1 Import Module

[2]:
```
import os
import numpy as np
import matplotlib.pyplot as plt
import warnings
from tqdm.notebook import tqdm

import tensorflow as tf
from tensorflow import keras
from tensorflow.keras.preprocessing.image import load_img, array_to_img
from tensorflow.keras.models import Sequential, Model
from tensorflow.keras import layers
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.losses import BinaryCrossentropy

warnings.filterwarnings('ignore')
```

2 Load the data

[3]:
```
BASE_DIR = '/kaggle/input/anime-faces/data/'
```

[4]:
```
# load complete image paths to the list
image_paths = []
for image_name in os.listdir(BASE_DIR):
    image_path = os.path.join(BASE_DIR, image_name)
    image_paths.append(image_path)
```

[5]:
```
image_paths[:5]
```

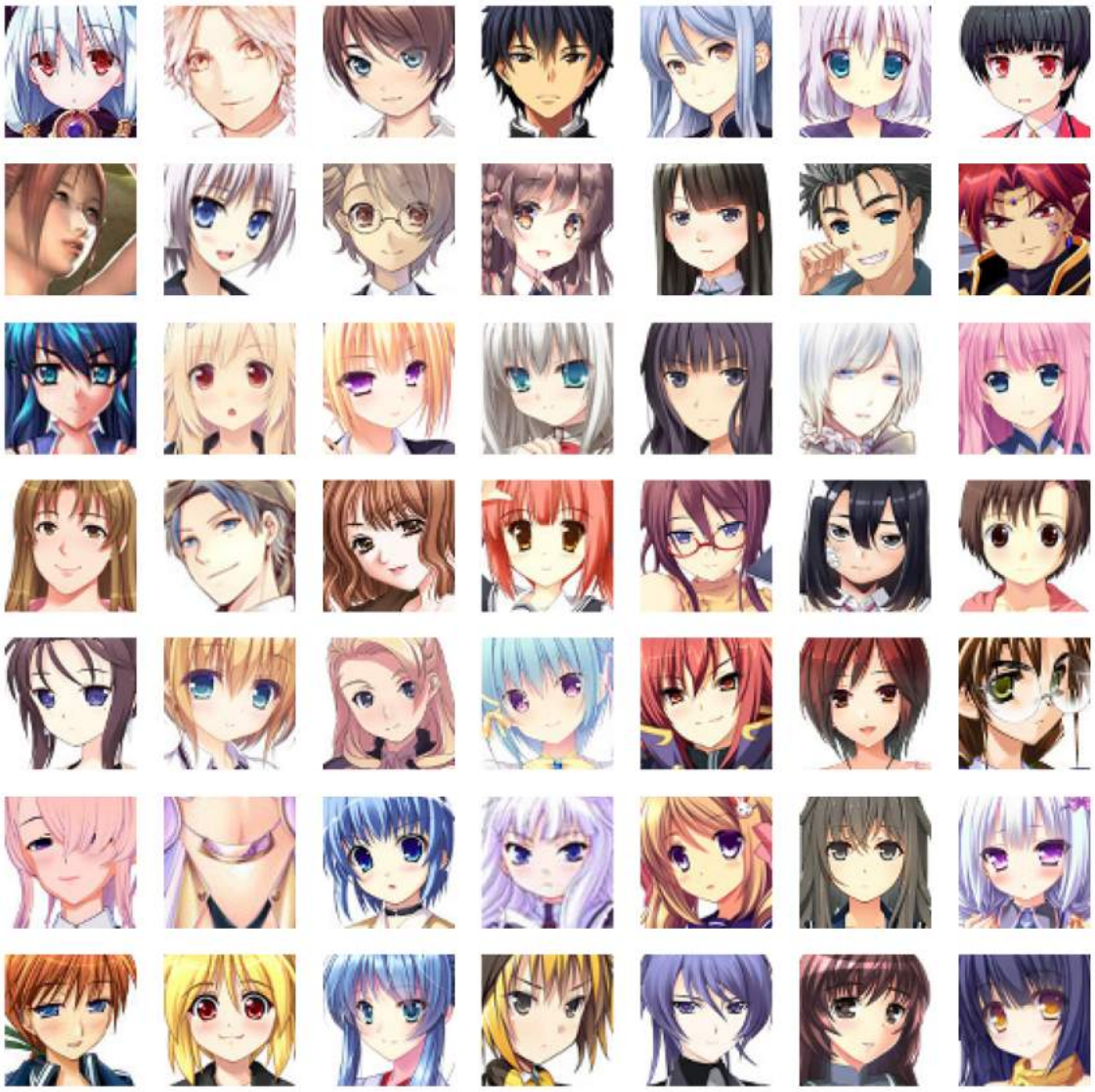
[5]:
```
['/kaggle/input/anime-faces/data/21130.png',
 '/kaggle/input/anime-faces/data/9273.png',
 '/kaggle/input/anime-faces/data/18966.png',
 '/kaggle/input/anime-faces/data/14127.png',
 '/kaggle/input/anime-faces/data/18054.png']
```

[6]:
```
# remove unnecessary file
image_paths.remove('/kaggle/input/anime-faces/data/data')
```

3 Visualize the Image Dataset

[7]:
```
# to display grid of images (7x7)
plt.figure(figsize=(20, 20))
temp_images = image_paths[:49]
index = 1

for image_path in temp_images:
    plt.subplot(7, 7, index)
    # load the image
    img = load_img(image_path)
    # convert to numpy array
    img = np.array(img)
    # show the image
    plt.imshow(img)
    plt.axis('off')
    # increment the index for next image
    index += 1
```



4 Preprocess the Images

[8]:
```
# load the image and convert to numpy array
train_images = [np.array(load_img(path)) for path in tqdm(image_paths)]
train_images = np.array(train_images)
```

```
  0%|          | 0/21551 [00:00<?, ?it/s]
```

[9]:
```
train_images[0].shape
```

[9]:
```
(64, 64, 3)
```

[10]:
```
# Reshape the array
train_images = train_images.reshape(train_images.shape[0], 64, 64, 3).␣
↪astype('float32')
```

[11]:
```
# Normalise the images
train_images = (train_images - 127.5) / 127.5
```

[12]:
```
train_images[0]
```

[12]:
```
array([[[-0.7254902 , -0.90588236, -0.7176471 ],
        [-0.70980394, -0.88235295, -0.6862745 ],
        [-0.7019608 , -1.        , -0.7176471 ],
        …,
        [-0.09019608,  0.01176471,  0.3882353 ],
        [-0.19215687,  0.05882353,  0.4509804 ],
        [-0.34117648, -0.13725491,  0.29411766]],

       [[-0.7254902 , -0.90588236, -0.7176471 ],
        [-0.73333335, -0.8666667 , -0.7019608 ],
        [-0.52156866, -0.8666667 , -0.5686275 ],
        …,
        [-0.29411766,  0.01176471,  0.3019608 ],
        [-0.08235294,  0.15294118,  0.5294118 ],
        [-0.23921569,  0.09019608,  0.45882353]],

       [[-0.7019608 , -0.8901961 , -0.7019608 ],
        [-0.7176471 , -0.8745098 , -0.69411767],
        [-0.5686275 , -0.8509804 , -0.58431375],
        …,
        [-0.22352941,  0.1764706 ,  0.48235294],
        [-0.13725491,  0.1764706 ,  0.4745098 ],
        [-0.06666667,  0.24705882,  0.5058824 ]],

       …,

       [[-0.30980393, -0.6862745 , -0.29411766],
        [-0.34901962, -0.6862745 , -0.37254903],
        [-0.30980393, -0.5372549 , -0.35686275],
        …,
        [-0.79607844, -0.9843137 , -0.77254903],
        [-0.827451  , -0.8745098 , -0.85882354],
        [-0.7647059 , -0.92156863, -0.8039216 ]],

       [[-0.3254902 , -0.6392157 , -0.29411766],
        [-0.44313726, -0.7254902 , -0.43529412],
        [-0.54509807, -0.7882353 , -0.54509807],
        …,
```

```
        [-0.8352941 , -0.9764706 , -0.70980394],
        [-0.7254902 , -0.8745098 , -0.75686276],
        [-0.7411765 , -0.9137255 , -0.69411767]],

       [[-0.5764706 , -0.827451  , -0.54509807],
        [-0.56078434, -0.8666667 , -0.5294118 ],
        [-0.58431375, -0.85882354, -0.5764706 ],
        …,
        [-0.7411765 , -0.8666667 , -0.6156863 ],
        [-0.7254902 , -0.9372549 , -0.70980394],
        [-0.7647059 , -0.92941177, -0.7176471 ]]], dtype=float32)
```

4.1 Create Generator & Dicriminator Models

[13]:
```
# latent dimension for random noise (helps in variety of imgages to be␣
 ↪developed)
LATENT_DIM = 100
# Weight initializer
WEIGHT_INIT = keras.initializers.RandomNormal(mean = 0.0, stddev = 0.02)
# No. of channels of the image
CHANNELS = 3
```

5 Generator Model

[14]:
```
model = Sequential(name = 'generator')

# 1D random noise
model.add(layers.Dense(8 * 8 * 512, input_dim = LATENT_DIM))
model.add(layers.BatchNormalization())
model.add(layers.ReLU())

# Convert 1D to 3D
model.add(layers.Reshape((8, 8, 512)))

# Upsample to 16x16
model.add(layers.Conv2DTranspose(256, (4, 4), strides = (2, 2), padding =␣
 ↪'same', kernel_initializer = WEIGHT_INIT))
model.add(layers.BatchNormalization())
model.add(layers.ReLU())

# Upsample to 32x32
model.add(layers.Conv2DTranspose(256, (4, 4), strides = (2, 2), padding =␣
 ↪'same', kernel_initializer = WEIGHT_INIT))
model.add(layers.BatchNormalization())
model.add(layers.ReLU())
```

```
# Upsample to 64x64
model.add(layers.Conv2DTranspose(256, (4, 4), strides = (2, 2), padding =␣
 ↪'same', kernel_initializer = WEIGHT_INIT))
model.add(layers.BatchNormalization())
model.add(layers.ReLU())

model.add(layers.Conv2D(CHANNELS, (4,4), padding = 'same', activation = 'tanh'))

generator = model
generator.summary()
```

```
Model: "generator"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense (Dense)               (None, 32768)             3309568

 batch_normalization (BatchN  (None, 32768)            131072
 ormalization)

 re_lu (ReLU)                (None, 32768)             0

 reshape (Reshape)           (None, 8, 8, 512)         0

 conv2d_transpose (Conv2DTra  (None, 16, 16, 256)      2097408
 nspose)

 batch_normalization_1 (Batc  (None, 16, 16, 256)      1024
 hNormalization)

 re_lu_1 (ReLU)              (None, 16, 16, 256)       0

 conv2d_transpose_1 (Conv2DT  (None, 32, 32, 256)      1048832
 ranspose)

 batch_normalization_2 (Batc  (None, 32, 32, 256)      1024
 hNormalization)

 re_lu_2 (ReLU)              (None, 32, 32, 256)       0

 conv2d_transpose_2 (Conv2DT  (None, 64, 64, 256)      1048832
 ranspose)

 batch_normalization_3 (Batc  (None, 64, 64, 256)      1024
 hNormalization)

 re_lu_3 (ReLU)              (None, 64, 64, 256)       0
```

```
 conv2d (Conv2D)              (None, 64, 64, 3)         12291

=================================================================
Total params: 7,651,075
Trainable params: 7,584,003
Non-trainable params: 67,072
_________________________________________________________________
```

