E and F.

As mentioned earlier, the images belonging to a particular tissue are input to the proposed GAN architecture described in the previous section and "Appendix" section. The performance of GAN depends on the choice of objective functions. In this chapter, we consider two objective functions. In what follows, a GAN trained with the MSE objective function described in Eqs. (11.4), (11.5) is referred to as GAN_{MSE}, and a GAN trained with the MAE objective function described in Eqs. (11.7), (11.8) is denoted GAN_{MAE}. The output of the generative network should resemble that of the corresponding target images. The closeness of the images is quantified using the metrics MSE, PSNR, and the SSI. (We note that due to a lack of resources, validation by a pathologist was not possible.)

Sample sets of input, target, and output images in the case of lung lesion tissue are shown in Fig. 11.2 showing the results with the proposed GAN_{MSE} and GAN_{MAE}. We note an Adam optimizer has been used during the training process. (The chosen parameters are $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 10^{-7}$.) From Fig. 11.2, it is clear that there is a close approximation between the target and output images, respectively, shown in Fig. 11.2B and C with mildly blurry results. In contrast, it is evident from Fig. 11.2E and F that GAN_{MAE} generates deep blue images, which is less than ideal for our

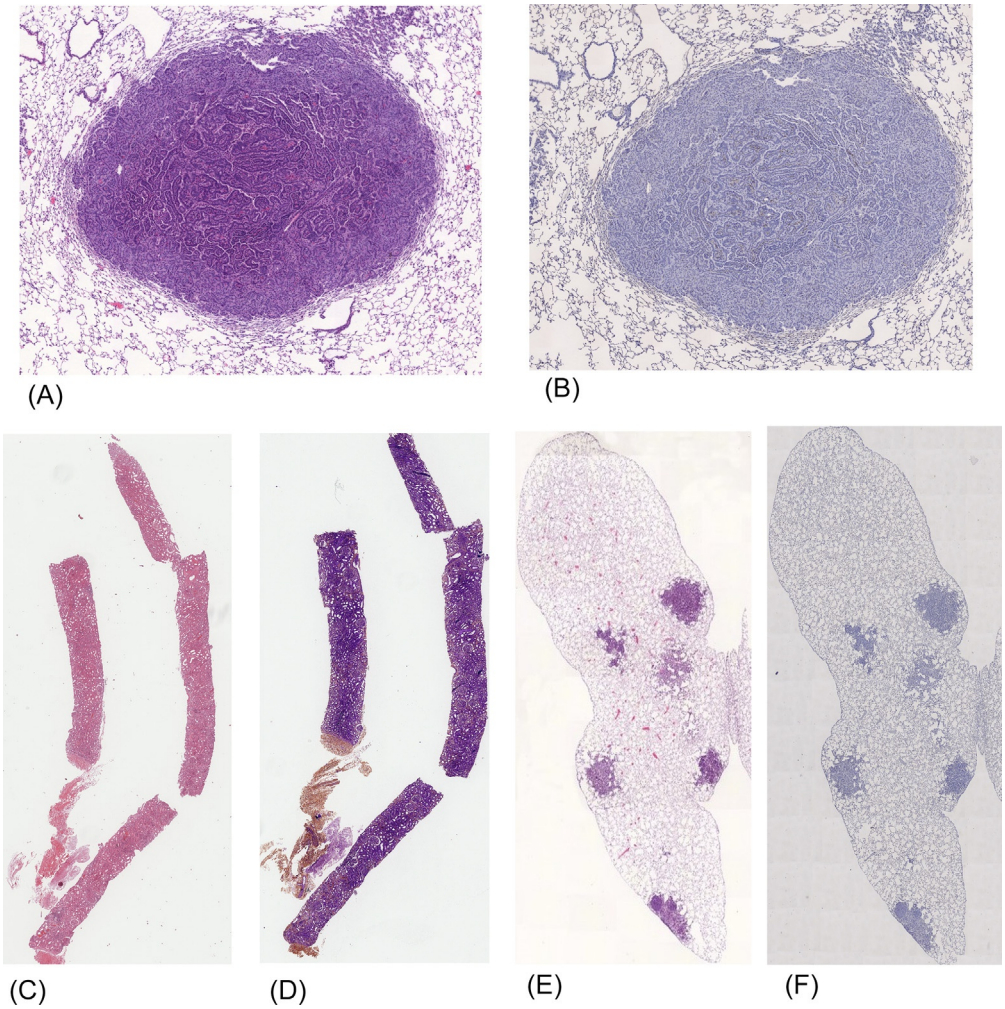

Fig. 11.1 Lung lesion tissue: (A) Input image with H&E stain. (B) Target image with CD31 stain. Kidney tissue: (C) Input image with H&E stain. (D) Target image with MAS stain. Lung lobe tissue: (E) Input image with H&E stain. (F) Target image with CD31 stain.

application. Thus, a closer comparison between target and output images clearly indicates that the MSE loss function provides better results.

These observations are further validated using the image-quality metrics. Indeed, with GAN_{MSE} the respective averaged values of SSI, PSNR, and MSE are 0.8455, 27.365, and 0.0650, and the corresponding values for GAN_{MAE} are 0.5938, 9.0791, and 0.1678. (These values are also depicted in Table 11.3.) All these measures indicate

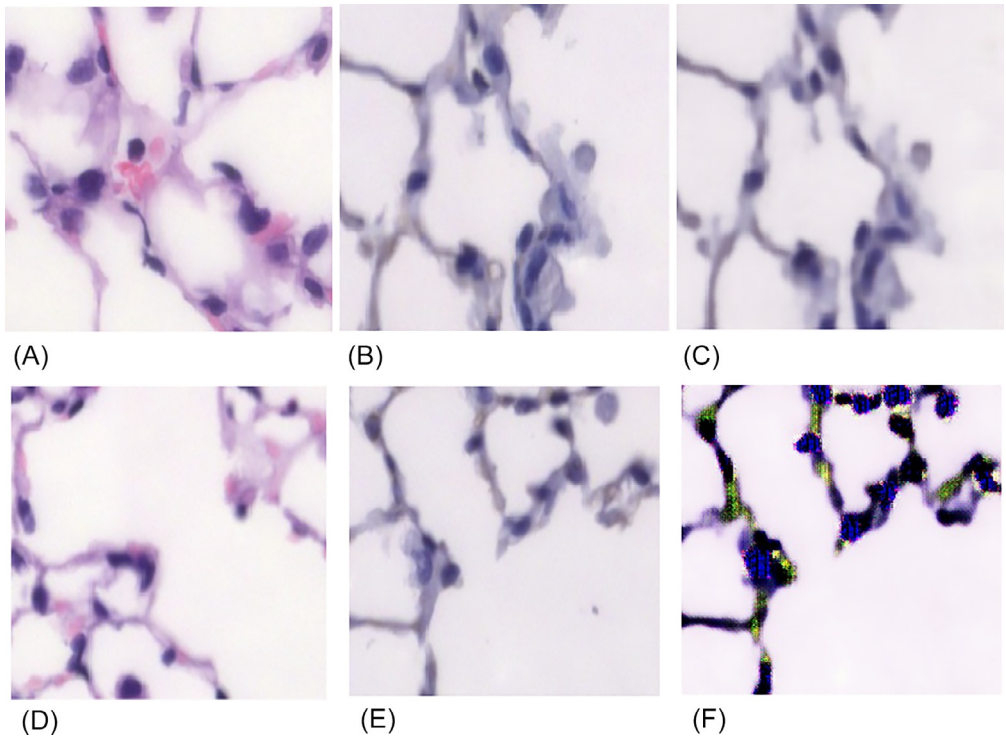

Fig. 11.2 Lung lesion tissue. With GAN_{MSE}: (A) Input image; (B) target image; (C) output image. With GAN_{MAE}: (D) Input image; (E) target image; (F) output image.

