

```
In [42]: import os
import keras
import numpy as np
from tqdm import tqdm
from glob import glob
import tensorflow as tf
import tensorflow.image as tfi
from keras.preprocessing.image import load_img, img_to_array
import matplotlib.pyplot as plt
```

Load Data

data --- <https://www.kaggle.com/code/utkarshsaxenadn/pix2pixgan-map-generator/notebook>

```
In [43]: def load_data(path, trim=None):
    paths = sorted(glob(path + "*.jpg"))

    if trim is not None:
        paths = sorted(glob(path + "*.jpg"))[:trim]

    images = np.zeros(shape=(len(paths), 256, 256, 3))
    masks = np.zeros(shape=(len(paths), 256, 256, 3))

    for i, path in tqdm(enumerate(paths), desc="Data"):
        image = tf.cast(img_to_array(load_img(path)), tf.float32)
        img = image[:, :600, :]
        mask = image[:, 600:, :]
        images[i] = tfi.resize(img, (256, 256))/255.
        masks[i] = tfi.resize(mask, (256, 256))/255.

    return images, masks

def show_image(image, title=None, alpha=1.0):
    plt.imshow(image, alpha=alpha)
    if title is not None:
        plt.title(title)
    plt.axis("off")

def show_mask(image, mask, alpha=0.4, title=None):
    image = tf.squeeze(tf.cast(image, tf.float32))
    mask = tf.squeeze(tf.cast(mask, tf.float32))
    show_image(image)
    show_image(mask, alpha=alpha, title=title)
```

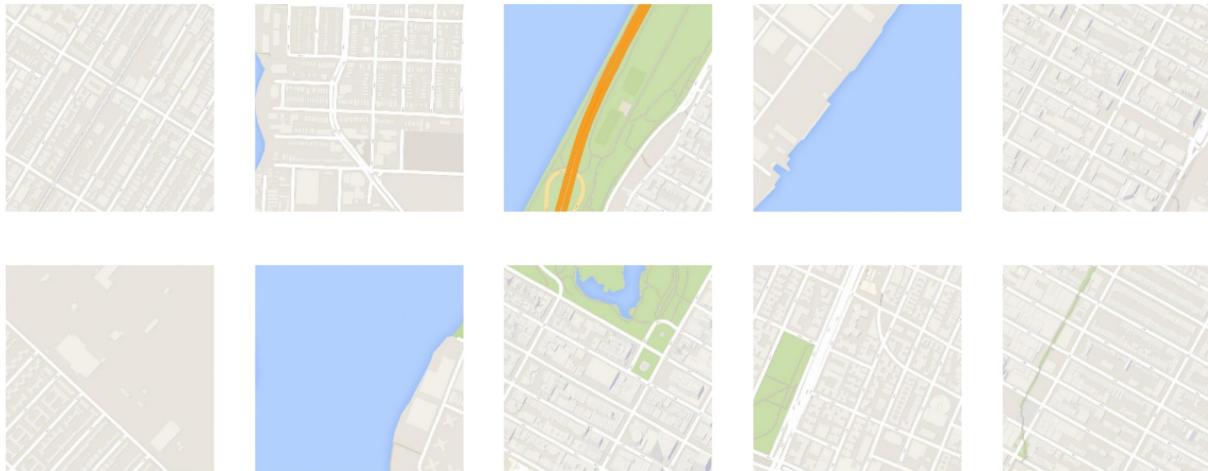
```
In [44]: # trim dataset to avoid ram issues
# vscode uses 22 gigs 🤦‍♂️🤦‍♂️🤦‍♂️
dataset = load_data('Data\\maps\\train\\', trim=4000)
images, masks = dataset
```

Data: 1096it [00:16, 64.84it/s]

```
In [ ]: data = tf.data.Dataset.from_tensor_slices(dataset).shuffle(2000).batch(32).prefetch
```

```
In [46]: for images, masks in data.take(1):
    sample_images = images
    sample_masks = masks
    break

    plt.figure(figsize=(20,8))
    for i in range(10):
        plt.subplot(2, 5, i + 1)
        show_image(sample_masks[i])
```



```
In [47]: plt.figure(figsize=(20,8))
for i in range(10):
    plt.subplot(2, 5, i + 1)
    show_image(sample_images[i])
```



Upsample and Downsample functions

```
In [48]: from keras.layers import ReLU
from keras.layers import Input
from keras.layers import Conv2D
from keras.layers import Dropout
```

```
from keras.layers import ZeroPadding2D
from keras.layers import LeakyReLU
from keras.layers import concatenate
from keras.layers import Conv2DTranspose

from keras.initializers import RandomNormal
from keras.layers import BatchNormalization
from tensorflow.keras.optimizers import Adam
from keras.models import Sequential, Model, load_model

# Model Viz
from tensorflow.keras.utils import plot_model

# Model Losses
from keras.losses import BinaryCrossentropy

def downsample(filters, size, batch_norm=True):

    layer = Sequential()
    layer.add(Conv2D(filters, kernel_size=size, strides=2, padding='same', kernel_initializer=RandomNormal()))
    if batch_norm:
        layer.add(BatchNormalization())
    layer.add(LeakyReLU())

    return layer

def upsample(filters, size, dropout=False):
    layer = Sequential()
    layer.add(Conv2DTranspose(filters, kernel_size=size, strides=2, padding='same', kernel_initializer=RandomNormal()))
    layer.add(BatchNormalization())
    if dropout:
        layer.add(Dropout(0.5))
    layer.add(ReLU())
    return layer
```

Create the Generator

The generator is a series of upsampling and downsampling ...

```
In [49]: # def Generator1():

    # inputs = Input(shape=(256, 256, 3))

    # downsample_stack = [
    #     downsample(64, 4, batch_norm=False), # why is batch false here
    #     downsample(128, 4),
    #     downsample(256, 4),
    #     downsample(512, 4),
    #     downsample(512, 4),
    #     downsample(512, 4),
    #     downsample(512, 4),
```

```
#     downsample(512, 4)
#     ]

#     upsample_stack=[
#         upsample(512, 4,dropout=True),
#         upsample(512, 4,dropout=True),
#         upsample(512, 4,dropout=True), # why is dropout true here
#         upsample(512, 4),
#         upsample(256, 4),
#         upsample(128, 4),
#         upsample(64, 4),
#     ]

#     last = Conv2DTranspose(3, 4, strides=2, padding='same', activation='tanh', kernel_size=4, use_bias=False)

#     x = inputs
#     skips = []
#     for down in downsample_stack:
#         x = down(x)
#         skips.append(x)

#     skips = reversed(skips[:-1])
#     for up, skip, in zip(upsample_stack, skips):
#         x = up(x)
#         x = concatenate([x, skip])

#     x = last(x)
#     return Model(inputs=inputs, outputs=x)
```

```
In [50]: # generator1 = Generator1()
# generator1.summary()
```

```
In [51]: class Generator:
    def __init__(self):
        inputs = Input(shape=(256, 256, 3))
        downsample_stack = [
            downsample(64, 4, batch_norm=False), # why is batch false here
            downsample(128, 4),
            downsample(256, 4),
            downsample(512, 4),
            downsample(512, 4),
            downsample(512, 4),
            downsample(512, 4),
            downsample(512, 4)]

        upsample_stack=[
            upsample(512, 4,dropout=True),
            upsample(512, 4,dropout=True),
            upsample(512, 4,dropout=True), # why is dropout true here
            upsample(512, 4),
            upsample(256, 4),
            upsample(128, 4),
            upsample(64, 4),
        ]
```

```
last = Conv2DTranspose(3, 4, strides=2, padding='same', activation='tanh', kernel_size=3, name='last')

x = inputs
skips = []
for down in downsample_stack:
    x = down(x)
    skips.append(x)

skips = reversed(skips[:-1])
for up, skip, in zip(upsample_stack, skips):
    x = up(x)
    x = concatenate([x, skip])

x = last(x)
self.model = Model(inputs=inputs, outputs=x)
```

```
In [52]: generator = Generator()
generator.model.summary()
```

```
Model: "functional_35"
```

Layer (type)	Output Shape	Param #	Connected to
input_layer_19 (InputLayer)	(None, 256, 256, 3)	0	-
sequential_18 (Sequential)	(None, 128, 128, 64)	3,072	input_layer_19[0]...
sequential_19 (Sequential)	(None, 64, 64, 128)	131,584	sequential_18[0]...
sequential_20 (Sequential)	(None, 32, 32, 256)	525,312	sequential_19[0]...
sequential_21 (Sequential)	(None, 16, 16, 512)	2,099,200	sequential_20[0]...
sequential_22 (Sequential)	(None, 8, 8, 512)	4,196,352	sequential_21[0]...
sequential_23 (Sequential)	(None, 4, 4, 512)	4,196,352	sequential_22[0]...
sequential_24 (Sequential)	(None, 2, 2, 512)	4,196,352	sequential_23[0]...
sequential_25 (Sequential)	(None, 1, 1, 512)	4,196,352	sequential_24[0]...
sequential_26 (Sequential)	(None, 2, 2, 512)	4,196,352	sequential_25[0]...
concatenate_8 (Concatenate)	(None, 2, 2, 1024)	0	sequential_26[0]... sequential_24[0]...
sequential_27 (Sequential)	(None, 4, 4, 512)	8,390,656	concatenate_8[0]...
concatenate_9 (Concatenate)	(None, 4, 4, 1024)	0	sequential_27[0]... sequential_23[0]...
sequential_28 (Sequential)	(None, 8, 8, 512)	8,390,656	concatenate_9[0]...
concatenate_10 (Concatenate)	(None, 8, 8, 1024)	0	sequential_28[0]... sequential_22[0]...
sequential_29 (Sequential)	(None, 16, 16, 512)	8,390,656	concatenate_10[0]...
concatenate_11 (Concatenate)	(None, 16, 16, 1024)	0	sequential_29[0]... sequential_21[0]...
sequential_30 (Sequential)	(None, 32, 32, 256)	4,195,328	concatenate_11[0]...
concatenate_12 (Concatenate)	(None, 32, 32, 512)	0	sequential_30[0]... sequential_20[0]...

sequential_31 (Sequential)	(None, 64, 64, 128)	1,049,088	concatenate_12[0...]
concatenate_13 (Concatenate)	(None, 64, 64, 256)	0	sequential_31[0]... sequential_19[0]...
sequential_32 (Sequential)	(None, 128, 128, 64)	262,400	concatenate_13[0...]
concatenate_14 (Concatenate)	(None, 128, 128, 128)	0	sequential_32[0]... sequential_18[0]...
conv2d_transpose_15 (Conv2DTranspose)	(None, 256, 256, 3)	6,147	concatenate_14[0...]

Total params: 54,425,859 (207.62 MB)

Trainable params: 54,414,979 (207.58 MB)

Non-trainable params: 10,880 (42.50 KB)

Random testin

```
In [53]: test_input = np.ones((256, 256, 3), dtype=np.float32)

# Define the circle's parameters
center = (128, 128)
radius = 80
color = (0, 0, 0)
for y in range(256):
    for x in range(256):
        if abs(np.sqrt((x - center[0]) ** 2 + (y - center[1]) ** 2) - radius) < 1.5:
            test_input[y, x] = color
test_input = tf.expand_dims(test_input, axis=0)

generated_image = generator.model(test_input, training=False)

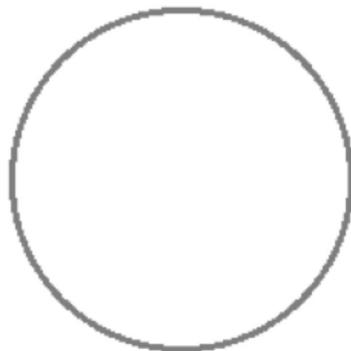
# # Remove the batch dimension and normalize values to [0, 1]
test_input = (test_input[0] + 1) / 2 # Normalizing from [-1, 1] to [0, 1]
generated_image = (generated_image[0] + 1) / 2 # Normalizing from [-1, 1] to [0, 1]

plt.figure(figsize=(10, 5))
plt.subplot(1, 2, 1)
plt.title("Input Image")
plt.imshow(test_input.numpy())
plt.axis("off")

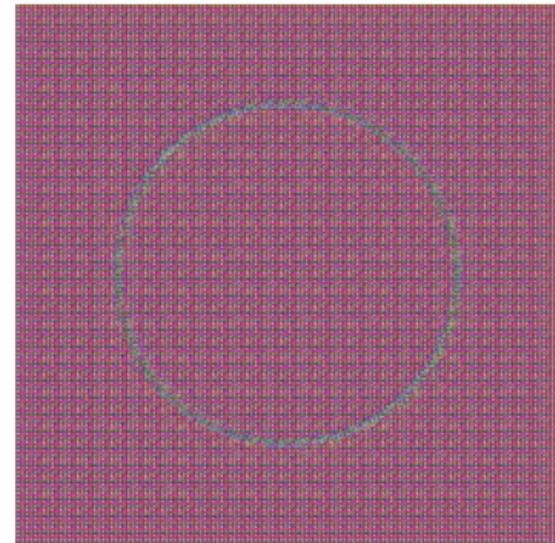
plt.subplot(1, 2, 2)
plt.title("Generated Image")
plt.imshow(generated_image.numpy())
plt.axis("off")

plt.show()
```

Input Image



Generated Image



Create Discriminator

```
In [54]: class Discriminator:  
    def __init__(self):  
        init = tf.random_normal_initializer(0.,0.02)  
        input = Input(shape=(256,256,3), name="input_image")  
        target = Input(shape=(256,256,3), name="target_image")  
  
        x = concatenate([input, target])  
        x = downsample(64, 4, batch_norm=False)(x)  
        x = downsample(128, 4)(x)  
        x = downsample(256, 4)(x)  
  
        x = ZeroPadding2D()(x)  
        x = Conv2D(512, 4, strides=1, kernel_initializer=init, use_bias=False)(x)  
        x = BatchNormalization()(x)  
        x = LeakyReLU()(x)  
        x = ZeroPadding2D()(x)  
  
        x = Conv2D(1, 4, kernel_initializer=init)(x)  
        self.model = Model(inputs=[input, target], outputs=[x])
```

```
In [55]: discriminator = Discriminator()  
discriminator.model.summary()
```

Model: "functional_39"

Layer (type)	Output Shape	Param #	Connected to
input_image (InputLayer)	(None, 256, 256, 3)	0	-
target)image (InputLayer)	(None, 256, 256, 3)	0	-
concatenate_15 (Concatenate)	(None, 256, 256, 6)	0	input_image[0][0]... target)image[0][...]
sequential_33 (Sequential)	(None, 128, 128, 64)	6,144	concatenate_15[0]...
sequential_34 (Sequential)	(None, 64, 64, 128)	131,584	sequential_33[0]...
sequential_35 (Sequential)	(None, 32, 32, 256)	525,312	sequential_34[0]...
zero_padding2d_2 (ZeroPadding2D)	(None, 34, 34, 256)	0	sequential_35[0]...
conv2d_24 (Conv2D)	(None, 31, 31, 512)	2,097,152	zero_padding2d_2...
batch_normalizatio... (BatchNormalizatio...)	(None, 31, 31, 512)	2,048	conv2d_24[0][0]
leaky_re_lu_23 (LeakyReLU)	(None, 31, 31, 512)	0	batch_normalizat...
zero_padding2d_3 (ZeroPadding2D)	(None, 33, 33, 512)	0	leaky_re_lu_23[0]...
conv2d_25 (Conv2D)	(None, 30, 30, 1)	8,193	zero_padding2d_3...