6 Disrciminator Model

[15]:
```
model = Sequential(name = 'discriminator')
input_shape = (64, 64, 3)
alpha = 0.2

# Create Convulsion Layers
model.add(layers.Conv2D(64, (4, 4), strides = (2,2), padding = 'same',␣
 ↪input_shape = input_shape))
model.add(layers.BatchNormalization())
model.add(layers.LeakyReLU(alpha = alpha))

model.add(layers.Conv2D(128, (4, 4), strides = (2,2), padding = 'same',␣
 ↪input_shape = input_shape))
model.add(layers.BatchNormalization())
model.add(layers.LeakyReLU(alpha = alpha))

model.add(layers.Conv2D(128, (4, 4), strides = (2,2), padding = 'same',␣
 ↪input_shape = input_shape))
model.add(layers.BatchNormalization())
model.add(layers.LeakyReLU(alpha = alpha))

model.add(layers.Flatten())
model.add(layers.Dropout(0.3))

# Output Class
model.add(layers.Dense(1, activation = 'sigmoid'))

discriminator = model
discriminator.summary()
```

```
Model: "discriminator"
_________________________________________________________________
 Layer (type)                 Output Shape              Param #
=================================================================
 conv2d_1 (Conv2D)            (None, 32, 32, 64)        3136

 batch_normalization_4 (Batc  (None, 32, 32, 64)       256
```

```
hNormalization)

leaky_re_lu (LeakyReLU)      (None, 32, 32, 64)        0

conv2d_2 (Conv2D)            (None, 16, 16, 128)       131200

batch_normalization_5 (Batc  (None, 16, 16, 128)       512
hNormalization)

leaky_re_lu_1 (LeakyReLU)    (None, 16, 16, 128)       0

conv2d_3 (Conv2D)            (None, 8, 8, 128)         262272

batch_normalization_6 (Batc  (None, 8, 8, 128)         512
hNormalization)

leaky_re_lu_2 (LeakyReLU)    (None, 8, 8, 128)         0

flatten (Flatten)            (None, 8192)              0

dropout (Dropout)            (None, 8192)              0

dense_1 (Dense)              (None, 1)                 8193

=================================================================
Total params: 406,081
Trainable params: 405,441
Non-trainable params: 640
_________________________________________________________________
```

7 Create DCGAN

[16]:
```python
class DCGAN(keras.Model):
    def __init__(self, generator, discriminator, latent_dim):
        super().__init__()
        self.generator = generator
        self.discriminator = discriminator
        self.latent_dim = latent_dim
        self.g_loss_metric = keras.metrics.Mean(name = 'g_loss')
        self.d_loss_metric = keras.metrics.Mean(name = 'd_loss')

    @property
    def metrics(self):
        return [self.g_loss_metric, self.d_loss_metric]

    def compile(self, g_optimizer, d_optimizer, loss_fn):
        super(DCGAN, self).compile()
```

```
        self.g_optimizer = g_optimizer
        self.d_optimizer = d_optimizer
        self.loss_fn = loss_fn

    def train_step(self, real_images):
        # get batch size from the data
        batch_size = tf.shape(real_images)[0]
        # generate random noise
        random_noise = tf.random.normal(shape=(batch_size, self.latent_dim))

        # train the discriminator with real (1) and fake (0) images
        with tf.GradientTape() as tape:
            # compute loss on real images
            pred_real = self.discriminator(real_images, training=True)
            # generate real image labels
            real_labels = tf.ones((batch_size, 1))
            # label smoothing
            real_labels += 0.05 * tf.random.uniform(tf.shape(real_labels))
            d_loss_real = self.loss_fn(real_labels, pred_real)

            # compute loss on fake images
            fake_images = self.generator(random_noise)
            pred_fake = self.discriminator(fake_images, training=True)
            # generate fake labels
            fake_labels = tf.zeros((batch_size, 1))
            d_loss_fake = self.loss_fn(fake_labels, pred_fake)

            # total discriminator loss
            d_loss = (d_loss_real + d_loss_fake) / 2

        # compute discriminator gradients
        gradients = tape.gradient(d_loss, self.discriminator.
↪trainable_variables)
        # update the gradients
        self.d_optimizer.apply_gradients(zip(gradients, self.discriminator.
↪trainable_variables))


        # train the generator model
        labels = tf.ones((batch_size, 1))
        # generator want discriminator to think that fake images are real
        with tf.GradientTape() as tape:
            # generate fake images from generator
            fake_images = self.generator(random_noise, training=True)
            # classify images as real or fake
            pred_fake = self.discriminator(fake_images, training=True)
            # compute loss
```

```
            g_loss = self.loss_fn(labels, pred_fake)

        # compute gradients
        gradients = tape.gradient(g_loss, self.generator.trainable_variables)
        # update the gradients
        self.g_optimizer.apply_gradients(zip(gradients, self.generator.
 ↪trainable_variables))

        # update states for both models
        self.d_loss_metric.update_state(d_loss)
        self.g_loss_metric.update_state(g_loss)

        return {'d_loss': self.d_loss_metric.result(), 'g_loss': self.
 ↪g_loss_metric.result()}
```

[17]:
```
class DCGANMonitor(keras.callbacks.Callback):
    def __init__(self, num_imgs=25, latent_dim=100):
        self.num_imgs = num_imgs
        self.latent_dim = latent_dim
        # create random noise for generating images
        self.noise = tf.random.normal([25, latent_dim])

    def on_epoch_end(self, epoch, logs=None):
        # generate the image from noise
        g_img = self.model.generator(self.noise)
        # denormalize the image
        g_img = (g_img * 127.5) + 127.5
        g_img.numpy()

        fig = plt.figure(figsize=(8, 8))
        for i in range(self.num_imgs):
            plt.subplot(5, 5, i+1)
            img = array_to_img(g_img[i])
            plt.imshow(img)
            plt.axis('off')
        # plt.savefig('epoch_{:03d}.png'.format(epoch))
        plt.show()

    def on_train_end(self, logs=None):
        self.model.generator.save('generator.h5')
```

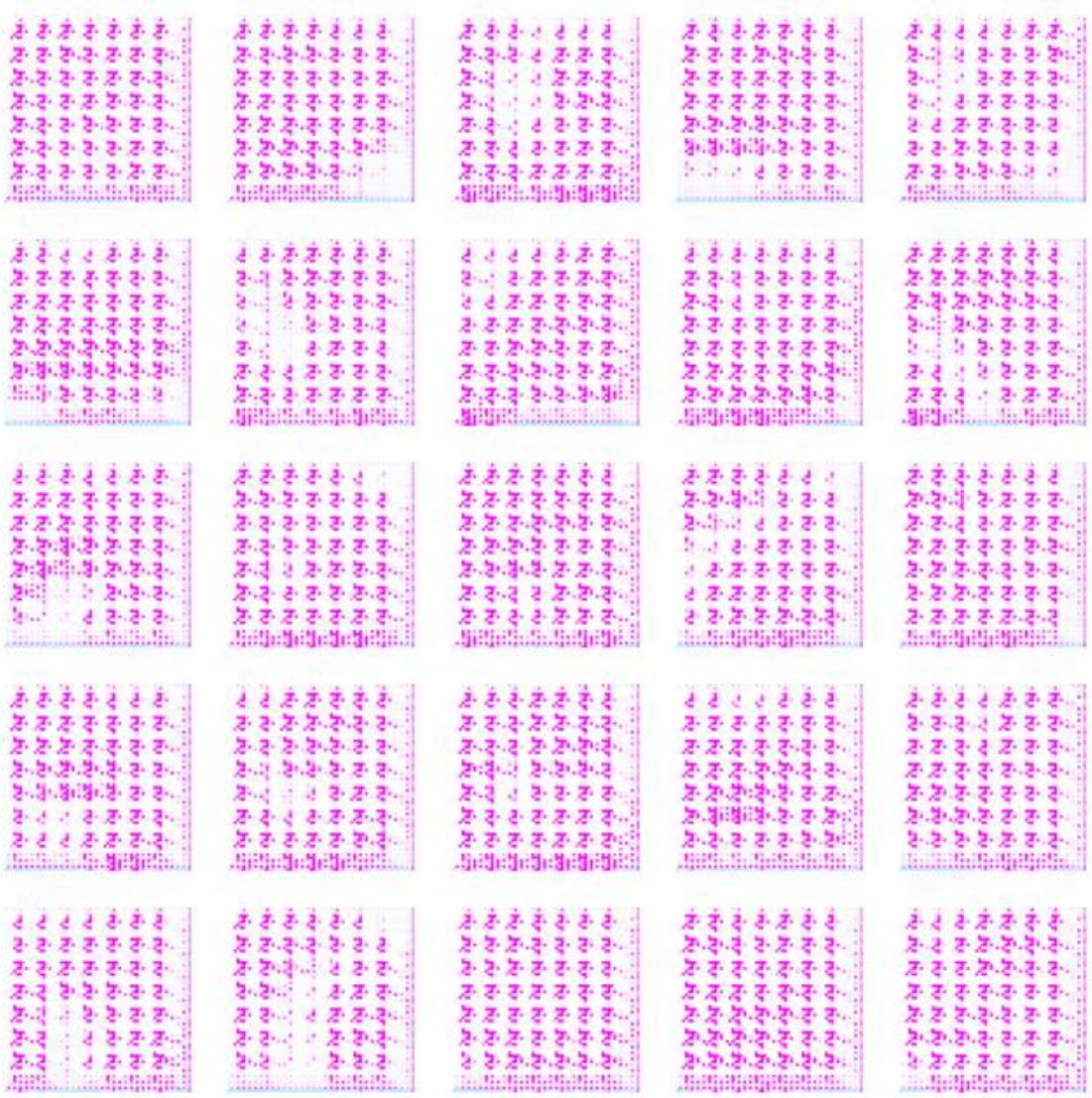
[18]:
```
dcgan = DCGAN(generator=generator, discriminator=discriminator,␣
 ↪latent_dim=LATENT_DIM)
```

[19]:
```
D_LR = 0.0001
G_LR = 0.0003
```

```
dcgan.compile(g_optimizer=Adam(learning_rate=G_LR, beta_1=0.5),␣
  ↪d_optimizer=Adam(learning_rate=D_LR, beta_1=0.5),␣
  ↪loss_fn=BinaryCrossentropy())
```

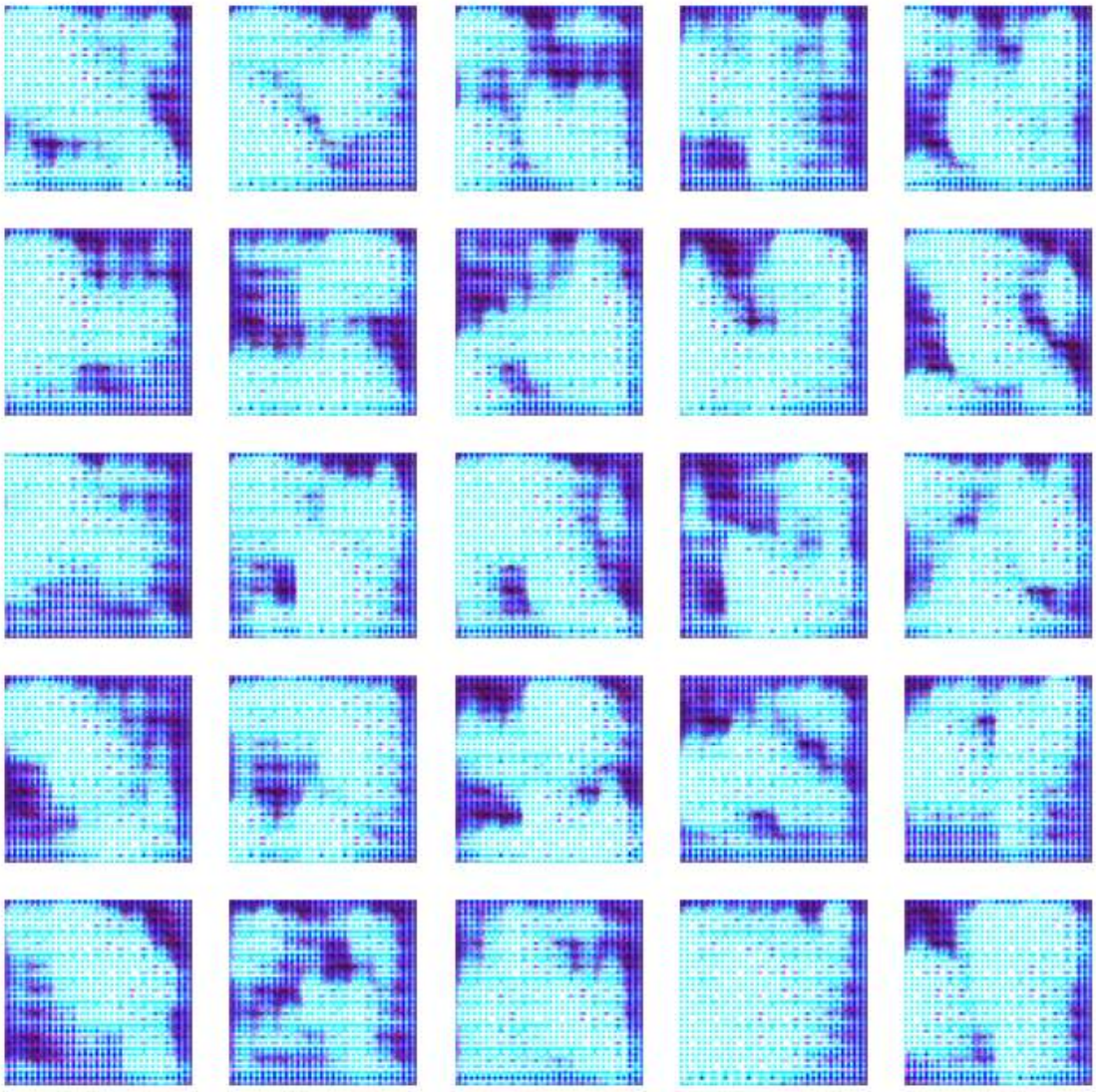
[ ]:
```
N_EPOCHS = 75
dcgan.fit(train_images, epochs=N_EPOCHS, callbacks=[DCGANMonitor()])
```

```
Epoch 1/75
674/674 [==============================] - ETA: 0s - d_loss: -0.4480 - g_loss:
0.7838
```