Table 11.3 Averaged image metric values.

Tissue	Type	SSI	PSNR	MSE
Lung lesion	GAN_{MSE}	0.8455	27.365	0.0650
	GAN_{MAE}	0.5938	9.0791	0.1678
Kidney	GAN_{MSE}	0.3757	12.6759	0.0052
	GAN_{MAE}	0.4038	12.0887	0.0739
Lung lobe	GAN_{MSE}	0.7213	21.0530	0.0046
	GAN_{MAE}	0.6931	21.1157	0.0140

that the output of the generator and the target image are reasonably close. However, GAN_{MSE} is performing better on the test images. Evidently, if an MSE loss function is used during training, the MSE image-quality metric yields a smaller value.

In addition to these observations, it is evident from Fig. 11.2B and C that some of the finer details have not been well captured by the network. Equivalently, the generative

network has not completely learned the transformation from one feature space to another. The primary reason for this is the smaller size of the dataset. It may be recalled that only 1250 images were available.

From the observations with reference to the lung lesion tissue, it is evident that the higher-level features of the images such as the overall structure are captured completely. However, the generated images can be blurry and there is a marked variations between the images with GAN_{MSE} and GAN_{MAE}. Therefore, our approach to histology staining posed an image-to-image translation problem is promising. Moreover, the quantitative measures SSIM and PSNR may not be the ideal metrics in our context. Despite the scores, the images are slightly blurry as observed in Fig. 11.2C. Further, there are artefacts introduced if the number of epochs is lesser than 100; these eventually disappear with additional training. As will be evident, these remarks hold for both kidney and lung lobe tissues. Some issues with these tissue images are exacerbated due to the smaller size of the datasets.

The results with kidney tissues are shown in Fig. 11.3 for both GAN_{MSE} and GAN_{MAE}. Similarly, the results with lung lobe tissues are shown in Fig. 11.4 for both