Total params: 2,770,433 (10.57 MB)

Trainable params: 2,768,641 (10.56 MB)

Non-trainable params: 1,792 (7.00 KB)

Testing discriminator

```
In [56]: input_test = sample_images[0]
target_test = sample_masks[0]

input_test = tf.expand_dims(input_test, axis=0)
target_test = tf.expand_dims(target_test, axis=0)

discriminator_output = discriminator.model([input_test, target_test], training=False)

input_test = (input_test[0] + 1) / 2
target_test = (target_test[0] + 1) / 2
```

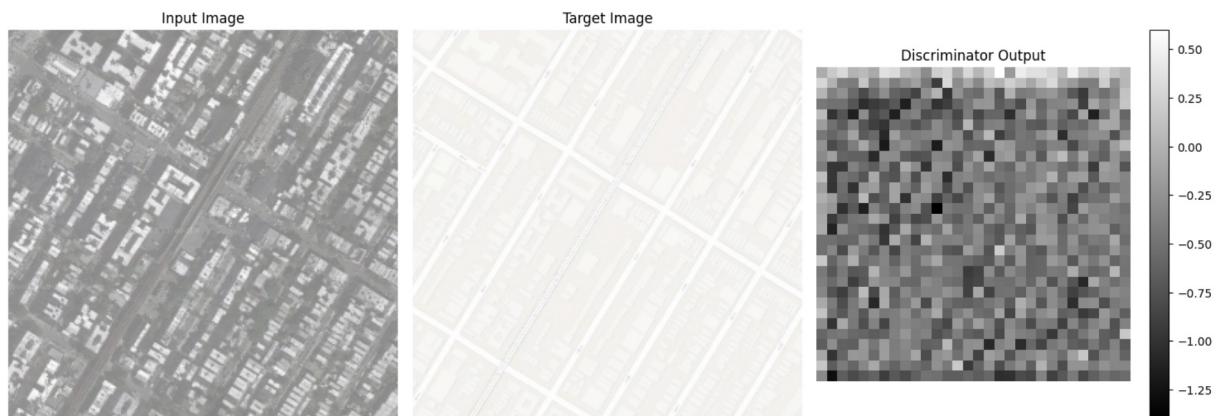
```
plt.figure(figsize=(15, 5))

plt.subplot(1, 3, 1)
plt.title("Input Image")
plt.imshow(input_test)
plt.axis("off")

plt.subplot(1, 3, 2)
plt.title("Target Image")
plt.imshow(target_test)
plt.axis("off")

plt.subplot(1, 3, 3)
plt.title("Discriminator Output")
plt.imshow(discriminator_output.numpy(), cmap="gray")
plt.colorbar()
plt.axis("off")

plt.tight_layout()
plt.show()
```



Discriminator is random so output is all over the place

Loss Functions

```
In [57]: cross_entropy = tf.keras.losses.BinaryCrossentropy(from_logits=True)

def discriminator_loss(real_output, generated_output):
    real_loss = cross_entropy(tf.ones_like(real_output), real_output)
    generated_loss = cross_entropy(tf.zeros_like(generated_output), generated_output)
    return real_loss + generated_loss

def generator_loss(generated_output, gen_output, target):
    adv_loss = cross_entropy(tf.ones_like(generated_output), generated_output) # F
    l1_loss = tf.reduce_mean(tf.abs(target - gen_output))
    return adv_loss + (100 * l1_loss)

generator_optimizer = tf.keras.optimizers.Adam(2e-4, beta_1=0.5)
discriminator_optimizer = tf.keras.optimizers.Adam(2e-4, beta_1=0.5)
```

Define training function

```
In [58]: def display_samples(epoch, input_images, target_images, generated_images):
    fig, axes = plt.subplots(1, 3, figsize=(15, 5))

    # Display the first image from the batch (index 0)
    axes[0].imshow(input_images[0])
    axes[1].imshow(target_images[0])
    axes[2].imshow(generated_images[0])

    # Set titles for each subplot
    axes[0].set_title("Input Image")
    axes[1].set_title("Target Image")
    axes[2].set_title("Generated Image")

    plt.show()
```

```
In [59]: def update_disc_gen(input_image, target_image):
    with tf.GradientTape() as gen_tape, tf.GradientTape() as disc_tape:

        # generate an image based on input
        generated_image = generator.model(input_image, training=True)

        # discriminator outputs one for input and target and one for input and generate
        real_output = discriminator.model([input_image, target_image], training=True)
        generated_output_disc = discriminator.model([input_image, generated_image], tra

        # calculate the loss of thee generator and the discriminator
        gen_loss = generator_loss(generated_output_disc, generated_image, target_image)
        disc_loss = discriminator_loss(real_output, generated_output_disc)

        # get gradient loss
        gradients_of_generator = gen_tape.gradient(gen_loss, generator.model.trainable_variables)
        gradients_of_discriminator = disc_tape.gradient(disc_loss, discriminator.model.trainable_variables)

        # apply the gradient loss to the variables
        generator_optimizer.apply_gradients(zip(gradients_of_generator, generator.model.trainable_variables))
        discriminator_optimizer.apply_gradients(zip(gradients_of_discriminator, discriminator.model.trainable_variables))

    return gen_loss, disc_loss

def train(dataset, epochs):
    for epoch in range(epochs):
        for input_image, target_image in dataset:
            gen_loss, disc_loss = update_disc_gen(input_image, target_image)

            print(f"Epoch {epoch+1}, Gen Loss: {gen_loss.numpy()}, Disc Loss: {disc_loss.numpy()}")

        # print samples of training images
        if ((epoch) % 10 == 0):
            sample_image = generator.model(input_image, training=False)
            display_samples(epoch, [input_image[0]], [target_image[0]], [sample_image[0]])
```

```
In [60]: # preprocess image data
def preprocess(input_image, target_image):
    input_image = tf.image.resize(input_image, [256, 256])
    target_image = tf.image.resize(target_image, [256, 256])
    return input_image, target_image

data = (
    tf.data.Dataset.from_tensor_slices((images, masks))
    .map(preprocess, num_parallel_calls=tf.data.AUTOTUNE)
    .shuffle(4000)
    .batch(32)
    .prefetch(tf.data.AUTOTUNE)
)
```

```
In [61]: train(data, 21)
```

Epoch 1, Gen Loss: 90.55968475341797, Disc Loss: 1.7320172786712646

Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers).



Epoch 2, Gen Loss: 87.78022766113281, Disc Loss: 2.9960546493530273

Epoch 3, Gen Loss: 83.77320098876953, Disc Loss: 1.4531104564666748

Epoch 4, Gen Loss: 80.70641326904297, Disc Loss: 1.5677103996276855

Epoch 5, Gen Loss: 77.44559478759766, Disc Loss: 1.3259824514389038

Epoch 6, Gen Loss: 74.3031997680664, Disc Loss: 1.2004390954971313

Epoch 7, Gen Loss: 71.2608642578125, Disc Loss: 1.1303706169128418

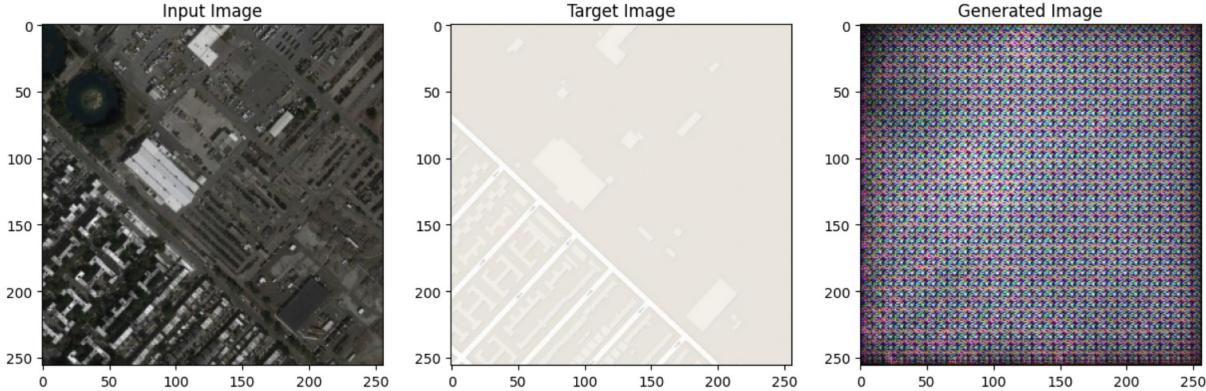
Epoch 8, Gen Loss: 68.24303436279297, Disc Loss: 1.0370888710021973

Epoch 9, Gen Loss: 65.27452087402344, Disc Loss: 0.968621015548706

Epoch 10, Gen Loss: 62.32588577270508, Disc Loss: 0.9052463173866272

Epoch 11, Gen Loss: 59.45484161376953, Disc Loss: 0.8562255501747131

Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers).



Epoch 12, Gen Loss: 56.6362190246582, Disc Loss: 0.8214210271835327

Epoch 13, Gen Loss: 53.91592788696289, Disc Loss: 0.8362501263618469

Epoch 14, Gen Loss: 51.31574249267578, Disc Loss: 0.9857845306396484

Epoch 15, Gen Loss: 48.67427444458008, Disc Loss: 0.9387552738189697

Epoch 16, Gen Loss: 45.88151550292969, Disc Loss: 0.7475816011428833

Epoch 17, Gen Loss: 43.288753509521484, Disc Loss: 0.7494750022888184

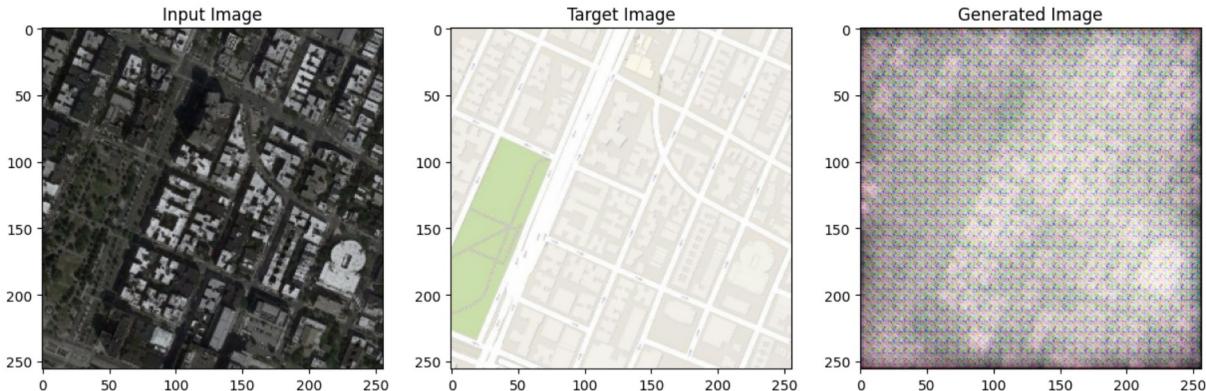
Epoch 18, Gen Loss: 40.830074310302734, Disc Loss: 0.7518161535263062

Epoch 19, Gen Loss: 38.4001579284668, Disc Loss: 0.7196889519691467

Epoch 20, Gen Loss: 36.084537506103516, Disc Loss: 0.722257137298584

Epoch 21, Gen Loss: 33.899024963378906, Disc Loss: 0.7352006435394287

Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers).