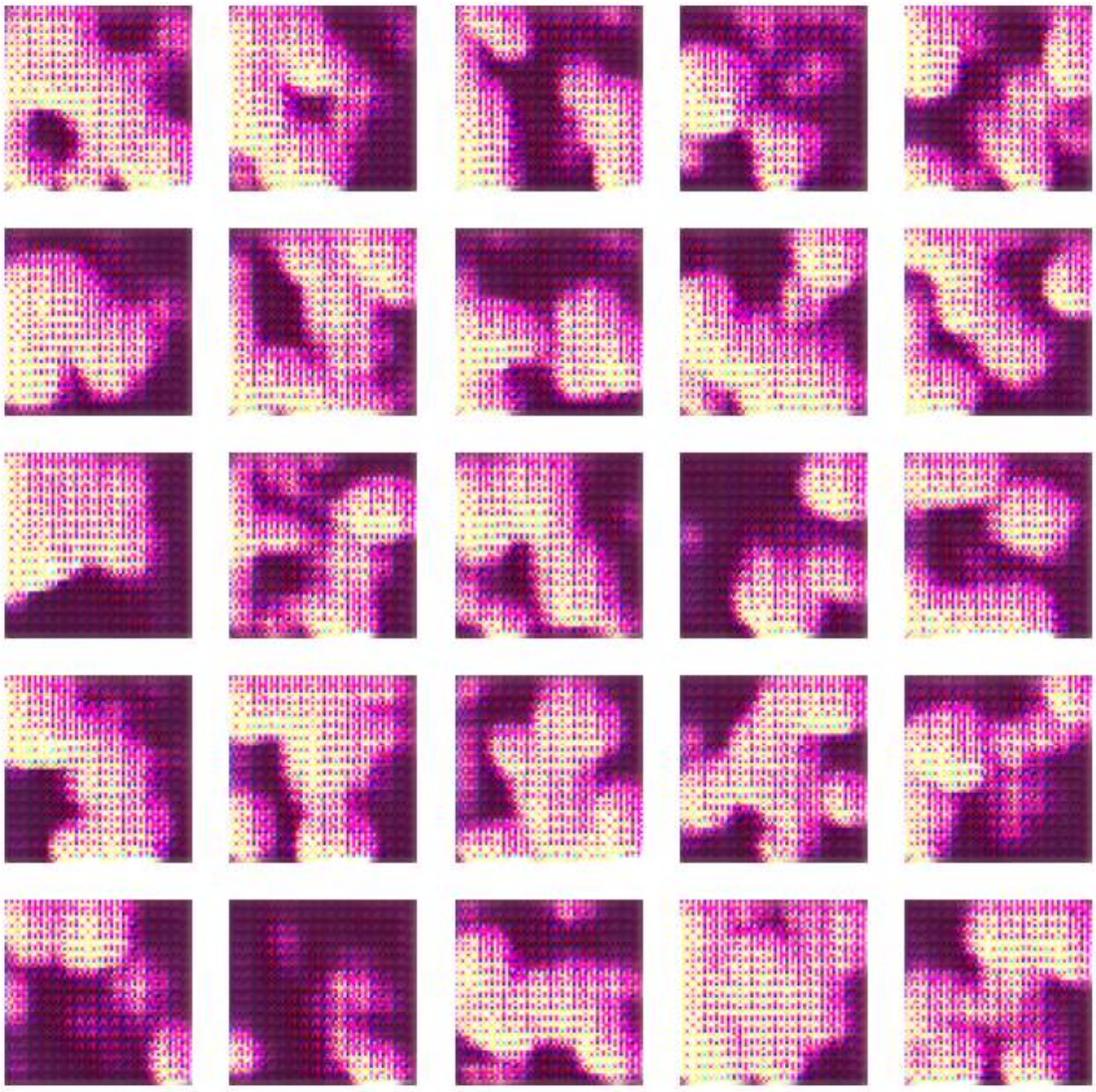


```
674/674 [==============================] - 86s 108ms/step - d_loss: -0.4480 -
g_loss: 0.7838
```

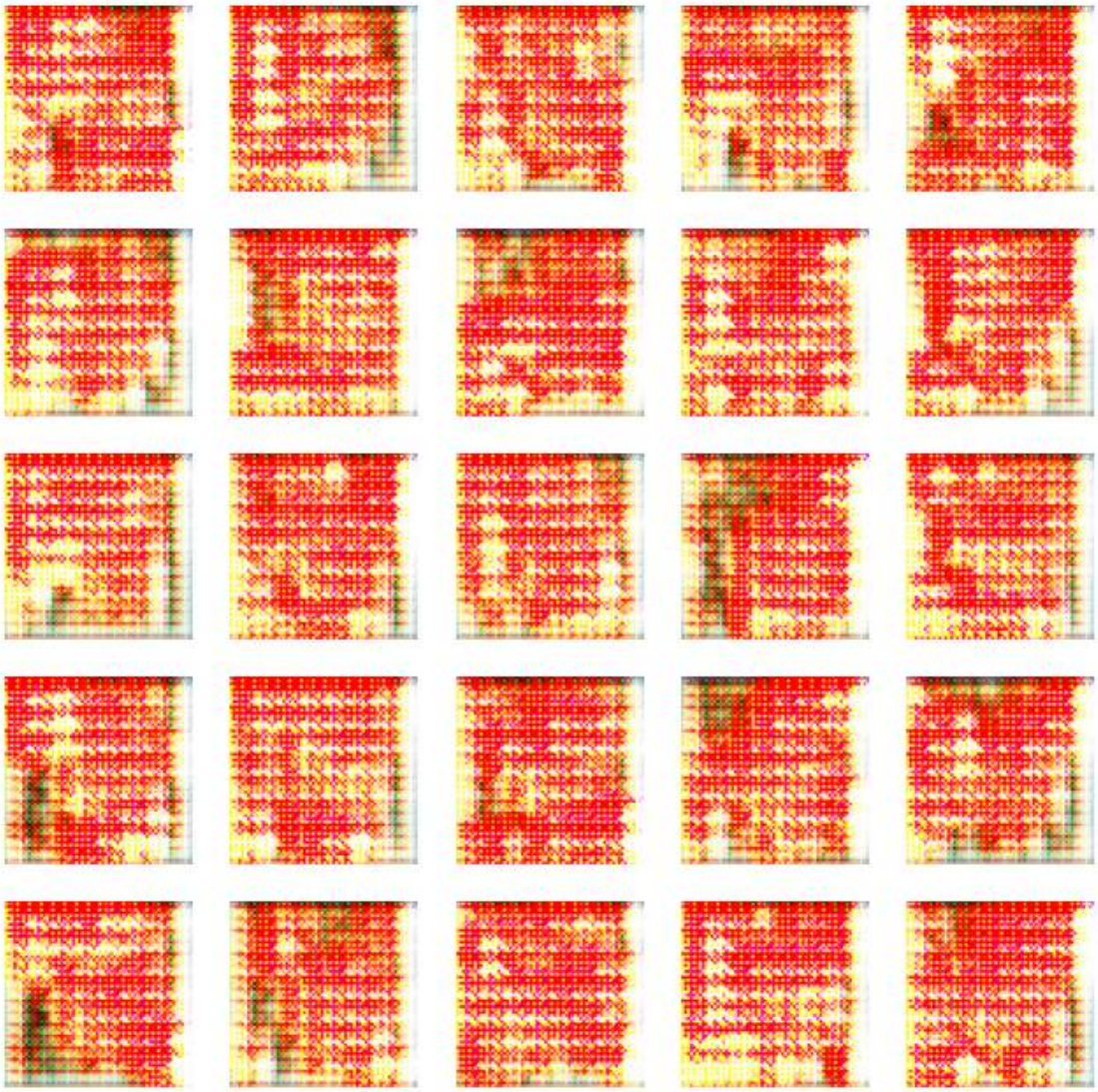
```
Epoch 2/75
674/674 [==============================] - ETA: 0s - d_loss: -1.2926 - g_loss:
0.2817
```