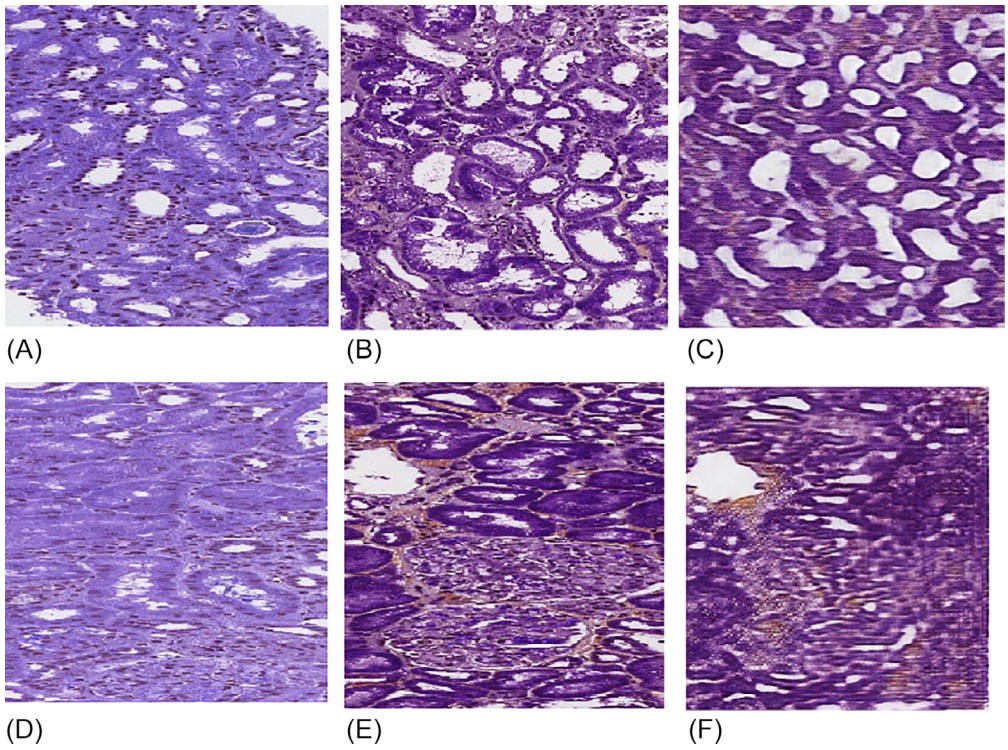

Fig. 11.3 Kidney tissue. With GAN_{MSE}: (A) Input image; (B) target image; (C) output image. With GAN_{MAE}: (D) Input image; (E) target image; (F) output image.

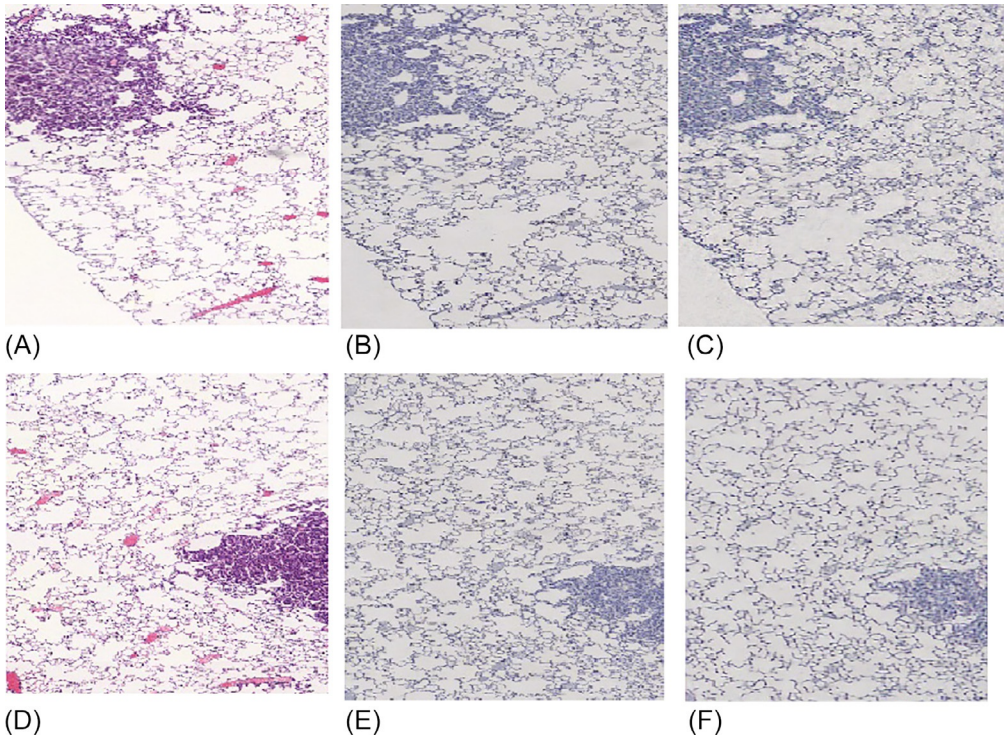

Fig. 11.4 Lung lobe tissue. With GAN_{MSE}: (A) Input image; (B) target image; (C) output image. With GAN_{MAE}: (D) Input image; (E) target image; (F) output image.

variants of GANs considered here. From Table 11.3, the differences between the two GANs are less pronounced for both kidney and lung lobe tissues, with the performances of GAN_{MSE} better than the other. (There is a negligible discrepancy in these observations with respect to PSNR for the lung lobe tissue.) Nonetheless, minor visual differences can be observed in the kidney tissue even with GAN_{MSE}. This is due to the relatively higher complexity of the kidney tissue when compared with the lung lesion or lung lobe tissues. An additional issue is the relatively smaller number of images in the dataset. At present we have only 368 images corresponding to the kidney tissue, which is rather small considering the complexity.