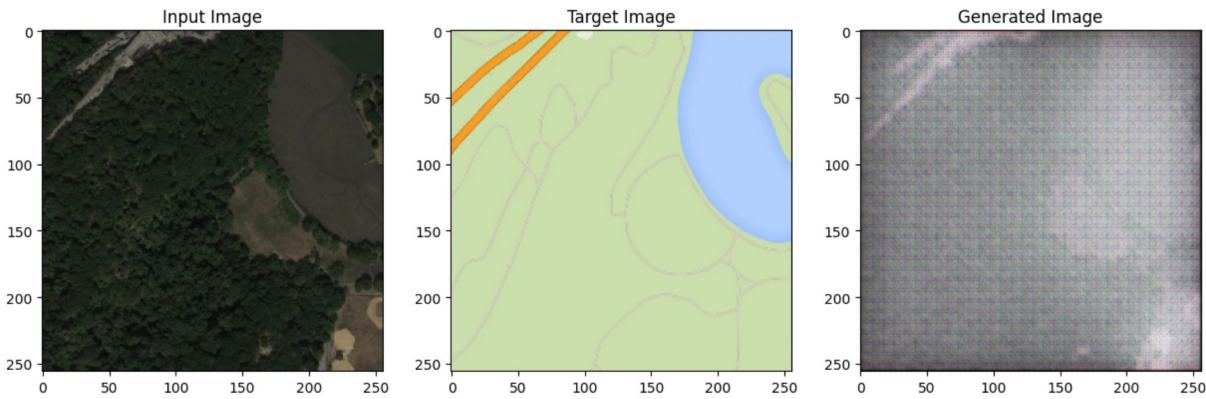
In [62]: `train(data, 21)`

Epoch 1, Gen Loss: 31.72567367553711, Disc Loss: 0.7482166886329651

Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers).



Epoch 2, Gen Loss: 29.787464141845703, Disc Loss: 0.7348878383636475
Epoch 3, Gen Loss: 27.802326202392578, Disc Loss: 0.734105110168457
Epoch 4, Gen Loss: 26.0485897064209, Disc Loss: 0.7239633798599243
Epoch 5, Gen Loss: 24.301570892333984, Disc Loss: 0.727171778678894
Epoch 6, Gen Loss: 22.822593688964844, Disc Loss: 0.7233973741531372
Epoch 7, Gen Loss: 21.304067611694336, Disc Loss: 0.74266517162323
Epoch 8, Gen Loss: 20.13869857788086, Disc Loss: 0.7533473372459412
Epoch 9, Gen Loss: 18.541866302490234, Disc Loss: 0.8032876253128052
Epoch 10, Gen Loss: 17.701805114746094, Disc Loss: 0.9106389880180359
Epoch 11, Gen Loss: 16.480806350708008, Disc Loss: 1.0348138809204102

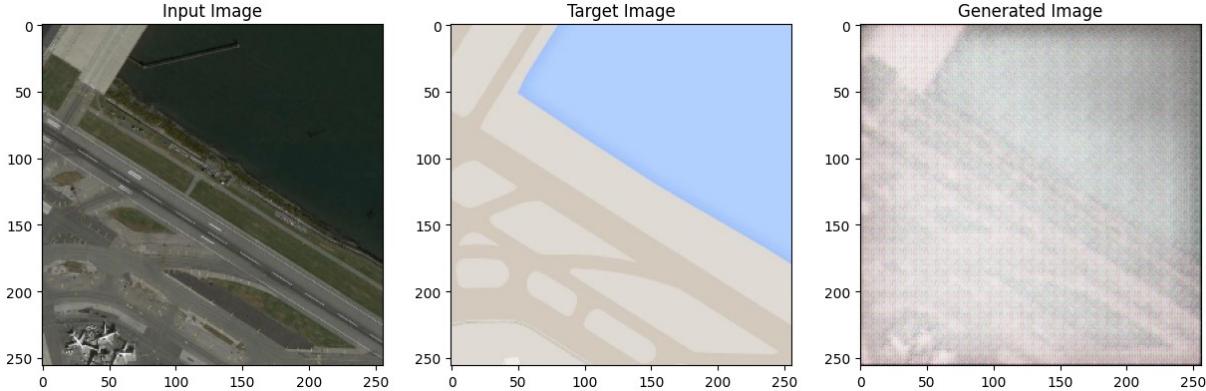


Epoch 12, Gen Loss: 15.738273620605469, Disc Loss: 0.8922258019447327
Epoch 13, Gen Loss: 14.432085037231445, Disc Loss: 0.8443456888198853
Epoch 14, Gen Loss: 13.843120574951172, Disc Loss: 0.8696218729019165
Epoch 15, Gen Loss: 12.795676231384277, Disc Loss: 0.8572313785552979
Epoch 16, Gen Loss: 12.272102355957031, Disc Loss: 0.8326388597488403
Epoch 17, Gen Loss: 11.311164855957031, Disc Loss: 0.86054527759552
Epoch 18, Gen Loss: 11.512260437011719, Disc Loss: 0.8881077170372009
Epoch 19, Gen Loss: 10.501047134399414, Disc Loss: 1.0621756315231323
Epoch 20, Gen Loss: 10.915243148803711, Disc Loss: 0.9266180992126465
Epoch 21, Gen Loss: 9.3489351272583, Disc Loss: 0.9385260343551636



In [63]: `train(data, 21)`

Epoch 1, Gen Loss: 9.247696876525879, Disc Loss: 0.8014193773269653



Epoch 2, Gen Loss: 9.038843154907227, Disc Loss: 0.79613196849823

Epoch 3, Gen Loss: 8.33626937866211, Disc Loss: 0.8397658467292786

Epoch 4, Gen Loss: 8.580901145935059, Disc Loss: 0.8064696788787842

Epoch 5, Gen Loss: 8.187509536743164, Disc Loss: 1.0283032655715942

Epoch 6, Gen Loss: 7.899869918823242, Disc Loss: 1.076673984527588

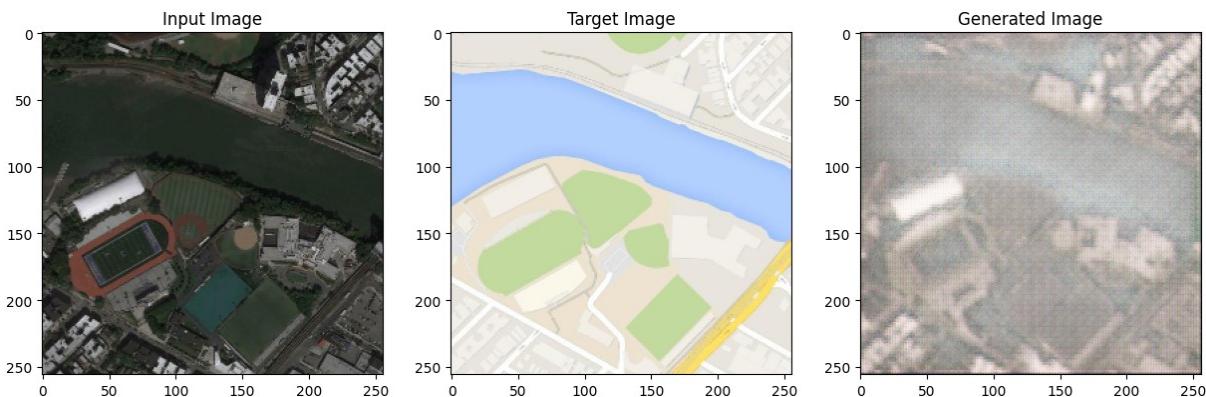
Epoch 7, Gen Loss: 7.325778007507324, Disc Loss: 0.879375696182251

Epoch 8, Gen Loss: 7.502342224121094, Disc Loss: 0.8643402457237244

Epoch 9, Gen Loss: 7.338085174560547, Disc Loss: 0.7916334271430969

Epoch 10, Gen Loss: 7.170848846435547, Disc Loss: 0.8263390064239502

Epoch 11, Gen Loss: 6.913078784942627, Disc Loss: 0.8048774003982544



Epoch 12, Gen Loss: 7.4210357666015625, Disc Loss: 0.8123117685317993

Epoch 13, Gen Loss: 6.302732467651367, Disc Loss: 0.9347323179244995

Epoch 14, Gen Loss: 7.625718116760254, Disc Loss: 0.9815316200256348

Epoch 15, Gen Loss: 6.388637065887451, Disc Loss: 1.1880221366882324

Epoch 16, Gen Loss: 7.626391410827637, Disc Loss: 1.2345582246780396

Epoch 17, Gen Loss: 7.568384170532227, Disc Loss: 0.8317245244979858

Epoch 18, Gen Loss: 6.717928409576416, Disc Loss: 0.8609824776649475

Epoch 19, Gen Loss: 7.048576831817627, Disc Loss: 0.7949750423431396

Epoch 20, Gen Loss: 6.876395225524902, Disc Loss: 0.689284086227417

Epoch 21, Gen Loss: 6.222752571105957, Disc Loss: 0.8765100240707397



```
In [64]: train(data, 21)
```

Epoch 1, Gen Loss: 7.166828632354736, Disc Loss: 1.056659460067749



Epoch 2, Gen Loss: 6.884897708892822, Disc Loss: 1.4339724779129028

Epoch 3, Gen Loss: 7.641903877258301, Disc Loss: 0.9673659801483154

Epoch 4, Gen Loss: 6.752223014831543, Disc Loss: 0.6817290186882019

Epoch 5, Gen Loss: 6.751965522766113, Disc Loss: 0.8272587060928345

Epoch 6, Gen Loss: 6.9242048263549805, Disc Loss: 0.7465325593948364

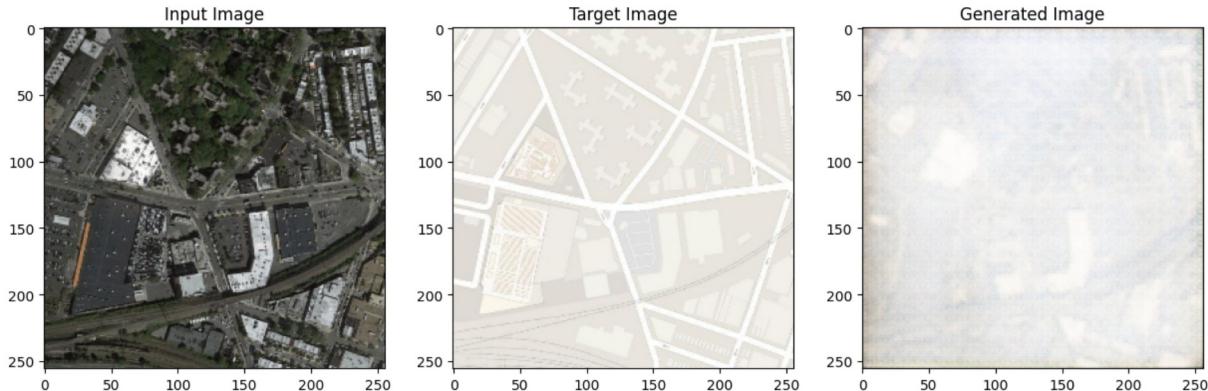
Epoch 7, Gen Loss: 6.196351528167725, Disc Loss: 0.7873330116271973

Epoch 8, Gen Loss: 6.749709129333496, Disc Loss: 0.8503270149230957

Epoch 9, Gen Loss: 5.934866905212402, Disc Loss: 1.006836175918579

Epoch 10, Gen Loss: 7.109516143798828, Disc Loss: 1.2839787006378174

Epoch 11, Gen Loss: 6.059560298919678, Disc Loss: 1.3786933422088623



Epoch 12, Gen Loss: 6.417690753936768, Disc Loss: 1.0106046199798584

Epoch 13, Gen Loss: 6.715047359466553, Disc Loss: 0.9478632807731628

Epoch 14, Gen Loss: 5.6870269775390625, Disc Loss: 1.2129766941070557

Epoch 15, Gen Loss: 6.604745388031006, Disc Loss: 1.3299674987792969

Epoch 16, Gen Loss: 5.303039073944092, Disc Loss: 1.1590096950531006

Epoch 17, Gen Loss: 5.722744941711426, Disc Loss: 1.022826075553894

Epoch 18, Gen Loss: 5.6650238037109375, Disc Loss: 0.9756796360015869

Epoch 19, Gen Loss: 5.645843029022217, Disc Loss: 0.9389742016792297

Epoch 20, Gen Loss: 5.668766975402832, Disc Loss: 0.9467642307281494

Epoch 21, Gen Loss: 5.189175605773926, Disc Loss: 0.9810123443603516



```
In [65]: train(data, 21)
```

Epoch 1, Gen Loss: 5.607276439666748, Disc Loss: 1.010812759399414



Epoch 2, Gen Loss: 4.89498233795166, Disc Loss: 1.0702787637710571

Epoch 3, Gen Loss: 6.003316879272461, Disc Loss: 1.2051559686660767

Epoch 4, Gen Loss: 4.477788925170898, Disc Loss: 1.5301018953323364

Epoch 5, Gen Loss: 5.696372985839844, Disc Loss: 1.1527800559997559

Epoch 6, Gen Loss: 4.907128810882568, Disc Loss: 1.0976190567016602

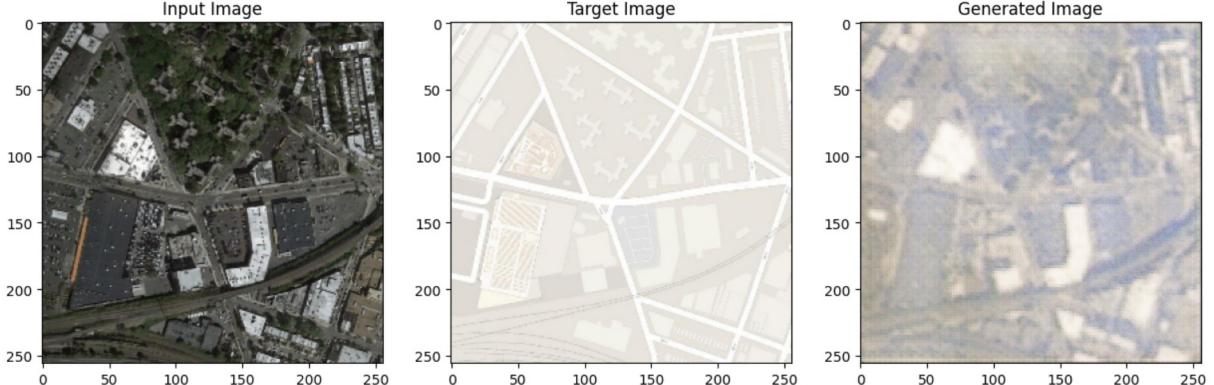
Epoch 7, Gen Loss: 5.327113628387451, Disc Loss: 1.1611706018447876