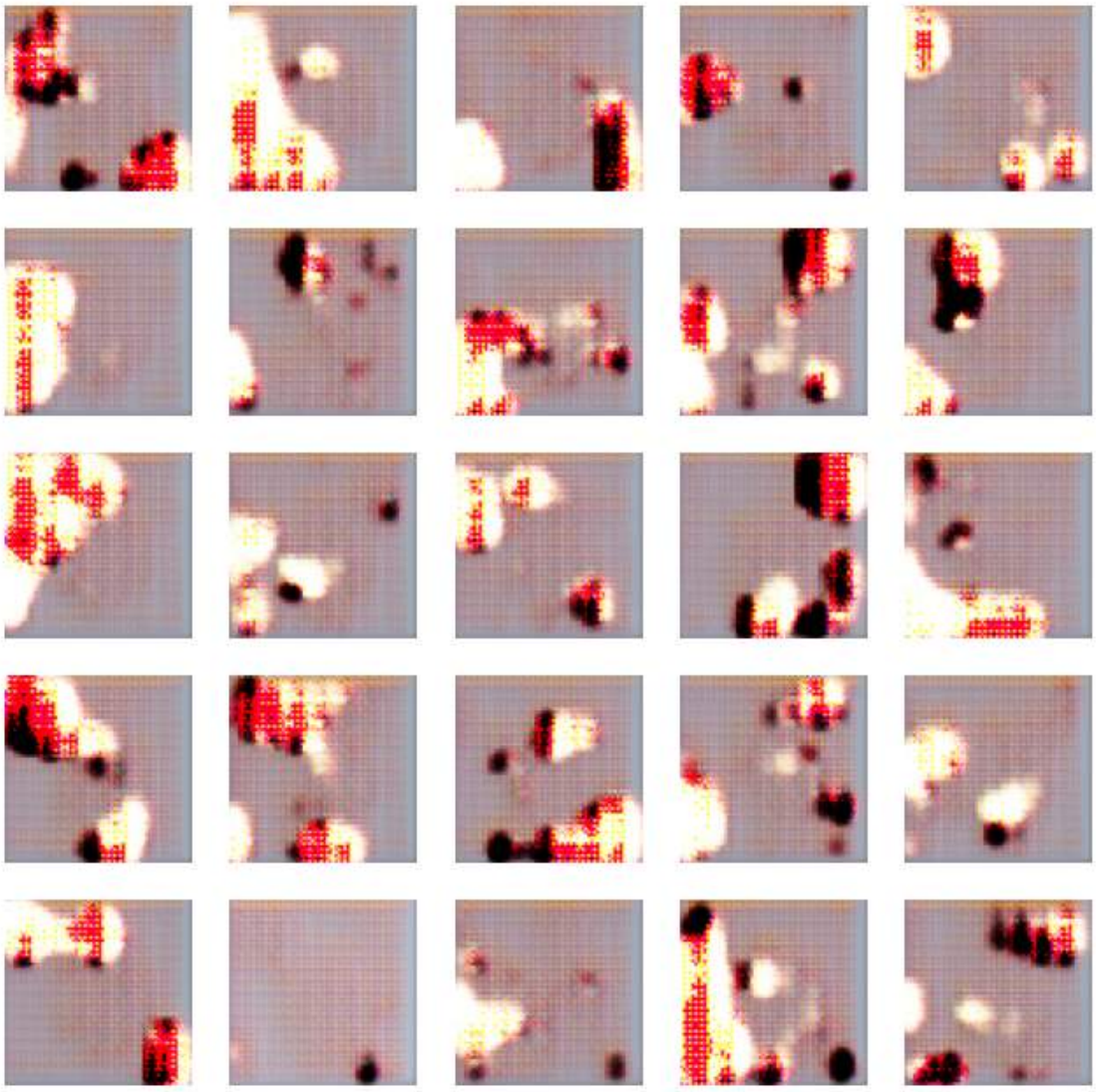
```
674/674 [==============================] - 72s 107ms/step - d_loss: -1.2926 -
g_loss: 0.2817
Epoch 3/75
674/674 [==============================] - ETA: 0s - d_loss: -2.5090 - g_loss:
2.4811
```



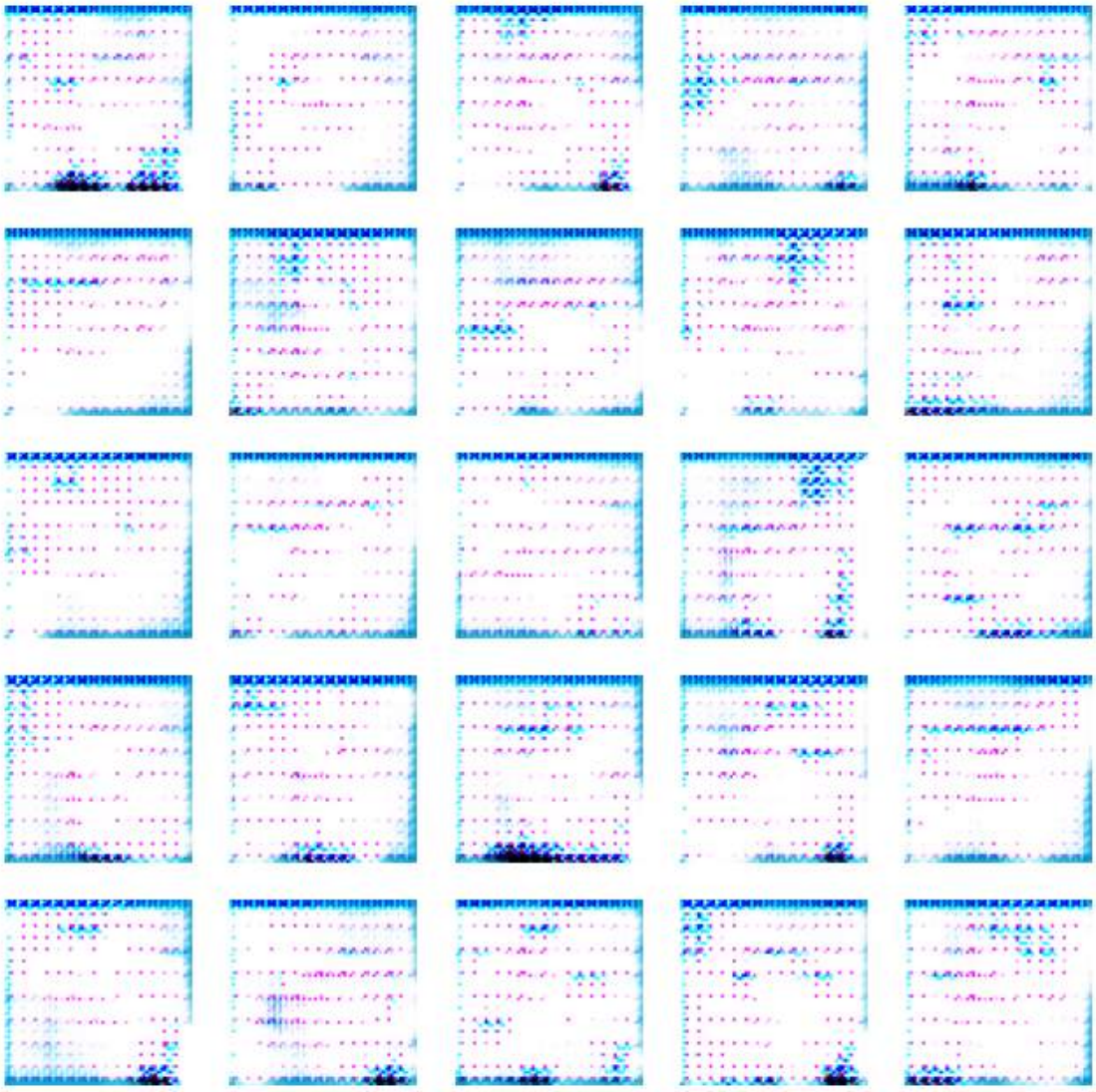
```
674/674 [==============================] - 72s 108ms/step - d_loss: -2.5090 -
g_loss: 2.4811
Epoch 4/75
674/674 [==============================] - ETA: 0s - d_loss: -2.4163 - g_loss:
1.1239
```



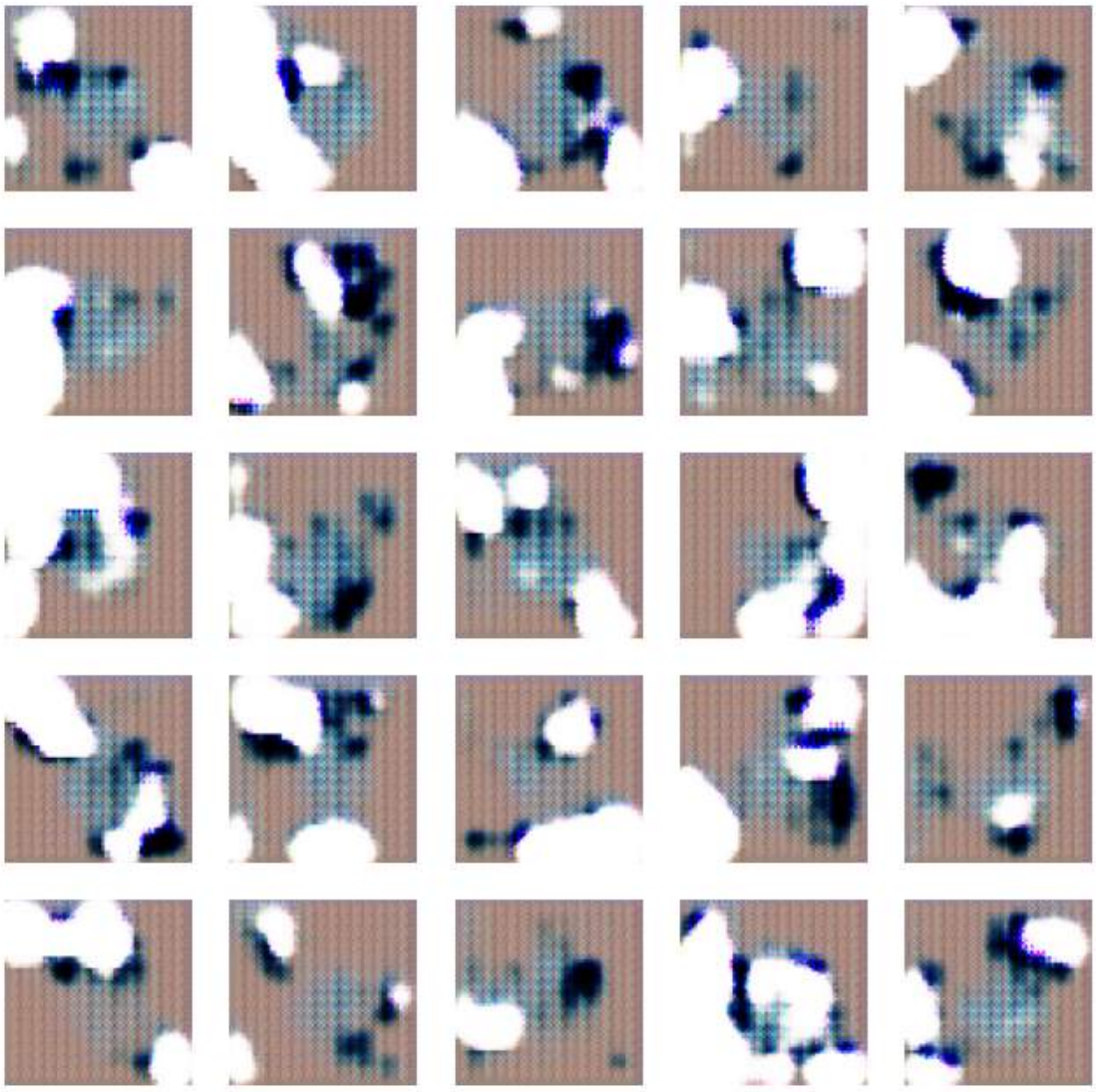
```
674/674 [==============================] - 73s 108ms/step - d_loss: -2.4163 -
g_loss: 1.1239
Epoch 5/75
674/674 [==============================] - ETA: 0s - d_loss: -3.8338 - g_loss:
1.1247
```