In contrast, even with a much lower dataset size of 151, the correlation between the output and target images for a lung lobe tissue is reasonably good for both GAN_{MSE} and GAN_{MAE} with the former outperforming the latter in both quantitative measures as depicted in Table 11.3, as well as visually. This is clearly due to the fact that these tissues are much less complex. An interesting point to highlight is that despite the dataset being smaller in size for the lung lobe tissue, the transformation is significantly better. This is

due to the fact that these images are far less complex when compared to the images corresponding to the kidney tissue.

Evidently, the network performance is highly varying with respect to the complexity of the tissue. The network showcases reasonably good performances in tissues such as lung lesions and lung lobes. However, the performance decreases with complex tissue samples such as kidneys. Accordingly, there is a requirement for a larger distribution of tissue for the generated samples to be modeled with a higher degree of accuracy.

An important point to note is the imperfections present due to the image capture of the histology and due to human error while fixing the histology. The imperfections present on the target image in Fig. 11.5A where the tissue is ripped and stringy is reflected in the output image Fig. 11.5B generated with GAN_{MAE}. This emphasizes the fact that the images used for training the GAN are to be chosen carefully, and requires domain knowledge from a pathologist.

To summarize, these results showcase the image-to-image transformation from an input image corresponding to one stain to another image corresponding to a different stain. When the number of samples in the dataset is large, the quality of the images is quite satisfactory. In some cases, however, the results are slightly blurry, which may be addressed if the dataset is larger. Variations in the objective function clearly have an effect on the output image. The images that resulted with an MSE objective function are better than those obtained with an MAE objective function. Due to the complexity of the image, the output images corresponding to the kidney tissue are rather diffuse as the intricate features have not been captured completely correctly. The finer details corresponding to the lung lobes have been well captured. As far as the metrics are concerned, there is a strong indication that both SSI and PSNR do not indicate visual quality of the images. Moreover, the imperfections present on the target tissue appear to be transferred as well to the output. This can prove problematic as the imperfections may be due to

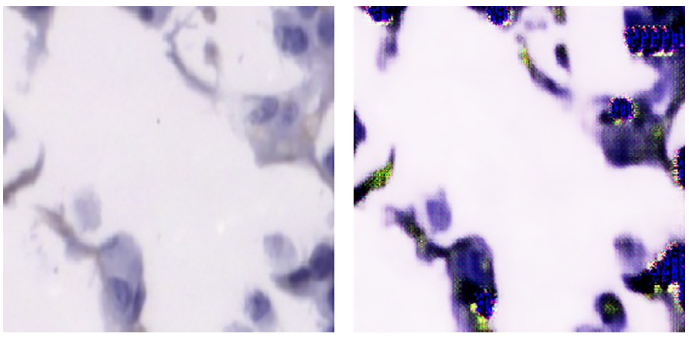

Fig. 11.5 Errors lung lobe tissue: (A) Target with imperfections; (B) output image with distortions.

human errors. Further, artefacts are sometimes introduced at lower epochs. However, with sufficient training, these artefacts are eventually removed.

11.7 Conclusions

GANs in the past have been used for image translation tasks. In our medical image translation task, we utilize the conditional GAN algorithm to generate differently stained images from paired input images. The results have been demonstrated to be satisfactory. If the size of the dataset is suitably large, the results are better as it captures both high-level and low-level features. Unfortunately, the imperfections in the images are captured as well by the network. Therefore, the training dataset ought to be carefully chosen. Additionally, histology staining is dependent on the presence of a particular compound/tissue on the slide. Accordingly, the network needs to have a large distribution of histology stains in different circumstances/stages. This is important as it needs to learn more settings for it to be able to learn the transformation sufficiently well to be viable clinically.

While the PSNR and the SSIM are popular image-quality metrics, a potential quality metric that could be utilized for future evaluation is the relative target registration error. Here, the landmarks can give the networks a better indication of whether the networks are performing better or worse. However, such a study requires hand labeling by experts.

