ArtGAN

April 8, 2023

1 ArtGAN: Art Style Transfer with Self-Attention Generative Adversarial Networks

1.0.1 Installations

[ ]:
```
pip install numpy opencv-python Pillow scikit-learn beautifulsoup4 requests␣
↪tensorflow tensorflow-addons matplotlib
```

1.1 Artist Image Dataset Creation

[ ]:
```
import os
import cv2
import requests
import numpy as np
from PIL import Image
from io import BytesIO
from bs4 import BeautifulSoup
from sklearn.model_selection import train_test_split

# Configuration
ARTIST_NAMES = ["vincent-van-gogh", "pablo-picasso"]
WIKIART_URL = "https://www.wikiart.org/en/{}/all-works/text-list"
IMAGE_SIZE = 256
TEST_SPLIT = 0.2

def preprocess_image(image_url, size):
    # Retrieve image from URL
    response = requests.get(image_url)
    image = Image.open(BytesIO(response.content))

    # Convert the image to RGB format
    image = image.convert('RGB')

    # Resize the image
    image = image.resize((size, size), Image.ANTIALIAS)

    # Convert the image to a NumPy array
```

```
    image = np.asarray(image)

    # Normalize the image to [-1, 1]
    image = (image - 127.5) / 127.5
    return image

def get_image_links(artist_name, page):
    image_links = []

    # Retrieve artist's paintings page
    artist_url = WIKIART_URL.format(artist_name) + f'?json=2&page={page}'
    response = requests.get(artist_url)

    if response.status_code == 200:
        response_json = response.json()

        # Extract image URLs from the response
        for item in response_json["Paintings"]:
            image_links.append(item["image"])

    return image_links

def load_dataset(artist_name, max_pages=10):
    images = []

    # Iterate over each page
    for page in range(1, max_pages + 1):
        print(f"Retrieving images from {artist_name} - Page {page}")
        image_links = get_image_links(artist_name, page)

        if not image_links:
            break

        # Iterate over each image link
        for image_url in image_links:
            try:
                print(f"Processing image: {image_url}")
                image = preprocess_image(image_url, IMAGE_SIZE)
                images.append(image)
            except:
                print(f"Error processing image: {image_url}")
                pass

    return np.array(images)

# Load datasets
print("Loading Van Gogh dataset...")
```

```
van_gogh_dataset = load_dataset("vincent-van-gogh")
print("Loading Picasso dataset...")
picasso_dataset = load_dataset("pablo-picasso")

# Split datasets
vg_train, vg_test = train_test_split(van_gogh_dataset, test_size=TEST_SPLIT,␣
 ↪random_state=42)
p_train, p_test = train_test_split(picasso_dataset, test_size=TEST_SPLIT,␣
 ↪random_state=42)

print("Van Gogh - Train:", vg_train.shape, "Test:", vg_test.shape)
print("Picasso - Train:", p_train.shape, "Test:", p_test.shape)
```

1.2 ArtGAN Architecture

[ ]:
```
import tensorflow as tf
from tensorflow.keras import layers

# Attention block for adding the self-attention mechanism
def attention_block(input_layer, filters):
    q = layers.Conv2D(filters, 1, padding='same')(input_layer)  # Query
    k = layers.Conv2D(filters, 1, padding='same')(input_layer)  # Key
    v = layers.Conv2D(filters, 1, padding='same')(input_layer)  # Value

    qk = layers.Multiply()([q, k])  # Element-wise multiplication of Query and␣
 ↪Key
    qk = layers.Activation('softmax')(qk)  # Apply softmax to obtain attention␣
 ↪weights

    attention = layers.Multiply()([v, qk])  # Element-wise multiplication of␣
 ↪Value and attention weights

    output = layers.Add()([input_layer, attention])  # Add input and attention␣
 ↪layers
    return output

# Define the generator network
def define_generator(is_attention, input_shape=(256, 256, 3)):
    input_layer = layers.Input(shape=input_shape)

    # Encoder
    enc1 = layers.Conv2D(64, 4, strides=2, padding='same')(input_layer)
    enc1 = layers.LeakyReLU(alpha=0.2)(enc1)

    enc2 = layers.Conv2D(128, 4, strides=2, padding='same')(enc1)
    enc2 = layers.BatchNormalization()(enc2)
```

```
    enc2 = layers.LeakyReLU(alpha=0.2)(enc2)

    enc3 = layers.Conv2D(256, 4, strides=2, padding='same')(enc2)
    enc3 = layers.BatchNormalization()(enc3)
    enc3 = layers.LeakyReLU(alpha=0.2)(enc3)

    # Bottleneck with attention mechanism
    bottleneck = layers.Conv2D(512, 4, strides=2, padding='same')(enc3)
    bottleneck = layers.BatchNormalization()(bottleneck)
    bottleneck = layers.LeakyReLU(alpha=0.2)(bottleneck)

    # Attention layer
    if is_attention:
      bottleneck = attention_block(bottleneck, 512)

    # Decoder
    dec1 = layers.Conv2DTranspose(256, 4, strides=2, padding='same')(bottleneck)
    dec1 = layers.BatchNormalization()(dec1)
    dec1 = layers.ReLU()(dec1)
    dec1 = layers.Concatenate()([dec1, enc3])

    dec2 = layers.Conv2DTranspose(128, 4, strides=2, padding='same')(dec1)
    dec2 = layers.BatchNormalization()(dec2)
    dec2 = layers.ReLU()(dec2)
    dec2 = layers.Concatenate()([dec2, enc2])

    dec3 = layers.Conv2DTranspose(64, 4, strides=2, padding='same')(dec2)
    dec3 = layers.BatchNormalization()(dec3)
    dec3 = layers.ReLU()(dec3)
    dec3 = layers.Concatenate()([dec3, enc1])

    output_layer = layers.Conv2DTranspose(3, 4, strides=2, padding='same',␣
 ↪activation='tanh')(dec3)

    return tf.keras.Model(inputs=input_layer, outputs=output_layer)

# Define the discriminator network
def define_discriminator(input_shape=(256, 256, 3)):
    input_layer = layers.Input(shape=input_shape)

    x = layers.Conv2D(64, 4, strides=2, padding='same')(input_layer)
    x = layers.LeakyReLU(alpha=0.2)(x)

    x = layers.Conv2D(128, 4, strides=2, padding='same')(x)
    x = layers.BatchNormalization()(x)
    x = layers.LeakyReLU(alpha=0.2)(x)
```

```
    x = layers.Conv2D(256, 4, strides=2, padding='same')(x)
    x = layers.BatchNormalization()(x)
    x = layers.LeakyReLU(alpha=0.2)(x)

    x = layers.Conv2D(512, 4, strides=2, padding='same')(x)
    x = layers.BatchNormalization()(x)
    x = layers.LeakyReLU(alpha=0.2)(x)

    x = layers.Conv2D(1, 4, padding='same')(x)  # Output layer

    return tf.keras.Model(inputs=input_layer, outputs=x)
```

1.2.1 ArtGAN Network Definition

[ ]:
```
# ArtGAN generator and discriminator networks
g_AB = define_generator(is_attention=True)
g_BA = define_generator(is_attention=True)

d_A = define_discriminator()
d_B = define_discriminator()
```

1.2.2 CycleGAN Network Definition

[ ]:
```
# ArtGAN generator and discriminator networks
g_AB_n = define_generator(is_attention=False)
g_BA_n = define_generator(is_attention=False)

d_A_n = define_discriminator()
d_B_n = define_discriminator()
```

1.3 Training the ArtGAN Model

1.3.1 Loss Function Definitions

[ ]:
```
import tensorflow_addons as tfa

# Loss functions
LAMBDA = 10  # Weight for cycle consistency loss
ALPHA = 0.5  # Weight for identity loss

# Binary cross-entropy loss for discriminator and generator loss functions
loss_object = tf.keras.losses.BinaryCrossentropy(from_logits=True)

# Discriminator loss function: it computes the average loss for real and␣
 ↪generated images
def discriminator_loss(real, generated):
    real_loss = loss_object(tf.ones_like(real), real)  # Loss for real images
```

```
    generated_loss = loss_object(tf.zeros_like(generated), generated)  # Loss␣
 ↪for generated images
    total_disc_loss = (real_loss + generated_loss) * 0.5  # Average loss
    return total_disc_loss

# Generator loss function: it aims to maximize the discriminator's error on␣
 ↪generated images
def generator_loss(generated):
    return loss_object(tf.ones_like(generated), generated)

# Cycle consistency loss function: it computes the mean absolute error between␣
 ↪the real and cycled images
def cycle_consistency_loss(real_image, cycled_image):
    return tf.reduce_mean(tf.abs(tf.cast(real_image, tf.float32) - tf.
 ↪cast(cycled_image, tf.float32)))

# Identity loss function: it computes the mean absolute error between the real␣
 ↪and same-domain generated images
def identity_loss(real_image, same_image):
    return tf.reduce_mean(tf.abs(tf.cast(real_image, tf.float32) - tf.
 ↪cast(same_image, tf.float32)))

# Optimizers
# Learning rate scheduler with exponential decay
lr_schedule = tf.keras.optimizers.schedules.ExponentialDecay(
    initial_learning_rate=2e-4,
    decay_steps=100000,
    decay_rate=0.96,
    staircase=True)
```

```
/usr/local/lib/python3.9/dist-
packages/tensorflow_addons/utils/tfa_eol_msg.py:23: UserWarning:

TensorFlow Addons (TFA) has ended development and introduction of new features.
TFA has entered a minimal maintenance and release mode until a planned end of
life in May 2024.
Please modify downstream libraries to take dependencies from other repositories
in our TensorFlow community (e.g. Keras, Keras-CV, and Keras-NLP).

For more information see: https://github.com/tensorflow/addons/issues/2807

  warnings.warn(
```

1.3.2 ArtGAN

ArtGAN Optimizer

```
[ ]: # Optimizers for generator and discriminator networks
all_trainable_variables = g_AB.trainable_variables + g_BA.trainable_variables +␣
 ↪d_A.trainable_variables + d_B.trainable_variables

generator_optimizer = tf.keras.optimizers.Adam(learning_rate=lr_schedule,␣
 ↪beta_1=0.5)
discriminator_optimizer = tf.keras.optimizers.Adam(learning_rate=lr_schedule,␣
 ↪beta_1=0.5)

# Build the optimizers with all trainable variables
generator_optimizer.build(all_trainable_variables)
discriminator_optimizer.build(all_trainable_variables)
```

Training

```
[ ]: import time
from IPython.display import clear_output

# Configuration
EPOCHS = 100  # Number of epochs for training
BATCH_SIZE = 1  # Batch size for training
BUFFER_SIZE = 1000  # Buffer size for shuffling the dataset

# Create TensorFlow datasets
train_A = tf.data.Dataset.from_tensor_slices(vg_train).shuffle(BUFFER_SIZE).
 ↪batch(BATCH_SIZE)
train_B = tf.data.Dataset.from_tensor_slices(p_train).shuffle(BUFFER_SIZE).
 ↪batch(BATCH_SIZE)

# Define the training step for one batch of real images from both domains
@tf.function
def train_step(real_A, real_B):
    with tf.GradientTape(persistent=True) as tape:
        # Generate fake images and cycle them back to the original domain
        fake_B = g_AB(real_A, training=True)
        cycled_A = g_BA(fake_B, training=True)
        fake_A = g_BA(real_B, training=True)
        cycled_B = g_AB(fake_A, training=True)

        # Generate same-domain images for identity loss
        same_A = g_BA(real_A, training=True)
        same_B = g_AB(real_B, training=True)

        # Compute the discriminator outputs for real and fake images
        disc_real_A = d_A(real_A, training=True)
        disc_real_B = d_B(real_B, training=True)
        disc_fake_A = d_A(fake_A, training=True)
```

```
        disc_fake_B = d_B(fake_B, training=True)

        # Compute the generator and discriminator losses
        gen_AB_loss = generator_loss(disc_fake_B)
        gen_BA_loss = generator_loss(disc_fake_A)
        total_cycle_loss = cycle_consistency_loss(real_A, cycled_A) +␣
↪cycle_consistency_loss(real_B, cycled_B)
        total_gen_AB_loss = gen_AB_loss + LAMBDA * total_cycle_loss + ALPHA *␣
↪identity_loss(real_B, same_B)
        total_gen_BA_loss = gen_BA_loss + LAMBDA * total_cycle_loss + ALPHA *␣
↪identity_loss(real_A, same_A)
        disc_A_loss = discriminator_loss(disc_real_A, disc_fake_A)
        disc_B_loss = discriminator_loss(disc_real_B, disc_fake_B)

    # Compute the gradients for the generator and discriminator networks
    generator_AB_gradients = tape.gradient(total_gen_AB_loss, g_AB.
↪trainable_variables)
    generator_BA_gradients = tape.gradient(total_gen_BA_loss, g_BA.
↪trainable_variables)
    discriminator_A_gradients = tape.gradient(disc_A_loss, d_A.
↪trainable_variables)
    discriminator_B_gradients = tape.gradient(disc_B_loss, d_B.
↪trainable_variables)

    # Apply the gradients using the optimizers
    generator_optimizer.apply_gradients(zip(generator_AB_gradients, g_AB.
↪trainable_variables))
    generator_optimizer.apply_gradients(zip(generator_BA_gradients, g_BA.
↪trainable_variables))
    discriminator_optimizer.apply_gradients(zip(discriminator_A_gradients, d_A.
↪trainable_variables))
    discriminator_optimizer.apply_gradients(zip(discriminator_B_gradients, d_B.
↪trainable_variables))

    return gen_AB_loss, gen_BA_loss, disc_A_loss, disc_B_loss

# Define the training loop for multiple epochs
def train(epochs):
    for epoch in range(epochs):
        start = time.time()

        n = 0
        gen_AB_losses = []
        gen_BA_losses = []
        disc_A_losses = []
        disc_B_losses = []
```

```
        # Iterate through the zipped dataset and perform a training step for␣
↪each batch
        for image_A, image_B in tf.data.Dataset.zip((train_A, train_B)):
            gen_AB_loss, gen_BA_loss, disc_A_loss, disc_B_loss =␣
↪train_step(image_A, image_B)
            gen_AB_losses.append(gen_AB_loss)
            gen_BA_losses.append(gen_BA_loss)
            disc_A_losses.append(disc_A_loss)
            disc_B_losses.append(disc_B_loss)

            if n % 10 == 0:
                print(".", end="")
            n += 1

        # Compute average losses for the current epoch
        avg_gen_AB_loss = tf.reduce_mean(gen_AB_losses)
        avg_gen_BA_loss = tf.reduce_mean(gen_BA_losses)
        avg_disc_A_loss = tf.reduce_mean(disc_A_losses)
        avg_disc_B_loss = tf.reduce_mean(disc_B_losses)

        # Logging
        print("Time taken for epoch {} is {} sec".format(epoch + 1, time.time()␣
↪- start))
        print("Generator AB loss: {:.4f}, Generator BA loss: {:.4f}".
↪format(avg_gen_AB_loss, avg_gen_BA_loss))
        print("Discriminator A loss: {:.4f}, Discriminator B loss: {:.4f}\n".
↪format(avg_disc_A_loss, avg_disc_B_loss))

        # if epoch % 10 == 0:
        #     for real_A, real_B in zip(vg_test, p_test):
        #         # Expand the dimensions of the input images to include the␣
↪batch size
        #         real_A = tf.expand_dims(real_A, axis=0)
        #         real_B = tf.expand_dims(real_B, axis=0)
        #         generated_B = g_AB(real_A, training=False)
        #         generated_A = g_BA(real_B, training=False)
        #         display_images(real_A, generated_B, real_B, generated_A)
        #         break  # Show only one batch of images


    # plot_loss_graphs(np.arange(1, EPOCHS), gen_AB_losses, gen_BA_losses,␣
↪disc_A_losses, disc_B_losses)


# Train the model
```

```
train(EPOCHS)
```

```
…Time taken for epoch 1 is
42.90241241455078 sec
Generator AB loss: 0.7859, Generator BA loss: 0.7892
Discriminator A loss: 0.6974, Discriminator B loss: 0.7017

…Time taken for epoch 2 is
15.394558191299438 sec
Generator AB loss: 0.7625, Generator BA loss: 0.7843
Discriminator A loss: 0.6776, Discriminator B loss: 0.6865

…Time taken for epoch 3 is
15.138753175735474 sec
Generator AB loss: 0.7708, Generator BA loss: 0.8236
Discriminator A loss: 0.6609, Discriminator B loss: 0.6802

…Time taken for epoch 4 is
15.16649317741394 sec
Generator AB loss: 0.7831, Generator BA loss: 0.8484
Discriminator A loss: 0.6472, Discriminator B loss: 0.6752

…Time taken for epoch 5 is
15.071443796157837 sec
Generator AB loss: 0.8038, Generator BA loss: 0.8256
Discriminator A loss: 0.6538, Discriminator B loss: 0.6712

…Time taken for epoch 6 is
15.114604234695435 sec
Generator AB loss: 0.8027, Generator BA loss: 0.8771
Discriminator A loss: 0.6370, Discriminator B loss: 0.6674

…Time taken for epoch 7 is
15.035118103027344 sec
Generator AB loss: 0.8207, Generator BA loss: 0.9115
Discriminator A loss: 0.6216, Discriminator B loss: 0.6617

…Time taken for epoch 8 is
15.070462703704834 sec
Generator AB loss: 0.8241, Generator BA loss: 0.8996
Discriminator A loss: 0.6267, Discriminator B loss: 0.6596

…Time taken for epoch 9 is
15.248525381088257 sec
Generator AB loss: 0.8384, Generator BA loss: 0.9262
Discriminator A loss: 0.6143, Discriminator B loss: 0.6544
```

```
…Time taken for epoch 10 is
15.231810092926025 sec
Generator AB loss: 0.8241, Generator BA loss: 0.9374
Discriminator A loss: 0.6164, Discriminator B loss: 0.6625

…Time taken for epoch 11 is
15.307421445846558 sec
Generator AB loss: 0.8379, Generator BA loss: 0.9552
Discriminator A loss: 0.6074, Discriminator B loss: 0.6537

…Time taken for epoch 12 is
15.387251615524292 sec
Generator AB loss: 0.8386, Generator BA loss: 0.9518
Discriminator A loss: 0.6123, Discriminator B loss: 0.6534

…Time taken for epoch 13 is
15.734796285629272 sec
Generator AB loss: 0.8349, Generator BA loss: 0.9279
Discriminator A loss: 0.6216, Discriminator B loss: 0.6571

…Time taken for epoch 14 is
15.552163124084473 sec
Generator AB loss: 0.8426, Generator BA loss: 0.9626
Discriminator A loss: 0.6025, Discriminator B loss: 0.6500

…Time taken for epoch 15 is
15.510040521621704 sec
Generator AB loss: 0.8552, Generator BA loss: 1.0013
Discriminator A loss: 0.5884, Discriminator B loss: 0.6485

…Time taken for epoch 16 is
15.577072858810425 sec
Generator AB loss: 0.8566, Generator BA loss: 0.9793
Discriminator A loss: 0.6044, Discriminator B loss: 0.6433

…Time taken for epoch 17 is
15.474993705749512 sec
Generator AB loss: 0.8575, Generator BA loss: 0.9567
Discriminator A loss: 0.6079, Discriminator B loss: 0.6445

…Time taken for epoch 18 is
15.558225870132446 sec
Generator AB loss: 0.8720, Generator BA loss: 0.9820
Discriminator A loss: 0.6022, Discriminator B loss: 0.6369

…Time taken for epoch 19 is
15.530194282531738 sec
Generator AB loss: 0.8741, Generator BA loss: 0.9759
```

```
Discriminator A loss: 0.6053, Discriminator B loss: 0.6391

…Time taken for epoch 20 is
15.307716608047485 sec
Generator AB loss: 0.8914, Generator BA loss: 0.9889
Discriminator A loss: 0.5978, Discriminator B loss: 0.6294

…Time taken for epoch 21 is
15.435723304748535 sec
Generator AB loss: 0.8951, Generator BA loss: 1.0127
Discriminator A loss: 0.5920, Discriminator B loss: 0.6272

…Time taken for epoch 22 is
15.326812505722046 sec
Generator AB loss: 0.8900, Generator BA loss: 0.9705
Discriminator A loss: 0.6064, Discriminator B loss: 0.6308

…Time taken for epoch 23 is
15.256537914276123 sec
Generator AB loss: 0.9070, Generator BA loss: 1.0131
Discriminator A loss: 0.5855, Discriminator B loss: 0.6248

…Time taken for epoch 24 is
15.164127349853516 sec
Generator AB loss: 0.9036, Generator BA loss: 1.0093
Discriminator A loss: 0.5871, Discriminator B loss: 0.6182

…Time taken for epoch 25 is
15.25506043434143 sec
Generator AB loss: 0.9268, Generator BA loss: 1.0114
Discriminator A loss: 0.5870, Discriminator B loss: 0.6138

…Time taken for epoch 26 is
15.30364465713501 sec
Generator AB loss: 0.9421, Generator BA loss: 1.0204
Discriminator A loss: 0.5871, Discriminator B loss: 0.6061

…Time taken for epoch 27 is
15.108559608459473 sec
Generator AB loss: 0.9547, Generator BA loss: 0.9905
Discriminator A loss: 0.6016, Discriminator B loss: 0.6044

…Time taken for epoch 28 is
15.193425178527832 sec
Generator AB loss: 0.9637, Generator BA loss: 1.0093
Discriminator A loss: 0.5871, Discriminator B loss: 0.5990

…Time taken for epoch 29 is
```

```
15.298278570175171 sec
Generator AB loss: 0.9889, Generator BA loss: 1.0146
Discriminator A loss: 0.5859, Discriminator B loss: 0.5902

…Time taken for epoch 30 is
15.25620150566101 sec
Generator AB loss: 1.0056, Generator BA loss: 1.0627
Discriminator A loss: 0.5689, Discriminator B loss: 0.5789

…Time taken for epoch 31 is
15.204922676086426 sec
Generator AB loss: 1.0183, Generator BA loss: 1.0653
Discriminator A loss: 0.5784, Discriminator B loss: 0.5779

…Time taken for epoch 32 is
15.262465476989746 sec
Generator AB loss: 1.0541, Generator BA loss: 1.0467
Discriminator A loss: 0.5783, Discriminator B loss: 0.5715

…Time taken for epoch 33 is
15.230950832366943 sec
Generator AB loss: 1.0775, Generator BA loss: 1.0438
Discriminator A loss: 0.5813, Discriminator B loss: 0.5617

…Time taken for epoch 34 is
15.595855712890625 sec
Generator AB loss: 1.0765, Generator BA loss: 1.0498
Discriminator A loss: 0.5750, Discriminator B loss: 0.5523

…Time taken for epoch 35 is
15.29367446899414 sec
Generator AB loss: 1.1048, Generator BA loss: 1.0754
Discriminator A loss: 0.5665, Discriminator B loss: 0.5513

…Time taken for epoch 36 is
15.279838800430298 sec
Generator AB loss: 1.1283, Generator BA loss: 1.0927
Discriminator A loss: 0.5642, Discriminator B loss: 0.5386

…Time taken for epoch 37 is
15.40403437614441 sec
Generator AB loss: 1.1452, Generator BA loss: 1.0810
Discriminator A loss: 0.5666, Discriminator B loss: 0.5351

…Time taken for epoch 38 is
15.684983968734741 sec
Generator AB loss: 1.1710, Generator BA loss: 1.0889
Discriminator A loss: 0.5556, Discriminator B loss: 0.5291
```

```
…Time taken for epoch 39 is
15.518184900283813 sec
Generator AB loss: 1.1972, Generator BA loss: 1.0734
Discriminator A loss: 0.5837, Discriminator B loss: 0.5241

…Time taken for epoch 40 is
15.252344369888306 sec
Generator AB loss: 1.2177, Generator BA loss: 1.0724
Discriminator A loss: 0.5603, Discriminator B loss: 0.5127

…Time taken for epoch 41 is
15.377226829528809 sec
Generator AB loss: 1.2494, Generator BA loss: 1.1323
Discriminator A loss: 0.5430, Discriminator B loss: 0.5004

…Time taken for epoch 42 is
15.137111902236938 sec
Generator AB loss: 1.2811, Generator BA loss: 1.1679
Discriminator A loss: 0.5424, Discriminator B loss: 0.4955

…Time taken for epoch 43 is
15.218085527420044 sec
Generator AB loss: 1.2747, Generator BA loss: 1.1500
Discriminator A loss: 0.5425, Discriminator B loss: 0.4971

…Time taken for epoch 44 is
15.26904559135437 sec
Generator AB loss: 1.3126, Generator BA loss: 1.1517
Discriminator A loss: 0.5460, Discriminator B loss: 0.4917

…Time taken for epoch 45 is
15.17368769645691 sec
Generator AB loss: 1.3376, Generator BA loss: 1.1374
Discriminator A loss: 0.5426, Discriminator B loss: 0.4818

…Time taken for epoch 46 is
15.090906381607056 sec
Generator AB loss: 1.3683, Generator BA loss: 1.1534
Discriminator A loss: 0.5384, Discriminator B loss: 0.4718

…Time taken for epoch 47 is
15.252392292022705 sec
Generator AB loss: 1.3940, Generator BA loss: 1.1761
Discriminator A loss: 0.5345, Discriminator B loss: 0.4586

…Time taken for epoch 48 is
15.130277872085571 sec
```

```
Generator AB loss: 1.4148, Generator BA loss: 1.1933
Discriminator A loss: 0.5345, Discriminator B loss: 0.4674

…Time taken for epoch 49 is
15.53298044204712 sec
Generator AB loss: 1.4375, Generator BA loss: 1.2389
Discriminator A loss: 0.5237, Discriminator B loss: 0.4436

…Time taken for epoch 50 is
15.409507274627686 sec
Generator AB loss: 1.4641, Generator BA loss: 1.2616
Discriminator A loss: 0.5049, Discriminator B loss: 0.4436

…Time taken for epoch 51 is
15.526890277862549 sec
Generator AB loss: 1.4973, Generator BA loss: 1.2283
Discriminator A loss: 0.5213, Discriminator B loss: 0.4330

…Time taken for epoch 52 is
15.644936323165894 sec
Generator AB loss: 1.5564, Generator BA loss: 1.2898
Discriminator A loss: 0.5007, Discriminator B loss: 0.4155

…Time taken for epoch 53 is
15.58890962600708 sec
Generator AB loss: 1.6012, Generator BA loss: 1.2737
Discriminator A loss: 0.5123, Discriminator B loss: 0.4184

…Time taken for epoch 54 is
15.53183388710022 sec
Generator AB loss: 1.5943, Generator BA loss: 1.2433
Discriminator A loss: 0.5155, Discriminator B loss: 0.4270

…Time taken for epoch 55 is
15.53901219367981 sec
Generator AB loss: 1.6207, Generator BA loss: 1.2621
Discriminator A loss: 0.5193, Discriminator B loss: 0.4067

…Time taken for epoch 56 is
15.512933731079102 sec
Generator AB loss: 1.6656, Generator BA loss: 1.2437
Discriminator A loss: 0.5159, Discriminator B loss: 0.4018

…Time taken for epoch 57 is
15.500202417373657 sec
Generator AB loss: 1.6797, Generator BA loss: 1.2623
Discriminator A loss: 0.5145, Discriminator B loss: 0.4024
```

```
…Time taken for epoch 58 is
15.262176036834717 sec
Generator AB loss: 1.6898, Generator BA loss: 1.2983
Discriminator A loss: 0.5042, Discriminator B loss: 0.3950

…Time taken for epoch 59 is
15.39287519454956 sec
Generator AB loss: 1.7333, Generator BA loss: 1.2587
Discriminator A loss: 0.5137, Discriminator B loss: 0.3858

…Time taken for epoch 60 is
15.522581577301025 sec
Generator AB loss: 1.7535, Generator BA loss: 1.2949
Discriminator A loss: 0.5114, Discriminator B loss: 0.3857

…Time taken for epoch 61 is
15.714993000030518 sec
Generator AB loss: 1.7442, Generator BA loss: 1.2798
Discriminator A loss: 0.5009, Discriminator B loss: 0.3986

…Time taken for epoch 62 is
15.772754907608032 sec
Generator AB loss: 1.7782, Generator BA loss: 1.2796
Discriminator A loss: 0.5053, Discriminator B loss: 0.3690

…Time taken for epoch 63 is
15.477714538574219 sec
Generator AB loss: 1.8379, Generator BA loss: 1.3217
Discriminator A loss: 0.5066, Discriminator B loss: 0.3883

…Time taken for epoch 64 is
15.467346429824829 sec
Generator AB loss: 1.8480, Generator BA loss: 1.3156
Discriminator A loss: 0.4896, Discriminator B loss: 0.3634

…Time taken for epoch 65 is
15.155029296875 sec
Generator AB loss: 1.9819, Generator BA loss: 1.3740
Discriminator A loss: 0.4695, Discriminator B loss: 0.3297

…Time taken for epoch 66 is
14.918633699417114 sec
Generator AB loss: 1.9405, Generator BA loss: 1.3458
Discriminator A loss: 0.4928, Discriminator B loss: 0.3515

…Time taken for epoch 67 is
14.96434736251831 sec
Generator AB loss: 1.9302, Generator BA loss: 1.3149
```

```
Discriminator A loss: 0.5044, Discriminator B loss: 0.3598

…Time taken for epoch 68 is
15.170149564743042 sec
Generator AB loss: 1.9582, Generator BA loss: 1.3630
Discriminator A loss: 0.4752, Discriminator B loss: 0.3618

…Time taken for epoch 69 is
15.296250820159912 sec
Generator AB loss: 1.9147, Generator BA loss: 1.3479
Discriminator A loss: 0.4933, Discriminator B loss: 0.3669

…Time taken for epoch 70 is
15.00644040107727 sec
Generator AB loss: 1.9824, Generator BA loss: 1.3920
Discriminator A loss: 0.4806, Discriminator B loss: 0.3427

…Time taken for epoch 71 is
14.993796348571777 sec
Generator AB loss: 1.9736, Generator BA loss: 1.3326
Discriminator A loss: 0.5020, Discriminator B loss: 0.3489

…Time taken for epoch 72 is
15.009693145751953 sec
Generator AB loss: 2.0113, Generator BA loss: 1.3546
Discriminator A loss: 0.4804, Discriminator B loss: 0.3423

…Time taken for epoch 73 is
14.866349697113037 sec
Generator AB loss: 2.0237, Generator BA loss: 1.3920
Discriminator A loss: 0.4748, Discriminator B loss: 0.3528

…Time taken for epoch 74 is
14.85451865196228 sec
Generator AB loss: 2.0262, Generator BA loss: 1.3941
Discriminator A loss: 0.4759, Discriminator B loss: 0.3447

…Time taken for epoch 75 is
15.216313600540161 sec
Generator AB loss: 2.0630, Generator BA loss: 1.4176
Discriminator A loss: 0.4779, Discriminator B loss: 0.3293

…Time taken for epoch 76 is
15.236567258834839 sec
Generator AB loss: 2.0206, Generator BA loss: 1.3559
Discriminator A loss: 0.4932, Discriminator B loss: 0.3546

…Time taken for epoch 77 is
```

```
15.277345657348633 sec
Generator AB loss: 2.0926, Generator BA loss: 1.4196
Discriminator A loss: 0.4763, Discriminator B loss: 0.3373

…Time taken for epoch 78 is
15.303292751312256 sec
Generator AB loss: 2.1042, Generator BA loss: 1.4410
Discriminator A loss: 0.4559, Discriminator B loss: 0.3274

…Time taken for epoch 79 is
15.338149070739746 sec
Generator AB loss: 2.0774, Generator BA loss: 1.4341
Discriminator A loss: 0.4737, Discriminator B loss: 0.3524

…Time taken for epoch 80 is
15.349163055419922 sec
Generator AB loss: 2.1177, Generator BA loss: 1.4733
Discriminator A loss: 0.4631, Discriminator B loss: 0.3414

…Time taken for epoch 81 is
15.323324203491211 sec
Generator AB loss: 2.1386, Generator BA loss: 1.4287
Discriminator A loss: 0.4647, Discriminator B loss: 0.3406

…Time taken for epoch 82 is
15.226031064987183 sec
Generator AB loss: 2.1252, Generator BA loss: 1.4484
Discriminator A loss: 0.4722, Discriminator B loss: 0.3247

…Time taken for epoch 83 is
15.390339612960815 sec
Generator AB loss: 2.2709, Generator BA loss: 1.5670
Discriminator A loss: 0.4278, Discriminator B loss: 0.3025

…Time taken for epoch 84 is
15.303881406784058 sec
Generator AB loss: 2.2029, Generator BA loss: 1.4670
Discriminator A loss: 0.4772, Discriminator B loss: 0.3336

…Time taken for epoch 85 is
15.326297998428345 sec
Generator AB loss: 2.1517, Generator BA loss: 1.4757
Discriminator A loss: 0.4554, Discriminator B loss: 0.3356

…Time taken for epoch 86 is
15.202396154403687 sec
Generator AB loss: 2.2000, Generator BA loss: 1.5512
Discriminator A loss: 0.4374, Discriminator B loss: 0.3326
```

```
…Time taken for epoch 87 is
15.318751573562622 sec
Generator AB loss: 2.1272, Generator BA loss: 1.5122
Discriminator A loss: 0.4453, Discriminator B loss: 0.3661

…Time taken for epoch 88 is
14.808402061462402 sec
Generator AB loss: 2.1674, Generator BA loss: 1.5609
Discriminator A loss: 0.4312, Discriminator B loss: 0.3422

…Time taken for epoch 89 is
14.936737298965454 sec
Generator AB loss: 2.3217, Generator BA loss: 1.6332
Discriminator A loss: 0.4222, Discriminator B loss: 0.2879

…Time taken for epoch 90 is
14.995625019073486 sec
Generator AB loss: 2.3276, Generator BA loss: 1.5849
Discriminator A loss: 0.4466, Discriminator B loss: 0.3055

…Time taken for epoch 91 is
14.865683794021606 sec
Generator AB loss: 2.2271, Generator BA loss: 1.5594
Discriminator A loss: 0.4422, Discriminator B loss: 0.3369

…Time taken for epoch 92 is
14.976512432098389 sec
Generator AB loss: 2.2255, Generator BA loss: 1.5987
Discriminator A loss: 0.4171, Discriminator B loss: 0.3289

…Time taken for epoch 93 is
15.076410055160522 sec
Generator AB loss: 2.2577, Generator BA loss: 1.7287
Discriminator A loss: 0.3920, Discriminator B loss: 0.3202

…Time taken for epoch 94 is
15.229826927185059 sec
Generator AB loss: 2.2584, Generator BA loss: 1.6496
Discriminator A loss: 0.4347, Discriminator B loss: 0.3384

…Time taken for epoch 95 is
15.105092525482178 sec
Generator AB loss: 2.2975, Generator BA loss: 1.6475
Discriminator A loss: 0.4131, Discriminator B loss: 0.3230

…Time taken for epoch 96 is
15.088930130004883 sec
```

```
Generator AB loss: 2.2099, Generator BA loss: 1.6679
Discriminator A loss: 0.4209, Discriminator B loss: 0.3469

…Time taken for epoch 97 is
15.43331265449524 sec
Generator AB loss: 2.2456, Generator BA loss: 1.6226
Discriminator A loss: 0.4320, Discriminator B loss: 0.3374

…Time taken for epoch 98 is
14.996917009353638 sec
Generator AB loss: 2.2926, Generator BA loss: 1.6454
Discriminator A loss: 0.4168, Discriminator B loss: 0.3060

…Time taken for epoch 99 is
14.957719087600708 sec
Generator AB loss: 2.2545, Generator BA loss: 1.6825
Discriminator A loss: 0.4148, Discriminator B loss: 0.3275

…Time taken for epoch 100 is
14.919346570968628 sec
Generator AB loss: 2.2271, Generator BA loss: 1.6342
Discriminator A loss: 0.4206, Discriminator B loss: 0.3533
```