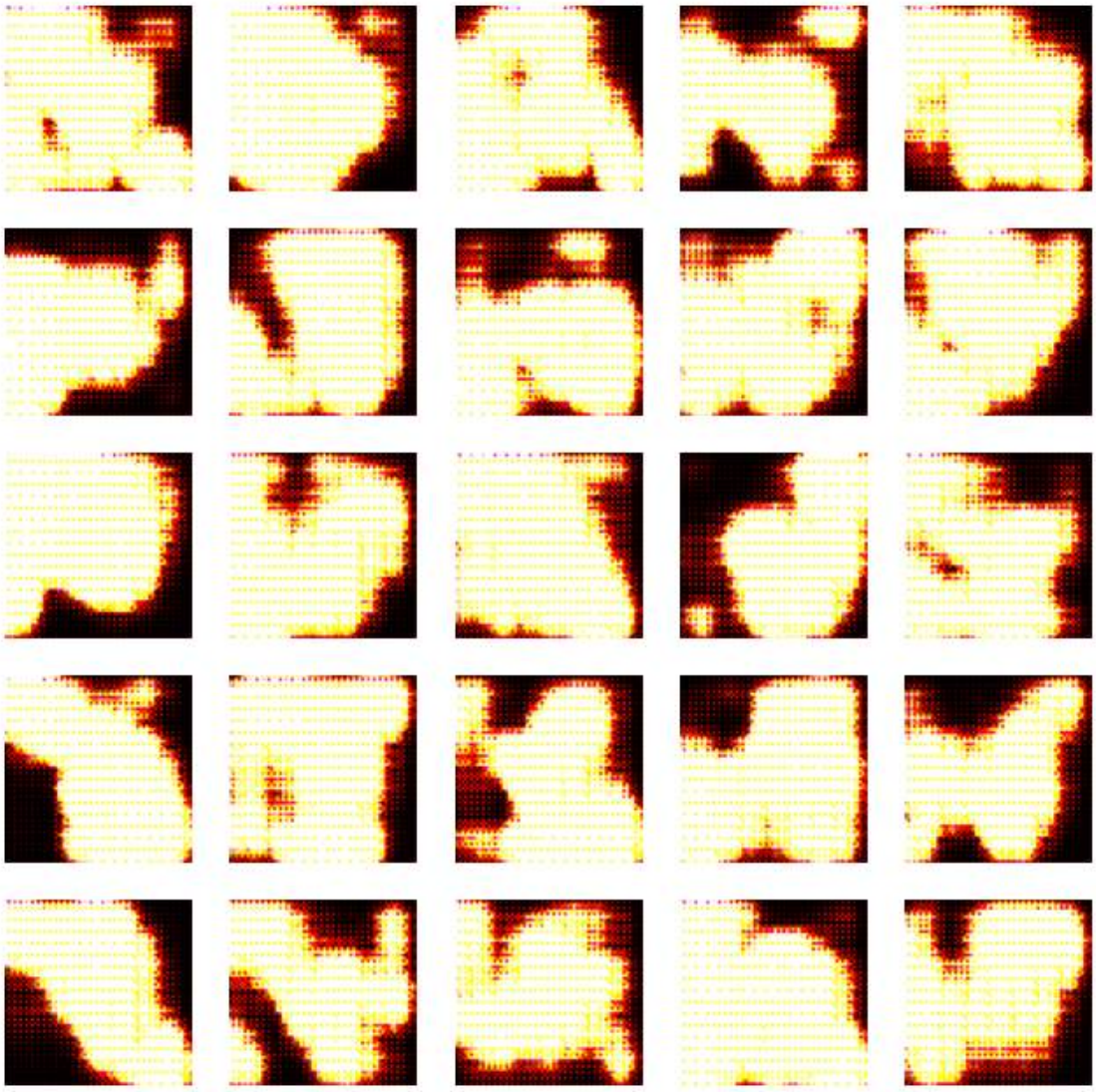
```
674/674 [==============================] - 72s 107ms/step - d_loss: -3.8338 -
g_loss: 1.1247
Epoch 6/75
674/674 [==============================] - ETA: 0s - d_loss: -6.2459 - g_loss:
2.0623
```



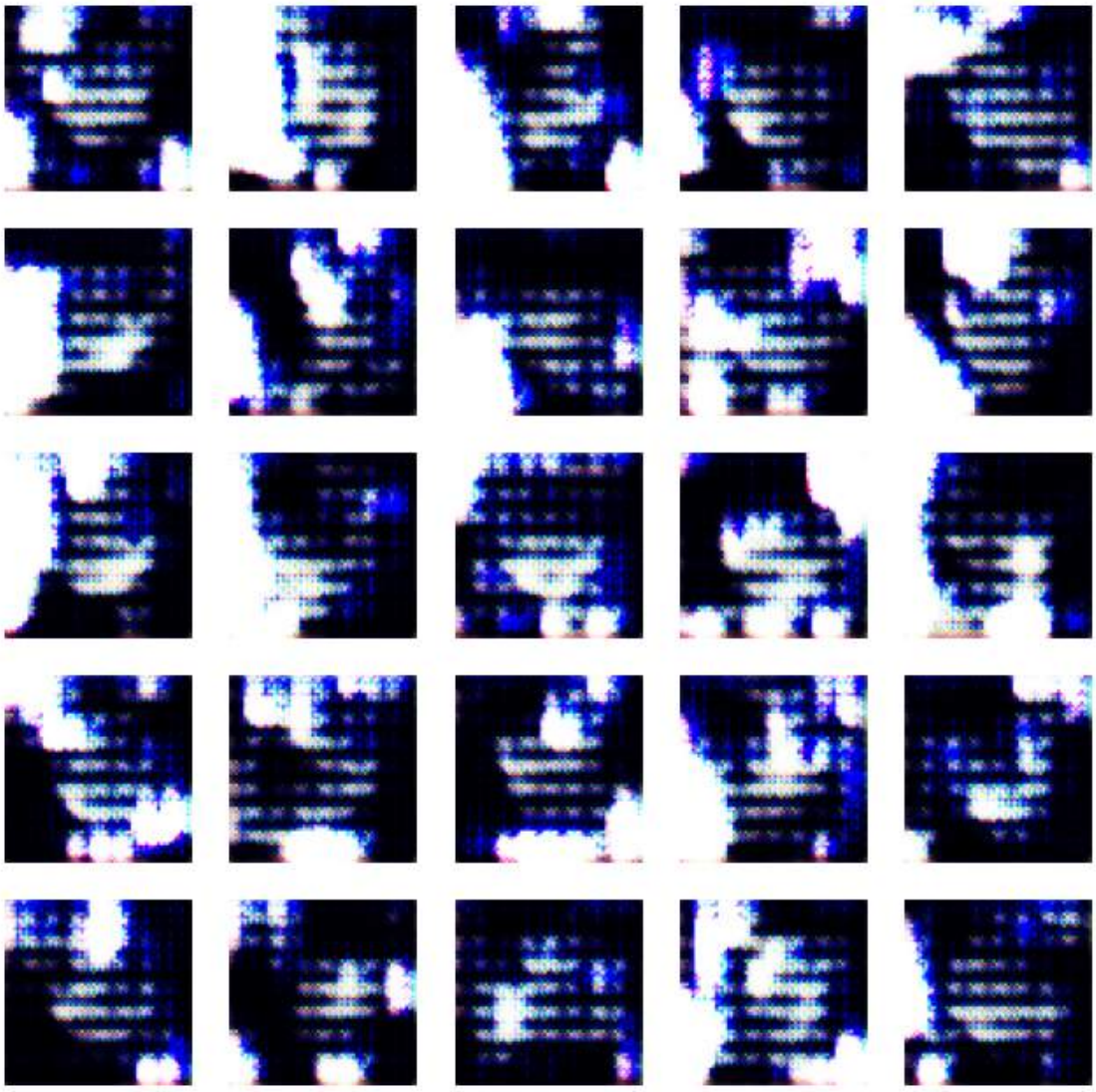
```
674/674 [==============================] - 72s 108ms/step - d_loss: -6.2459 -
g_loss: 2.0623
Epoch 7/75
674/674 [==============================] - ETA: 0s - d_loss: -9.3347 - g_loss:
1.2724
```



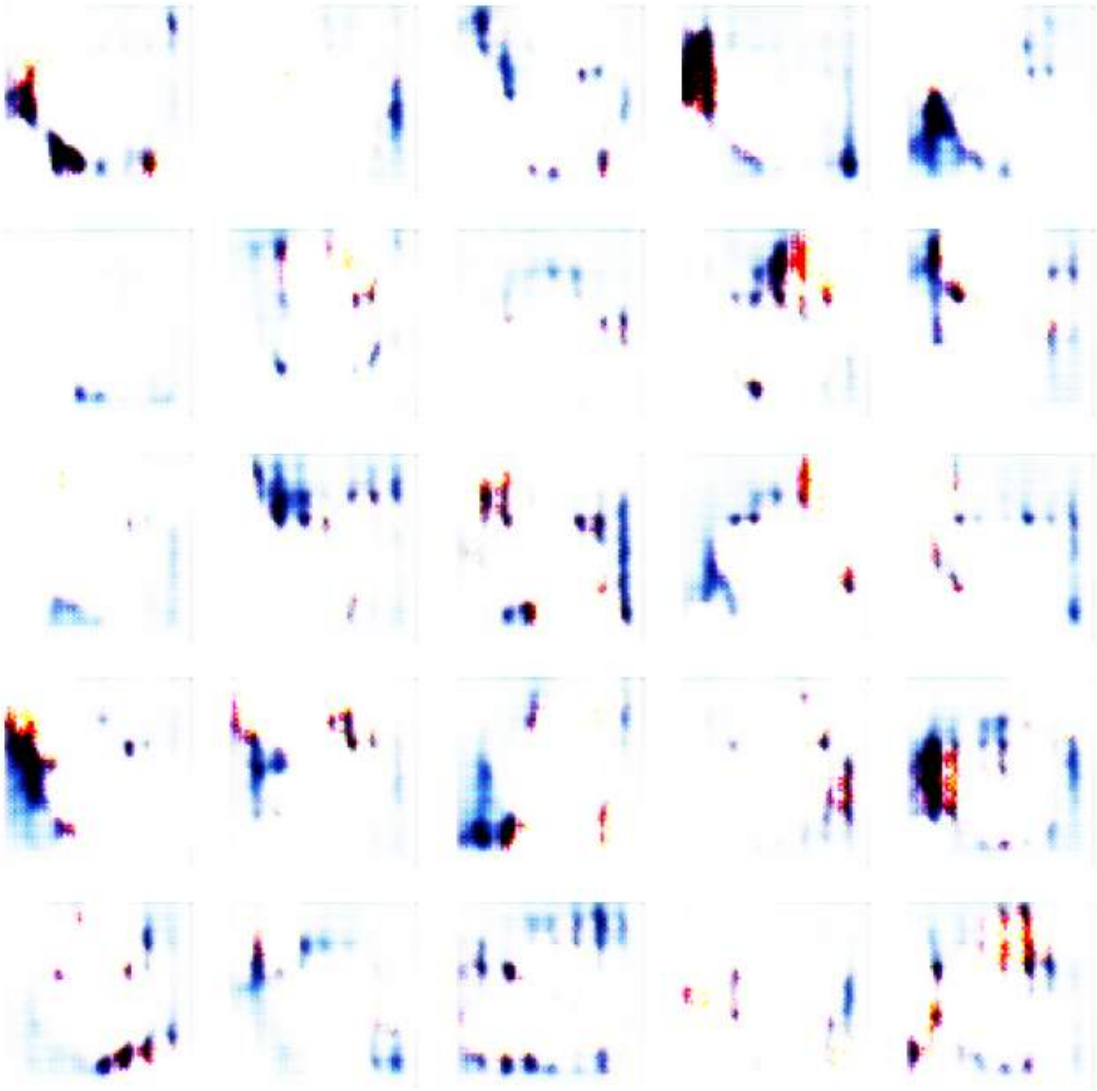
```
674/674 [==============================] - 72s 107ms/step - d_loss: -9.3347 -
g_loss: 1.2724
Epoch 8/75
674/674 [==============================] - ETA: 0s - d_loss: -13.2560 - g_loss:
4.1757
```