Epoch 8, Gen Loss: 4.857007026672363, Disc Loss: 1.24591064453125

Epoch 9, Gen Loss: 5.363732814788818, Disc Loss: 1.1985666751861572

Epoch 10, Gen Loss: 4.697413444519043, Disc Loss: 1.0677521228790283

Epoch 11, Gen Loss: 5.2771501541137695, Disc Loss: 1.0550247430801392



Epoch 12, Gen Loss: 4.688105583190918, Disc Loss: 1.0418082475662231
Epoch 13, Gen Loss: 5.253318786621094, Disc Loss: 1.0215883255004883
Epoch 14, Gen Loss: 4.629277229309082, Disc Loss: 1.015446424484253
Epoch 15, Gen Loss: 5.2694010734558105, Disc Loss: 1.0153272151947021
Epoch 16, Gen Loss: 4.562535285949707, Disc Loss: 1.0382812023162842
Epoch 17, Gen Loss: 5.412552833557129, Disc Loss: 1.0484299659729004
Epoch 18, Gen Loss: 4.486222743988037, Disc Loss: 1.1011418104171753
Epoch 19, Gen Loss: 5.495386123657227, Disc Loss: 1.091502070426941
Epoch 20, Gen Loss: 4.7653279304504395, Disc Loss: 1.0374473333358765
Epoch 21, Gen Loss: 5.232052803039551, Disc Loss: 0.9638056755065918

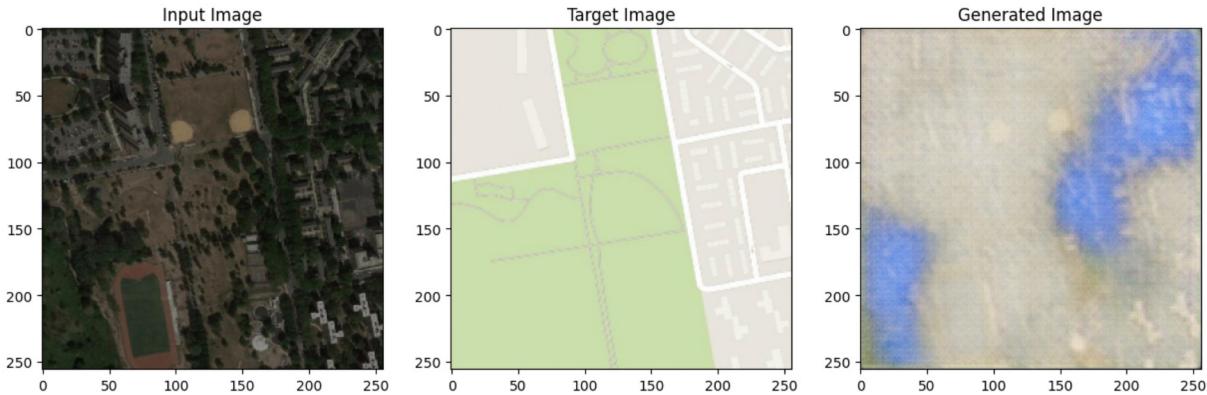


```
In [66]: generator_optimizer.learning_rate = 3e-4  
train(data, 21)
```

Epoch 1, Gen Loss: 4.985657691955566, Disc Loss: 0.9691625833511353



Epoch 2, Gen Loss: 4.995285987854004, Disc Loss: 0.915316104888916
Epoch 3, Gen Loss: 4.959049224853516, Disc Loss: 0.9374128580093384
Epoch 4, Gen Loss: 5.042456150054932, Disc Loss: 0.95871901512146
Epoch 5, Gen Loss: 5.157680511474609, Disc Loss: 1.018684983253479
Epoch 6, Gen Loss: 6.22879695892334, Disc Loss: 1.1493558883666992
Epoch 7, Gen Loss: 6.231919288635254, Disc Loss: 1.129786491394043
Epoch 8, Gen Loss: 5.493568420410156, Disc Loss: 1.194287896156311
Epoch 9, Gen Loss: 5.808053016662598, Disc Loss: 1.127016544342041
Epoch 10, Gen Loss: 5.359449863433838, Disc Loss: 0.9616712331771851
Epoch 11, Gen Loss: 6.010220527648926, Disc Loss: 1.1556856632232666



Epoch 12, Gen Loss: 5.603280067443848, Disc Loss: 1.1285210847854614

Epoch 13, Gen Loss: 4.942075729370117, Disc Loss: 1.3269002437591553

Epoch 14, Gen Loss: 4.993734359741211, Disc Loss: 1.4144541025161743

Epoch 15, Gen Loss: 4.758805751800537, Disc Loss: 1.4013310670852661

Epoch 16, Gen Loss: 4.699134349822998, Disc Loss: 1.3619606494903564

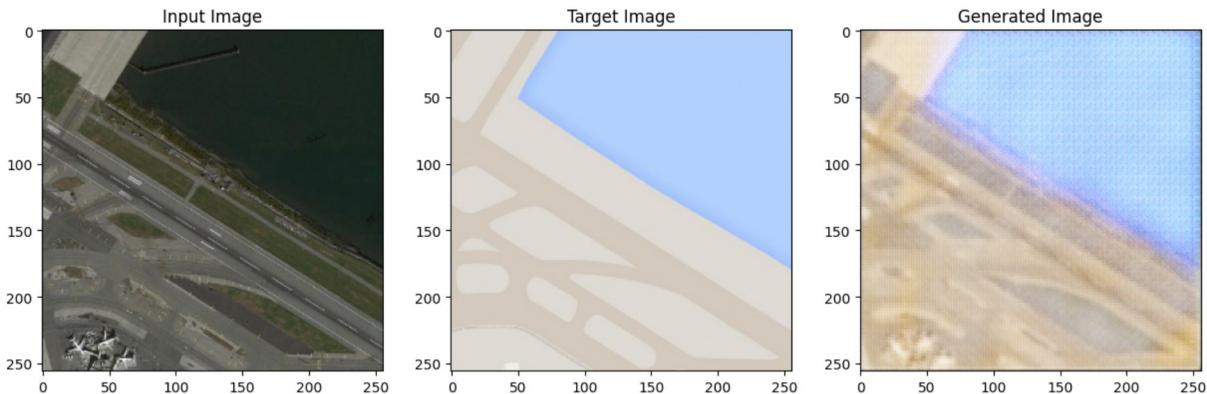
Epoch 17, Gen Loss: 4.868474006652832, Disc Loss: 1.3779492378234863

Epoch 18, Gen Loss: 4.894190788269043, Disc Loss: 1.3038654327392578

Epoch 19, Gen Loss: 4.7633280754089355, Disc Loss: 1.296945571899414

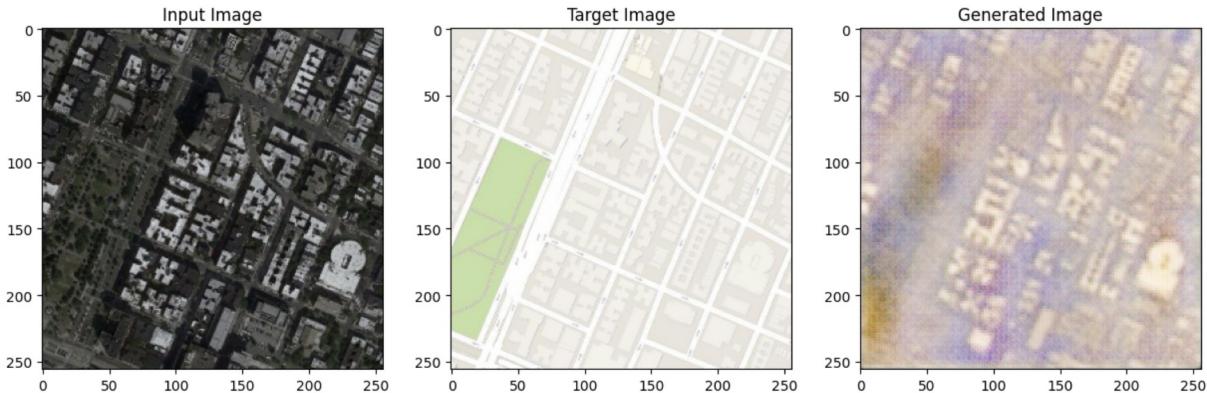
Epoch 20, Gen Loss: 4.586258888244629, Disc Loss: 1.218764305114746

Epoch 21, Gen Loss: 4.431421279907227, Disc Loss: 1.1877145767211914



In [67]: `train(data, 21)`

Epoch 1, Gen Loss: 4.441211700439453, Disc Loss: 1.2202250957489014

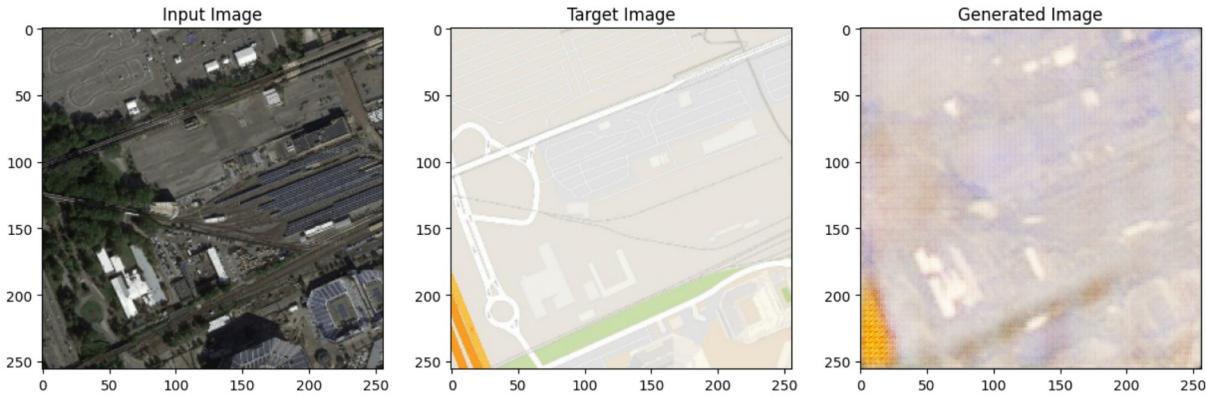


Epoch 2, Gen Loss: 4.308872222900391, Disc Loss: 1.2786072492599487
Epoch 3, Gen Loss: 4.413944721221924, Disc Loss: 1.3124301433563232
Epoch 4, Gen Loss: 4.185634136199951, Disc Loss: 1.4054667949676514
Epoch 5, Gen Loss: 4.250639915466309, Disc Loss: 1.4299561977386475
Epoch 6, Gen Loss: 4.211755752563477, Disc Loss: 1.4422203302383423
Epoch 7, Gen Loss: 4.31046199798584, Disc Loss: 1.4129940271377563
Epoch 8, Gen Loss: 4.30525541305542, Disc Loss: 1.4046847820281982
Epoch 9, Gen Loss: 4.365664005279541, Disc Loss: 1.3675020933151245
Epoch 10, Gen Loss: 4.177990436553955, Disc Loss: 1.345321536064148
Epoch 11, Gen Loss: 4.24305534362793, Disc Loss: 1.3274004459381104



Epoch 12, Gen Loss: 4.137127876281738, Disc Loss: 1.3422658443450928
Epoch 13, Gen Loss: 4.229859828948975, Disc Loss: 1.3695683479309082
Epoch 14, Gen Loss: 4.086915969848633, Disc Loss: 1.3648555278778076
Epoch 15, Gen Loss: 4.153689384460449, Disc Loss: 1.3018815517425537
Epoch 16, Gen Loss: 4.070995330810547, Disc Loss: 1.2689003944396973
Epoch 17, Gen Loss: 4.116881847381592, Disc Loss: 1.2995465993881226
Epoch 18, Gen Loss: 4.026427268981934, Disc Loss: 1.3517799377441406
Epoch 19, Gen Loss: 4.0947723388671875, Disc Loss: 1.4158251285552979
Epoch 20, Gen Loss: 3.871514320373535, Disc Loss: 1.4754259586334229
Epoch 21, Gen Loss: 3.9610438346862793, Disc Loss: 1.476680040359497

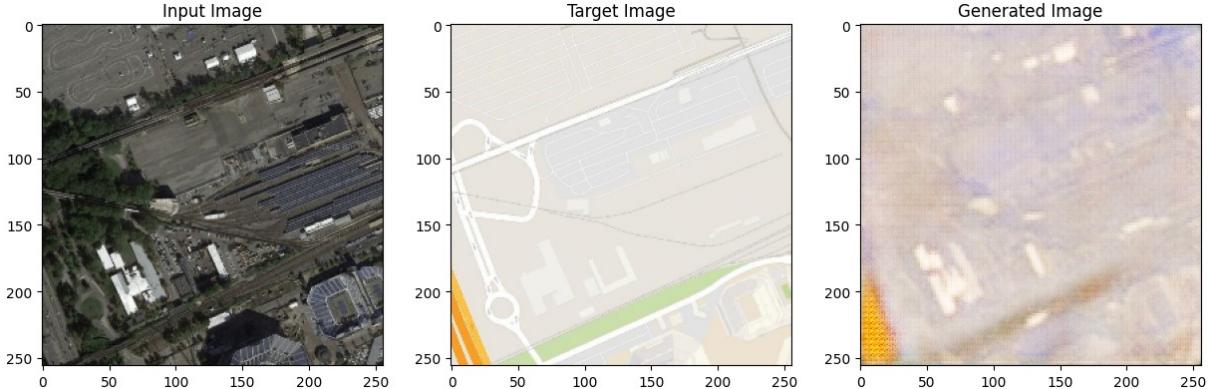
Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers).