1.3.3 CycleGAN

CycleGAN Optimizer

[ ]:
```
# Optimizers for generator and discriminator networks
all_trainable_variables_n = g_AB_n.trainable_variables + g_BA_n.
 ↪trainable_variables + d_A_n.trainable_variables + d_B_n.trainable_variables

generator_optimizer_n = tf.keras.optimizers.Adam(learning_rate=lr_schedule,␣
 ↪beta_1=0.5)
discriminator_optimizer_n = tf.keras.optimizers.Adam(learning_rate=lr_schedule,␣
 ↪beta_1=0.5)

# Build the optimizers with all trainable variables
generator_optimizer_n.build(all_trainable_variables_n)
discriminator_optimizer_n.build(all_trainable_variables_n)
```

Training

[ ]:
```
import time
from IPython.display import clear_output

# Configuration
EPOCHS = 100  # Number of epochs for training
```

```
BATCH_SIZE = 1  # Batch size for training
BUFFER_SIZE = 1000  # Buffer size for shuffling the dataset

# Create TensorFlow datasets
train_A = tf.data.Dataset.from_tensor_slices(vg_train).shuffle(BUFFER_SIZE).
 ↪batch(BATCH_SIZE)
train_B = tf.data.Dataset.from_tensor_slices(p_train).shuffle(BUFFER_SIZE).
 ↪batch(BATCH_SIZE)

# Define the training step for one batch of real images from both domains
@tf.function
def train_step(real_A, real_B):
    with tf.GradientTape(persistent=True) as tape:
        # Generate fake images and cycle them back to the original domain
        fake_B = g_AB_n(real_A, training=True)
        cycled_A = g_BA_n(fake_B, training=True)
        fake_A = g_BA_n(real_B, training=True)
        cycled_B = g_AB_n(fake_A, training=True)

        # Generate same-domain images for identity loss
        same_A = g_BA_n(real_A, training=True)
        same_B = g_AB_n(real_B, training=True)

        # Compute the discriminator outputs for real and fake images
        disc_real_A = d_A_n(real_A, training=True)
        disc_real_B = d_B_n(real_B, training=True)
        disc_fake_A = d_A_n(fake_A, training=True)
        disc_fake_B = d_B_n(fake_B, training=True)

        # Compute the generator and discriminator losses
        gen_AB_loss = generator_loss(disc_fake_B)
        gen_BA_loss = generator_loss(disc_fake_A)
        total_cycle_loss = cycle_consistency_loss(real_A, cycled_A) +␣
 ↪cycle_consistency_loss(real_B, cycled_B)
        total_gen_AB_loss = gen_AB_loss + LAMBDA * total_cycle_loss + ALPHA *␣
 ↪identity_loss(real_B, same_B)
        total_gen_BA_loss = gen_BA_loss + LAMBDA * total_cycle_loss + ALPHA *␣
 ↪identity_loss(real_A, same_A)
        disc_A_loss = discriminator_loss(disc_real_A, disc_fake_A)
        disc_B_loss = discriminator_loss(disc_real_B, disc_fake_B)

    # Compute the gradients for the generator and discriminator networks
    generator_AB_gradients = tape.gradient(total_gen_AB_loss, g_AB_n.
 ↪trainable_variables)
    generator_BA_gradients = tape.gradient(total_gen_BA_loss, g_BA_n.
 ↪trainable_variables)
```

```
    discriminator_A_gradients = tape.gradient(disc_A_loss, d_A_n.
↪trainable_variables)
    discriminator_B_gradients = tape.gradient(disc_B_loss, d_B_n.
↪trainable_variables)

    # Apply the gradients using the optimizers
    generator_optimizer_n.apply_gradients(zip(generator_AB_gradients, g_AB_n.
↪trainable_variables))
    generator_optimizer_n.apply_gradients(zip(generator_BA_gradients, g_BA_n.
↪trainable_variables))
    discriminator_optimizer_n.apply_gradients(zip(discriminator_A_gradients,␣
↪d_A_n.trainable_variables))
    discriminator_optimizer_n.apply_gradients(zip(discriminator_B_gradients,␣
↪d_B_n.trainable_variables))

    return gen_AB_loss, gen_BA_loss, disc_A_loss, disc_B_loss

# Define the training loop for multiple epochs
def train(epochs):
    for epoch in range(epochs):
        start = time.time()

        n = 0
        gen_AB_losses = []
        gen_BA_losses = []
        disc_A_losses = []
        disc_B_losses = []

        # Iterate through the zipped dataset and perform a training step for␣
↪each batch
        for image_A, image_B in tf.data.Dataset.zip((train_A, train_B)):
            gen_AB_loss, gen_BA_loss, disc_A_loss, disc_B_loss =␣
↪train_step(image_A, image_B)
            gen_AB_losses.append(gen_AB_loss)
            gen_BA_losses.append(gen_BA_loss)
            disc_A_losses.append(disc_A_loss)
            disc_B_losses.append(disc_B_loss)

            if n % 10 == 0:
                print(".", end="")
            n += 1

        # Compute average losses for the current epoch
        avg_gen_AB_loss = tf.reduce_mean(gen_AB_losses)
        avg_gen_BA_loss = tf.reduce_mean(gen_BA_losses)
        avg_disc_A_loss = tf.reduce_mean(disc_A_losses)
```

```
        avg_disc_B_loss = tf.reduce_mean(disc_B_losses)

        # Logging
        print("Time taken for epoch {} is {} sec".format(epoch + 1, time.time()␣
 ↪- start))
        print("Generator AB loss: {:.4f}, Generator BA loss: {:.4f}".
 ↪format(avg_gen_AB_loss, avg_gen_BA_loss))
        print("Discriminator A loss: {:.4f}, Discriminator B loss: {:.4f}\n".
 ↪format(avg_disc_A_loss, avg_disc_B_loss))


# Train the model
train(EPOCHS)
```

```
…Time taken for epoch 1 is
27.488462686538696 sec
Generator AB loss: 0.7765, Generator BA loss: 0.7680
Discriminator A loss: 0.6977, Discriminator B loss: 0.6990

…Time taken for epoch 2 is
13.07184910774231 sec
Generator AB loss: 0.7679, Generator BA loss: 0.7771
Discriminator A loss: 0.6840, Discriminator B loss: 0.6861

…Time taken for epoch 3 is
13.128509759902954 sec
Generator AB loss: 0.7757, Generator BA loss: 0.8128
Discriminator A loss: 0.6662, Discriminator B loss: 0.6812

…Time taken for epoch 4 is
13.198266744613647 sec
Generator AB loss: 0.7829, Generator BA loss: 0.8489
Discriminator A loss: 0.6466, Discriminator B loss: 0.6764

…Time taken for epoch 5 is
13.061321258544922 sec
Generator AB loss: 0.7945, Generator BA loss: 0.8624
Discriminator A loss: 0.6465, Discriminator B loss: 0.6712

…Time taken for epoch 6 is
13.006491899490356 sec
Generator AB loss: 0.7970, Generator BA loss: 0.8850
Discriminator A loss: 0.6322, Discriminator B loss: 0.6714

…Time taken for epoch 7 is
12.808730125427246 sec
Generator AB loss: 0.8161, Generator BA loss: 0.9400
```

```
Discriminator A loss: 0.6201, Discriminator B loss: 0.6638

…Time taken for epoch 8 is
12.740618467330933 sec
Generator AB loss: 0.8126, Generator BA loss: 0.9375
Discriminator A loss: 0.6138, Discriminator B loss: 0.6640

…Time taken for epoch 9 is
12.939184665679932 sec
Generator AB loss: 0.8205, Generator BA loss: 0.9422
Discriminator A loss: 0.6109, Discriminator B loss: 0.6645

…Time taken for epoch 10 is
12.783109664916992 sec
Generator AB loss: 0.8168, Generator BA loss: 0.9566
Discriminator A loss: 0.6108, Discriminator B loss: 0.6616

…Time taken for epoch 11 is
12.843762159347534 sec
Generator AB loss: 0.8182, Generator BA loss: 0.9687
Discriminator A loss: 0.6015, Discriminator B loss: 0.6622

…Time taken for epoch 12 is
12.880480289459229 sec
Generator AB loss: 0.8239, Generator BA loss: 0.9704
Discriminator A loss: 0.6059, Discriminator B loss: 0.6580

…Time taken for epoch 13 is
12.693547010421753 sec
Generator AB loss: 0.8334, Generator BA loss: 0.9437
Discriminator A loss: 0.6178, Discriminator B loss: 0.6598

…Time taken for epoch 14 is
12.674412250518799 sec
Generator AB loss: 0.8384, Generator BA loss: 0.9816
Discriminator A loss: 0.5974, Discriminator B loss: 0.6527

…Time taken for epoch 15 is
12.76650333404541 sec
Generator AB loss: 0.8312, Generator BA loss: 0.9234
Discriminator A loss: 0.6229, Discriminator B loss: 0.6522

…Time taken for epoch 16 is
13.208204984664917 sec
Generator AB loss: 0.8446, Generator BA loss: 1.0015
Discriminator A loss: 0.5932, Discriminator B loss: 0.6489

…Time taken for epoch 17 is
```

```
13.231647253036499 sec
Generator AB loss: 0.8457, Generator BA loss: 0.9846
Discriminator A loss: 0.5951, Discriminator B loss: 0.6496

…Time taken for epoch 18 is
13.08447790145874 sec
Generator AB loss: 0.8487, Generator BA loss: 0.9777
Discriminator A loss: 0.6008, Discriminator B loss: 0.6450

…Time taken for epoch 19 is
13.151406526565552 sec
Generator AB loss: 0.8684, Generator BA loss: 1.0002
Discriminator A loss: 0.5958, Discriminator B loss: 0.6419

…Time taken for epoch 20 is
13.184776782989502 sec
Generator AB loss: 0.8777, Generator BA loss: 1.0277
Discriminator A loss: 0.5848, Discriminator B loss: 0.6287

…Time taken for epoch 21 is
13.083128213882446 sec
Generator AB loss: 0.8837, Generator BA loss: 1.0161
Discriminator A loss: 0.5864, Discriminator B loss: 0.6373

…Time taken for epoch 22 is
13.048650026321411 sec
Generator AB loss: 0.8838, Generator BA loss: 1.0170
Discriminator A loss: 0.5880, Discriminator B loss: 0.6281

…Time taken for epoch 23 is
13.188129186630249 sec
Generator AB loss: 0.9121, Generator BA loss: 1.0251
Discriminator A loss: 0.5908, Discriminator B loss: 0.6198

…Time taken for epoch 24 is
13.145785093307495 sec
Generator AB loss: 0.9005, Generator BA loss: 1.0211
Discriminator A loss: 0.5890, Discriminator B loss: 0.6228

…Time taken for epoch 25 is
13.072012662887573 sec
Generator AB loss: 0.9282, Generator BA loss: 1.0830
Discriminator A loss: 0.5695, Discriminator B loss: 0.6144

…Time taken for epoch 26 is
12.97522783279419 sec
Generator AB loss: 0.9373, Generator BA loss: 1.0364
Discriminator A loss: 0.5778, Discriminator B loss: 0.6103
```

```
…Time taken for epoch 27 is
12.924783706665039 sec
Generator AB loss: 0.9423, Generator BA loss: 1.0612
Discriminator A loss: 0.5745, Discriminator B loss: 0.6067

…Time taken for epoch 28 is
12.803432941436768 sec
Generator AB loss: 0.9611, Generator BA loss: 1.0540
Discriminator A loss: 0.5712, Discriminator B loss: 0.5977

…Time taken for epoch 29 is
12.828327178955078 sec
Generator AB loss: 0.9821, Generator BA loss: 1.0828
Discriminator A loss: 0.5688, Discriminator B loss: 0.5926

…Time taken for epoch 30 is
12.86509108543396 sec
Generator AB loss: 0.9942, Generator BA loss: 1.0845
Discriminator A loss: 0.5774, Discriminator B loss: 0.5882

…Time taken for epoch 31 is
12.889779090881348 sec
Generator AB loss: 1.0004, Generator BA loss: 1.0744
Discriminator A loss: 0.5694, Discriminator B loss: 0.5828

…Time taken for epoch 32 is
12.922930479049683 sec
Generator AB loss: 1.0096, Generator BA loss: 1.0759
Discriminator A loss: 0.5663, Discriminator B loss: 0.5842

…Time taken for epoch 33 is
13.318272113800049 sec
Generator AB loss: 1.0328, Generator BA loss: 1.1005
Discriminator A loss: 0.5648, Discriminator B loss: 0.5792

…Time taken for epoch 34 is
13.116933584213257 sec
Generator AB loss: 1.0508, Generator BA loss: 1.0848
Discriminator A loss: 0.5610, Discriminator B loss: 0.5671

…Time taken for epoch 35 is
13.166071891784668 sec
Generator AB loss: 1.0606, Generator BA loss: 1.0828
Discriminator A loss: 0.5808, Discriminator B loss: 0.5669

…Time taken for epoch 36 is
13.374498128890991 sec
```

```
Generator AB loss: 1.0693, Generator BA loss: 1.0400
Discriminator A loss: 0.5851, Discriminator B loss: 0.5572

…Time taken for epoch 37 is
13.12476396560669 sec
Generator AB loss: 1.0919, Generator BA loss: 1.1004
Discriminator A loss: 0.5572, Discriminator B loss: 0.5568

…Time taken for epoch 38 is
13.213222742080688 sec
Generator AB loss: 1.1305, Generator BA loss: 1.1144
Discriminator A loss: 0.5567, Discriminator B loss: 0.5405

…Time taken for epoch 39 is
13.392103433609009 sec
Generator AB loss: 1.1523, Generator BA loss: 1.1394
Discriminator A loss: 0.5528, Discriminator B loss: 0.5383

…Time taken for epoch 40 is
13.565938472747803 sec
Generator AB loss: 1.1483, Generator BA loss: 1.1333
Discriminator A loss: 0.5474, Discriminator B loss: 0.5400

…Time taken for epoch 41 is
13.553267002105713 sec
Generator AB loss: 1.1968, Generator BA loss: 1.1402
Discriminator A loss: 0.5514, Discriminator B loss: 0.5230

…Time taken for epoch 42 is
13.352603435516357 sec
Generator AB loss: 1.1902, Generator BA loss: 1.1161
Discriminator A loss: 0.5515, Discriminator B loss: 0.5240

…Time taken for epoch 43 is
13.099270820617676 sec
Generator AB loss: 1.2112, Generator BA loss: 1.1425
Discriminator A loss: 0.5482, Discriminator B loss: 0.5153

…Time taken for epoch 44 is
13.375960111618042 sec
Generator AB loss: 1.2351, Generator BA loss: 1.0995
Discriminator A loss: 0.5806, Discriminator B loss: 0.5127

…Time taken for epoch 45 is
13.129218816757202 sec
Generator AB loss: 1.2768, Generator BA loss: 1.1607
Discriminator A loss: 0.5358, Discriminator B loss: 0.4957
```

```
…Time taken for epoch 46 is
13.100391864776611 sec
Generator AB loss: 1.2886, Generator BA loss: 1.1445
Discriminator A loss: 0.5512, Discriminator B loss: 0.4996

…Time taken for epoch 47 is
13.366199731826782 sec
Generator AB loss: 1.2984, Generator BA loss: 1.1757
Discriminator A loss: 0.5379, Discriminator B loss: 0.4858

…Time taken for epoch 48 is
13.383051633834839 sec
Generator AB loss: 1.3282, Generator BA loss: 1.1504
Discriminator A loss: 0.5408, Discriminator B loss: 0.4812

…Time taken for epoch 49 is
13.025627613067627 sec
Generator AB loss: 1.3578, Generator BA loss: 1.1563
Discriminator A loss: 0.5506, Discriminator B loss: 0.4785

…Time taken for epoch 50 is
13.203929901123047 sec
Generator AB loss: 1.3866, Generator BA loss: 1.1408
Discriminator A loss: 0.5424, Discriminator B loss: 0.4689

…Time taken for epoch 51 is
12.850569248199463 sec
Generator AB loss: 1.4020, Generator BA loss: 1.1975
Discriminator A loss: 0.5260, Discriminator B loss: 0.4666

…Time taken for epoch 52 is
12.821491718292236 sec
Generator AB loss: 1.4235, Generator BA loss: 1.1694
Discriminator A loss: 0.5327, Discriminator B loss: 0.4618

…Time taken for epoch 53 is
13.080695629119873 sec
Generator AB loss: 1.4091, Generator BA loss: 1.1750
Discriminator A loss: 0.5342, Discriminator B loss: 0.4645

…Time taken for epoch 54 is
13.101016283035278 sec
Generator AB loss: 1.4518, Generator BA loss: 1.1764
Discriminator A loss: 0.5360, Discriminator B loss: 0.4507

…Time taken for epoch 55 is
13.31496787071228 sec
Generator AB loss: 1.4959, Generator BA loss: 1.2087
```

```
Discriminator A loss: 0.5244, Discriminator B loss: 0.4393

…Time taken for epoch 56 is
13.03330135345459 sec
Generator AB loss: 1.5220, Generator BA loss: 1.2380
Discriminator A loss: 0.5150, Discriminator B loss: 0.4347

…Time taken for epoch 57 is
13.083192586898804 sec
Generator AB loss: 1.5579, Generator BA loss: 1.2388
Discriminator A loss: 0.5187, Discriminator B loss: 0.4391

…Time taken for epoch 58 is
13.074130773544312 sec
Generator AB loss: 1.5447, Generator BA loss: 1.2309
Discriminator A loss: 0.5250, Discriminator B loss: 0.4337

…Time taken for epoch 59 is
12.94373893737793 sec
Generator AB loss: 1.5938, Generator BA loss: 1.2291
Discriminator A loss: 0.5216, Discriminator B loss: 0.4205

…Time taken for epoch 60 is
12.734158992767334 sec
Generator AB loss: 1.6021, Generator BA loss: 1.2307
Discriminator A loss: 0.5278, Discriminator B loss: 0.4149

…Time taken for epoch 61 is
12.8080153465271 sec
Generator AB loss: 1.6207, Generator BA loss: 1.2495
Discriminator A loss: 0.5035, Discriminator B loss: 0.4159

…Time taken for epoch 62 is
12.975268602371216 sec
Generator AB loss: 1.6643, Generator BA loss: 1.2806
Discriminator A loss: 0.5090, Discriminator B loss: 0.4147

…Time taken for epoch 63 is
13.07485055923462 sec
Generator AB loss: 1.6533, Generator BA loss: 1.2625
Discriminator A loss: 0.5147, Discriminator B loss: 0.4061

…Time taken for epoch 64 is
12.96437692642212 sec
Generator AB loss: 1.6838, Generator BA loss: 1.2768
Discriminator A loss: 0.5027, Discriminator B loss: 0.3998

…Time taken for epoch 65 is
```

```
12.934196472167969 sec
Generator AB loss: 1.7173, Generator BA loss: 1.2597
Discriminator A loss: 0.5283, Discriminator B loss: 0.3966

…Time taken for epoch 66 is
13.271772146224976 sec
Generator AB loss: 1.7276, Generator BA loss: 1.2760
Discriminator A loss: 0.4912, Discriminator B loss: 0.3951

…Time taken for epoch 67 is
12.870438814163208 sec
Generator AB loss: 1.7560, Generator BA loss: 1.3074
Discriminator A loss: 0.5012, Discriminator B loss: 0.4086

…Time taken for epoch 68 is
12.903080224990845 sec
Generator AB loss: 1.7509, Generator BA loss: 1.2858
Discriminator A loss: 0.5055, Discriminator B loss: 0.3987

…Time taken for epoch 69 is
12.803889036178589 sec
Generator AB loss: 1.7840, Generator BA loss: 1.3310
Discriminator A loss: 0.4751, Discriminator B loss: 0.3704

…Time taken for epoch 70 is
12.761464357376099 sec
Generator AB loss: 1.7794, Generator BA loss: 1.3192
Discriminator A loss: 0.4979, Discriminator B loss: 0.3936

…Time taken for epoch 71 is
12.802840948104858 sec
Generator AB loss: 1.7987, Generator BA loss: 1.3315
Discriminator A loss: 0.4896, Discriminator B loss: 0.3964

…Time taken for epoch 72 is
12.808837413787842 sec
Generator AB loss: 1.7987, Generator BA loss: 1.3213
Discriminator A loss: 0.4952, Discriminator B loss: 0.3821

…Time taken for epoch 73 is
12.866992473602295 sec
Generator AB loss: 1.8419, Generator BA loss: 1.3222
Discriminator A loss: 0.5129, Discriminator B loss: 0.3703

…Time taken for epoch 74 is
12.890188217163086 sec
Generator AB loss: 1.8668, Generator BA loss: 1.3104
Discriminator A loss: 0.4920, Discriminator B loss: 0.3971
```

```
…Time taken for epoch 75 is
12.935905933380127 sec
Generator AB loss: 1.8361, Generator BA loss: 1.3475
Discriminator A loss: 0.4787, Discriminator B loss: 0.3731

…Time taken for epoch 76 is
13.09743618965149 sec
Generator AB loss: 1.8572, Generator BA loss: 1.3636
Discriminator A loss: 0.4779, Discriminator B loss: 0.3774

…Time taken for epoch 77 is
12.828179359436035 sec
Generator AB loss: 1.8495, Generator BA loss: 1.3564
Discriminator A loss: 0.4888, Discriminator B loss: 0.3860

…Time taken for epoch 78 is
12.97624659538269 sec
Generator AB loss: 1.8517, Generator BA loss: 1.3605
Discriminator A loss: 0.4920, Discriminator B loss: 0.3807

…Time taken for epoch 79 is
13.221158266067505 sec
Generator AB loss: 1.8884, Generator BA loss: 1.4109
Discriminator A loss: 0.4690, Discriminator B loss: 0.3711

…Time taken for epoch 80 is
13.22184944152832 sec
Generator AB loss: 1.9258, Generator BA loss: 1.3892
Discriminator A loss: 0.4750, Discriminator B loss: 0.3779

…Time taken for epoch 81 is
13.167560815811157 sec
Generator AB loss: 1.9311, Generator BA loss: 1.3706
Discriminator A loss: 0.4912, Discriminator B loss: 0.3757

…Time taken for epoch 82 is
12.91163945198059 sec
Generator AB loss: 1.9430, Generator BA loss: 1.4094
Discriminator A loss: 0.4751, Discriminator B loss: 0.3701

…Time taken for epoch 83 is
12.93082070350647 sec
Generator AB loss: 1.9183, Generator BA loss: 1.4103
Discriminator A loss: 0.4714, Discriminator B loss: 0.3687

…Time taken for epoch 84 is
12.899925231933594 sec
```

```
Generator AB loss: 1.9573, Generator BA loss: 1.4789
Discriminator A loss: 0.4536, Discriminator B loss: 0.3691

…Time taken for epoch 85 is
13.31040644645691 sec
Generator AB loss: 1.9712, Generator BA loss: 1.4336
Discriminator A loss: 0.4546, Discriminator B loss: 0.3597

…Time taken for epoch 86 is
13.437985181808472 sec
Generator AB loss: 1.9728, Generator BA loss: 1.4506
Discriminator A loss: 0.4713, Discriminator B loss: 0.3843

…Time taken for epoch 87 is
13.231847763061523 sec
Generator AB loss: 1.9561, Generator BA loss: 1.4637
Discriminator A loss: 0.4537, Discriminator B loss: 0.3675

…Time taken for epoch 88 is
13.12296724319458 sec
Generator AB loss: 1.9558, Generator BA loss: 1.4406
Discriminator A loss: 0.4708, Discriminator B loss: 0.3702

…Time taken for epoch 89 is
13.398867845535278 sec
Generator AB loss: 1.9724, Generator BA loss: 1.4466
Discriminator A loss: 0.4653, Discriminator B loss: 0.3749

…Time taken for epoch 90 is
13.414485692977905 sec
Generator AB loss: 1.9948, Generator BA loss: 1.4357
Discriminator A loss: 0.4764, Discriminator B loss: 0.3653

…Time taken for epoch 91 is
13.201572179794312 sec
Generator AB loss: 1.9798, Generator BA loss: 1.4520
Discriminator A loss: 0.4582, Discriminator B loss: 0.3647

…Time taken for epoch 92 is
13.15543818473816 sec
Generator AB loss: 1.9599, Generator BA loss: 1.4526
Discriminator A loss: 0.4705, Discriminator B loss: 0.3783

…Time taken for epoch 93 is
13.239192008972168 sec
Generator AB loss: 1.9903, Generator BA loss: 1.4229
Discriminator A loss: 0.4799, Discriminator B loss: 0.3707
```

```
…Time taken for epoch 94 is
13.112773895263672 sec
Generator AB loss: 2.0097, Generator BA loss: 1.4884
Discriminator A loss: 0.4453, Discriminator B loss: 0.3712

…Time taken for epoch 95 is
13.092725038528442 sec
Generator AB loss: 1.9909, Generator BA loss: 1.4956
Discriminator A loss: 0.4463, Discriminator B loss: 0.3619

…Time taken for epoch 96 is
13.095580577850342 sec
Generator AB loss: 2.0075, Generator BA loss: 1.5051
Discriminator A loss: 0.4560, Discriminator B loss: 0.3650

…Time taken for epoch 97 is
13.171940803527832 sec
Generator AB loss: 2.0454, Generator BA loss: 1.4782
Discriminator A loss: 0.4757, Discriminator B loss: 0.3832

…Time taken for epoch 98 is
13.06265139579773 sec
Generator AB loss: 2.0189, Generator BA loss: 1.4511
Discriminator A loss: 0.4732, Discriminator B loss: 0.3709

…Time taken for epoch 99 is
13.158701658248901 sec
Generator AB loss: 2.0252, Generator BA loss: 1.5086
Discriminator A loss: 0.4361, Discriminator B loss: 0.3662

…Time taken for epoch 100 is
13.208929300308228 sec
Generator AB loss: 2.0385, Generator BA loss: 1.5007
Discriminator A loss: 0.4621, Discriminator B loss: 0.3594
```