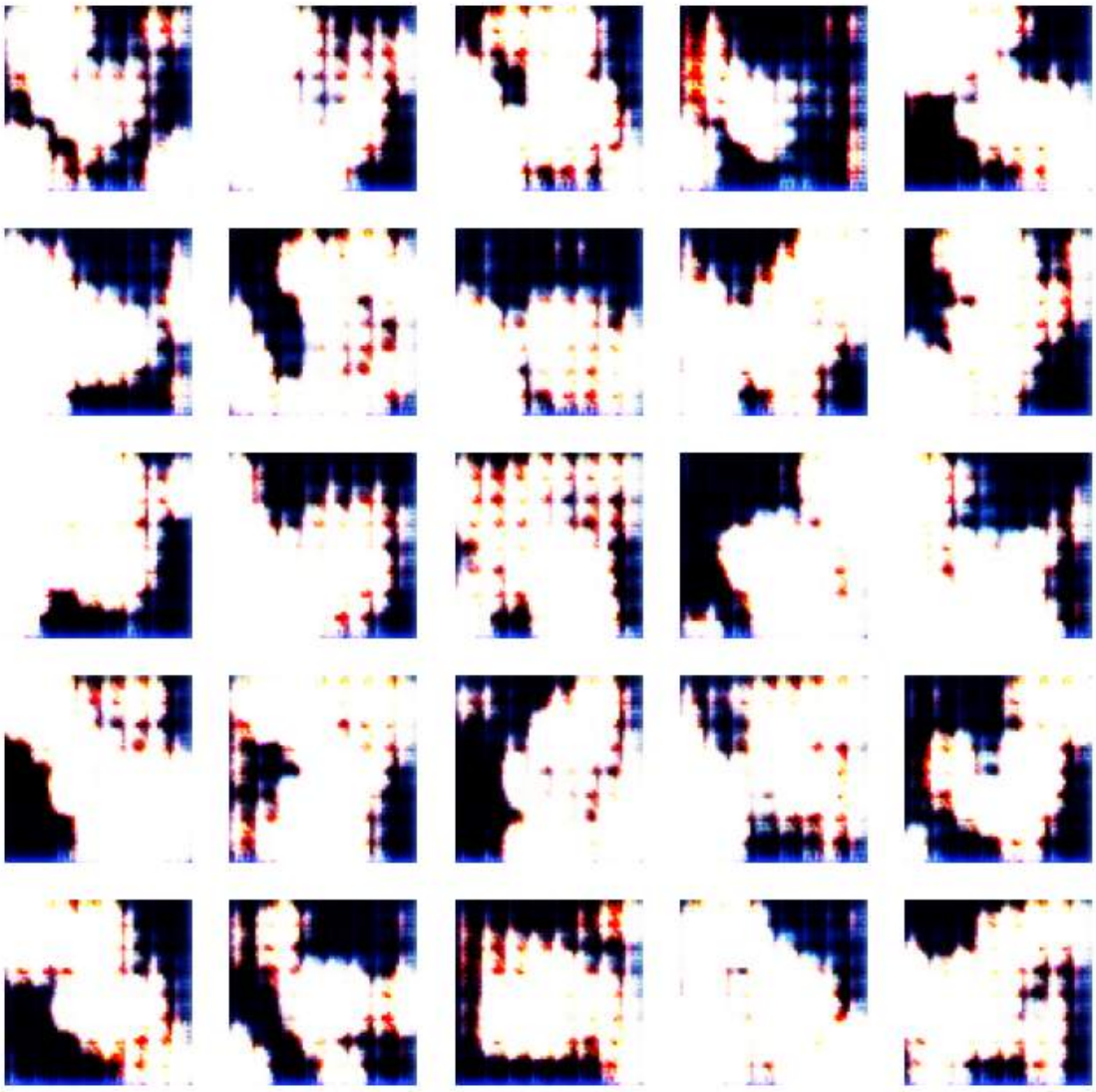
```
674/674 [==============================] - 72s 107ms/step - d_loss: -13.2560 -
g_loss: 4.1757
Epoch 9/75
674/674 [==============================] - ETA: 0s - d_loss: -16.6312 - g_loss:
5.2230
```



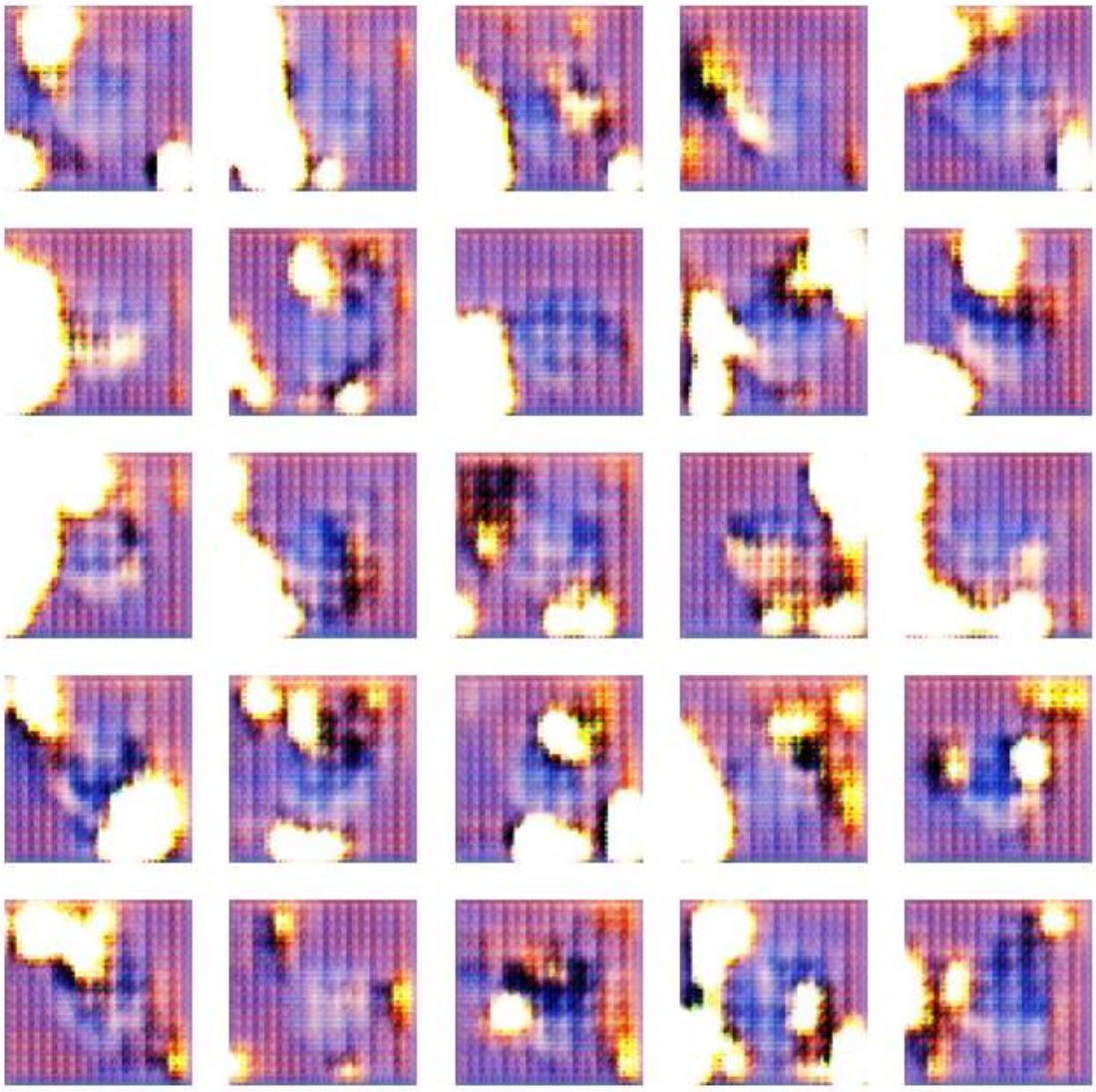
```
674/674 [==============================] - 73s 108ms/step - d_loss: -16.6312 -
g_loss: 5.2230
Epoch 10/75
674/674 [==============================] - ETA: 0s - d_loss: -22.2393 - g_loss:
31.8688
```



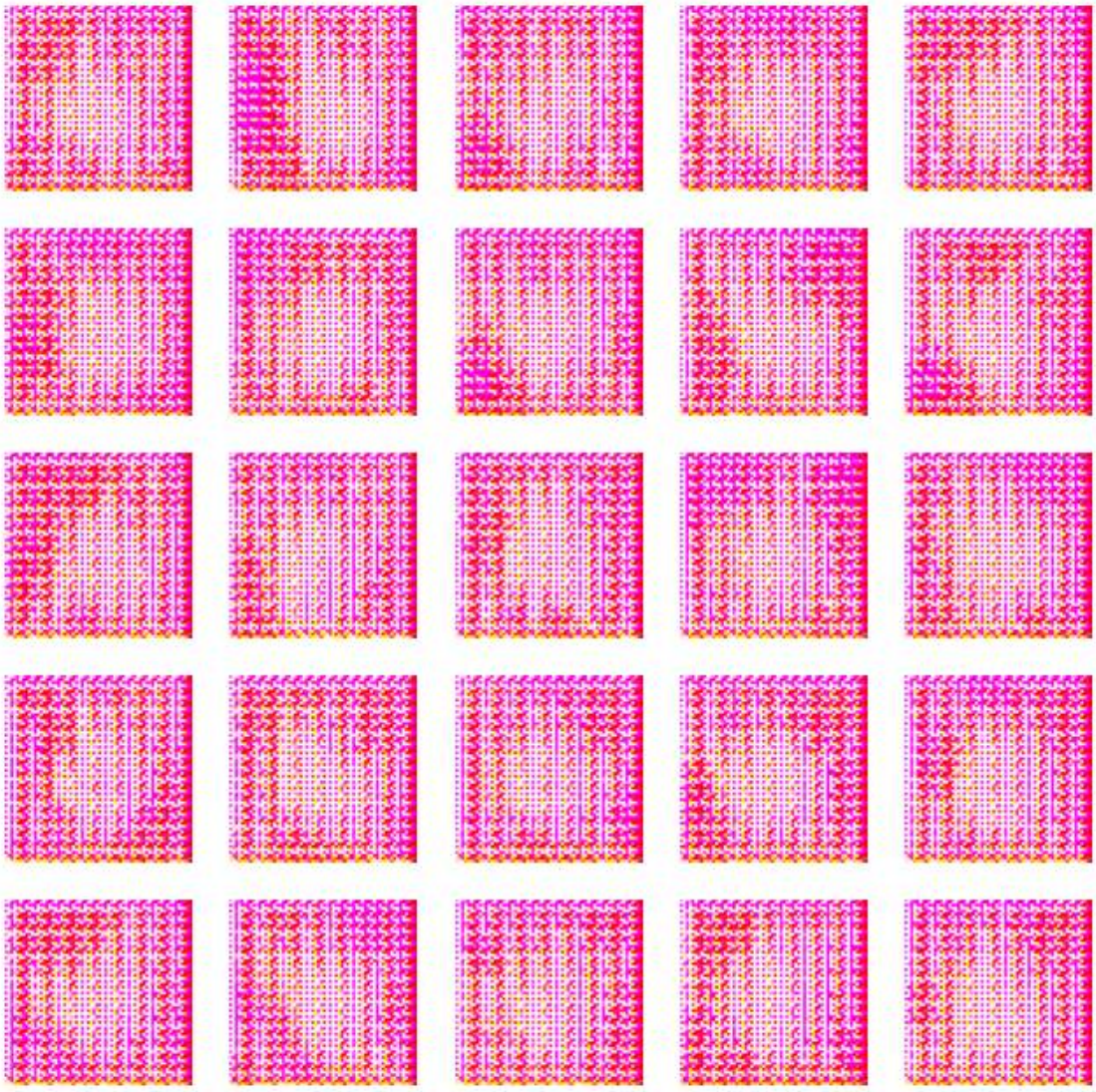
```
674/674 [==============================] - 72s 107ms/step - d_loss: -22.2393 -
g_loss: 31.8688
Epoch 11/75
674/674 [==============================] - ETA: 0s - d_loss: -27.4840 - g_loss:
12.7982
```