In [68]: `train(data, 21)`

Epoch 1, Gen Loss: 3.762789011001587, Disc Loss: 1.5314273834228516

Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers).

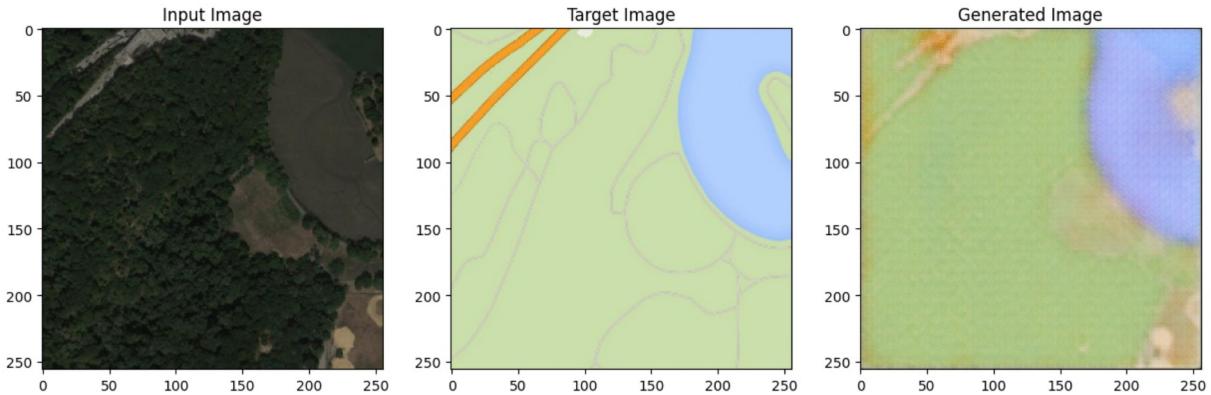


Epoch 2, Gen Loss: 3.895963430404663, Disc Loss: 1.5194758176803589
Epoch 3, Gen Loss: 3.7267284393310547, Disc Loss: 1.5432214736938477
Epoch 4, Gen Loss: 3.8253047466278076, Disc Loss: 1.5042692422866821
Epoch 5, Gen Loss: 3.7476112842559814, Disc Loss: 1.504315972328186
Epoch 6, Gen Loss: 3.8222129344940186, Disc Loss: 1.4629734754562378
Epoch 7, Gen Loss: 3.8018107414245605, Disc Loss: 1.4651978015899658
Epoch 8, Gen Loss: 3.8698630332946777, Disc Loss: 1.4410065412521362
Epoch 9, Gen Loss: 3.8463010787963867, Disc Loss: 1.4510232210159302
Epoch 10, Gen Loss: 3.7800137996673584, Disc Loss: 1.4595332145690918
Epoch 11, Gen Loss: 3.712737560272217, Disc Loss: 1.4772802591323853

Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers).

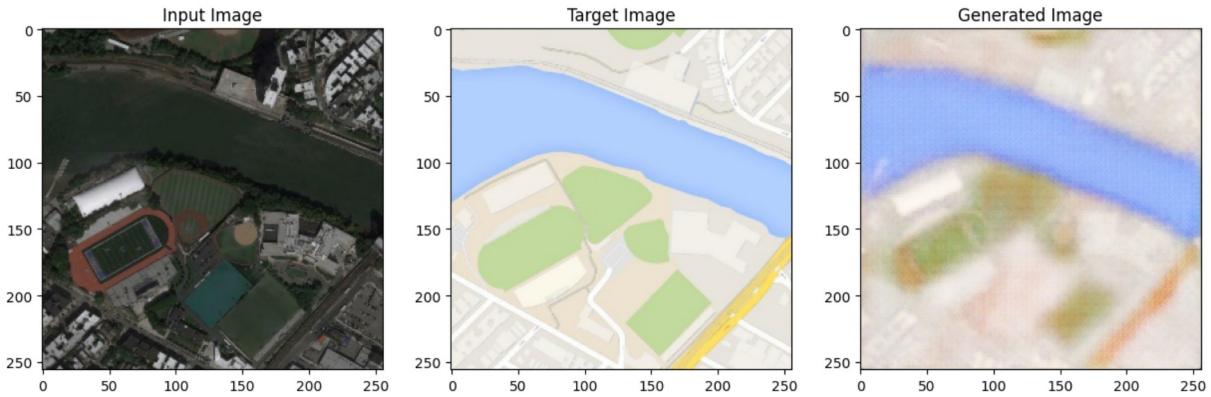


Epoch 12, Gen Loss: 3.704549789428711, Disc Loss: 1.4714632034301758
Epoch 13, Gen Loss: 3.661909818649292, Disc Loss: 1.46830415725708
Epoch 14, Gen Loss: 3.648505926132202, Disc Loss: 1.43630051612854
Epoch 15, Gen Loss: 3.653550624847412, Disc Loss: 1.4364569187164307
Epoch 16, Gen Loss: 3.6714413166046143, Disc Loss: 1.4123700857162476
Epoch 17, Gen Loss: 3.673006534576416, Disc Loss: 1.4209368228912354
Epoch 18, Gen Loss: 3.6879453659057617, Disc Loss: 1.397095799446106
Epoch 19, Gen Loss: 3.669893741607666, Disc Loss: 1.402268409729004
Epoch 20, Gen Loss: 3.685434579849243, Disc Loss: 1.3922629356384277
Epoch 21, Gen Loss: 3.627053737640381, Disc Loss: 1.403312087059021



```
In [69]: train(data, 41)
```

Epoch 1, Gen Loss: 3.6181886196136475, Disc Loss: 1.40055251121521



Epoch 2, Gen Loss: 3.617732048034668, Disc Loss: 1.4362609386444092

Epoch 3, Gen Loss: 3.6268234252929688, Disc Loss: 1.4552803039550781

Epoch 4, Gen Loss: 3.6085517406463623, Disc Loss: 1.4601569175720215

Epoch 5, Gen Loss: 3.5777626037597656, Disc Loss: 1.4250433444976807

Epoch 6, Gen Loss: 3.532135486602783, Disc Loss: 1.4057559967041016

Epoch 7, Gen Loss: 3.544095039367676, Disc Loss: 1.3852713108062744

Epoch 8, Gen Loss: 3.5300323963165283, Disc Loss: 1.3951070308685303

Epoch 9, Gen Loss: 3.5377414226531982, Disc Loss: 1.3780720233917236

Epoch 10, Gen Loss: 3.5221550464630127, Disc Loss: 1.3883826732635498

Epoch 11, Gen Loss: 3.534773588180542, Disc Loss: 1.3735411167144775



Epoch 12, Gen Loss: 3.5276455879211426, Disc Loss: 1.3873411417007446
Epoch 13, Gen Loss: 3.5727765560150146, Disc Loss: 1.3646713495254517
Epoch 14, Gen Loss: 3.5377085208892822, Disc Loss: 1.3809356689453125
Epoch 15, Gen Loss: 3.517442226409912, Disc Loss: 1.3719923496246338
Epoch 16, Gen Loss: 3.474393606185913, Disc Loss: 1.3740311861038208
Epoch 17, Gen Loss: 3.4891655445098877, Disc Loss: 1.371692180633545
Epoch 18, Gen Loss: 3.459841728210449, Disc Loss: 1.3772636651992798
Epoch 19, Gen Loss: 3.470602035522461, Disc Loss: 1.3718098402023315
Epoch 20, Gen Loss: 3.4578280448913574, Disc Loss: 1.377748727798462
Epoch 21, Gen Loss: 3.479086399078369, Disc Loss: 1.363218069076538



Epoch 22, Gen Loss: 3.4657394886016846, Disc Loss: 1.3822516202926636
Epoch 23, Gen Loss: 3.4705586433410645, Disc Loss: 1.367061734199524
Epoch 24, Gen Loss: 3.410533905029297, Disc Loss: 1.3874658346176147
Epoch 25, Gen Loss: 3.4285855293273926, Disc Loss: 1.3779890537261963
Epoch 26, Gen Loss: 3.3804659843444824, Disc Loss: 1.398555874824524
Epoch 27, Gen Loss: 3.4260270595550537, Disc Loss: 1.3885977268218994
Epoch 28, Gen Loss: 3.3889434337615967, Disc Loss: 1.4048182964324951
Epoch 29, Gen Loss: 3.4295825958251953, Disc Loss: 1.3946443796157837
Epoch 30, Gen Loss: 3.418572425842285, Disc Loss: 1.3930425643920898
Epoch 31, Gen Loss: 3.4643638134002686, Disc Loss: 1.381009817123413



Epoch 32, Gen Loss: 3.431386709213257, Disc Loss: 1.380225419998169
Epoch 33, Gen Loss: 3.438751220703125, Disc Loss: 1.3636460304260254
Epoch 34, Gen Loss: 3.388833999633789, Disc Loss: 1.3788154125213623
Epoch 35, Gen Loss: 3.379103422164917, Disc Loss: 1.3589632511138916
Epoch 36, Gen Loss: 3.335887908935547, Disc Loss: 1.3780517578125
Epoch 37, Gen Loss: 3.346510410308838, Disc Loss: 1.363372564315796
Epoch 38, Gen Loss: 3.322589635848999, Disc Loss: 1.3754870891571045
Epoch 39, Gen Loss: 3.3323416709899902, Disc Loss: 1.3665878772735596
Epoch 40, Gen Loss: 3.303720235824585, Disc Loss: 1.376399040222168
Epoch 41, Gen Loss: 3.3279500007629395, Disc Loss: 1.362355351448059

Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers).



```
In [70]: train(data, 41)
```

Epoch 1, Gen Loss: 3.2870240211486816, Disc Loss: 1.3731200695037842



Epoch 2, Gen Loss: 3.320012092590332, Disc Loss: 1.3605396747589111

Epoch 3, Gen Loss: 3.307335138320923, Disc Loss: 1.3719090223312378

Epoch 4, Gen Loss: 3.3590173721313477, Disc Loss: 1.362344741821289

Epoch 5, Gen Loss: 3.316124677658081, Disc Loss: 1.3762550354003906

Epoch 6, Gen Loss: 3.3261265754699707, Disc Loss: 1.37057363986969

Epoch 7, Gen Loss: 3.317150831222534, Disc Loss: 1.3681323528289795

Epoch 8, Gen Loss: 3.331265449523926, Disc Loss: 1.3733549118041992

Epoch 9, Gen Loss: 3.408881187438965, Disc Loss: 1.35912024974823

Epoch 10, Gen Loss: 3.411746025085449, Disc Loss: 1.371948003768921

Epoch 11, Gen Loss: 3.413745403289795, Disc Loss: 1.358940839767456



Epoch 12, Gen Loss: 3.3428001403808594, Disc Loss: 1.3624409437179565
Epoch 13, Gen Loss: 3.349527597427368, Disc Loss: 1.3547757863998413
Epoch 14, Gen Loss: 3.3206634521484375, Disc Loss: 1.369087815284729
Epoch 15, Gen Loss: 3.3450565338134766, Disc Loss: 1.3809728622436523
Epoch 16, Gen Loss: 3.3356523513793945, Disc Loss: 1.4264328479766846
Epoch 17, Gen Loss: 3.3648486137390137, Disc Loss: 1.427596092224121
Epoch 18, Gen Loss: 3.2865920066833496, Disc Loss: 1.3938426971435547
Epoch 19, Gen Loss: 3.277703046798706, Disc Loss: 1.3723177909851074
Epoch 20, Gen Loss: 3.3091273307800293, Disc Loss: 1.3737565279006958
Epoch 21, Gen Loss: 3.248065948486328, Disc Loss: 1.358078956604004

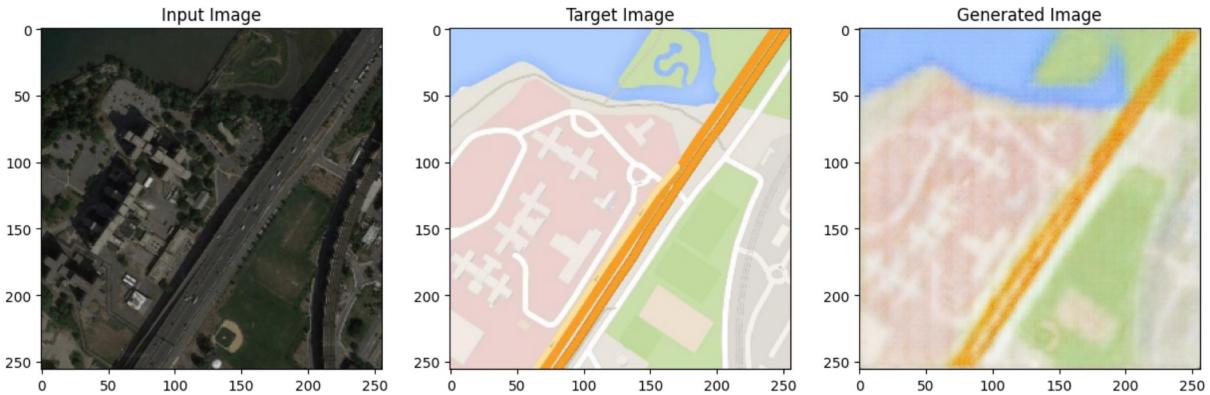


Epoch 22, Gen Loss: 3.232203245162964, Disc Loss: 1.3605492115020752
Epoch 23, Gen Loss: 3.2667629718780518, Disc Loss: 1.3456226587295532
Epoch 24, Gen Loss: 3.1997222900390625, Disc Loss: 1.3695597648620605
Epoch 25, Gen Loss: 3.3156557083129883, Disc Loss: 1.3459696769714355
Epoch 26, Gen Loss: 3.20531964302063, Disc Loss: 1.369200587272644
Epoch 27, Gen Loss: 3.2940096855163574, Disc Loss: 1.350928783416748
Epoch 28, Gen Loss: 3.163276195526123, Disc Loss: 1.3550360202789307
Epoch 29, Gen Loss: 3.2872283458709717, Disc Loss: 1.3446779251098633
Epoch 30, Gen Loss: 3.206969976425171, Disc Loss: 1.3491663932800293
Epoch 31, Gen Loss: 3.355809211730957, Disc Loss: 1.3487091064453125