1.4 Plotting

1.4.1 Loss Graph Plotting Function

[ ]:
```python
import matplotlib.pyplot as plt

def plot_loss_graphs(epochs, gen_AB_losses, gen_BA_losses, disc_A_losses,␣
 ↪disc_B_losses):
    plt.figure(figsize=(12, 6))

    plt.subplot(2, 2, 1)
    plt.plot(range(epochs), gen_AB_losses, label="Generator AB Loss")
```

```
    plt.xlabel("Epochs")
    plt.ylabel("Loss")
    plt.legend()

    plt.subplot(2, 2, 2)
    plt.plot(range(epochs), gen_BA_losses, label="Generator BA Loss")
    plt.xlabel("Epochs")
    plt.ylabel("Loss")
    plt.legend()

    plt.subplot(2, 2, 3)
    plt.plot(range(epochs), disc_A_losses, label="Discriminator A Loss")
    plt.xlabel("Epochs")
    plt.ylabel("Loss")
    plt.legend()

    plt.subplot(2, 2, 4)
    plt.plot(range(epochs), disc_B_losses, label="Discriminator B Loss")
    plt.xlabel("Epochs")
    plt.ylabel("Loss")
    plt.legend()

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

1.4.2 Image Display Plotting Function

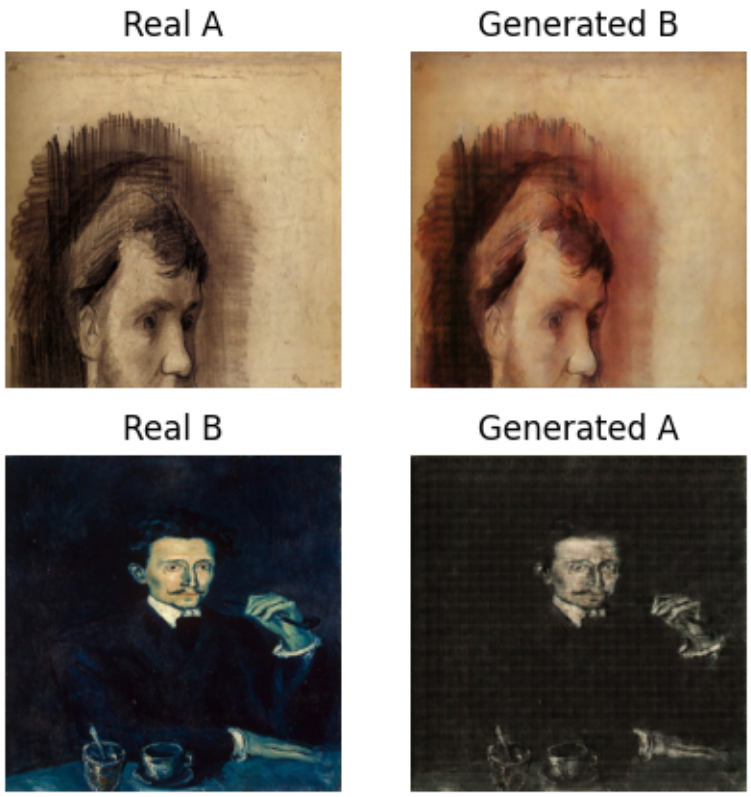
[ ]:
```
import matplotlib.pyplot as plt

def display_images(real_A, generated_B, real_B, generated_A):
    plt.figure(figsize=(5, 5))

    display_list = [real_A[0], generated_B[0], real_B[0], generated_A[0]]
    title = ['Real A', 'Generated B', 'Real B', 'Generated A']

    for i in range(4):
        plt.subplot(2, 2, i + 1)
        plt.title(title[i])
        plt.imshow(display_list[i] * 0.5 + 0.5)
        plt.axis('off')
    plt.show()
```

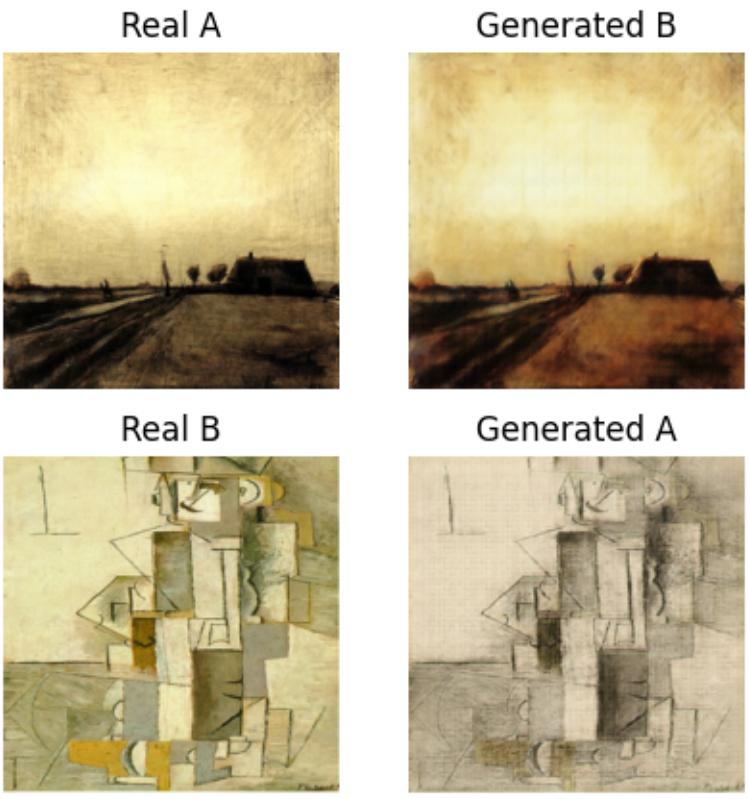
1.5 Evaluation

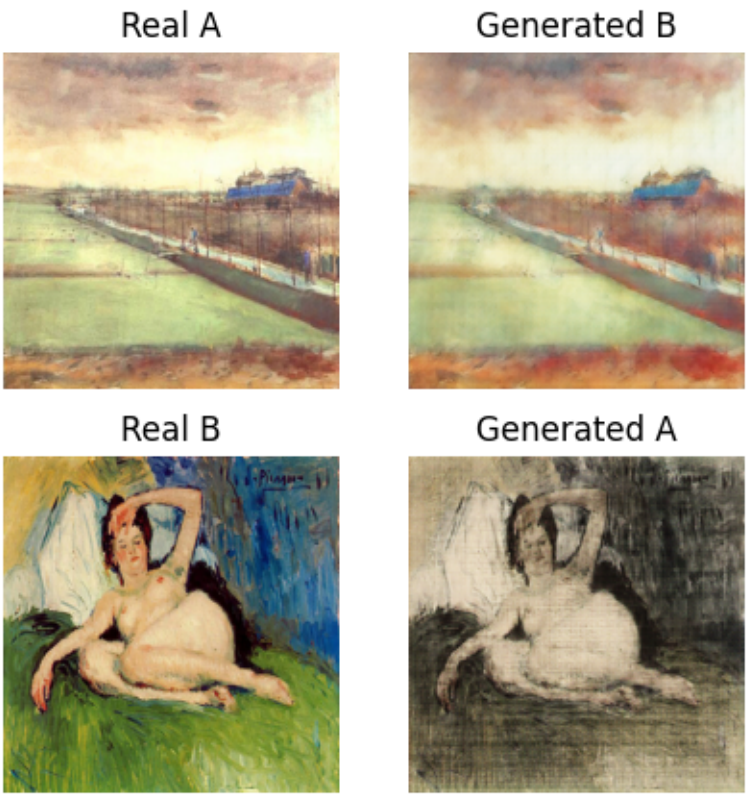
1.5.1 ArtGAN Evaluation

[ ]:
```
import tensorflow as tf

print("***** ArtGAN Images *****")

# Convert test datasets to TensorFlow datasets (take the first 50 images)
test_A = tf.data.Dataset.from_tensor_slices(vg_test).take(50).batch(1)
test_B = tf.data.Dataset.from_tensor_slices(p_test).take(50).batch(1)

# Test the model on the withheld dataset
for real_A, real_B in zip(test_A, test_B):
    generated_B = g_AB(real_A, training=False)
    generated_A = g_BA(real_B, training=False)
    display_images(real_A, generated_B, real_B, generated_A)
```