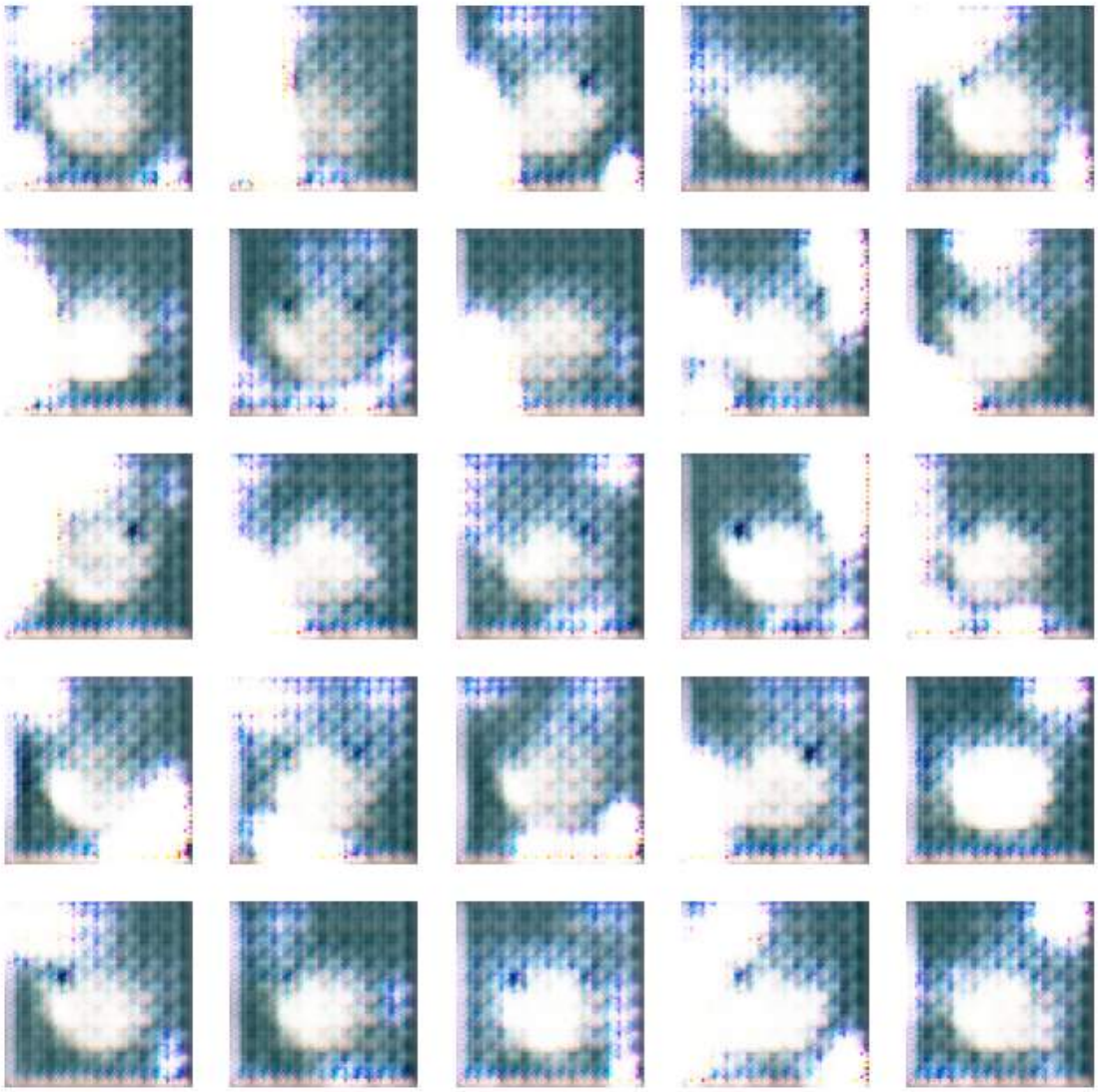
```
674/674 [==============================] - 72s 107ms/step - d_loss: -27.4840 -
g_loss: 12.7982
Epoch 12/75
674/674 [==============================] - ETA: 0s - d_loss: -26.8489 - g_loss:
4.9392
```



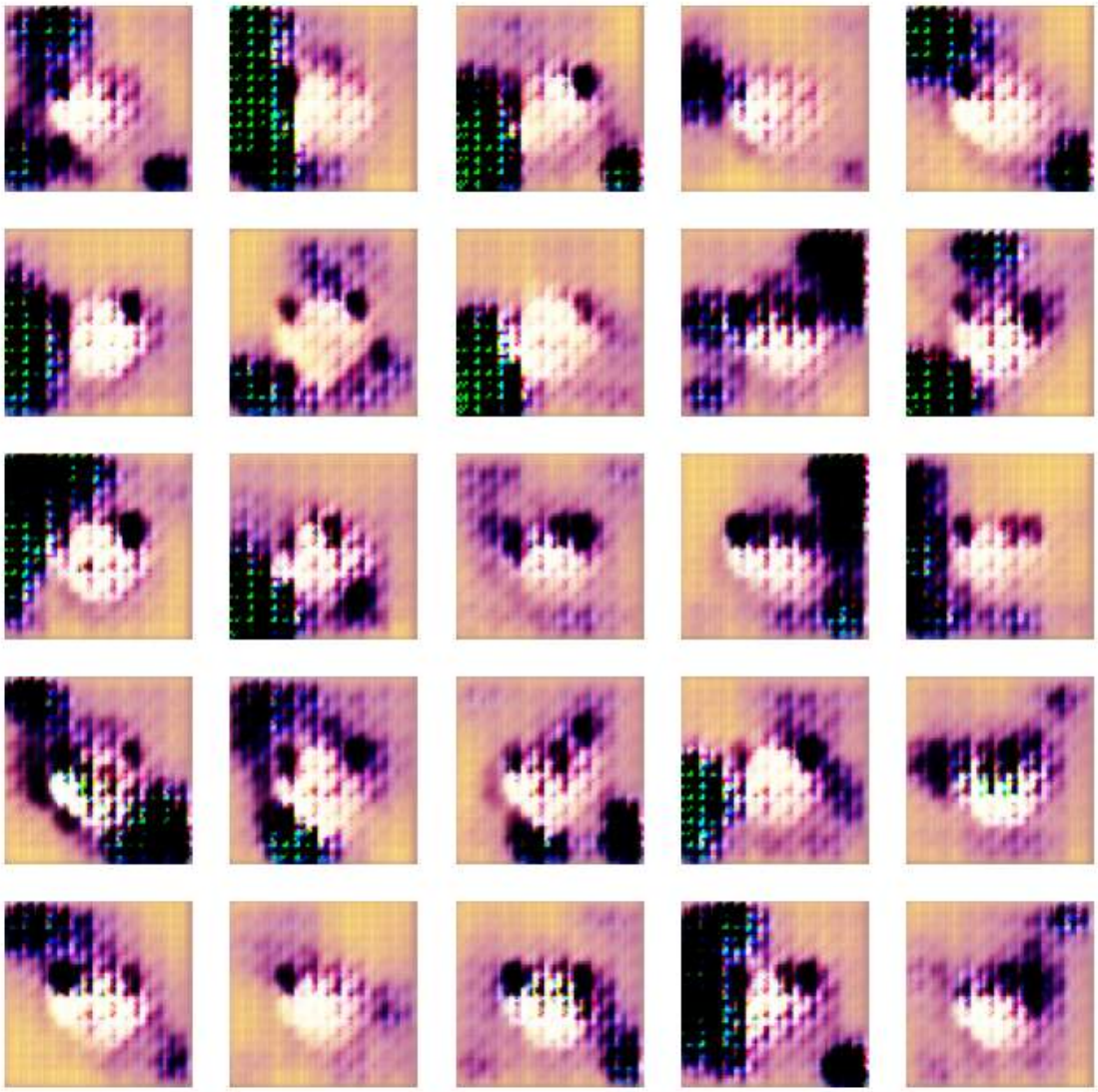
```
674/674 [==============================] - 73s 108ms/step - d_loss: -26.8489 -
g_loss: 4.9392
Epoch 13/75
674/674 [==============================] - ETA: 0s - d_loss: -37.6425 - g_loss:
27.7081
```



```
674/674 [==============================] - 72s 107ms/step - d_loss: -37.6425 -
g_loss: 27.7081
Epoch 14/75
674/674 [==============================] - ETA: 0s - d_loss: -43.4236 - g_loss:
95.3185
```



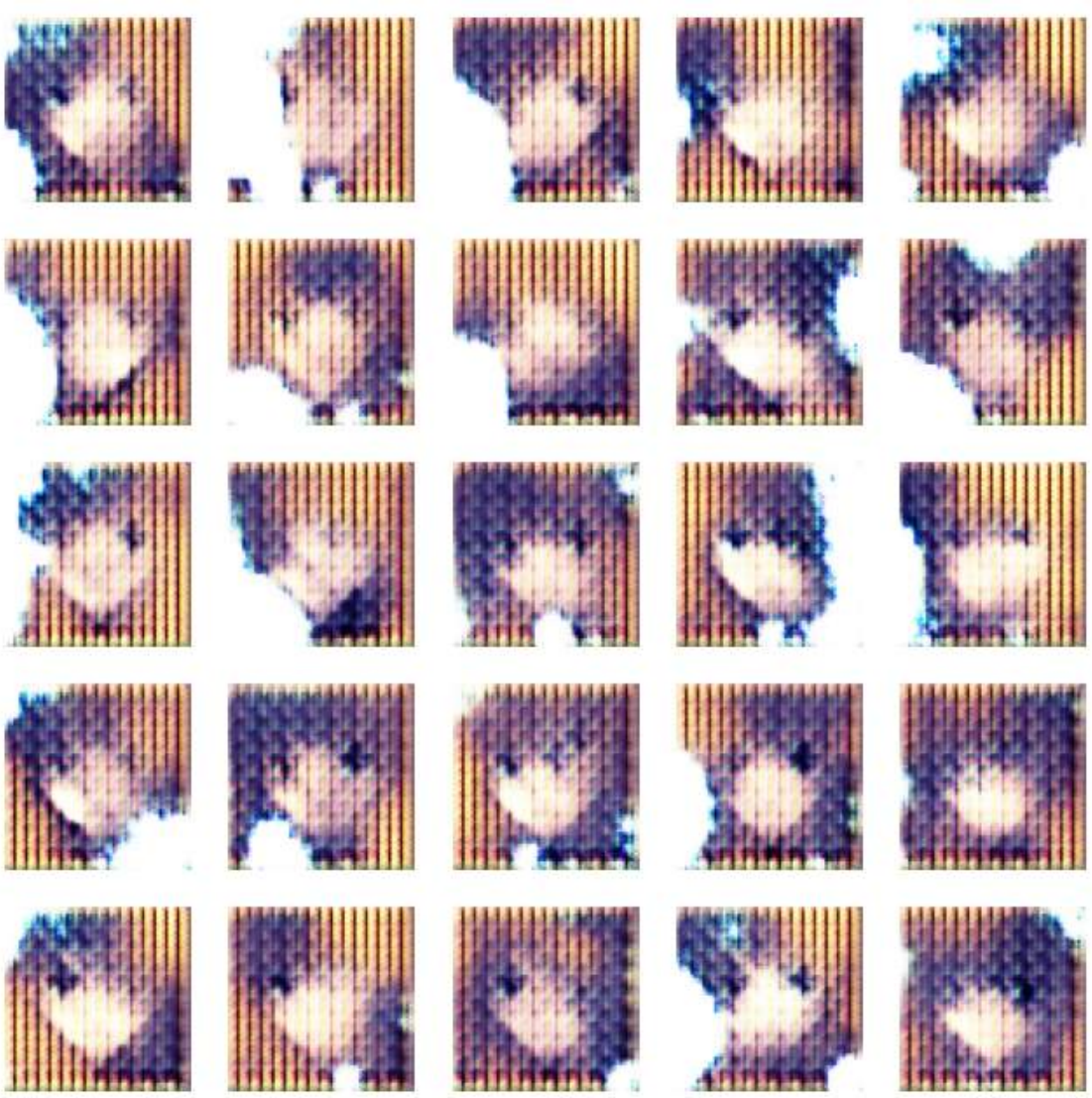
```
674/674 [==============================] - 72s 107ms/step - d_loss: -43.4236 -
g_loss: 95.3185
Epoch 15/75
674/674 [==============================] - ETA: 0s - d_loss: -42.1060 - g_loss:
16.2202
```