Epoch 32, Gen Loss: 3.2327919006347656, Disc Loss: 1.3418548107147217
Epoch 33, Gen Loss: 3.354914665222168, Disc Loss: 1.3359318971633911
Epoch 34, Gen Loss: 3.245469570159912, Disc Loss: 1.3353776931762695
Epoch 35, Gen Loss: 3.339447498321533, Disc Loss: 1.3234999179840088
Epoch 36, Gen Loss: 3.2003934383392334, Disc Loss: 1.3315458297729492
Epoch 37, Gen Loss: 3.2825167179107666, Disc Loss: 1.3284025192260742
Epoch 38, Gen Loss: 3.1527066230773926, Disc Loss: 1.3343075513839722
Epoch 39, Gen Loss: 3.2680535316467285, Disc Loss: 1.3320508003234863
Epoch 40, Gen Loss: 3.123180866241455, Disc Loss: 1.3443197011947632
Epoch 41, Gen Loss: 3.290069103240967, Disc Loss: 1.331114411354065

Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers).



```
In [71]: train(data, 41)
```

Epoch 1, Gen Loss: 3.1266415119171143, Disc Loss: 1.3446929454803467



Epoch 2, Gen Loss: 3.284548044204712, Disc Loss: 1.337277889251709

Epoch 3, Gen Loss: 3.1007206439971924, Disc Loss: 1.3327165842056274

Epoch 4, Gen Loss: 3.231480836868286, Disc Loss: 1.3300745487213135

Epoch 5, Gen Loss: 3.1287858486175537, Disc Loss: 1.3420329093933105

Epoch 6, Gen Loss: 3.2656657695770264, Disc Loss: 1.3399708271026611

Epoch 7, Gen Loss: 3.1219534873962402, Disc Loss: 1.3490562438964844

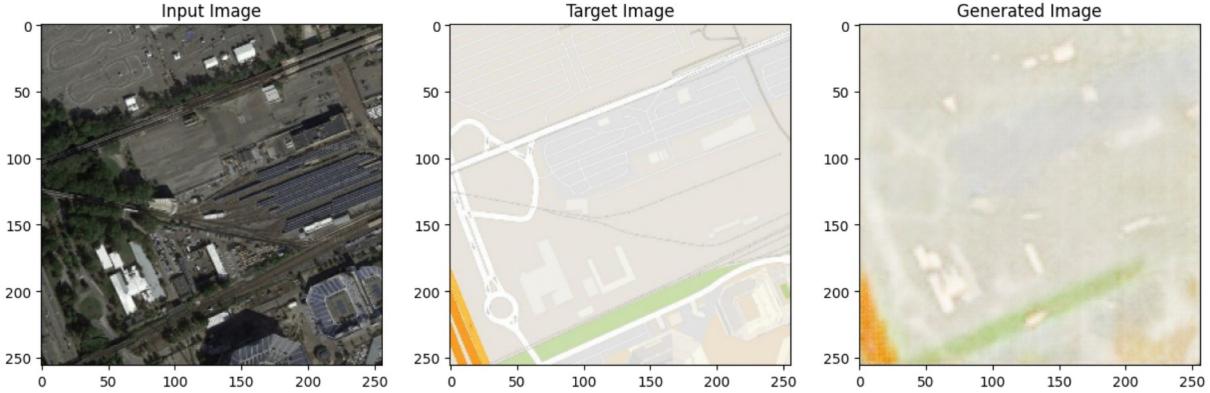
Epoch 8, Gen Loss: 3.2275102138519287, Disc Loss: 1.3371597528457642

Epoch 9, Gen Loss: 3.1162726879119873, Disc Loss: 1.3448147773742676

Epoch 10, Gen Loss: 3.2123396396636963, Disc Loss: 1.3400386571884155

Epoch 11, Gen Loss: 3.1500730514526367, Disc Loss: 1.3517260551452637

Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers).



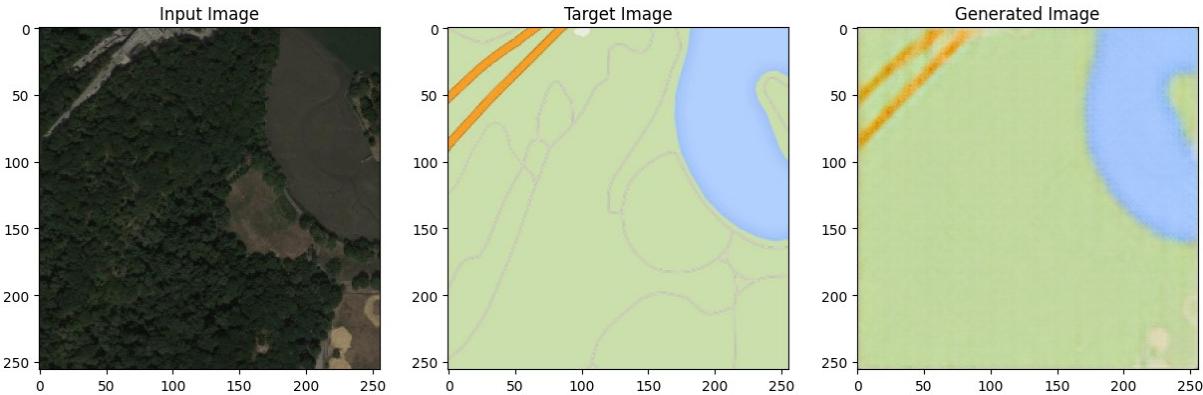
Epoch 12, Gen Loss: 3.2426037788391113, Disc Loss: 1.3750994205474854
Epoch 13, Gen Loss: 3.343109130859375, Disc Loss: 1.4378271102905273
Epoch 14, Gen Loss: 3.2138595581054688, Disc Loss: 1.4528536796569824
Epoch 15, Gen Loss: 3.2981812953948975, Disc Loss: 1.3964250087738037
Epoch 16, Gen Loss: 3.142219066619873, Disc Loss: 1.3796117305755615
Epoch 17, Gen Loss: 3.2683041095733643, Disc Loss: 1.3432186841964722
Epoch 18, Gen Loss: 3.212082862854004, Disc Loss: 1.3576490879058838
Epoch 19, Gen Loss: 3.2553961277008057, Disc Loss: 1.3278913497924805
Epoch 20, Gen Loss: 3.2348544597625732, Disc Loss: 1.3588273525238037
Epoch 21, Gen Loss: 3.2373480796813965, Disc Loss: 1.3522753715515137

Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers).

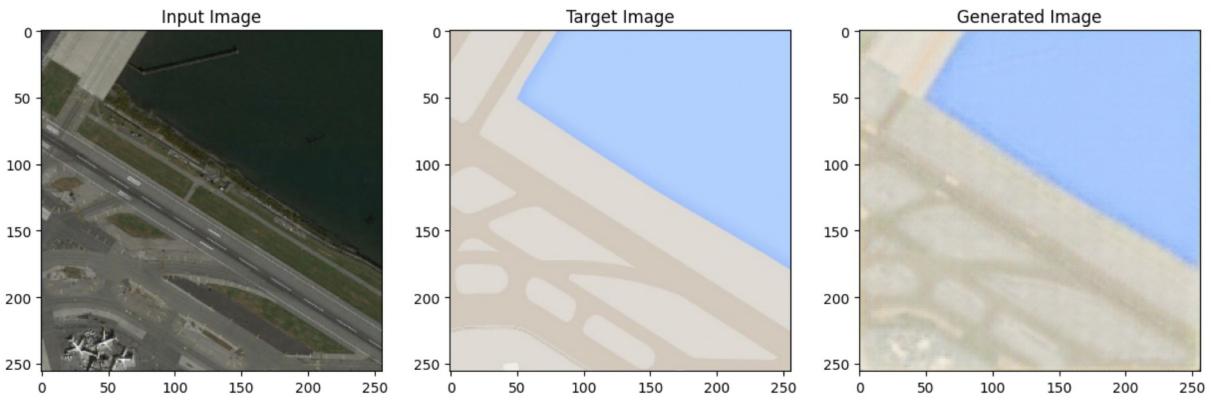


Epoch 22, Gen Loss: 3.1210122108459473, Disc Loss: 1.354752540588379
Epoch 23, Gen Loss: 3.2651257514953613, Disc Loss: 1.3674383163452148
Epoch 24, Gen Loss: 3.1802127361297607, Disc Loss: 1.358041763305664
Epoch 25, Gen Loss: 3.17922306060791, Disc Loss: 1.3667285442352295
Epoch 26, Gen Loss: 3.1061058044433594, Disc Loss: 1.3503260612487793
Epoch 27, Gen Loss: 3.0976250171661377, Disc Loss: 1.3587628602981567
Epoch 28, Gen Loss: 3.0875983238220215, Disc Loss: 1.3494155406951904
Epoch 29, Gen Loss: 3.0565733909606934, Disc Loss: 1.356964349746704
Epoch 30, Gen Loss: 3.106093406677246, Disc Loss: 1.3478920459747314
Epoch 31, Gen Loss: 3.065887451171875, Disc Loss: 1.3683518171310425

Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers).



```
Epoch 32, Gen Loss: 3.174800395965576, Disc Loss: 1.3566687107086182
Epoch 33, Gen Loss: 3.0097758769989014, Disc Loss: 1.3757827281951904
Epoch 34, Gen Loss: 3.170609474182129, Disc Loss: 1.3700836896896362
Epoch 35, Gen Loss: 3.025989532470703, Disc Loss: 1.384293556213379
Epoch 36, Gen Loss: 3.170562982559204, Disc Loss: 1.3847334384918213
Epoch 37, Gen Loss: 3.013366222381592, Disc Loss: 1.3972358703613281
Epoch 38, Gen Loss: 3.1458215713500977, Disc Loss: 1.3928987979888916
Epoch 39, Gen Loss: 3.0551235675811768, Disc Loss: 1.3787193298339844
Epoch 40, Gen Loss: 3.088038444519043, Disc Loss: 1.3508597612380981
Epoch 41, Gen Loss: 3.041661262512207, Disc Loss: 1.3435640335083008
```



```
In [72]: train(data, 41)
```

```
Epoch 1, Gen Loss: 3.0908212661743164, Disc Loss: 1.3415358066558838
```



```
Epoch 2, Gen Loss: 2.995311975479126, Disc Loss: 1.3422434329986572
Epoch 3, Gen Loss: 3.0740561485290527, Disc Loss: 1.3364959955215454
Epoch 4, Gen Loss: 3.005453109741211, Disc Loss: 1.345696210861206
Epoch 5, Gen Loss: 3.054384231567383, Disc Loss: 1.3388131856918335
Epoch 6, Gen Loss: 2.983888864517212, Disc Loss: 1.3483645915985107
Epoch 7, Gen Loss: 3.1270556449890137, Disc Loss: 1.3480966091156006
Epoch 8, Gen Loss: 3.0292654037475586, Disc Loss: 1.3524640798568726
Epoch 9, Gen Loss: 3.1197400093078613, Disc Loss: 1.3643709421157837
Epoch 10, Gen Loss: 2.9893746376037598, Disc Loss: 1.3785185813903809
Epoch 11, Gen Loss: 3.103727102279663, Disc Loss: 1.3829314708709717
```



Epoch 12, Gen Loss: 2.996309757232666, Disc Loss: 1.3744926452636719

Epoch 13, Gen Loss: 3.1375913619995117, Disc Loss: 1.3596473932266235

Epoch 14, Gen Loss: 2.946934700012207, Disc Loss: 1.3570947647094727

Epoch 15, Gen Loss: 3.1453332901000977, Disc Loss: 1.3736517429351807

Epoch 16, Gen Loss: 2.9183349609375, Disc Loss: 1.3676995038986206

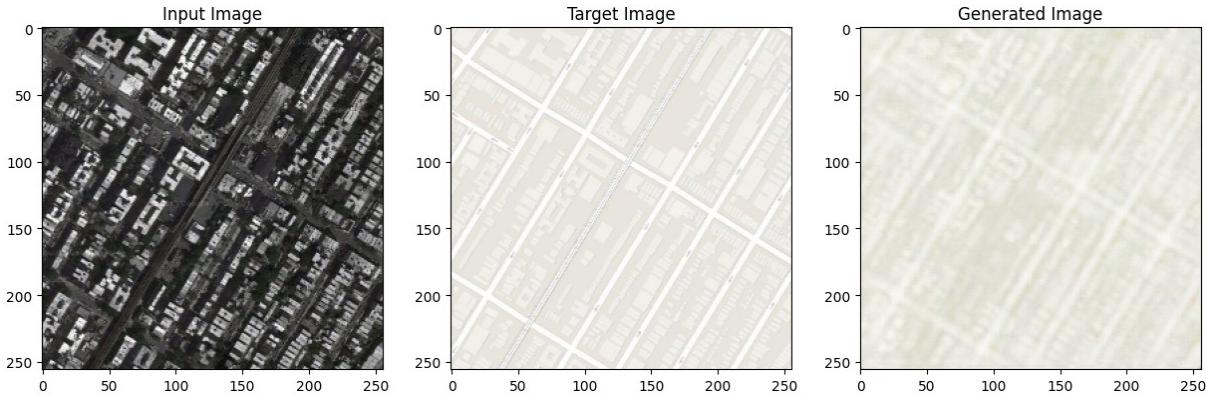
Epoch 17, Gen Loss: 3.07143235206604, Disc Loss: 1.3529467582702637