```
***** ArtGAN Images *****
```




Real A
Generated B
Real B
Generated A


Real A
Generated B
Real B
Generated A

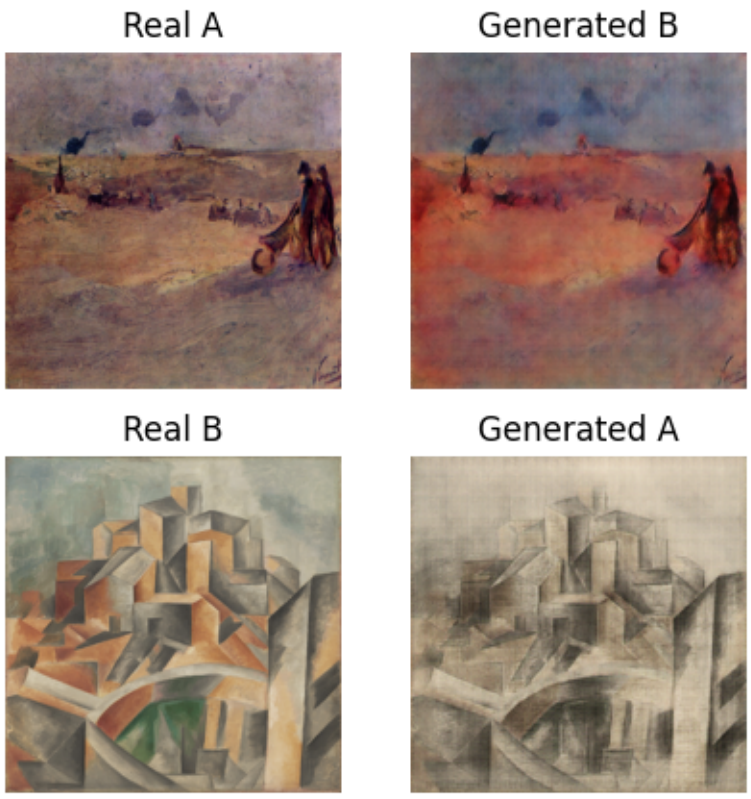
Real A

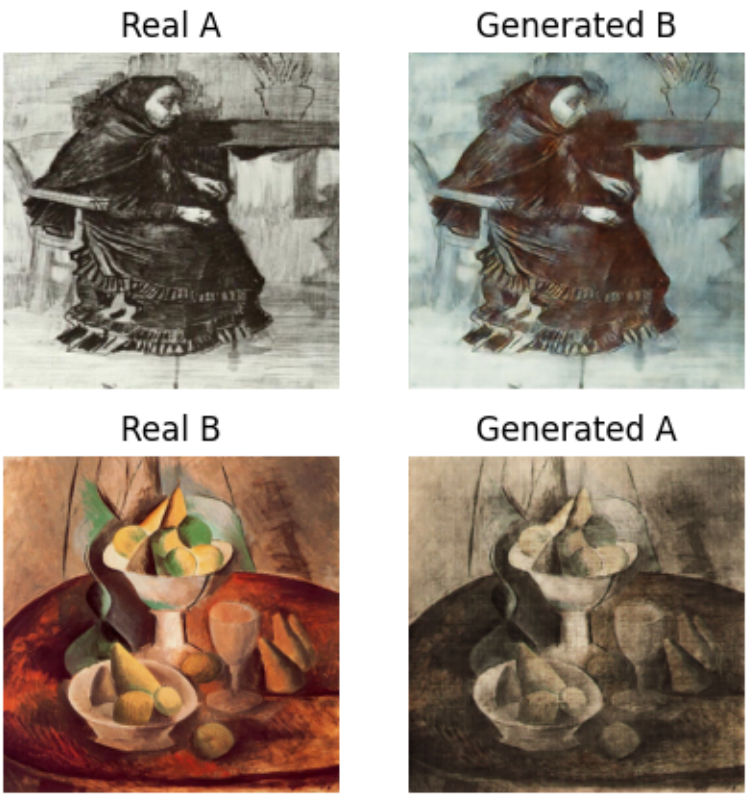
Generated B

Real B

Generated A


Real A
Generated B
Real B
Generated A


Real A
Generated B
Real B
Generated A


Real A
Generated B
Real B
Generated A

Real A

Generated B

Real B

Generated A

Real A

Generated B

Real B

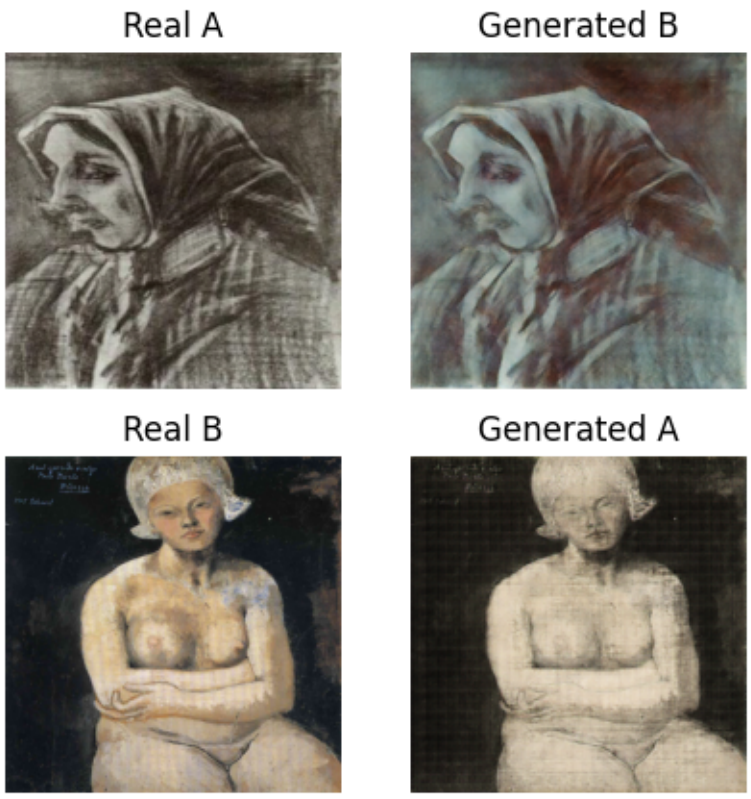
Generated A

Real A

Generated B

Real B

Generated A

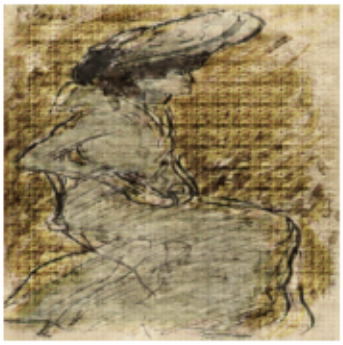

Real A
Generated B
Real B
Generated A

Real A

Generated B

Real B

Generated A



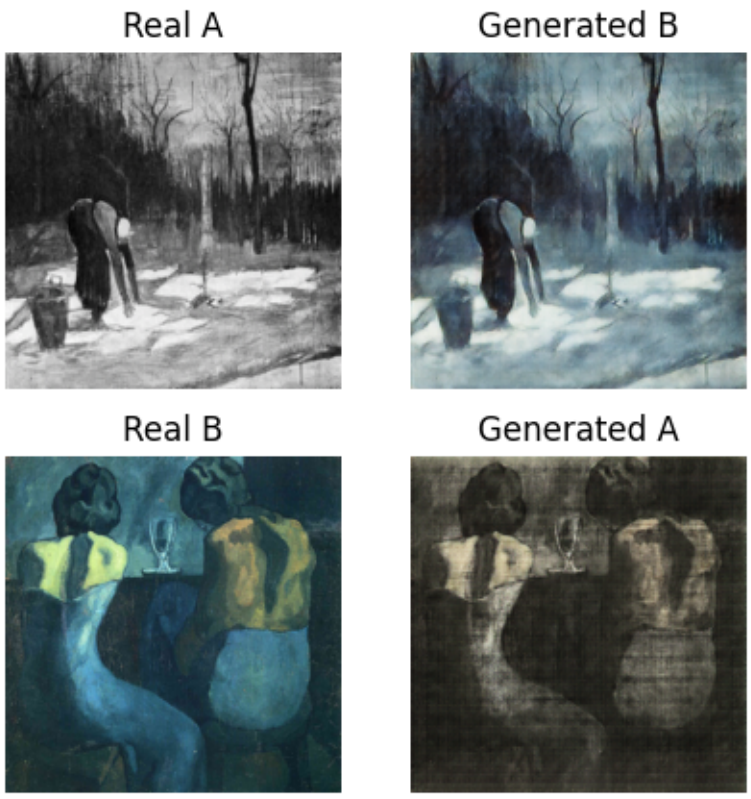

Real A
Generated B
Real B
Generated A

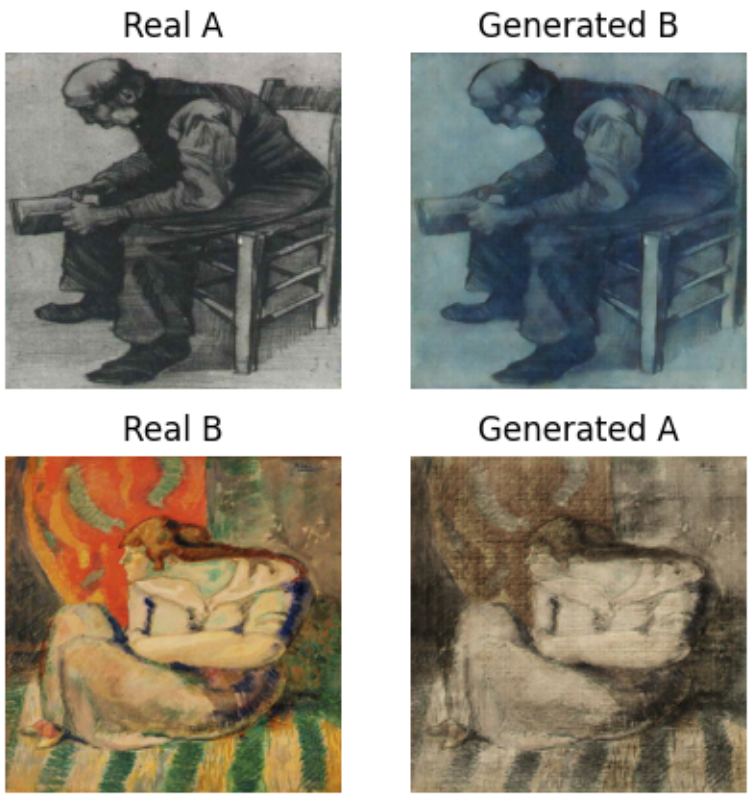
Real A

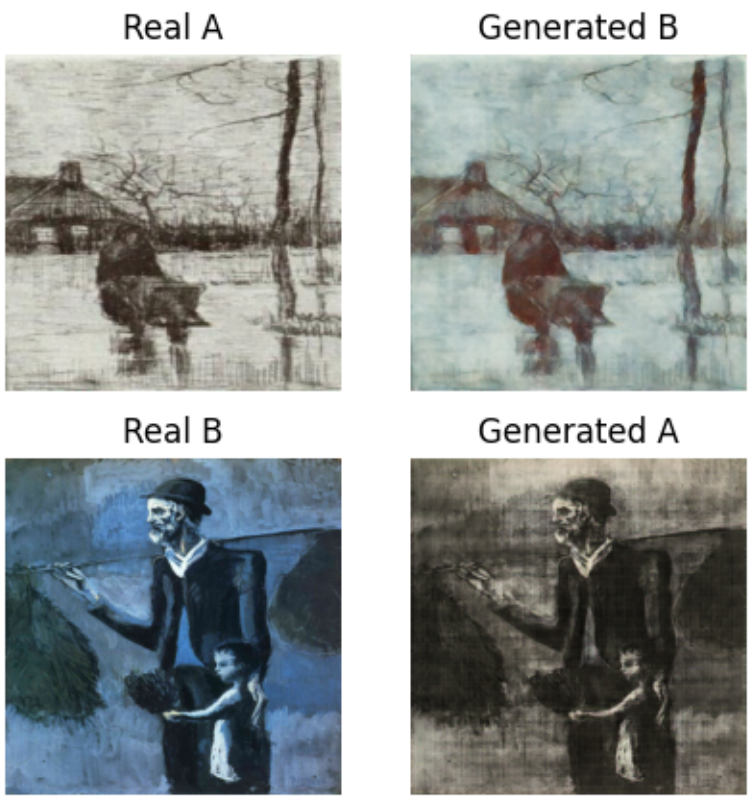
Generated B

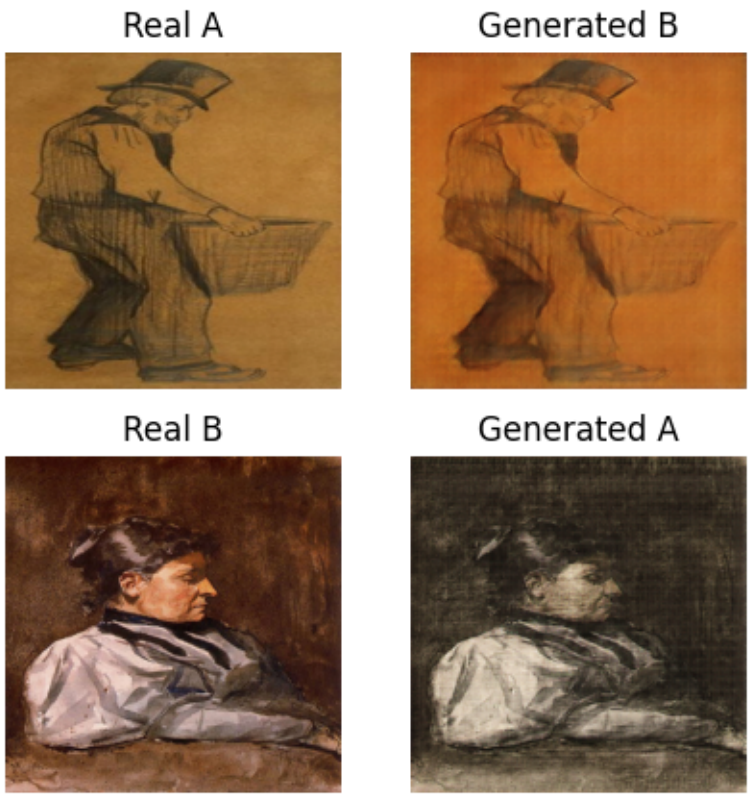
Real B

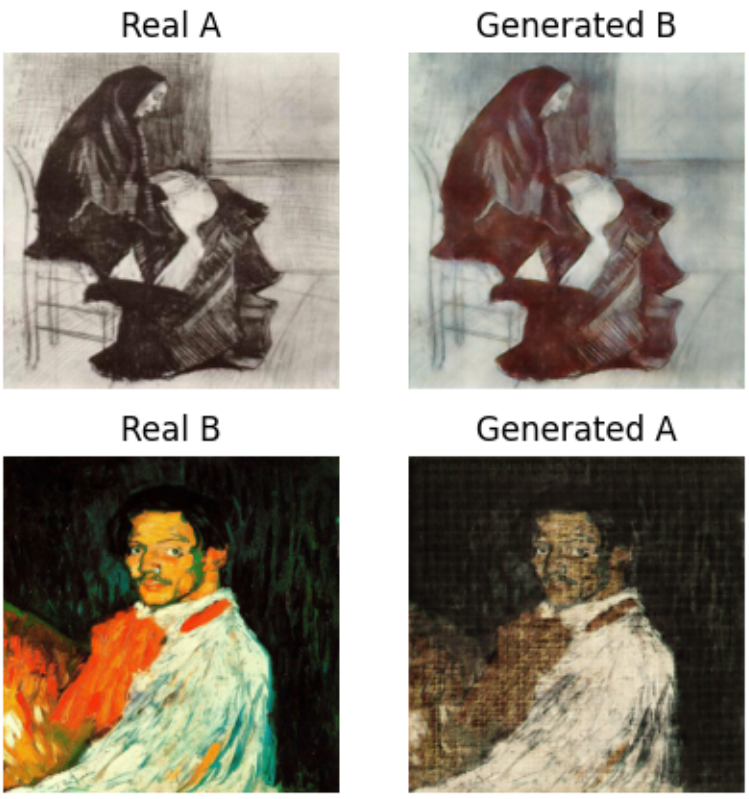
Generated A

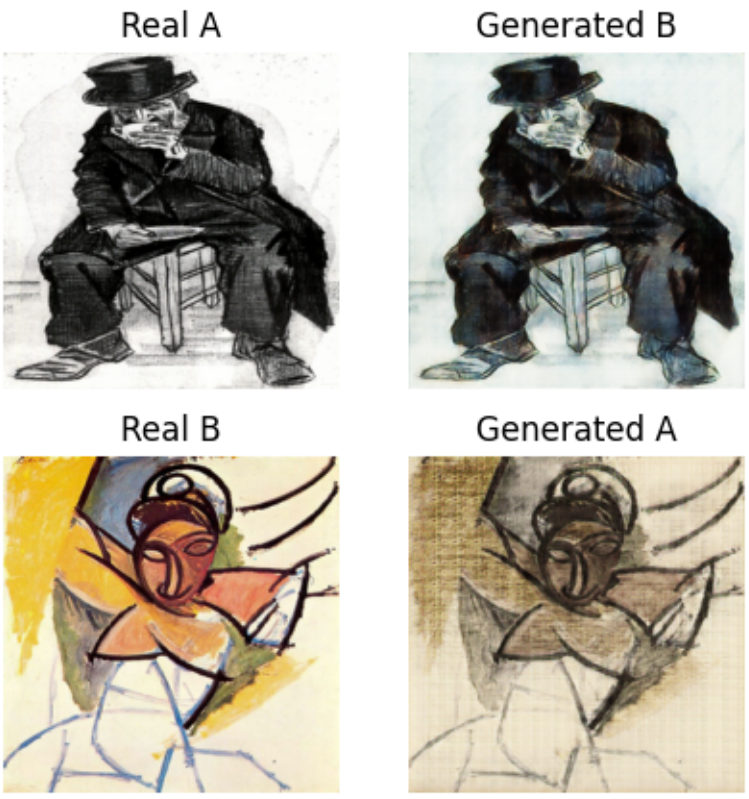

Real A
Generated B
Real B
Generated A

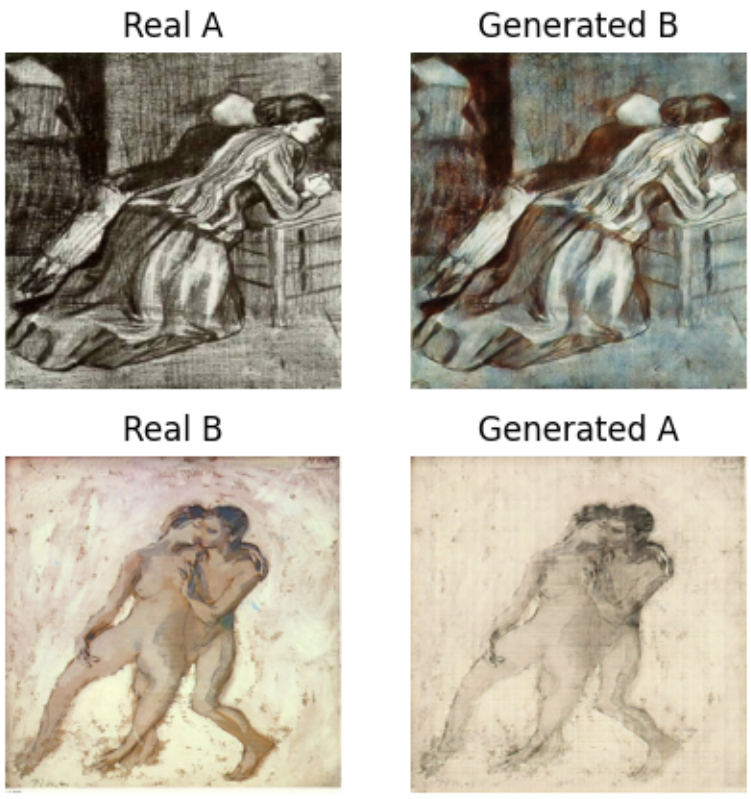

Real A
Generated B
Real B
Generated A


Real A
Generated B
Real B
Generated A


Real A
Generated B
Real B
Generated A


Real A
Generated B
Real B
Generated A


Real A
Generated B
Real B
Generated A


Real A
Generated B
Real B
Generated A


Real A
Generated B
Real B
Generated A

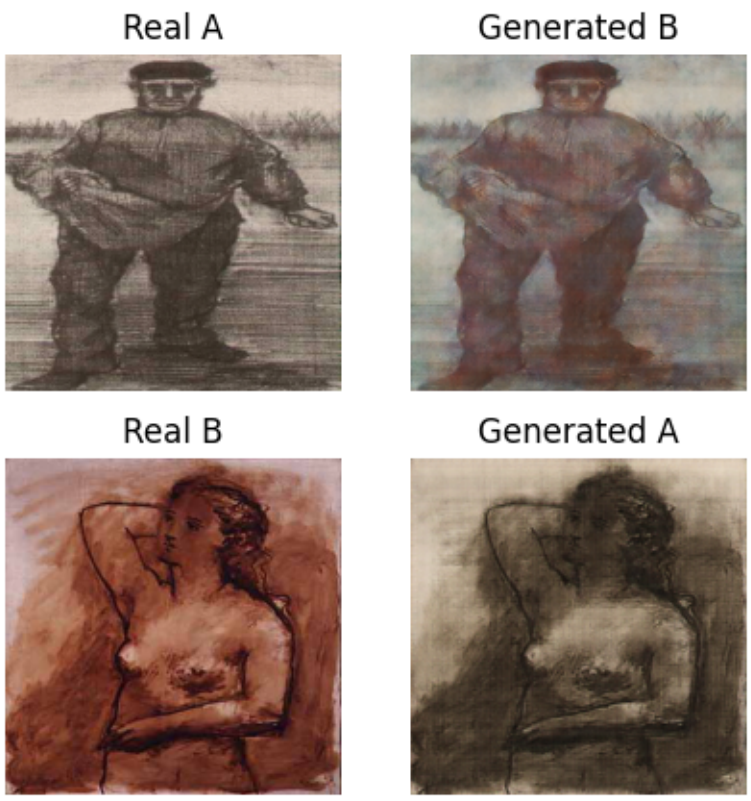

Real A
Generated B
Real B
Generated A

Real A

Generated B

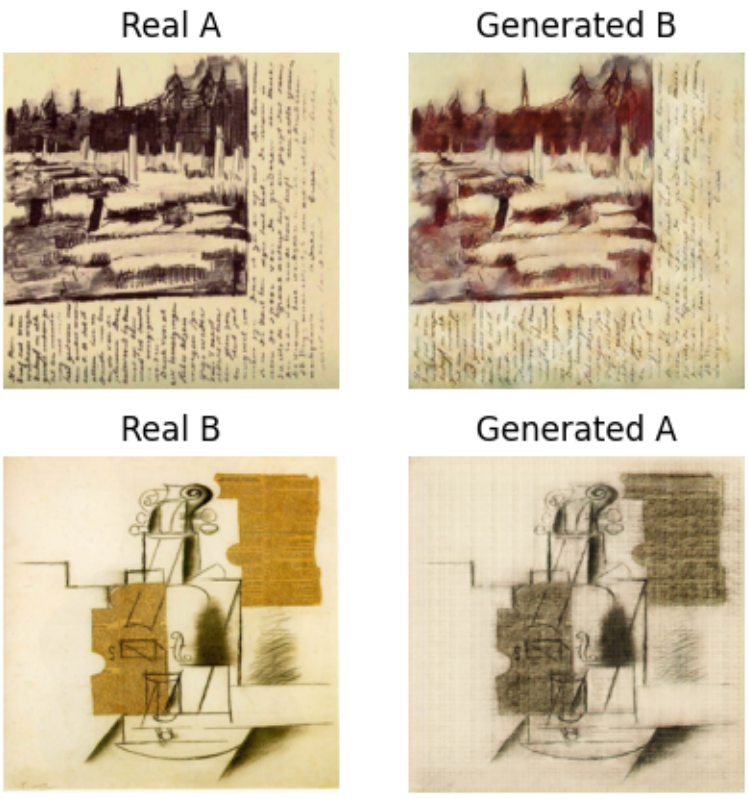
Real B

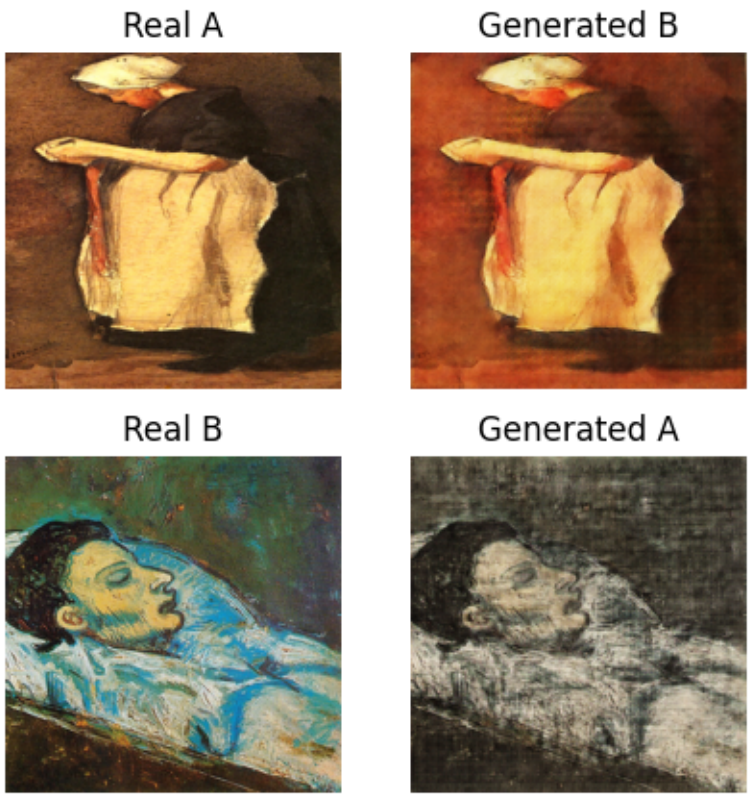
Generated A