We have showcased the use of CGANs for the image-to-image translation between two stained tissues. Therefore, it has the potential for transforming images of unstained tissues to stained ones given enough data with different settings. However, the issues highlighted earlier are to be addressed before a more complete end-to-end system is available that can transform an unstained tissue to a stained one. Finally, the networks should also learn a general mapping of how to transform stained tissues. Since the images are paired from one image to another, the network might learn a map where it transforms the input image to the target image, rather than altering the stain. This issue is a known problem when using a one-to-one mapping between inputs and targets. Thus future work in this space should explore unpaired translation of such images to make an effective end-to-end staining algorithm.

Appendix: Network architectures

The generator and discriminator networks are described here. The generator network consists of 38 layers with several sets of convolutional and max pool layers, and listed in Tables 11.4 and 11.5. Each layer is characterized by a triplet (n_1, n_2, n_3) which indicates an image of dimensions $n_1 \times n_2$ with n_3 the number of filters. There are four skip connections. Specifically, the outputs of Layers 3, 6, 9, and 13 are as well inputs to Layers 35, 30, 25, and 20, respectively. The 12 layers of the discriminator network are listed in Table 11.6.

Table 11.4 Generator architecture: Part A.

No.	Layer	Input	Output	Remark
1	Input layer	(256, 256, 3)	(256, 256, 3)	
2	Conv2D	(256, 256, 3)	(256, 256, 64)	
3	Conv2D	(256, 256, 64)	(256, 256, 64)	To Layer 35
4	MaxPooling2D	(256, 256, 64)	(128, 128, 64)	
5	Conv2D	(128, 128, 64)	(128, 128, 128)	
6	Conv2D	(128, 128, 128)	(128, 128, 128)	To Layer 30
7	MaxPooling2D	(128, 128, 128)	(64, 64, 128)	
8	Conv2D	(64, 64, 128)	(64, 64, 256)	
9	Conv2D	(64, 64, 256)	(64, 64, 256)	To Layer 25
10	MaxPooling2D	(64, 64, 256)	(32, 32, 256)	
11	Conv2D	(32, 32, 256)	(32, 32, 512)	
12	Conv2D	(32, 32, 512)	(32, 32, 512)	
13	Dropout	(32, 32, 512)	(32, 32, 512)	To Layer 20
14	MaxPooling2D	(32, 32, 512)	(16, 16, 512)	
15	Conv2D	(16, 16, 512)	(16, 16, 1024)	
16	Conv2D	(16, 16, 1024)	(16, 16, 1024)	
17	Dropout	(16, 16, 1024)	(16, 16, 1024)	
18	Upsampling2D	(16, 16, 1024)	(32, 32, 1024)	
19	Conv2D	(32, 32, 1024)	(32, 32, 512)	

Table 11.5 Generator architecture: Part B.

No.	Layer	Input	Output	Remark
20	Concatenate	(32, 32, 512) (32, 32, 512)	(32, 32, 1024)	From Layer 13
21	Conv2D	(32, 32, 1024)	(32, 32, 512)	
22	Conv2D	(32, 32, 512)	(32, 32, 512)	
23	Upsampling2D	(32, 32, 512)	(64, 64, 512)	
24	Conv2D	(64, 64, 512)	(64, 64, 256)	
25	Concatenate	(64, 64, 256) (64, 64, 256)	(64, 64, 512)	From Layer 9
26	Conv2D	(64, 64, 512)	(64, 64, 256)	
27	Conv2D	(64, 64, 256)	(64, 64, 256)	
28	Upsampling2D	(64, 64, 256)	(128, 128, 256)	
29	Conv2D	(128, 128, 256)	(128, 128, 128)	
30	Concatenate	(128, 128, 128) (128, 128, 128)	(128, 128, 256)	From Layer 6
31	Conv2D	(128, 128, 256)	(128, 128, 128)	
32	Conv2D	(128, 128, 128)	(128, 128, 128)	
33	Upsampling2D	(128, 128, 128)	(256, 256, 128)	
34	Conv2D	(256, 256, 128)	(256, 256, 64)	
35	Concatenate	(256, 256, 64) (256, 256, 64)	(256, 256, 128)	From Layer 3
36	Conv2D	(256, 256, 128)	(256, 256, 64)	
37	Conv2D	(256, 256, 64)	(256, 256, 64)	
38	Conv2D	(256, 256, 64)	(256, 256, 3)	

Table 11.6 Discriminator architecture.