Epoch 18, Gen Loss: 2.880626916885376, Disc Loss: 1.3482609987258911

Epoch 19, Gen Loss: 3.0364418029785156, Disc Loss: 1.3329933881759644

Epoch 20, Gen Loss: 2.9182329177856445, Disc Loss: 1.3456072807312012

Epoch 21, Gen Loss: 3.0496866703033447, Disc Loss: 1.3348655700683594



Epoch 22, Gen Loss: 2.8555245399475098, Disc Loss: 1.3495140075683594

Epoch 23, Gen Loss: 3.0038914680480957, Disc Loss: 1.349084496498108

Epoch 24, Gen Loss: 2.8713011741638184, Disc Loss: 1.3630573749542236

Epoch 25, Gen Loss: 3.0278971195220947, Disc Loss: 1.3742592334747314

Epoch 26, Gen Loss: 2.932981491088867, Disc Loss: 1.3793234825134277

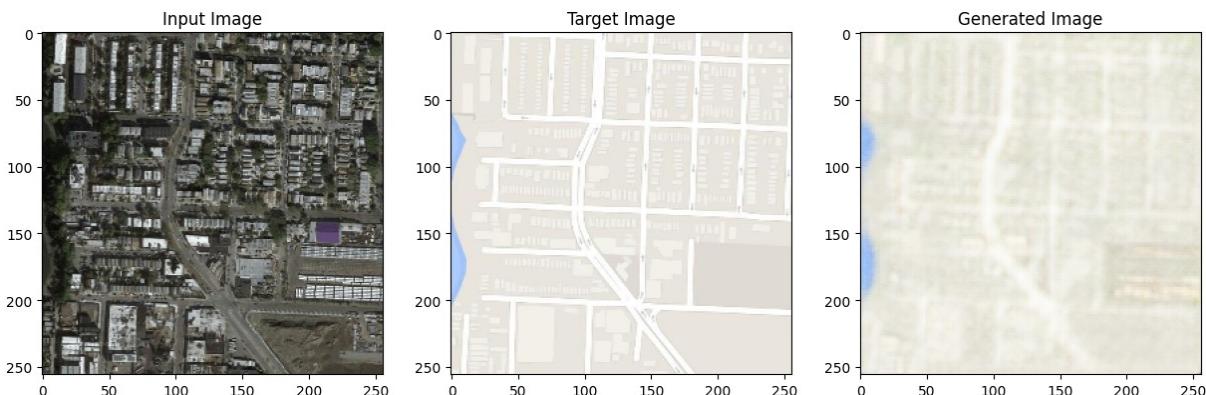
Epoch 27, Gen Loss: 2.9669978618621826, Disc Loss: 1.3594858646392822

Epoch 28, Gen Loss: 2.950587749481201, Disc Loss: 1.3396670818328857

Epoch 29, Gen Loss: 2.9917845726013184, Disc Loss: 1.3415223360061646

Epoch 30, Gen Loss: 2.9114181995391846, Disc Loss: 1.3362135887145996

Epoch 31, Gen Loss: 3.035146951675415, Disc Loss: 1.347489833831787



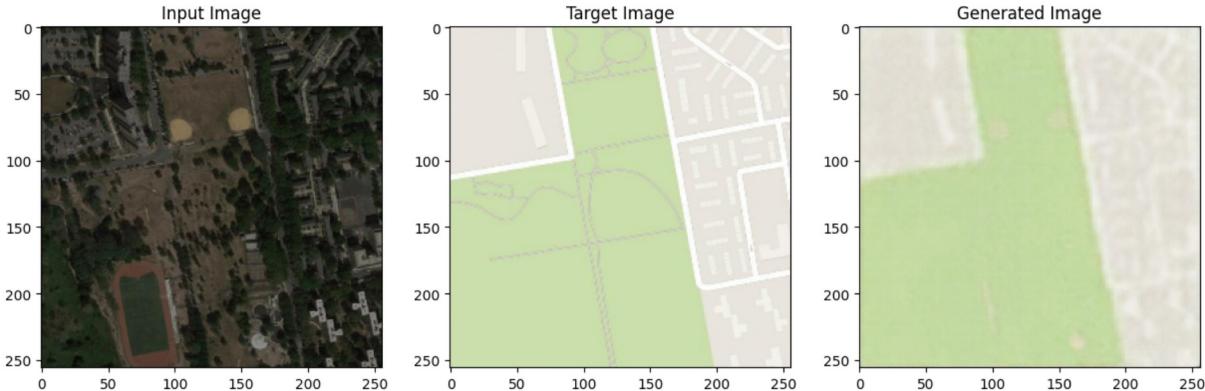
Epoch 32, Gen Loss: 2.849148750305176, Disc Loss: 1.3476085662841797
Epoch 33, Gen Loss: 3.102604866027832, Disc Loss: 1.3620562553405762
Epoch 34, Gen Loss: 2.8239264488220215, Disc Loss: 1.3500897884368896
Epoch 35, Gen Loss: 3.0522778034210205, Disc Loss: 1.354878544807434
Epoch 36, Gen Loss: 2.882420063018799, Disc Loss: 1.3450219631195068
Epoch 37, Gen Loss: 3.091399908065796, Disc Loss: 1.3399558067321777
Epoch 38, Gen Loss: 2.901998996734619, Disc Loss: 1.3384051322937012
Epoch 39, Gen Loss: 3.0306389331817627, Disc Loss: 1.3376420736312866
Epoch 40, Gen Loss: 2.829378604888916, Disc Loss: 1.3290611505508423
Epoch 41, Gen Loss: 2.9725608825683594, Disc Loss: 1.3359737396240234

Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers).



```
In [75]: generator.learning_rate = 3e-4  
train(data, 11)
```

Epoch 1, Gen Loss: 2.8238797187805176, Disc Loss: 1.3221399784088135



Epoch 2, Gen Loss: 2.9804351329803467, Disc Loss: 1.3290655612945557
Epoch 3, Gen Loss: 2.834095001220703, Disc Loss: 1.3256714344024658
Epoch 4, Gen Loss: 3.0166709423065186, Disc Loss: 1.3346805572509766
Epoch 5, Gen Loss: 2.7640466690063477, Disc Loss: 1.3460392951965332
Epoch 6, Gen Loss: 3.0190858840942383, Disc Loss: 1.3525316715240479
Epoch 7, Gen Loss: 2.703486919403076, Disc Loss: 1.3800827264785767
Epoch 8, Gen Loss: 3.062532901763916, Disc Loss: 1.3749088048934937
Epoch 9, Gen Loss: 2.714521884918213, Disc Loss: 1.3613440990447998
Epoch 10, Gen Loss: 2.9605817794799805, Disc Loss: 1.349210262298584
Epoch 11, Gen Loss: 2.8152570724487305, Disc Loss: 1.3453692197799683

Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers).



```
In [76]: generator.learning_rate = 4e-4  
train(data, 11)
```

Epoch 1, Gen Loss: 2.907069444656372, Disc Loss: 1.376901626586914



Epoch 2, Gen Loss: 2.852583408355713, Disc Loss: 1.3536725044250488

Epoch 3, Gen Loss: 2.872340679168701, Disc Loss: 1.3507187366485596

Epoch 4, Gen Loss: 2.790876626968384, Disc Loss: 1.3320685625076294

Epoch 5, Gen Loss: 2.8650662899017334, Disc Loss: 1.3426862955093384

Epoch 6, Gen Loss: 2.7621233463287354, Disc Loss: 1.3322107791900635

Epoch 7, Gen Loss: 2.89520525932312, Disc Loss: 1.328613519668579

Epoch 8, Gen Loss: 2.795985221862793, Disc Loss: 1.3239388465881348

Epoch 9, Gen Loss: 3.0289647579193115, Disc Loss: 1.340348482131958

Epoch 10, Gen Loss: 2.7841243743896484, Disc Loss: 1.3420042991638184

Epoch 11, Gen Loss: 3.102708339691162, Disc Loss: 1.36599600315094



```
In [77]: discriminator.learning_rate = 3e-4  
train(data, 11)
```

Epoch 1, Gen Loss: 2.71850848197937, Disc Loss: 1.367140531539917



Epoch 2, Gen Loss: 3.1464459896087646, Disc Loss: 1.369004726409912

Epoch 3, Gen Loss: 2.7422595024108887, Disc Loss: 1.3417245149612427

Epoch 4, Gen Loss: 2.9709856510162354, Disc Loss: 1.3350476026535034

Epoch 5, Gen Loss: 2.74575138092041, Disc Loss: 1.3169012069702148

Epoch 6, Gen Loss: 2.7861032485961914, Disc Loss: 1.3190014362335205

Epoch 7, Gen Loss: 2.751307964324951, Disc Loss: 1.3218963146209717

Epoch 8, Gen Loss: 2.76025390625, Disc Loss: 1.319526195526123

Epoch 9, Gen Loss: 2.7930381298065186, Disc Loss: 1.3436434268951416

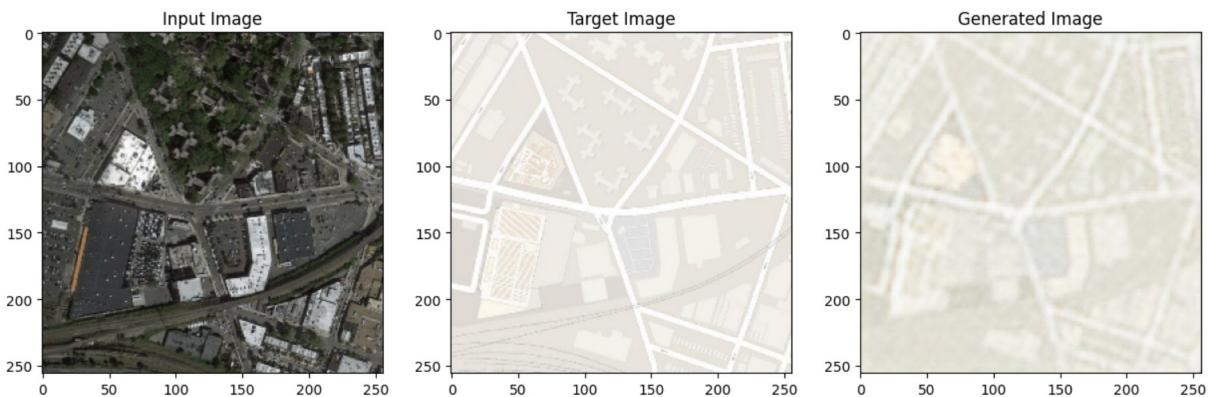
Epoch 10, Gen Loss: 2.808316707611084, Disc Loss: 1.3394168615341187

Epoch 11, Gen Loss: 2.795701503753662, Disc Loss: 1.3693630695343018



In [78]: `train(data, 21)`

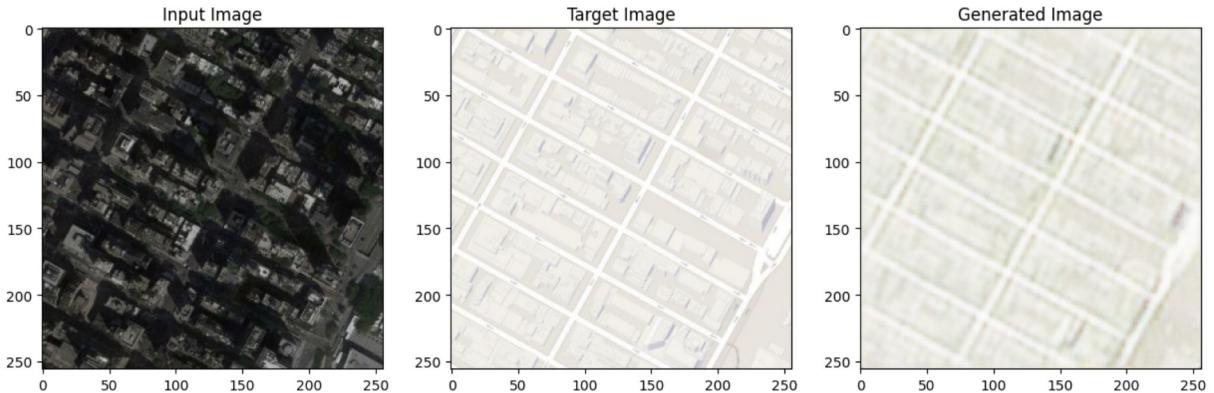
Epoch 1, Gen Loss: 2.8126208782196045, Disc Loss: 1.3619271516799927