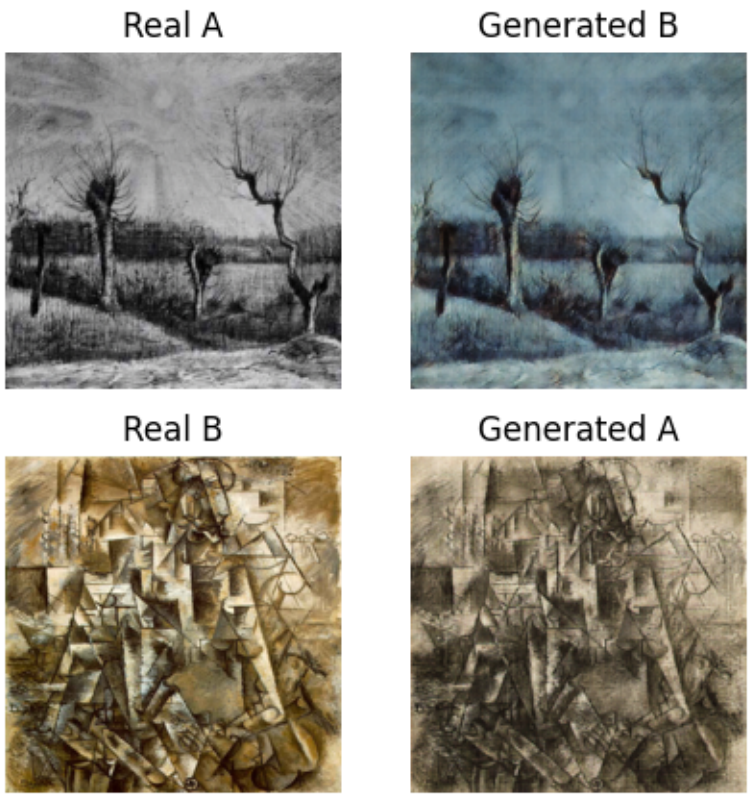

Real A
Generated B
Real B
Generated A

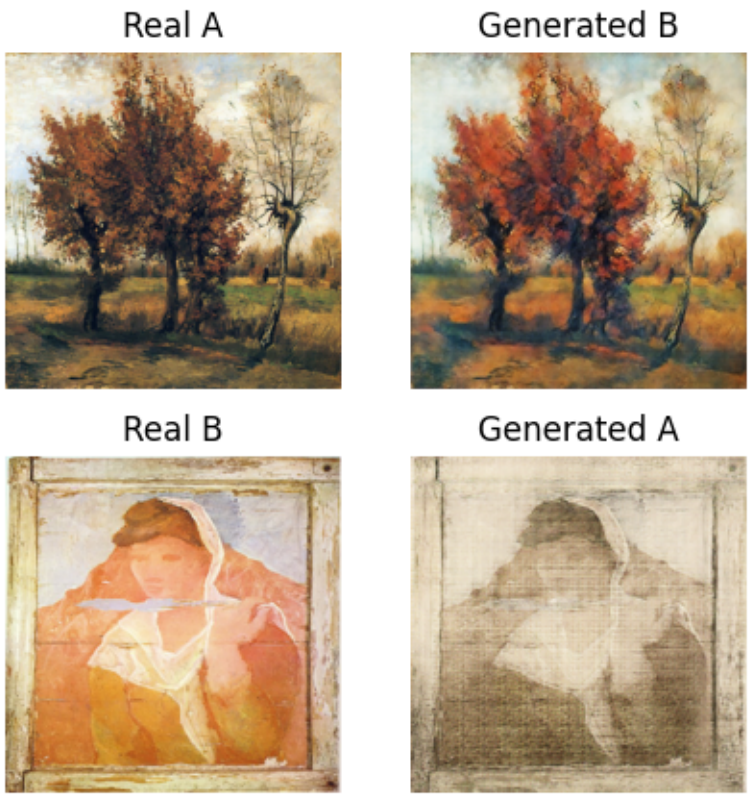

Real A
Generated B
Real B
Generated A

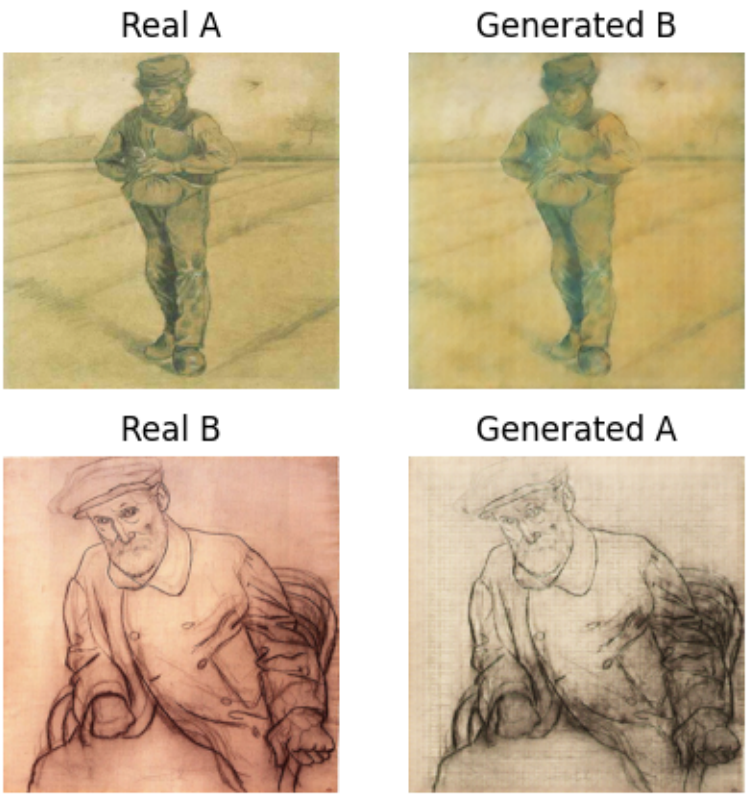

Real A
Generated B
Real B
Generated A


Real A
Generated B
Real B
Generated A


Real A
Generated B
Real B
Generated A


Real A
Generated B
Real B
Generated A


Real A
Generated B
Real B
Generated A

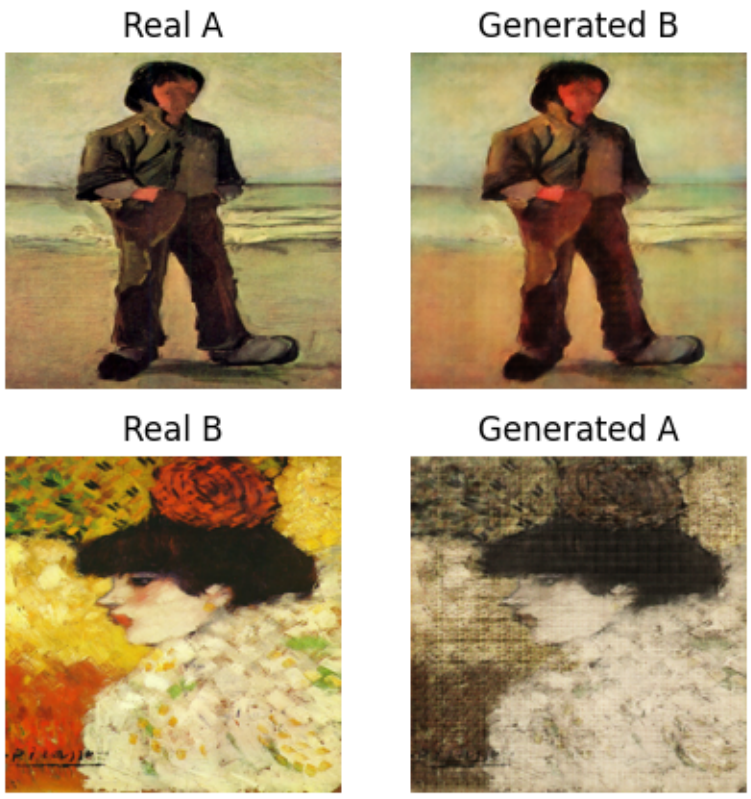

Real A
Generated B
Real B
Generated A

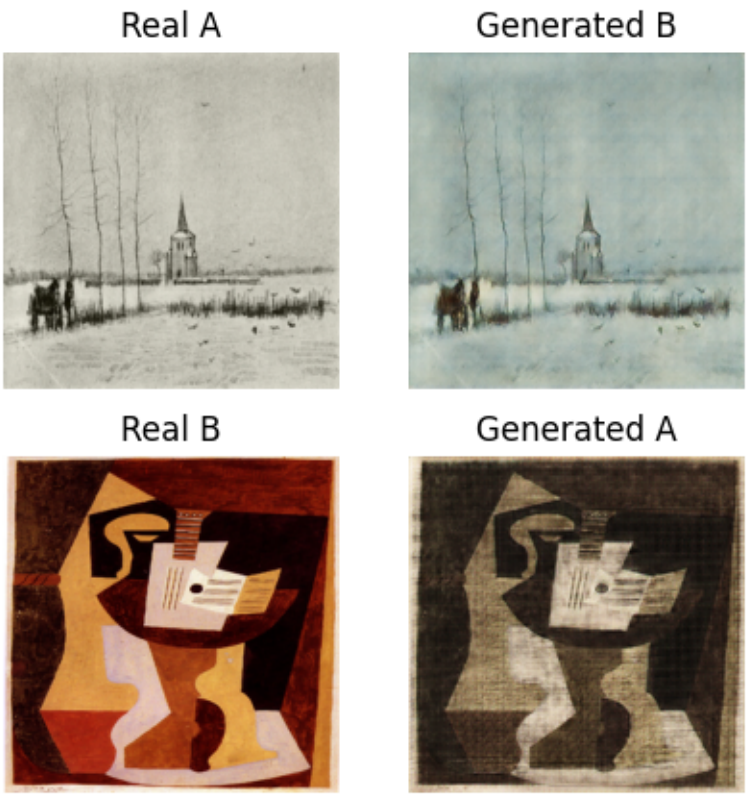
Real A

Generated B

Real B

Generated A


Real A
Generated B
Real B
Generated A


Real A
Generated B
Real B
Generated A


Real A
Generated B
Real B
Generated A

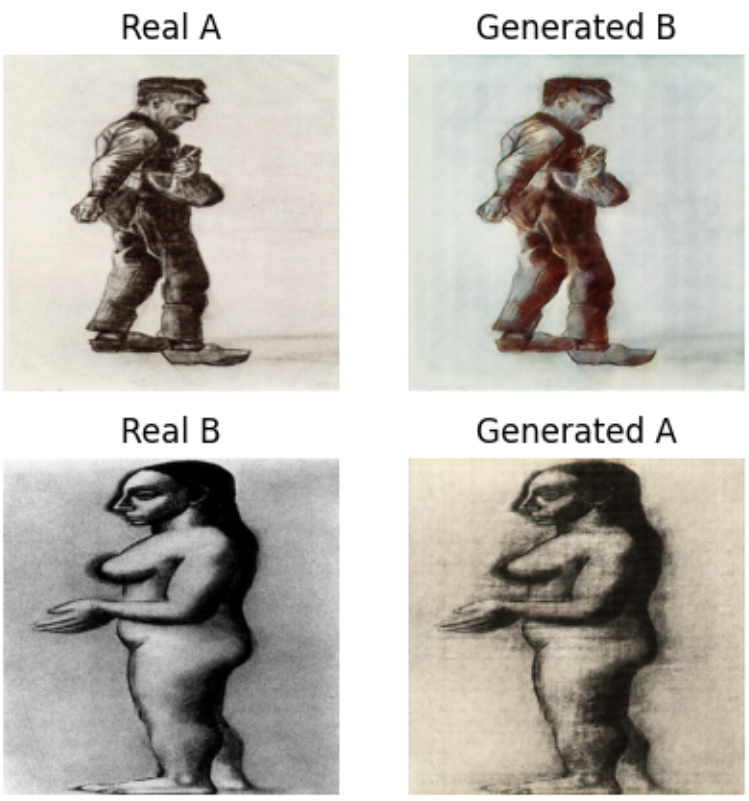

Real A
Generated B
Real B
Generated A

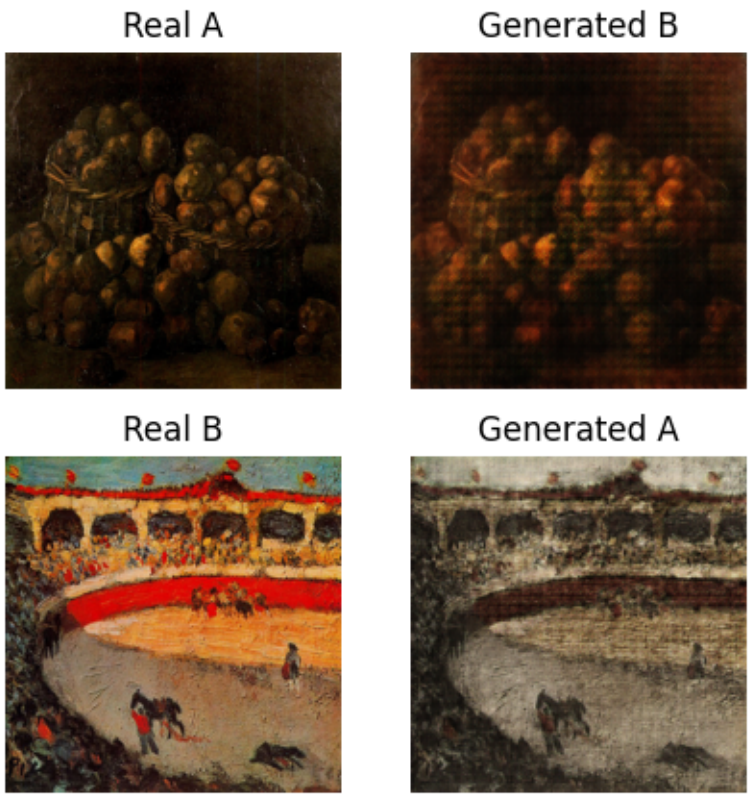
Real A

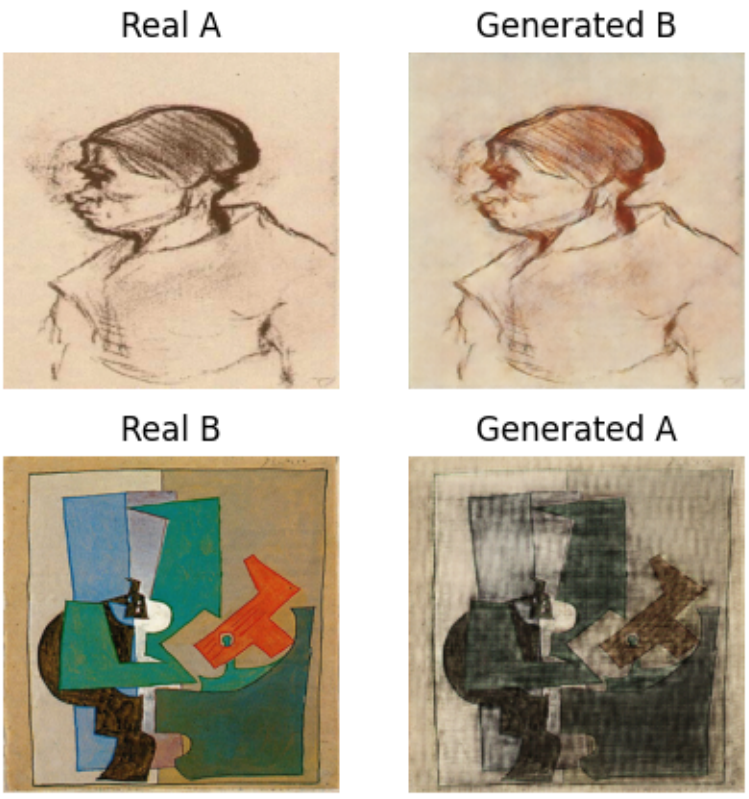
Generated B

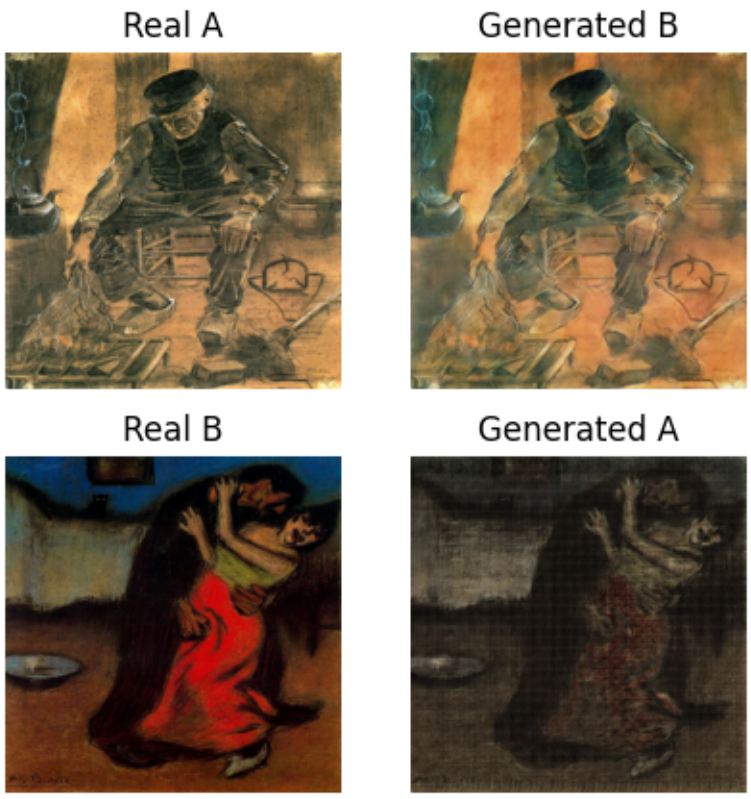
Real B

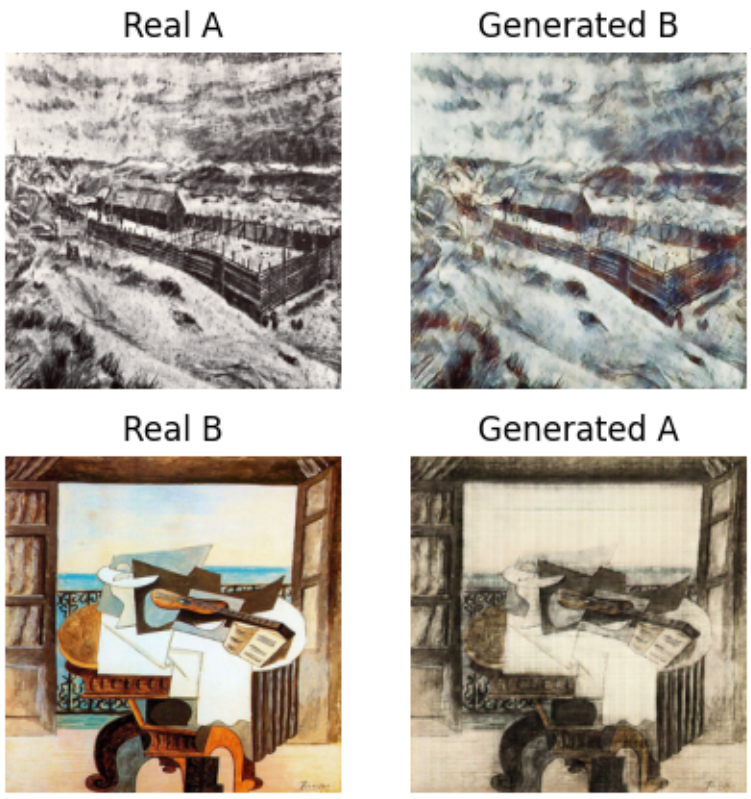
Generated A

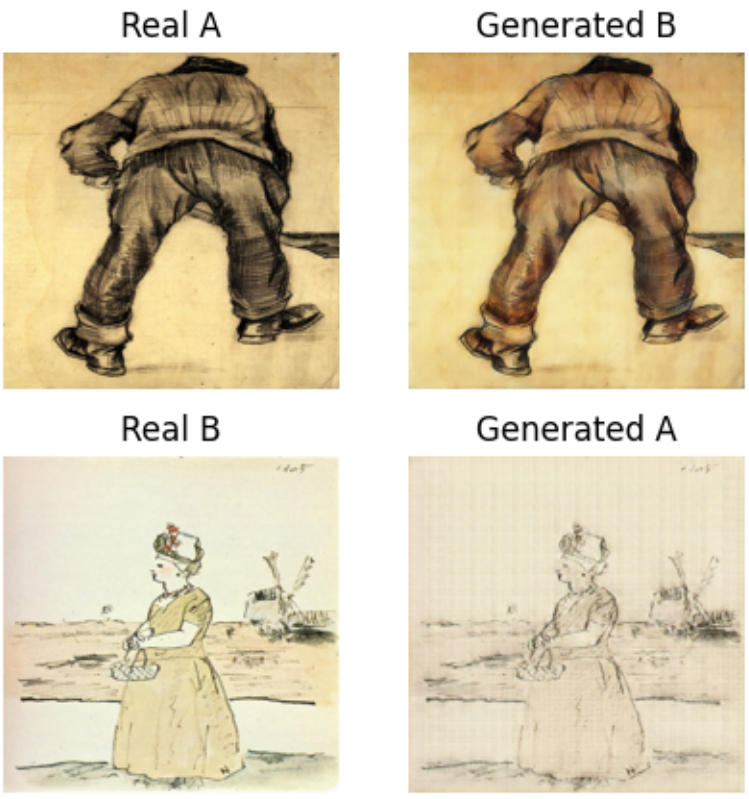

Real A
Generated B
Real B
Generated A

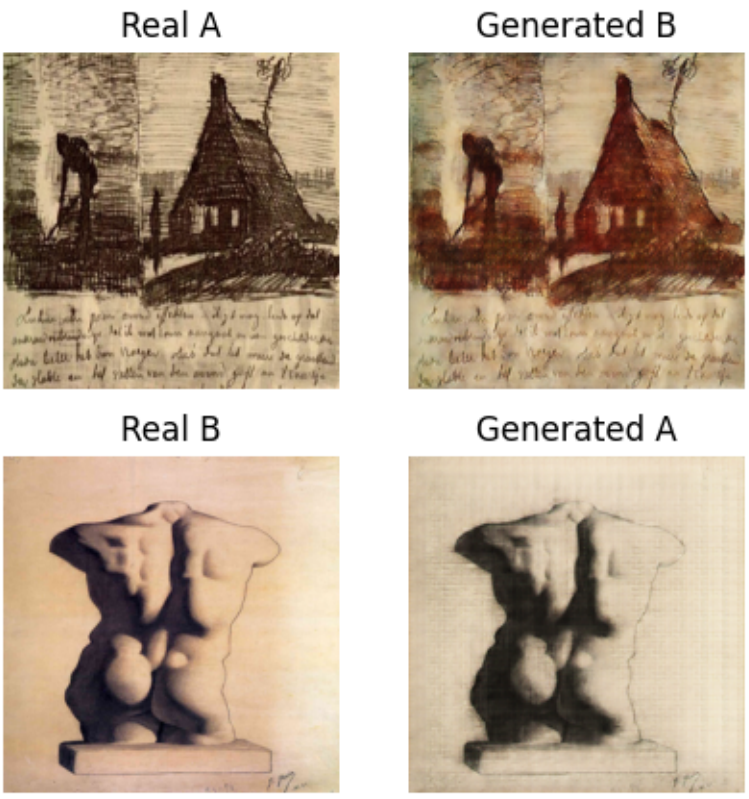

Real A
Generated B
Real B
Generated A

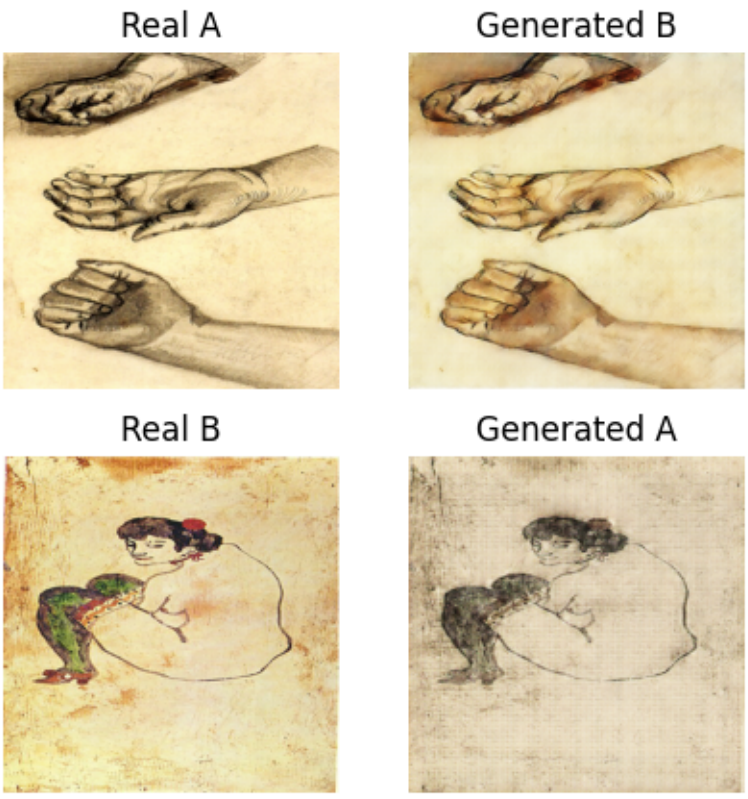

Real A
Generated B
Real B
Generated A


Real A
Generated B
Real B
Generated A


Real A
Generated B
Real B
Generated A


Real A
Generated B
Real B
Generated A

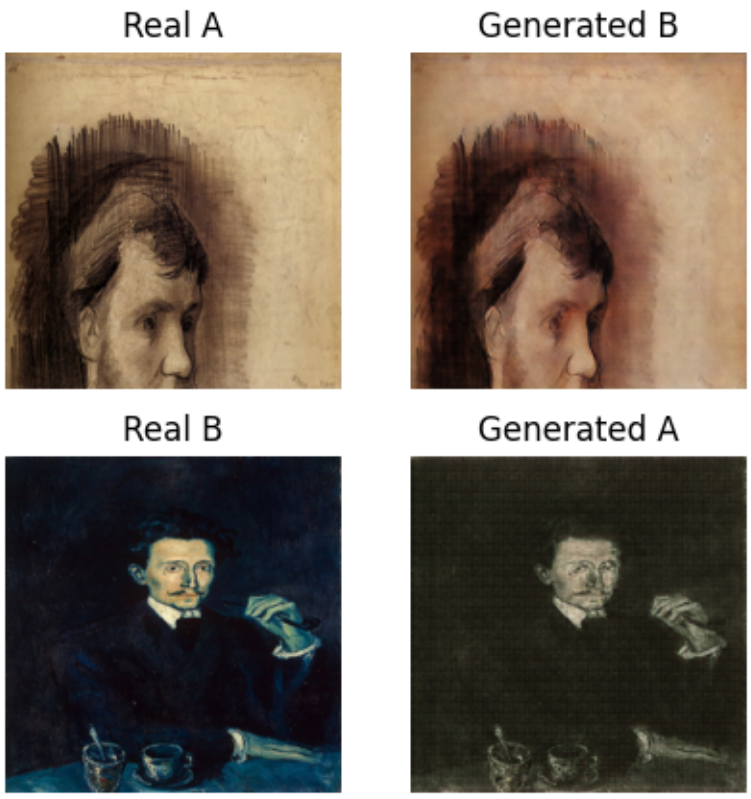

Real A
Generated B
Real B
Generated A

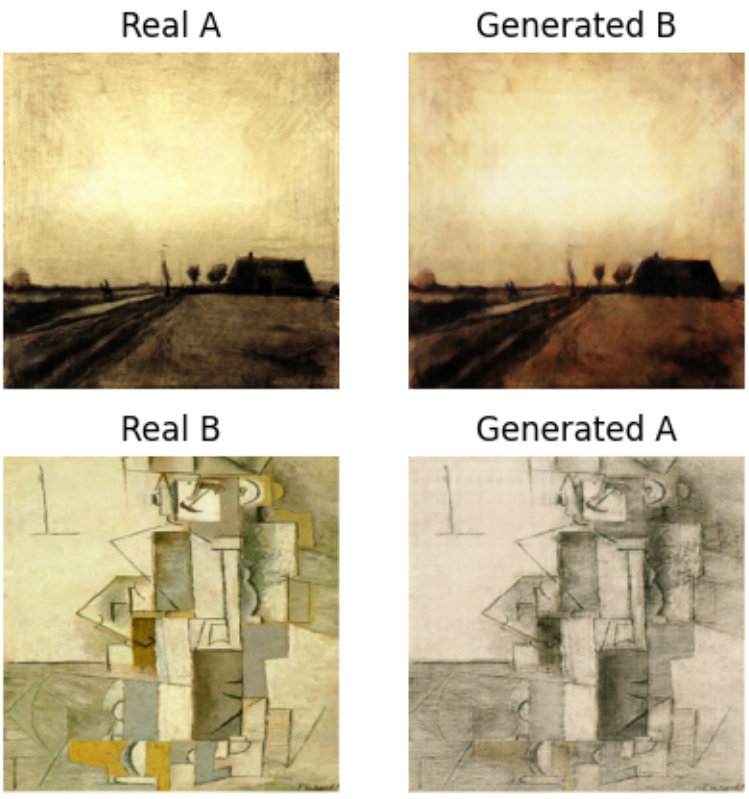


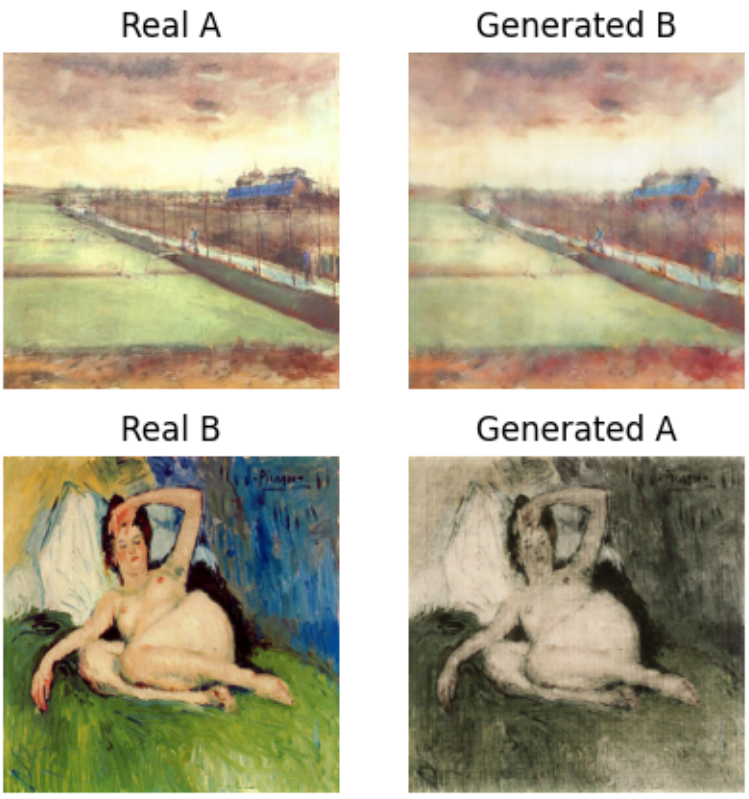
1.5.2 CycleGAN Evaluation

[ ]:
```
print("***** CycleGAN Images *****")