No.	Layer	Input	Output
1	Input layer	(256, 256, 3)	(256, 256, 3)
2	Conv2D	(256, 256, 3)	(128, 128, 64)
3	Leaky ReLU	(128, 128, 64)	(128, 128, 64)
4	Dropout	(128, 128, 64)	(128, 128, 64)
5	Conv2D	(128, 128, 64)	(64, 64, 128)
6	Leaky ReLU	(64, 64, 128)	(64, 64, 128)
7	Dropout	(64, 64, 128)	(64, 64, 128)
8	Conv2D	(64, 64, 128)	(32, 32, 256)
9	Leaky ReLU	(32, 32, 256)	(32, 32, 256)
10	Dropout	(32, 32, 256)	(32, 32, 256)
11	Flatten	(32, 32, 256)	(262144)
12	Dense	(262144)	(1)

References

[1] I.J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial networks, in: Proceedings of the 28th International Conference on Neural Information Processing Systems (NIPS'14), Montréal, Quebec, Canada, 2014, pp. 2672–2680.

[2] A.Y. Ng, M.I. Jordan, On discriminative vs. generative classifiers: a comparison of logistic regression and naive Bayes, in: Proceedings of the 27th International Conference on Neural Information Processing Systems (NIPS'01), Vancouver, British Columbia, Canada, 2001, pp. 841–848.

[3] D. Foster, Generative Deep Learning: Teaching Machines to Paint, Write, Compose, and Play, O'Reilly Media, Sebastopol, CA, 2019.

[4] K. Ganguly, Learning Generative Adversarial Networks: Next-Generation Deep Learning Simplified, Packt Publishing, Birmingham, UK, 2017.

[5] J. Langr, V. Bok, GANs in Action: Deep Learning With Generative Adversarial Networks, Manning Publications, Shelter Island, NY, 2019.

[6] J.F. Nash, Jr., Equilibrium points in n-person game, Proc. Natl. Acad. Sci. 36 (1) (1950) 48–49.

[7] G. Hinton, L. Deng, D. Yu, G.E. Dahl, A.-R. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T.N. Sainath, B. Kingsbury, Deep neural networks for acoustic modeling in speech recognition: the shared views of four research groups, IEEE Signal Process. Mag. 29 (6) (2012) 82–97.

[8] A. Krizhevsky, I. Sutskever, G.E. Hinton, ImageNet classification with deep convolutional neural networks, Commun. ACM 60 (6) (2017) 84–90.

[9] R. Yamashita, M. Nishio, R.K.G. Do, K. Togashi, Convolutional neural networks: an overview and application to radiology, Insights Imaging 9 (2018) 611–629.

[10] K. Jarrett, K. Kavukcuoglu, M. Ranzato, Y. LeCun, What is the best multi-stage architecture for object recognition? in: Proceedings of the 12th International Conference on Computer Vision (ICCV'09), Kyoto, Japan, 2009, pp. 2146–2153.

[11] X. Glorot, A. Bordes, Y. Bengio, Deep sparse rectifier neural networks, in: Proceedings of the 14th International Conference on Artificial Intelligence and Statistics (AISTATS'11), Ft. Lauderdale, FL, USA, 2011, pp. 315–323.

[12] F. Agostinelli, M. Hoffman, P. Sadowski, P. Baldi, Learning activation functions to improve deep neural networks, in: Proceedings of the 3rd International Conference on Learning Representations Workshop (ICLR), San Diego, CA, USA, 2015.

[13] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting, J. Mach. Learn. Res. 15 (1) (2014) 1929–1958.