Epoch 2, Gen Loss: 2.802135467529297, Disc Loss: 1.368411898612976
Epoch 3, Gen Loss: 2.768827199935913, Disc Loss: 1.3677220344543457
Epoch 4, Gen Loss: 2.8719332218170166, Disc Loss: 1.4037439823150635
Epoch 5, Gen Loss: 2.7568302154541016, Disc Loss: 1.4251267910003662
Epoch 6, Gen Loss: 2.9575576782226562, Disc Loss: 1.4036293029785156
Epoch 7, Gen Loss: 2.676023006439209, Disc Loss: 1.3484092950820923
Epoch 8, Gen Loss: 2.9404361248016357, Disc Loss: 1.3866112232208252
Epoch 9, Gen Loss: 2.60135817527771, Disc Loss: 1.414668321609497
Epoch 10, Gen Loss: 3.078502655029297, Disc Loss: 1.4569772481918335
Epoch 11, Gen Loss: 2.5542995929718018, Disc Loss: 1.3975244760513306



Epoch 12, Gen Loss: 2.8408775329589844, Disc Loss: 1.3491655588150024
Epoch 13, Gen Loss: 2.680039644241333, Disc Loss: 1.341081142425537
Epoch 14, Gen Loss: 2.7157628536224365, Disc Loss: 1.3383299112319946
Epoch 15, Gen Loss: 2.6844310760498047, Disc Loss: 1.3303483724594116
Epoch 16, Gen Loss: 2.682481527328491, Disc Loss: 1.3332655429840088
Epoch 17, Gen Loss: 2.686704397201538, Disc Loss: 1.3242018222808838
Epoch 18, Gen Loss: 2.704421043395996, Disc Loss: 1.3355131149291992
Epoch 19, Gen Loss: 2.7497053146362305, Disc Loss: 1.3180173635482788
Epoch 20, Gen Loss: 2.7509207725524902, Disc Loss: 1.3398581743240356
Epoch 21, Gen Loss: 2.787059783935547, Disc Loss: 1.3199357986450195

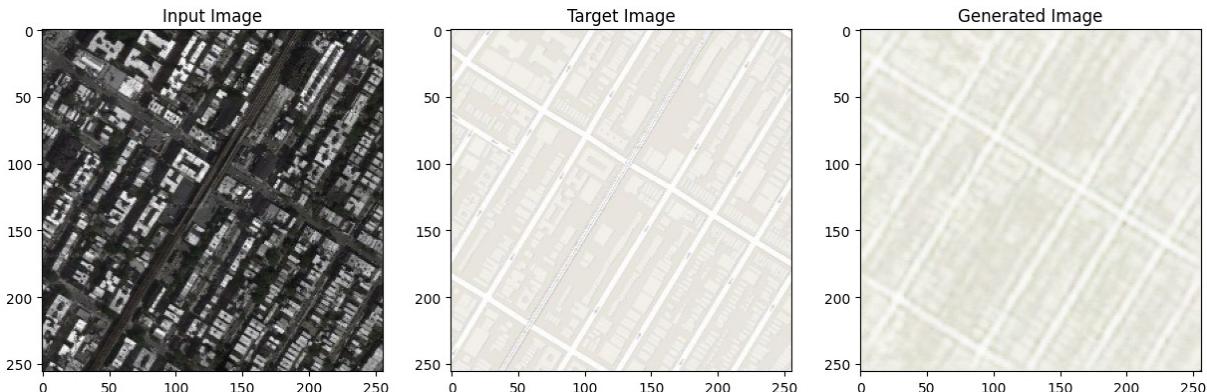


In [79]: `train(data, 41)`

Epoch 1, Gen Loss: 2.805370330810547, Disc Loss: 1.3354394435882568



Epoch 2, Gen Loss: 2.7553839683532715, Disc Loss: 1.328115701675415
Epoch 3, Gen Loss: 2.7706921100616455, Disc Loss: 1.333922266960144
Epoch 4, Gen Loss: 2.657832384109497, Disc Loss: 1.3378604650497437
Epoch 5, Gen Loss: 2.8333992958068848, Disc Loss: 1.359419345855713
Epoch 6, Gen Loss: 2.564467668533325, Disc Loss: 1.3738172054290771
Epoch 7, Gen Loss: 2.962024450302124, Disc Loss: 1.4207743406295776
Epoch 8, Gen Loss: 2.4946770668029785, Disc Loss: 1.4015843868255615
Epoch 9, Gen Loss: 2.867117166519165, Disc Loss: 1.3844027519226074
Epoch 10, Gen Loss: 2.5823471546173096, Disc Loss: 1.361936330795288
Epoch 11, Gen Loss: 2.7563564777374268, Disc Loss: 1.3558778762817383



Epoch 12, Gen Loss: 2.625767230987549, Disc Loss: 1.3489964008331299
Epoch 13, Gen Loss: 2.745650291442871, Disc Loss: 1.34184730052948
Epoch 14, Gen Loss: 2.629891872406006, Disc Loss: 1.3352715969085693
Epoch 15, Gen Loss: 2.7677297592163086, Disc Loss: 1.340085506439209
Epoch 16, Gen Loss: 2.614872932434082, Disc Loss: 1.333598256111145
Epoch 17, Gen Loss: 2.7396318912506104, Disc Loss: 1.3407278060913086
Epoch 18, Gen Loss: 2.5471315383911133, Disc Loss: 1.3410534858703613
Epoch 19, Gen Loss: 2.7296857833862305, Disc Loss: 1.349168300628662
Epoch 20, Gen Loss: 2.511603593826294, Disc Loss: 1.3513761758804321
Epoch 21, Gen Loss: 2.764533519744873, Disc Loss: 1.3616912364959717

Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers).



Epoch 22, Gen Loss: 2.497265100479126, Disc Loss: 1.3633265495300293

Epoch 23, Gen Loss: 2.797337532043457, Disc Loss: 1.3719816207885742

Epoch 24, Gen Loss: 2.5528416633605957, Disc Loss: 1.357858419418335

Epoch 25, Gen Loss: 2.7766201496124268, Disc Loss: 1.3604137897491455

Epoch 26, Gen Loss: 2.595667839050293, Disc Loss: 1.3381268978118896

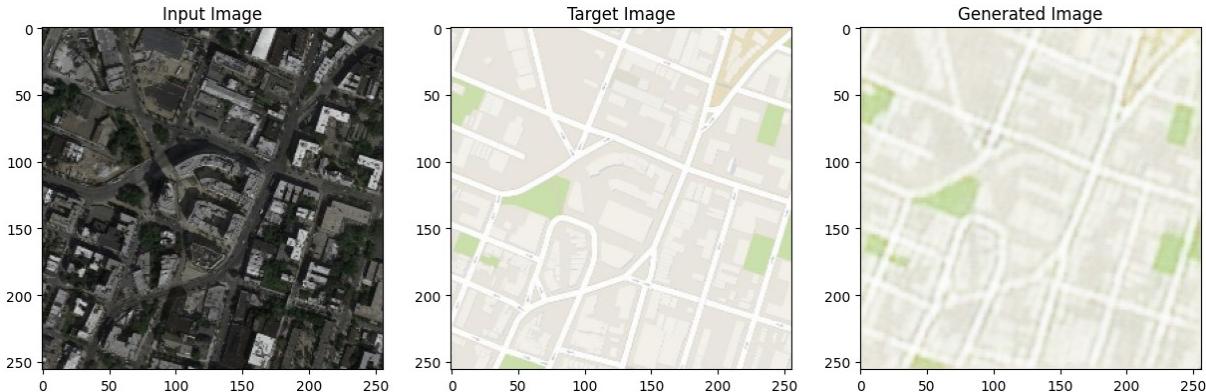
Epoch 27, Gen Loss: 2.720716953277588, Disc Loss: 1.3533673286437988

Epoch 28, Gen Loss: 2.6809215545654297, Disc Loss: 1.3349566459655762

Epoch 29, Gen Loss: 2.712698221206665, Disc Loss: 1.352043867111206

Epoch 30, Gen Loss: 2.724724769592285, Disc Loss: 1.3352282047271729

Epoch 31, Gen Loss: 2.7045626640319824, Disc Loss: 1.357330560684204



Epoch 32, Gen Loss: 2.8074915409088135, Disc Loss: 1.3413002490997314

Epoch 33, Gen Loss: 2.767451286315918, Disc Loss: 1.374819278717041

Epoch 34, Gen Loss: 3.03181791305542, Disc Loss: 1.3458342552185059

Epoch 35, Gen Loss: 2.9043924808502197, Disc Loss: 1.3660125732421875

Epoch 36, Gen Loss: 3.775266647338867, Disc Loss: 1.30167818069458

Epoch 37, Gen Loss: 3.923442840576172, Disc Loss: 1.3696867227554321

Epoch 38, Gen Loss: 5.709847927093506, Disc Loss: 1.1840324401855469

Epoch 39, Gen Loss: 5.933039665222168, Disc Loss: 1.4509377479553223

Epoch 40, Gen Loss: 6.852659225463867, Disc Loss: 1.569488525390625

Epoch 41, Gen Loss: 5.489511489868164, Disc Loss: 1.2253499031066895

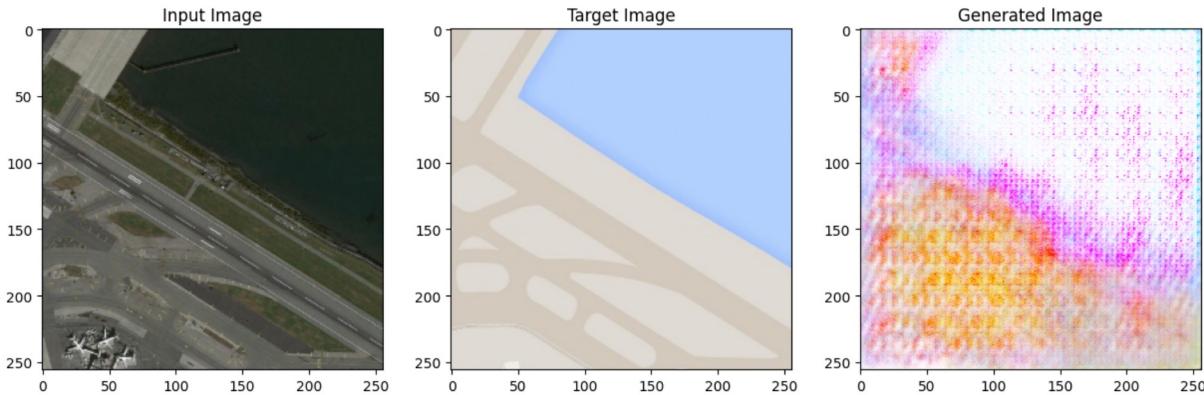


Generator EXPLDED

```
In [80]: train(data, 11)
```

Epoch 1, Gen Loss: 5.091287136077881, Disc Loss: 1.3296048641204834

Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers).



Epoch 2, Gen Loss: 5.456643104553223, Disc Loss: 1.56837797164917

Epoch 3, Gen Loss: 5.9257330894470215, Disc Loss: 1.298153042793274

Epoch 4, Gen Loss: 4.531567096710205, Disc Loss: 1.370591640472412

Epoch 5, Gen Loss: 4.844128608703613, Disc Loss: 1.3852447271347046

Epoch 6, Gen Loss: 4.880160331726074, Disc Loss: 1.2009007930755615

Epoch 7, Gen Loss: 4.006290435791016, Disc Loss: 1.1915569305419922

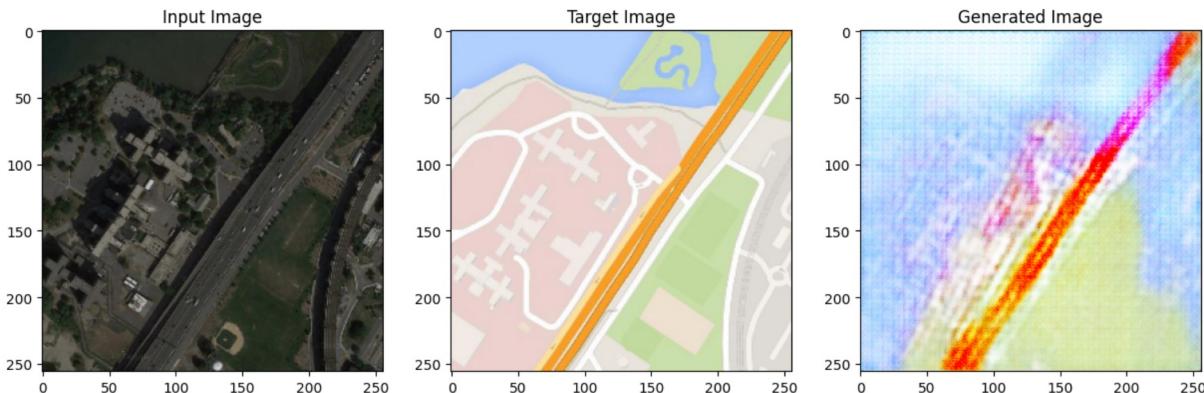
Epoch 8, Gen Loss: 4.07930326461792, Disc Loss: 1.2254670858383179

Epoch 9, Gen Loss: 3.7088570594787598, Disc Loss: 1.1889851093292236

Epoch 10, Gen Loss: 3.6602916717529297, Disc Loss: 1.2001758813858032

Epoch 11, Gen Loss: 3.383741855621338, Disc Loss: 1.2327601909637451

Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers).



Bring back saved model of generator and discriminator

```
In [84]: from tensorflow.keras.models import load_model
```

```
generator.model = load_model('Models\second_generator_model.keras')
discriminator.model = load_model("Models\second_discriminator_model.keras")
```

```
generator_optimizer = tf.keras.optimizers.Adam(2e-4, beta_1=0.5)
discriminator_optimizer = tf.keras.optimizers.Adam(2e-4, beta_1=0.5)
```

In [85]: `train(data, 1)`

Epoch 1, Gen Loss: 3.053156614303589, Disc Loss: 1.1078063249588013