# Convert test datasets to TensorFlow datasets (take the first 50 images)
test_A = tf.data.Dataset.from_tensor_slices(vg_test).take(50).batch(1)
test_B = tf.data.Dataset.from_tensor_slices(p_test).take(50).batch(1)

# Test the model on the withheld dataset
for real_A, real_B in zip(test_A, test_B):
    generated_B = g_AB_n(real_A, training=False)
    generated_A = g_BA_n(real_B, training=False)
    display_images(real_A, generated_B, real_B, generated_A)
```

```
***** CycleGAN Images *****
```


Real A
Generated B
Real B
Generated A


Real A
Generated B
Real B
Generated A


Real A
Generated B
Real B
Generated A

Real A

Generated B

Real B

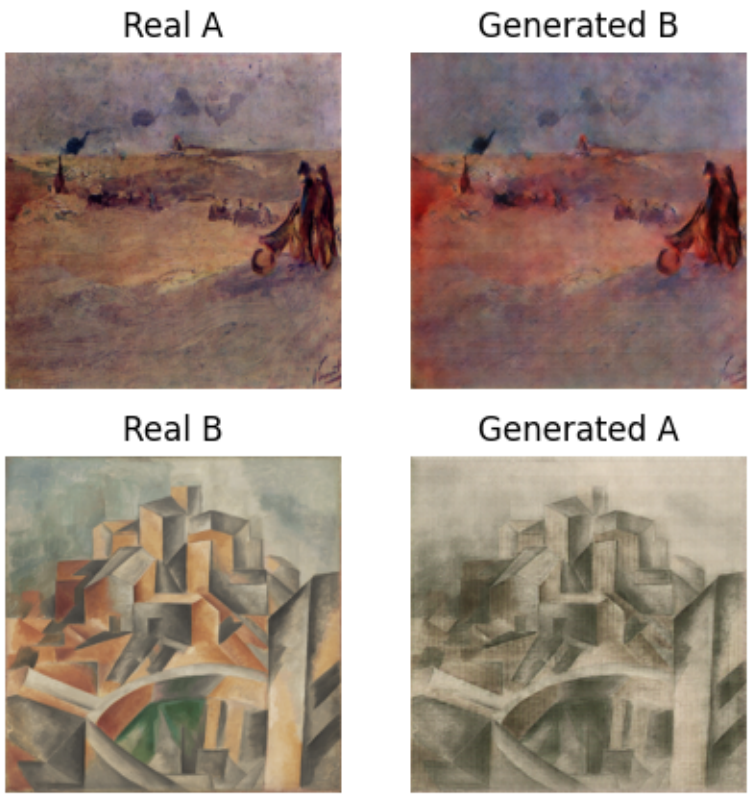
Generated A

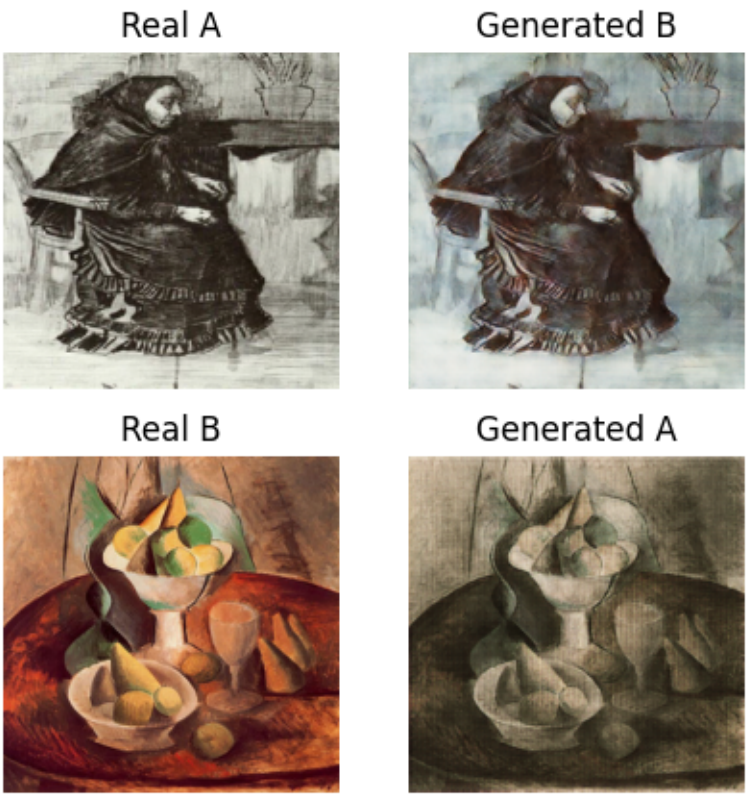
Real A

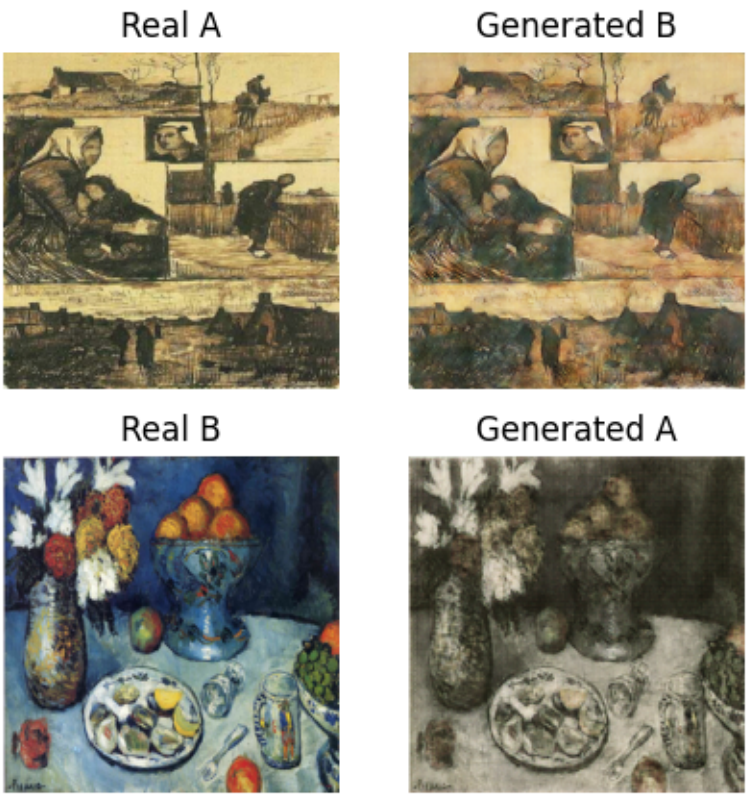
Generated B

Real B

Generated A


Real A
Generated B
Real B
Generated A


Real A
Generated B
Real B
Generated A

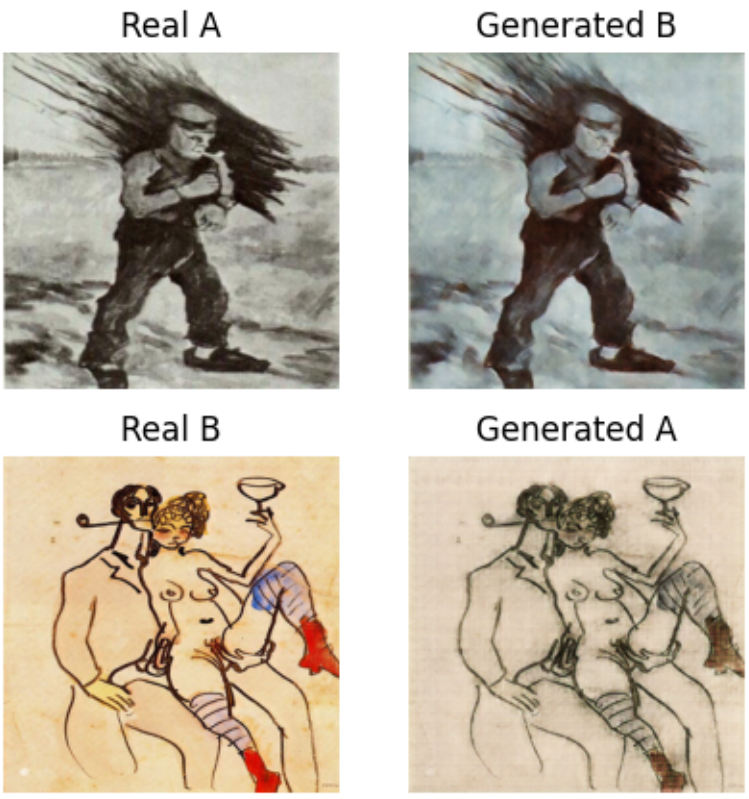

Real A
Generated B
Real B
Generated A

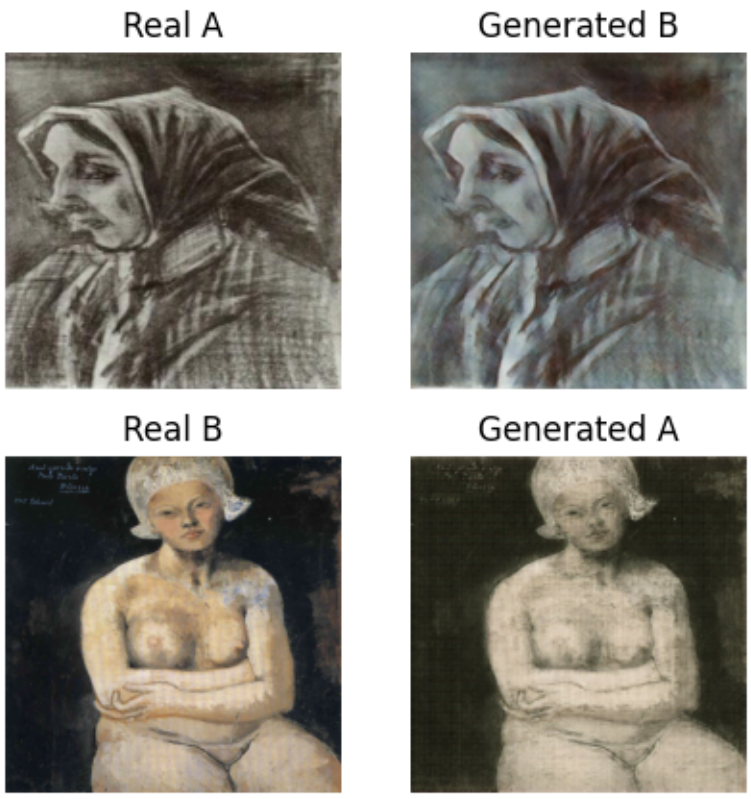
Real A

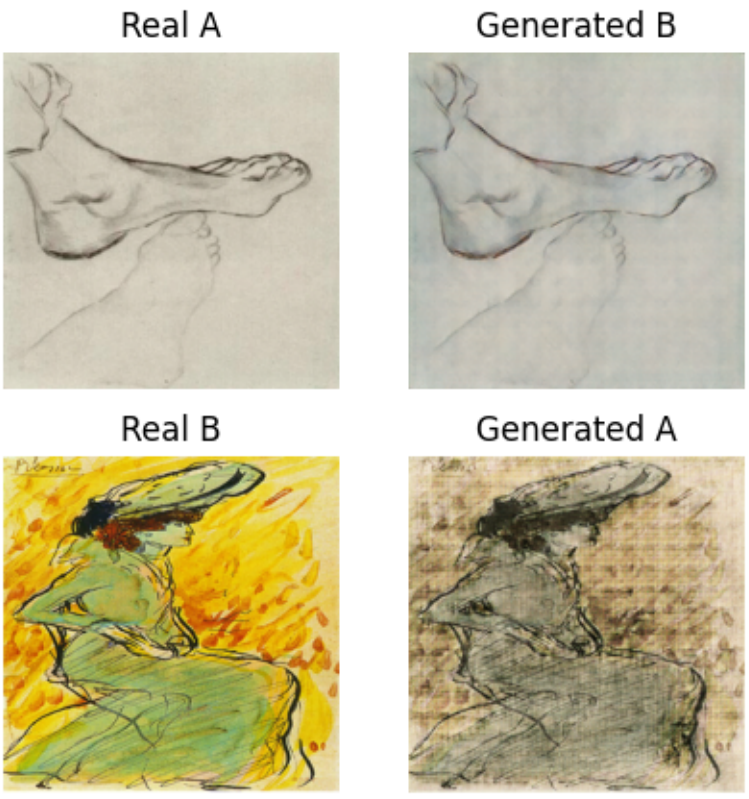
Generated B

Real B

Generated A


Real A
Generated B
Real B
Generated A


Real A
Generated B
Real B
Generated A


Real A
Generated B
Real B
Generated A

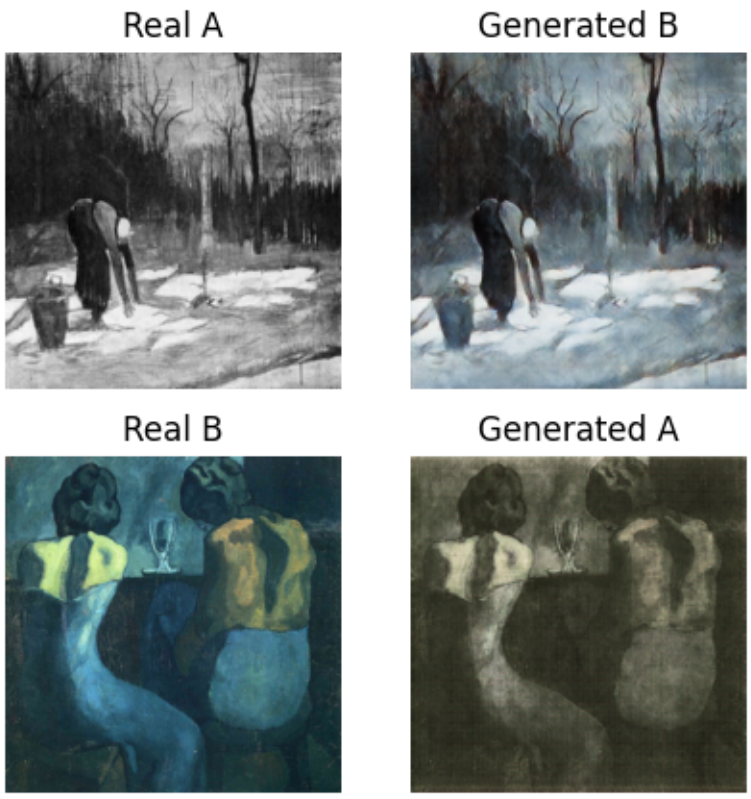

Real A
Generated B
Real B
Generated A

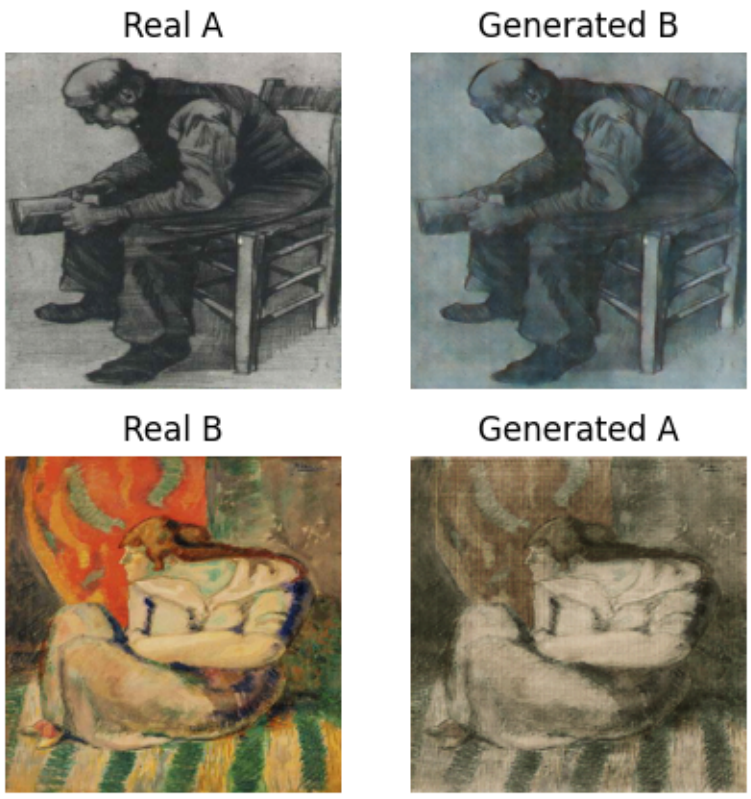
Real A

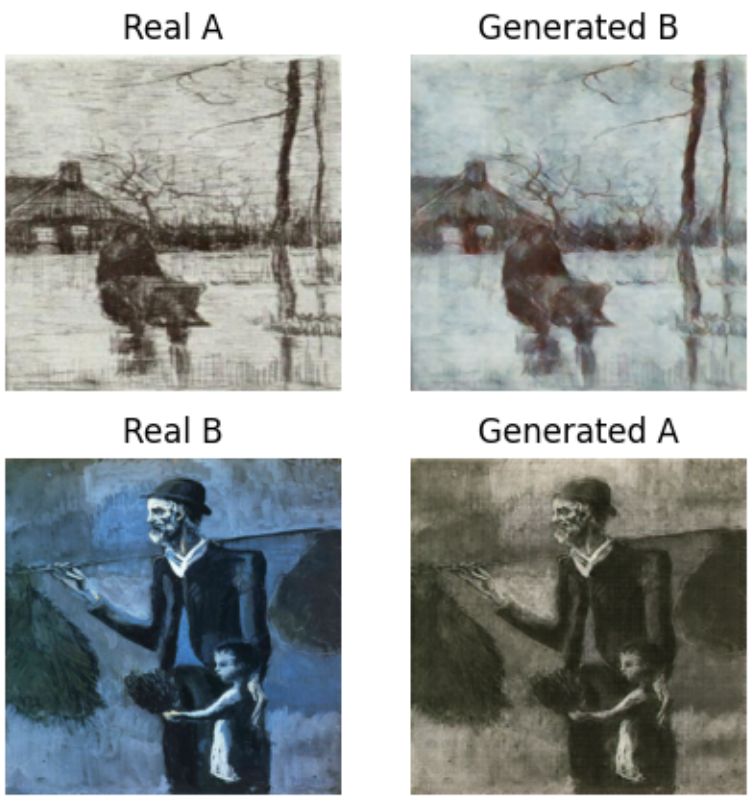
Generated B

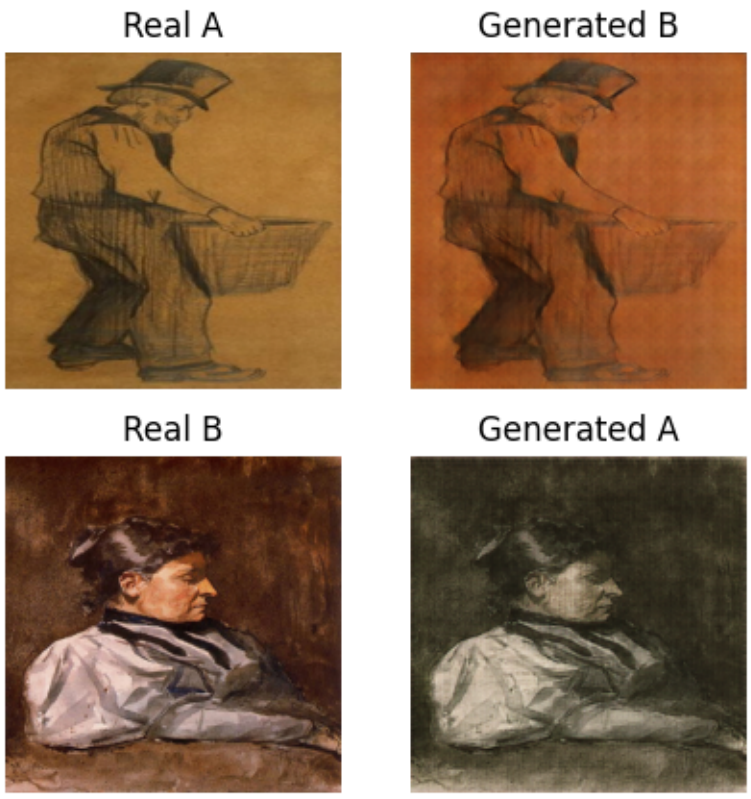
Real B

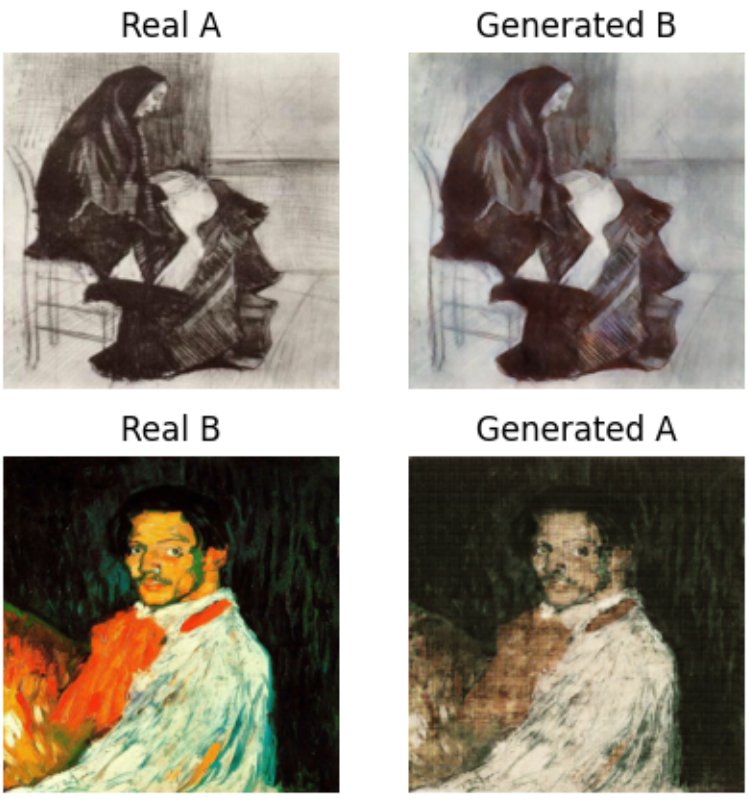
Generated A

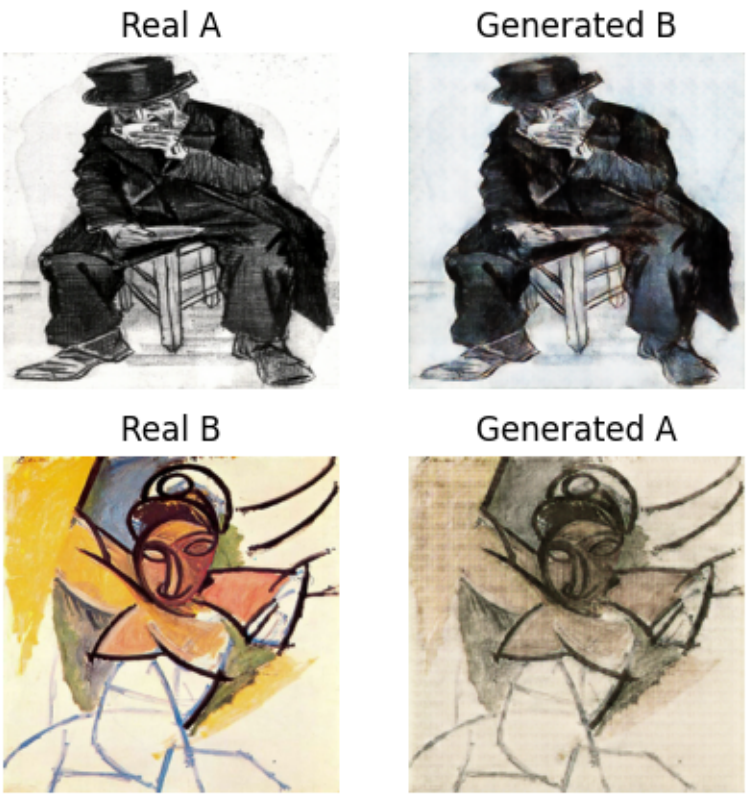

Real A
Generated B
Real B
Generated A

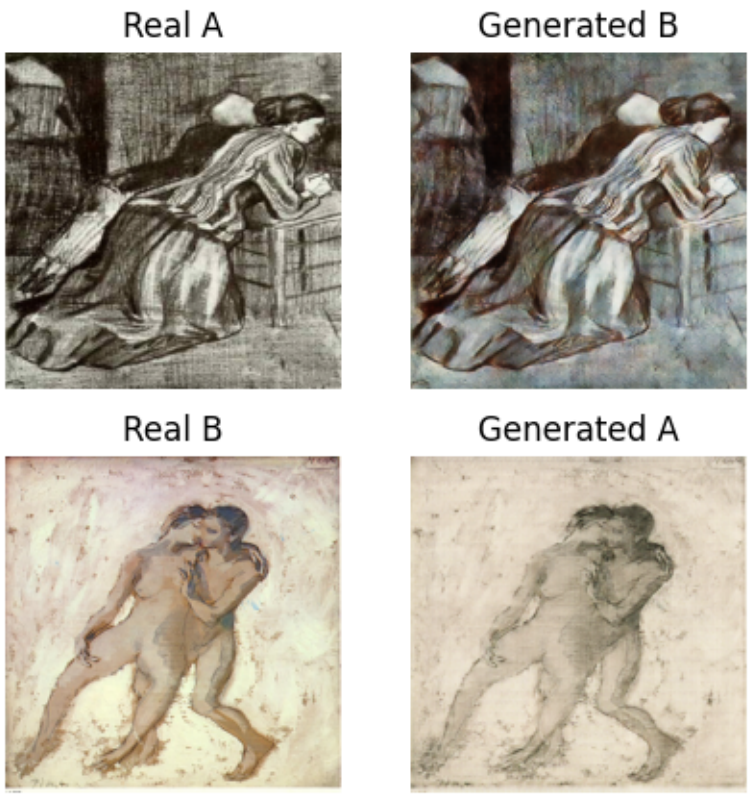

Real A
Generated B
Real B
Generated A


Real A
Generated B
Real B
Generated A


Real A
Generated B
Real B
Generated A

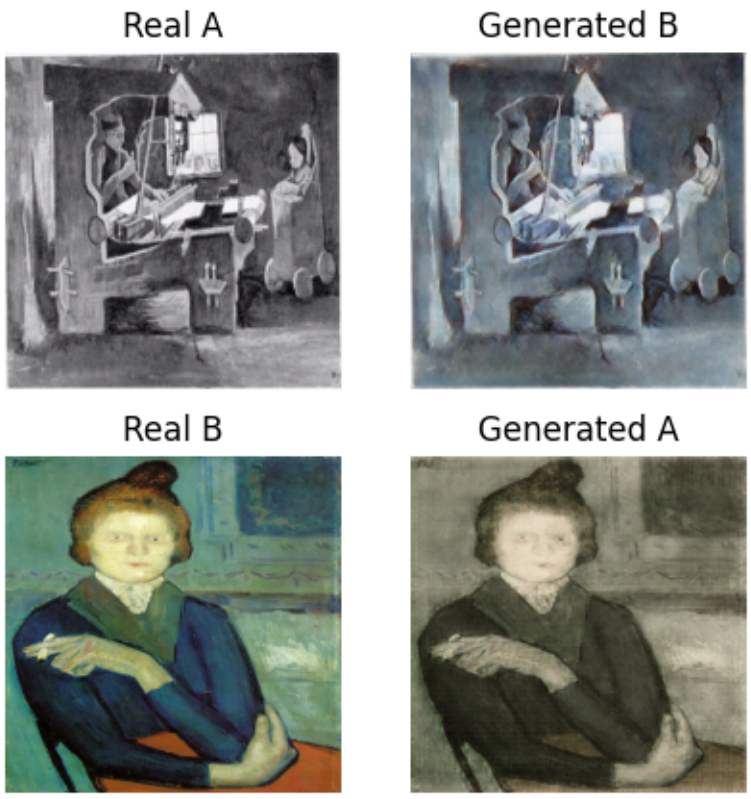

Real A
Generated B
Real B
Generated A

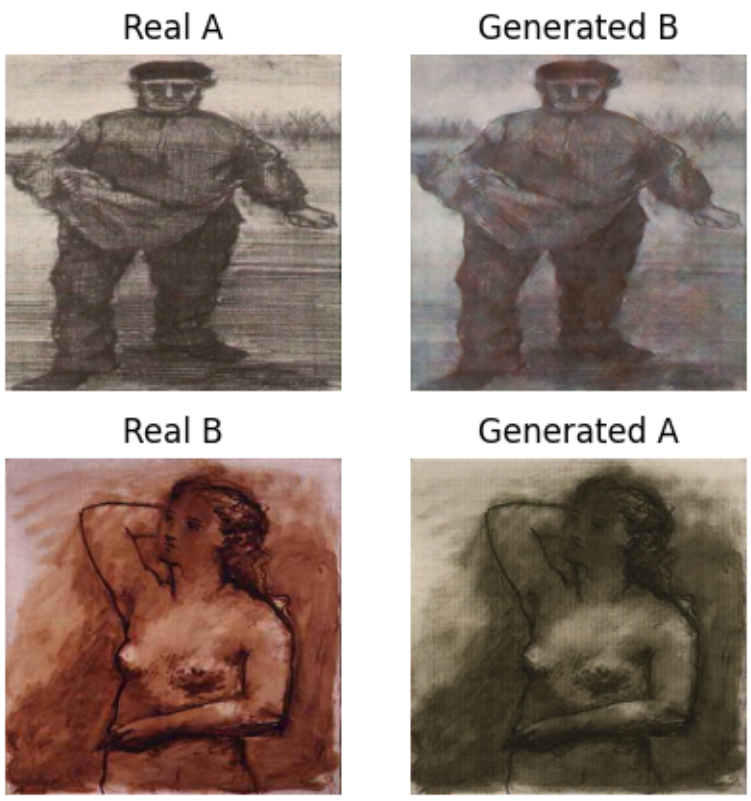

Real A
Generated B
Real B
Generated A


Real A
Generated B
Real B
Generated A


Real A
Generated B
Real B
Generated A

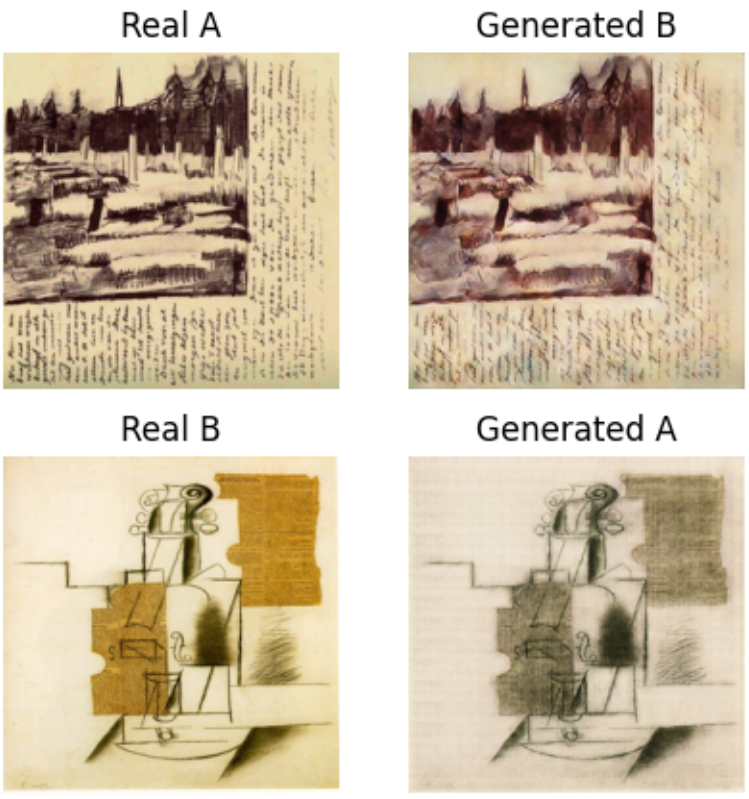

Real A
Generated B
Real B
Generated A

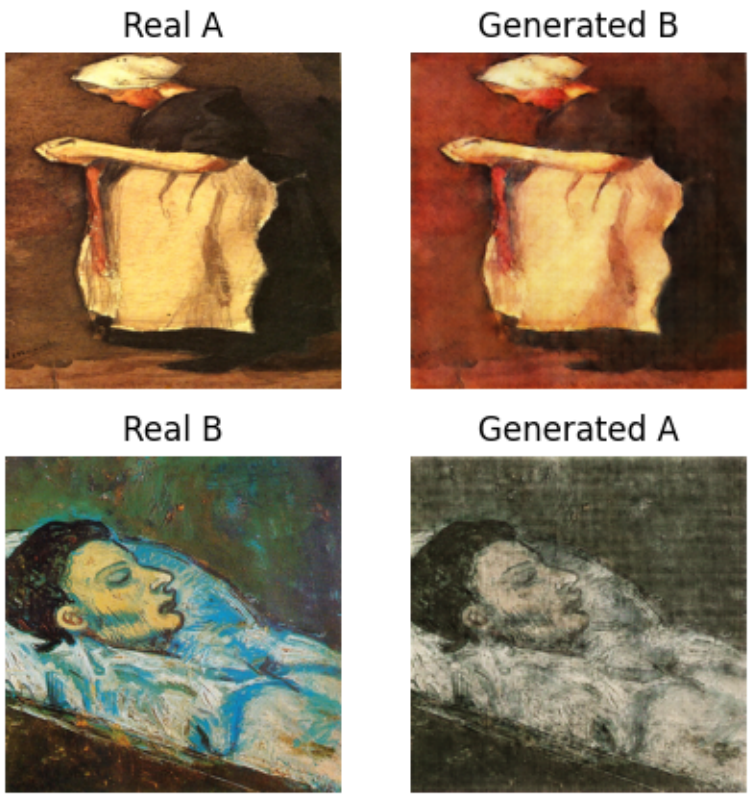

Real A
Generated B
Real B
Generated A

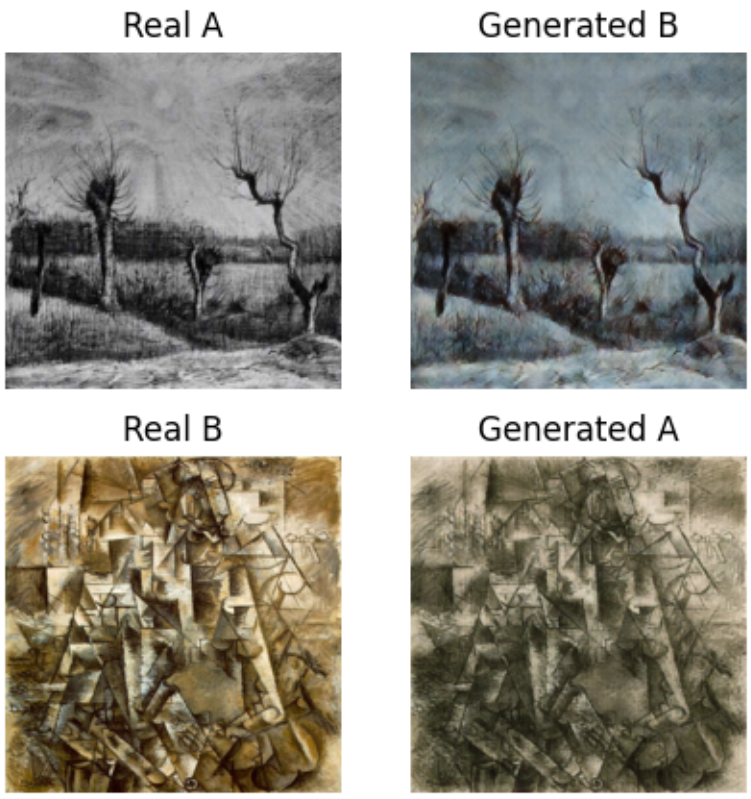

Real A
Generated B
Real B
Generated A

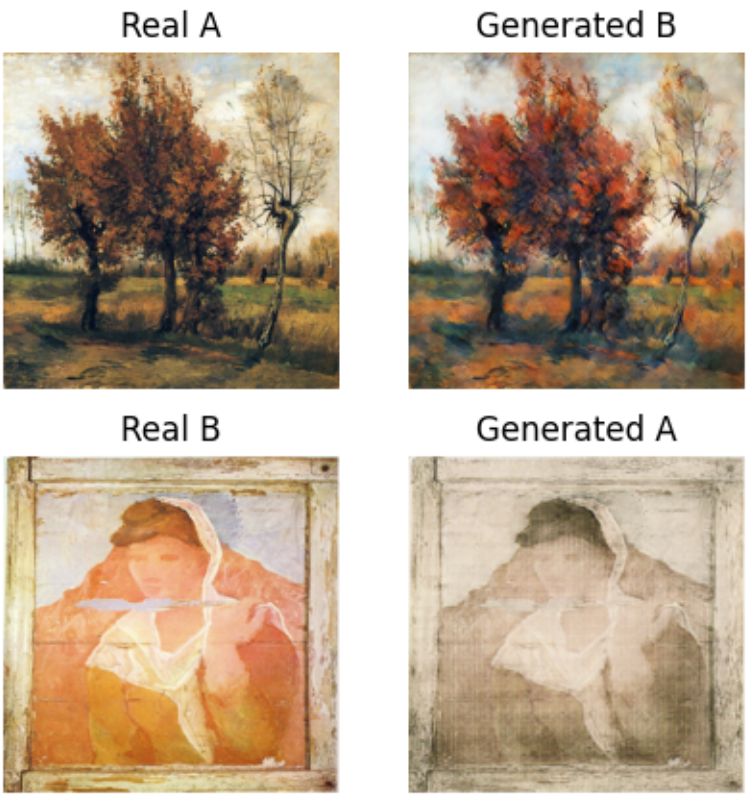

Real A
Generated B
Real B
Generated A

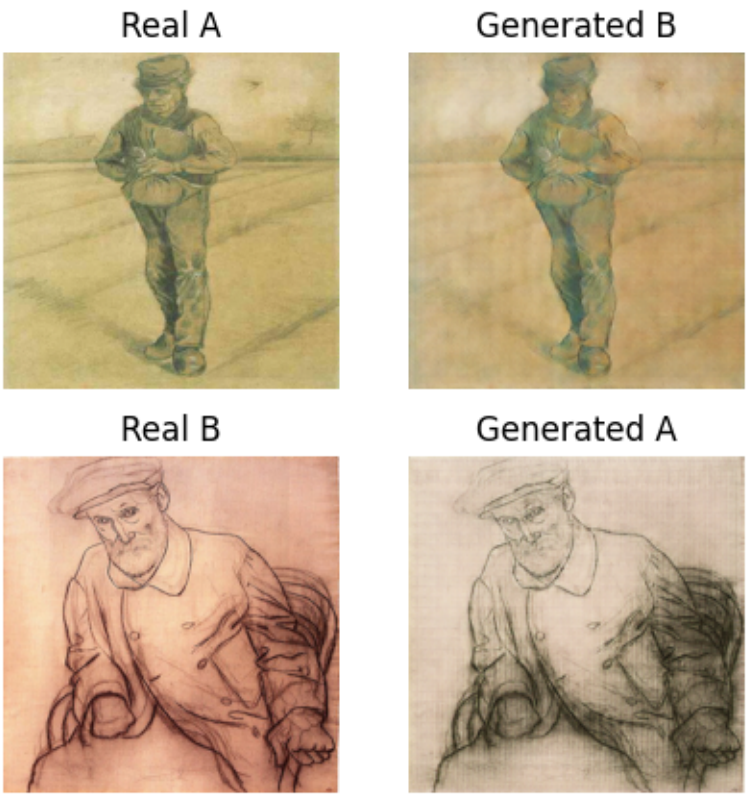

Real A
Generated B
Real B
Generated A

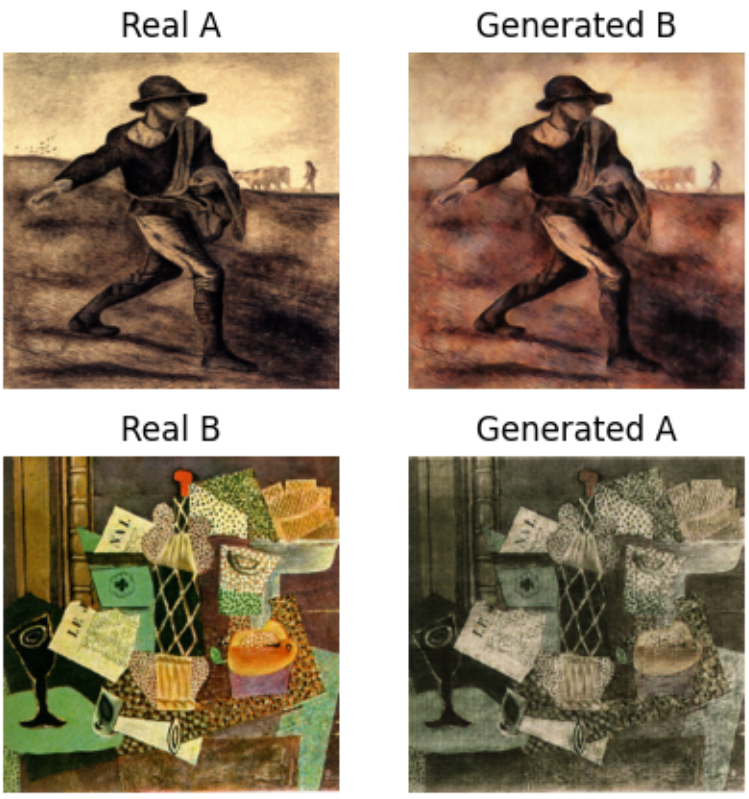

Real A
Generated B
Real B
Generated A

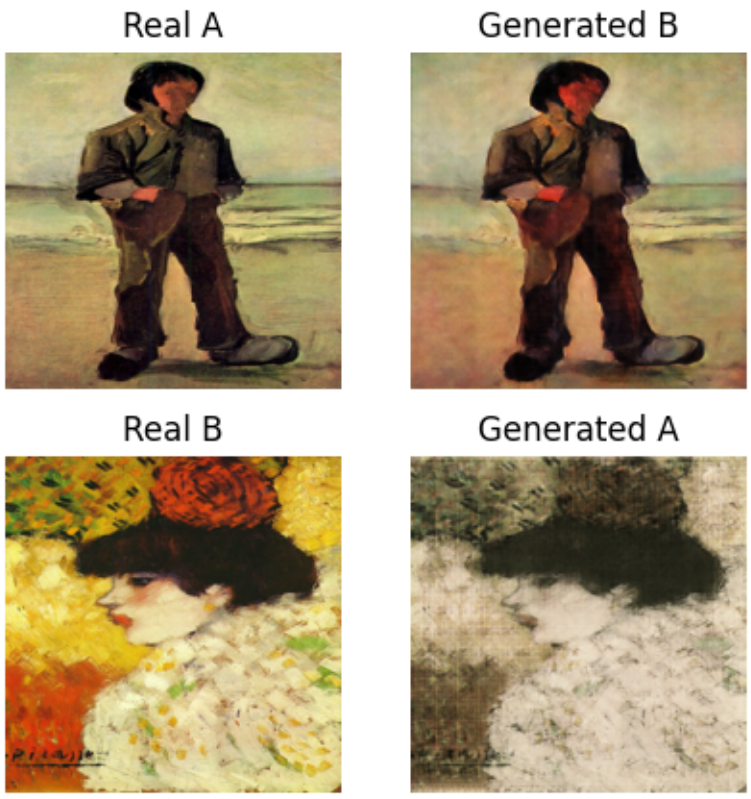

Real A
Generated B
Real B
Generated A

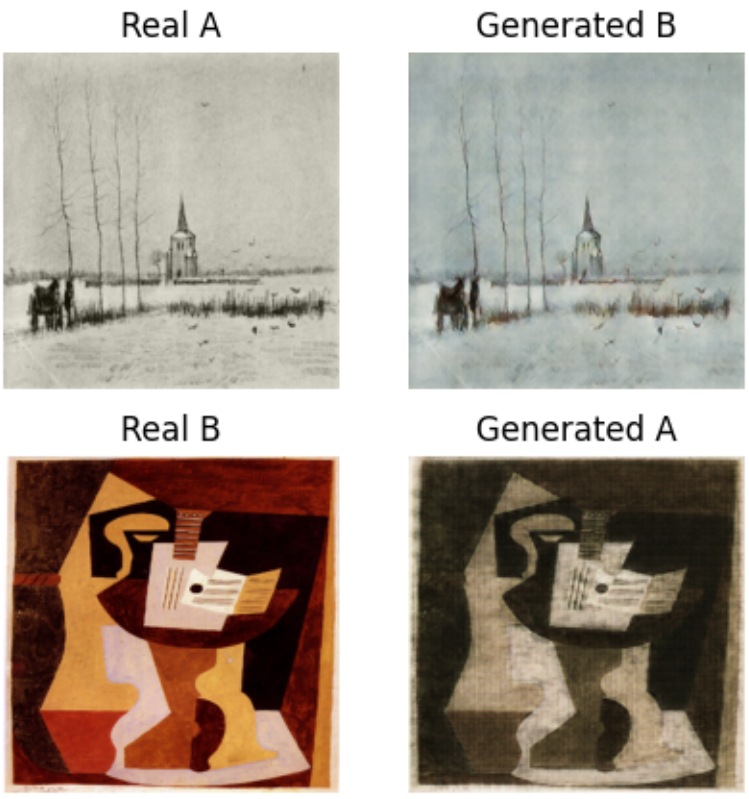

Real A
Generated B
Real B
Generated A


Real A
Generated B
Real B
Generated A


Real A
Generated B
Real B
Generated A


Real A
Generated B
Real B
Generated A


Real A
Generated B
Real B
Generated A


Real A
Generated B
Real B
Generated A


Real A
Generated B
Real B
Generated A

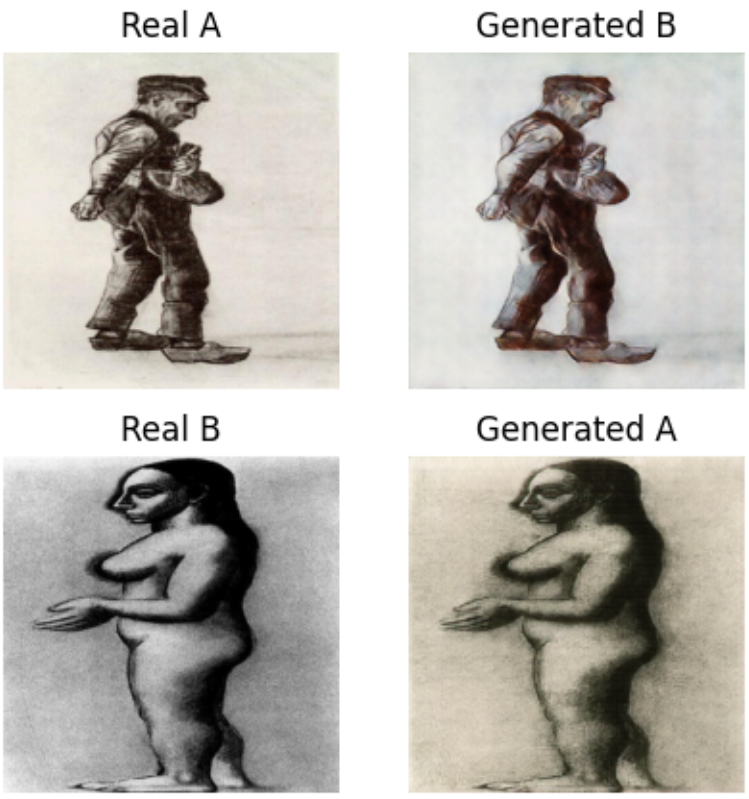

Real A
Generated B
Real B
Generated A

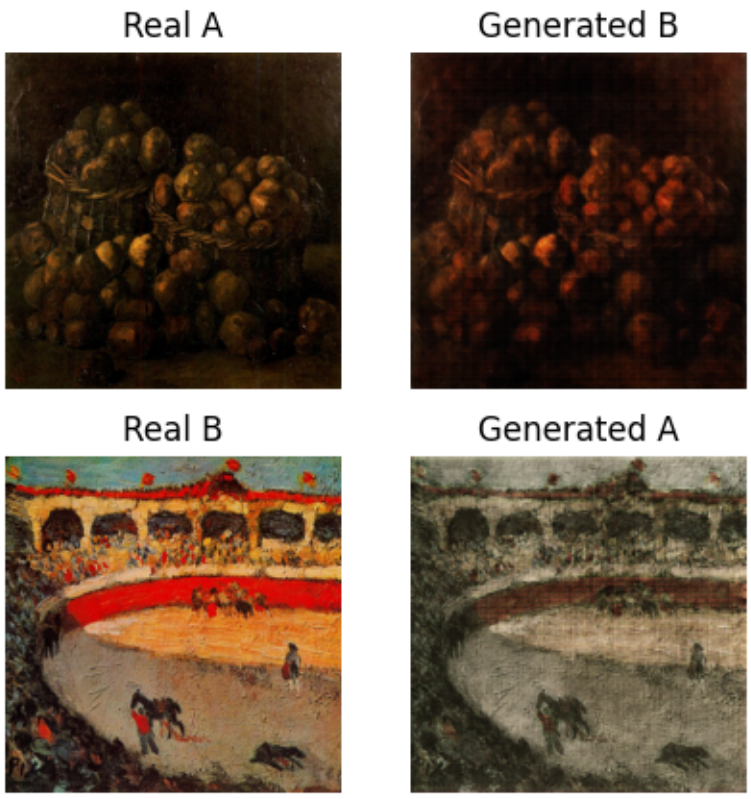
Real A

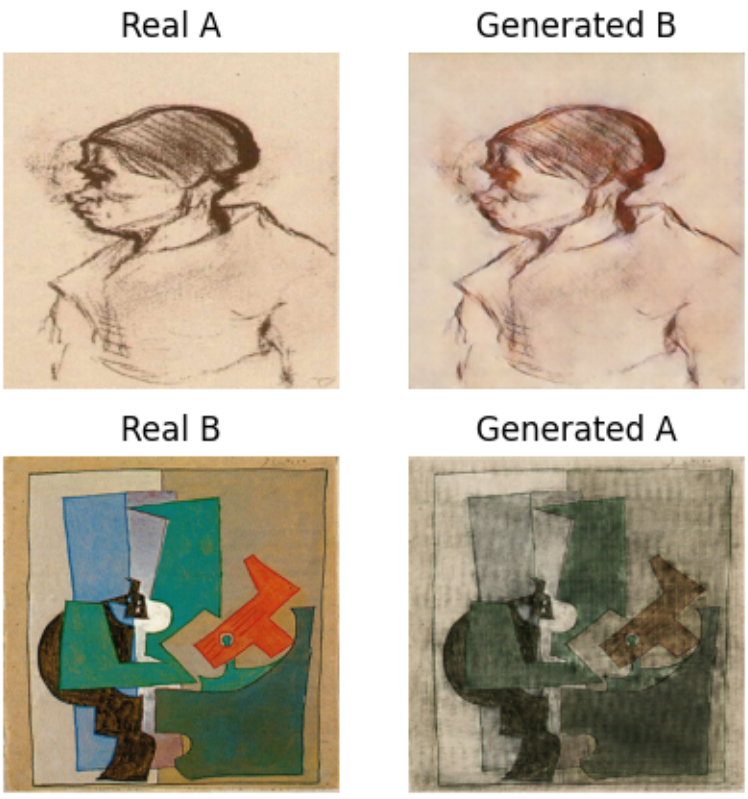
Generated B

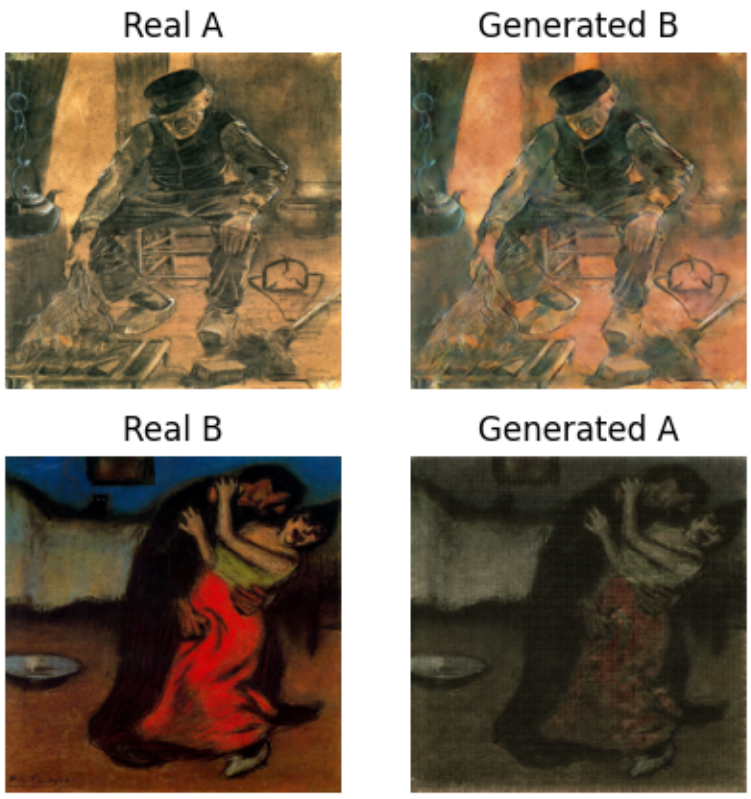
Real B

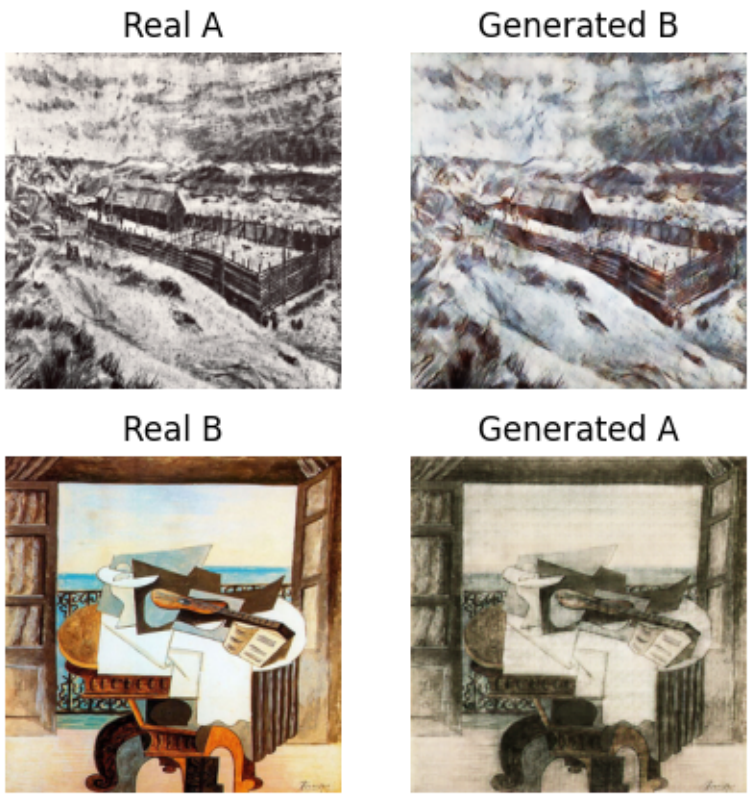
Generated A


Real A
Generated B
Real B
Generated A

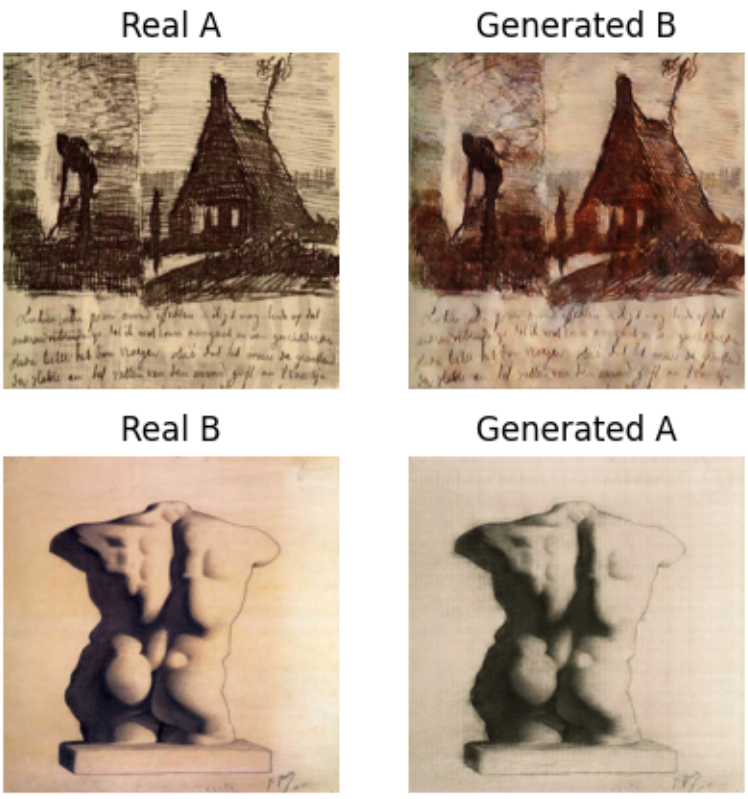

Real A
Generated B
Real B
Generated A

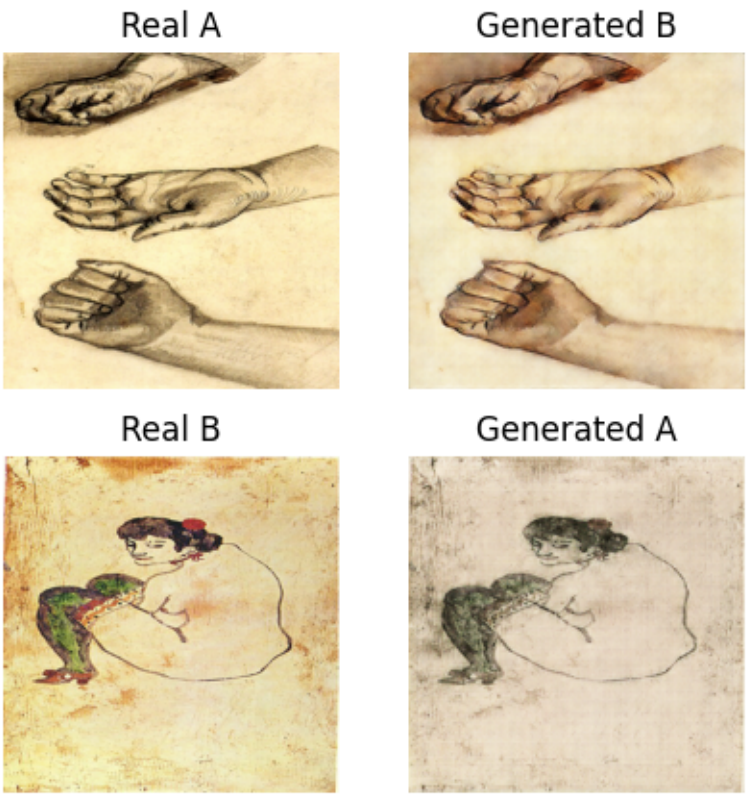

Real A
Generated B
Real B
Generated A

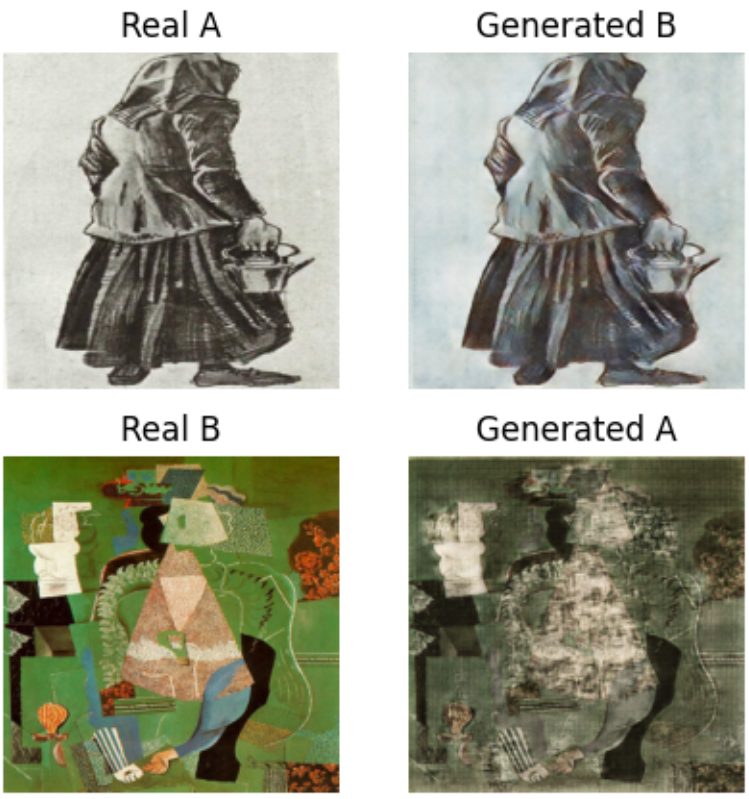

Real A
Generated B
Real B
Generated A

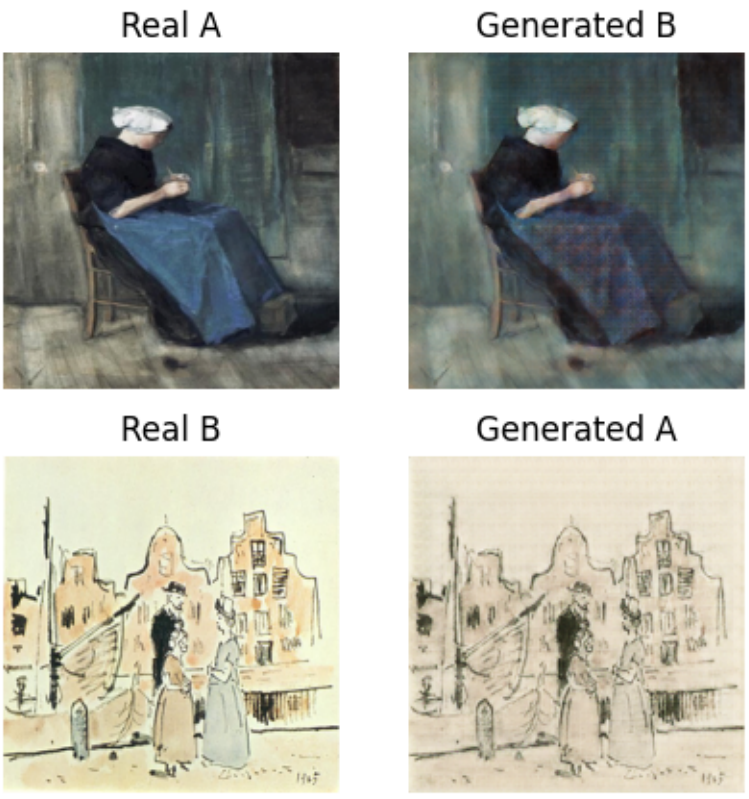

Real A
Generated B
Real B
Generated A


Real A
Generated B
Real B
Generated A


Real A
Generated B
Real B
Generated A


Real A
Generated B
Real B
Generated A


Real A
Generated B
Real B
Generated A

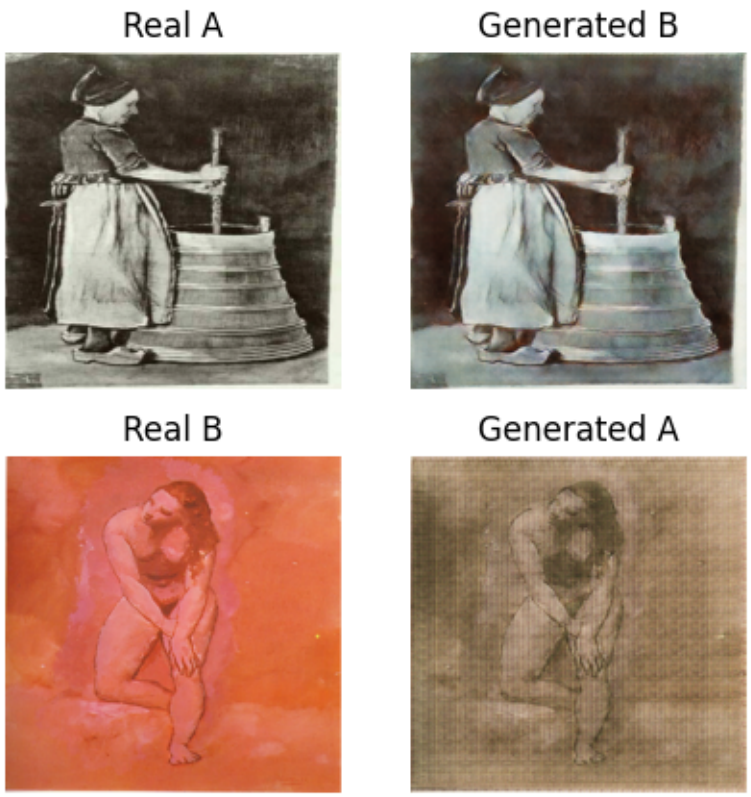

Real A
Generated B
Real B
Generated A

Real A

Generated B

Real B

Generated A



PDF

```
[ ]: !pip install nbconvert
     !sudo apt-get install texlive-xetex texlive-fonts-recommended␣
       ↪texlive-plain-generic

     !jupyter nbconvert --to pdf /content/ArtGAN.ipynb
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-
wheels/public/simple/
Requirement already satisfied: nbconvert in /usr/local/lib/python3.9/dist-
packages (6.5.4)
Requirement already satisfied: entrypoints>=0.2.2 in
/usr/local/lib/python3.9/dist-packages (from nbconvert) (0.4)
Requirement already satisfied: jinja2>=3.0 in /usr/local/lib/python3.9/dist-
packages (from nbconvert) (3.1.2)
Requirement already satisfied: lxml in /usr/local/lib/python3.9/dist-packages
(from nbconvert) (4.9.2)
Requirement already satisfied: nbformat>=5.1 in /usr/local/lib/python3.9/dist-
packages (from nbconvert) (5.8.0)
Requirement already satisfied: beautifulsoup4 in /usr/local/lib/python3.9/dist-
packages (from nbconvert) (4.11.2)
```

```
Requirement already satisfied: bleach in /usr/local/lib/python3.9/dist-packages
(from nbconvert) (6.0.0)
Requirement already satisfied: pandocfilters>=1.4.1 in
/usr/local/lib/python3.9/dist-packages (from nbconvert) (1.5.0)
Requirement already satisfied: mistune<2,>=0.8.1 in
/usr/local/lib/python3.9/dist-packages (from nbconvert) (0.8.4)
Requirement already satisfied: nbclient>=0.5.0 in /usr/local/lib/python3.9/dist-
packages (from nbconvert) (0.7.3)
Requirement already satisfied: jupyterlab-pygments in
/usr/local/lib/python3.9/dist-packages (from nbconvert) (0.2.2)
Requirement already satisfied: packaging in /usr/local/lib/python3.9/dist-
packages (from nbconvert) (23.0)
Requirement already satisfied: tinycss2 in /usr/local/lib/python3.9/dist-
packages (from nbconvert) (1.2.1)
Requirement already satisfied: pygments>=2.4.1 in /usr/local/lib/python3.9/dist-
packages (from nbconvert) (2.14.0)
Requirement already satisfied: jupyter-core>=4.7 in
/usr/local/lib/python3.9/dist-packages (from nbconvert) (5.3.0)
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.9/dist-
packages (from nbconvert) (2.1.2)
Requirement already satisfied: defusedxml in /usr/local/lib/python3.9/dist-
packages (from nbconvert) (0.7.1)
Requirement already satisfied: traitlets>=5.0 in /usr/local/lib/python3.9/dist-
packages (from nbconvert) (5.7.1)
Requirement already satisfied: platformdirs>=2.5 in
/usr/local/lib/python3.9/dist-packages (from jupyter-core>=4.7->nbconvert)
(3.2.0)
Requirement already satisfied: jupyter-client>=6.1.12 in
/usr/local/lib/python3.9/dist-packages (from nbclient>=0.5.0->nbconvert)
(6.1.12)
Requirement already satisfied: jsonschema>=2.6 in /usr/local/lib/python3.9/dist-
packages (from nbformat>=5.1->nbconvert) (4.3.3)
Requirement already satisfied: fastjsonschema in /usr/local/lib/python3.9/dist-
packages (from nbformat>=5.1->nbconvert) (2.16.3)
Requirement already satisfied: soupsieve>1.2 in /usr/local/lib/python3.9/dist-
packages (from beautifulsoup4->nbconvert) (2.4)
Requirement already satisfied: webencodings in /usr/local/lib/python3.9/dist-
packages (from bleach->nbconvert) (0.5.1)
Requirement already satisfied: six>=1.9.0 in /usr/local/lib/python3.9/dist-
packages (from bleach->nbconvert) (1.16.0)
Requirement already satisfied: pyrsistent!=0.17.0,!=0.17.1,!=0.17.2,>=0.14.0 in
/usr/local/lib/python3.9/dist-packages (from
jsonschema>=2.6->nbformat>=5.1->nbconvert) (0.19.3)
Requirement already satisfied: attrs>=17.4.0 in /usr/local/lib/python3.9/dist-
packages (from jsonschema>=2.6->nbformat>=5.1->nbconvert) (22.2.0)
Requirement already satisfied: pyzmq>=13 in /usr/local/lib/python3.9/dist-
packages (from jupyter-client>=6.1.12->nbclient>=0.5.0->nbconvert) (23.2.1)
Requirement already satisfied: python-dateutil>=2.1 in
```

```
/usr/local/lib/python3.9/dist-packages (from jupyter-
client>=6.1.12->nbclient>=0.5.0->nbconvert) (2.8.2)
Requirement already satisfied: tornado>=4.1 in /usr/local/lib/python3.9/dist-
packages (from jupyter-client>=6.1.12->nbclient>=0.5.0->nbconvert) (6.2)
Reading package lists… Done
Building dependency tree
Reading state information… Done
texlive-fonts-recommended is already the newest version (2019.20200218-1).
texlive-plain-generic is already the newest version (2019.202000218-1).
texlive-xetex is already the newest version (2019.20200218-1).
0 upgraded, 0 newly installed, 0 to remove and 24 not upgraded.
[NbConvertApp] Converting notebook /content/ArtGAN.ipynb to pdf
[NbConvertApp] Support files will be in ArtGAN_files/
[NbConvertApp] Making directory ./ArtGAN_files
[NbConvertApp] Making directory ./ArtGAN_files
[NbConvertApp] Making directory ./ArtGAN_files
[NbConvertApp] Making directory ./ArtGAN_files
[NbConvertApp] Making directory ./ArtGAN_files
[NbConvertApp] Making directory ./ArtGAN_files
[NbConvertApp] Making directory ./ArtGAN_files
[NbConvertApp] Making directory ./ArtGAN_files
[NbConvertApp] Making directory ./ArtGAN_files
[NbConvertApp] Making directory ./ArtGAN_files
[NbConvertApp] Making directory ./ArtGAN_files
[NbConvertApp] Making directory ./ArtGAN_files
[NbConvertApp] Making directory ./ArtGAN_files
[NbConvertApp] Making directory ./ArtGAN_files
[NbConvertApp] Making directory ./ArtGAN_files
[NbConvertApp] Making directory ./ArtGAN_files
[NbConvertApp] Making directory ./ArtGAN_files
[NbConvertApp] Making directory ./ArtGAN_files
[NbConvertApp] Making directory ./ArtGAN_files
[NbConvertApp] Making directory ./ArtGAN_files
[NbConvertApp] Making directory ./ArtGAN_files
[NbConvertApp] Making directory ./ArtGAN_files
[NbConvertApp] Making directory ./ArtGAN_files
[NbConvertApp] Making directory ./ArtGAN_files
[NbConvertApp] Making directory ./ArtGAN_files
[NbConvertApp] Making directory ./ArtGAN_files
[NbConvertApp] Making directory ./ArtGAN_files
[NbConvertApp] Making directory ./ArtGAN_files
[NbConvertApp] Making directory ./ArtGAN_files
[NbConvertApp] Making directory ./ArtGAN_files
[NbConvertApp] Making directory ./ArtGAN_files
[NbConvertApp] Making directory ./ArtGAN_files
[NbConvertApp] Making directory ./ArtGAN_files
[NbConvertApp] Making directory ./ArtGAN_files
```

```
[NbConvertApp] Making directory ./ArtGAN_files
[NbConvertApp] Making directory ./ArtGAN_files
[NbConvertApp] Making directory ./ArtGAN_files
[NbConvertApp] Making directory ./ArtGAN_files
[NbConvertApp] Making directory ./ArtGAN_files
[NbConvertApp] Making directory ./ArtGAN_files
[NbConvertApp] Making directory ./ArtGAN_files
[NbConvertApp] Making directory ./ArtGAN_files
[NbConvertApp] Making directory ./ArtGAN_files
[NbConvertApp] Making directory ./ArtGAN_files
[NbConvertApp] Making directory ./ArtGAN_files
[NbConvertApp] Making directory ./ArtGAN_files
[NbConvertApp] Making directory ./ArtGAN_files
[NbConvertApp] Making directory ./ArtGAN_files
[NbConvertApp] Making directory ./ArtGAN_files
[NbConvertApp] Making directory ./ArtGAN_files
[NbConvertApp] Making directory ./ArtGAN_files
[NbConvertApp] Making directory ./ArtGAN_files
[NbConvertApp] Making directory ./ArtGAN_files
[NbConvertApp] Making directory ./ArtGAN_files
[NbConvertApp] Making directory ./ArtGAN_files
[NbConvertApp] Making directory ./ArtGAN_files
[NbConvertApp] Making directory ./ArtGAN_files
[NbConvertApp] Making directory ./ArtGAN_files
[NbConvertApp] Making directory ./ArtGAN_files
[NbConvertApp] Making directory ./ArtGAN_files
[NbConvertApp] Making directory ./ArtGAN_files
[NbConvertApp] Making directory ./ArtGAN_files
[NbConvertApp] Making directory ./ArtGAN_files
[NbConvertApp] Making directory ./ArtGAN_files
[NbConvertApp] Making directory ./ArtGAN_files
[NbConvertApp] Making directory ./ArtGAN_files
[NbConvertApp] Making directory ./ArtGAN_files
[NbConvertApp] Making directory ./ArtGAN_files
[NbConvertApp] Making directory ./ArtGAN_files
[NbConvertApp] Making directory ./ArtGAN_files
[NbConvertApp] Making directory ./ArtGAN_files
[NbConvertApp] Making directory ./ArtGAN_files
[NbConvertApp] Making directory ./ArtGAN_files
[NbConvertApp] Making directory ./ArtGAN_files
[NbConvertApp] Making directory ./ArtGAN_files
[NbConvertApp] Making directory ./ArtGAN_files
[NbConvertApp] Making directory ./ArtGAN_files
[NbConvertApp] Making directory ./ArtGAN_files
[NbConvertApp] Making directory ./ArtGAN_files
[NbConvertApp] Making directory ./ArtGAN_files
[NbConvertApp] Making directory ./ArtGAN_files
[NbConvertApp] Making directory ./ArtGAN_files
```

```
[NbConvertApp] Making directory ./ArtGAN_files
[NbConvertApp] Making directory ./ArtGAN_files
[NbConvertApp] Making directory ./ArtGAN_files
[NbConvertApp] Making directory ./ArtGAN_files
[NbConvertApp] Making directory ./ArtGAN_files
[NbConvertApp] Making directory ./ArtGAN_files
[NbConvertApp] Making directory ./ArtGAN_files
[NbConvertApp] Making directory ./ArtGAN_files
[NbConvertApp] Making directory ./ArtGAN_files
[NbConvertApp] Making directory ./ArtGAN_files
[NbConvertApp] Making directory ./ArtGAN_files
[NbConvertApp] Making directory ./ArtGAN_files
[NbConvertApp] Making directory ./ArtGAN_files
[NbConvertApp] Writing 259050 bytes to notebook.tex
[NbConvertApp] Building PDF
[NbConvertApp] Running xelatex 3 times: ['xelatex', 'notebook.tex', '-quiet']
[NbConvertApp] Running bibtex 1 time: ['bibtex', 'notebook']
[NbConvertApp] WARNING | bibtex had problems, most likely because there were no
citations
[NbConvertApp] PDF successfully created
[NbConvertApp] Writing 23888065 bytes to /content/ArtGAN.pdf
```