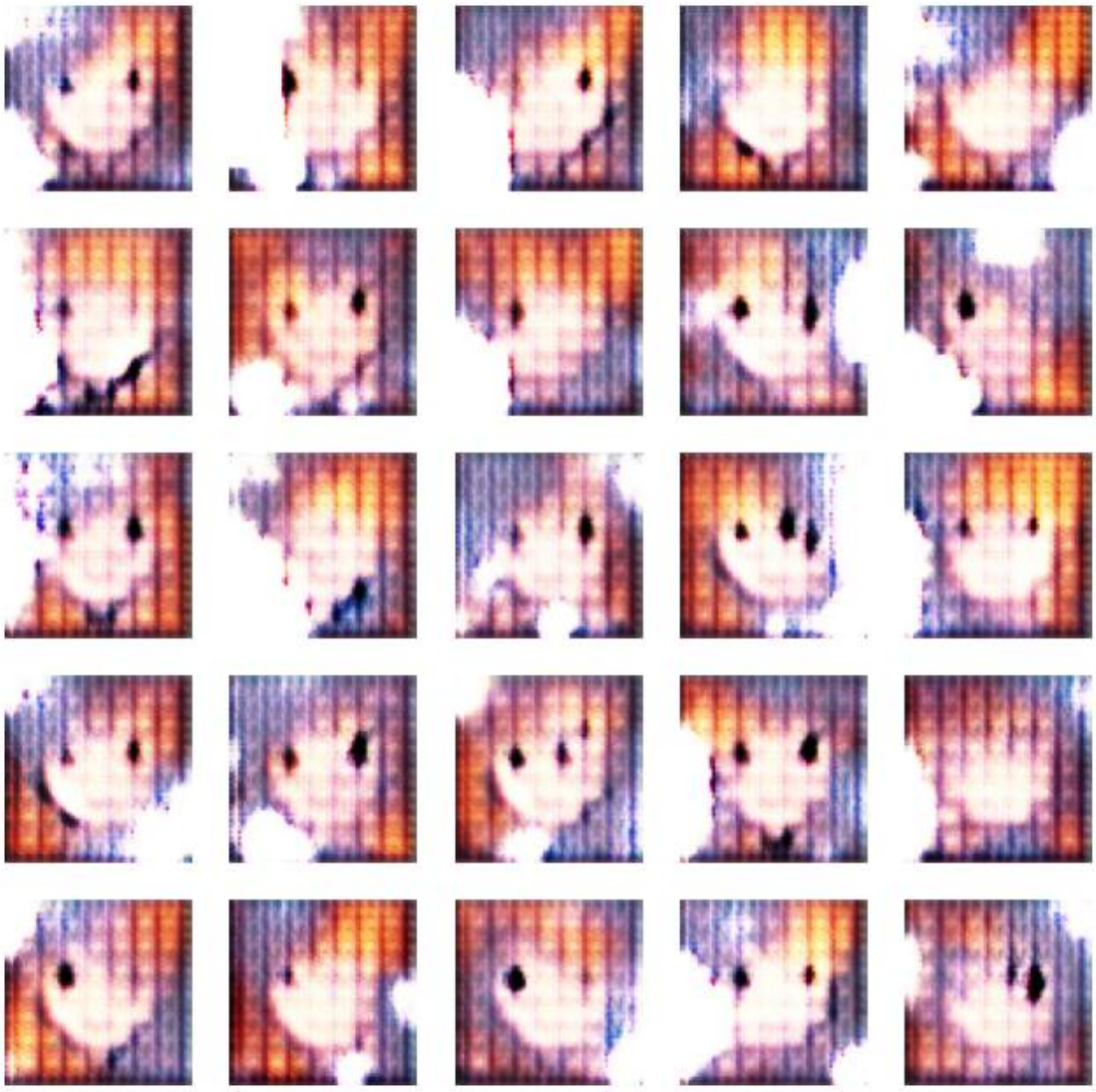
```
674/674 [==============================] - 72s 107ms/step - d_loss: -42.1060 -
g_loss: 16.2202
Epoch 16/75
674/674 [==============================] - ETA: 0s - d_loss: -45.5059 - g_loss:
25.1132
```

```
674/674 [==============================] - 72s 107ms/step - d_loss: -45.5059 -
g_loss: 25.1132
Epoch 17/75
674/674 [==============================] - ETA: 0s - d_loss: -46.3861 - g_loss:
19.0743
```

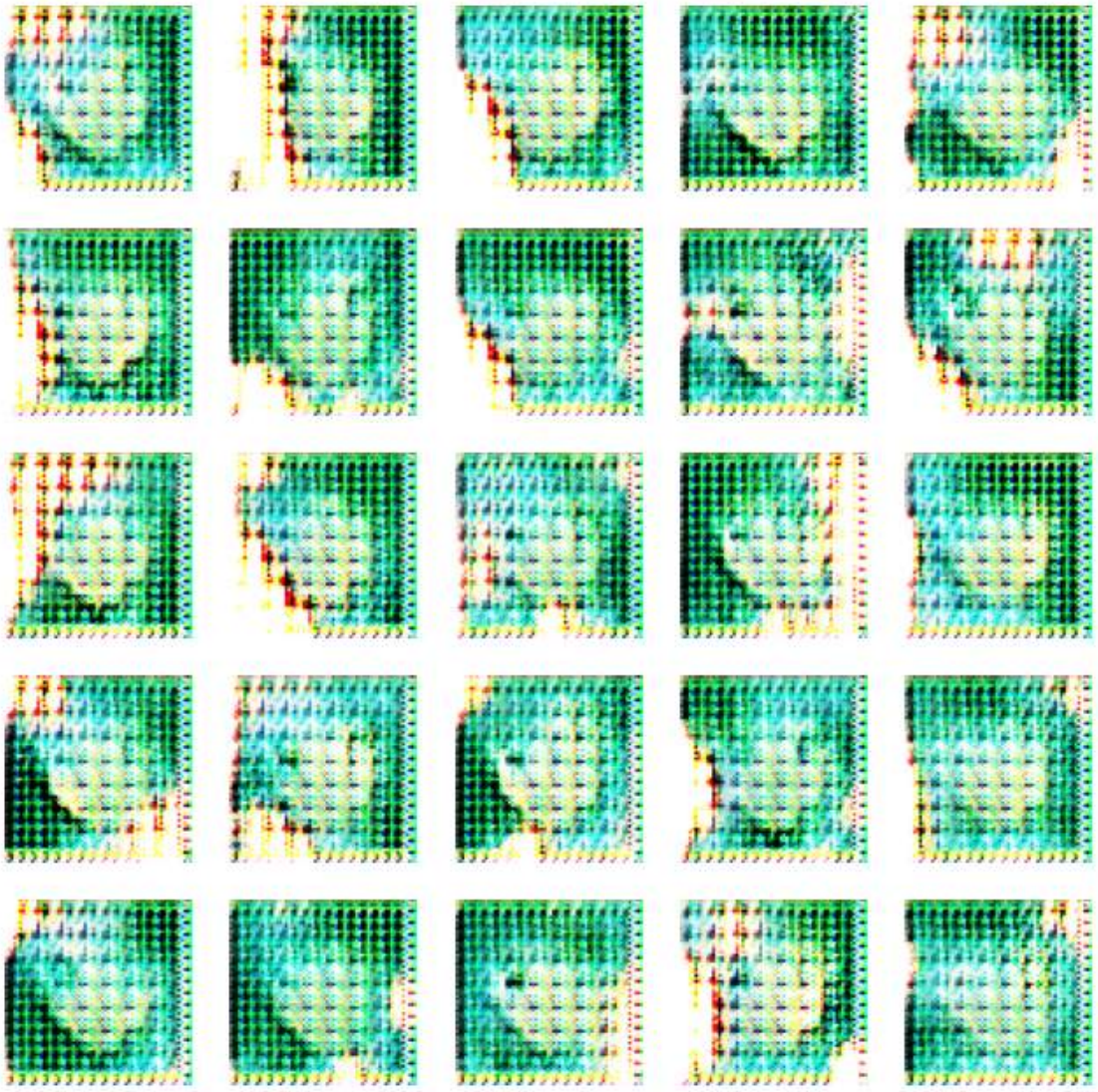
```
674/674 [==============================] - 72s 107ms/step - d_loss: -46.3861 -
g_loss: 19.0743
Epoch 18/75
674/674 [==============================] - ETA: 0s - d_loss: -58.6492 - g_loss:
21.6814
```



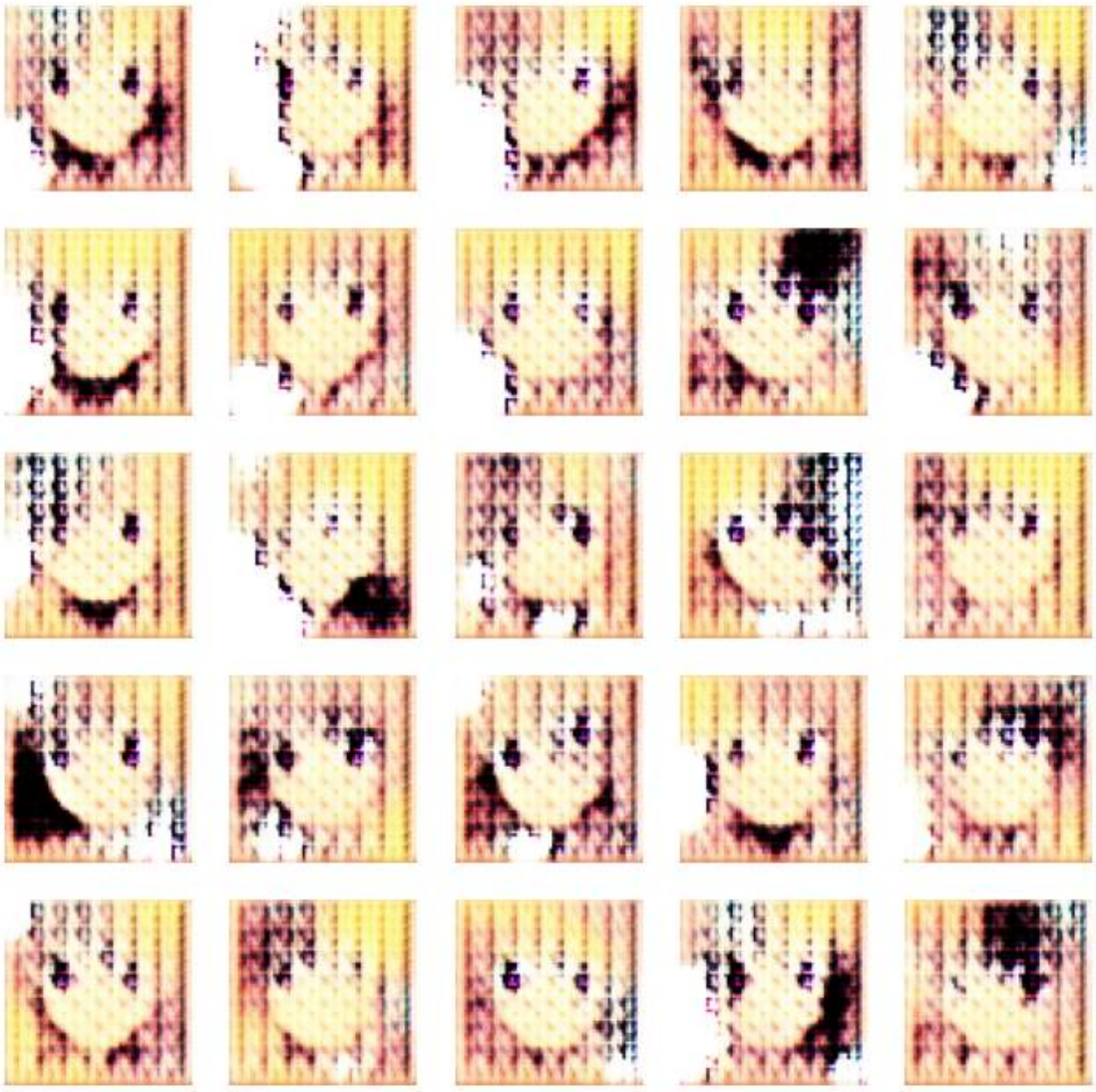
```
674/674 [==============================] - 72s 107ms/step - d_loss: -58.6492 -
g_loss: 21.6814
Epoch 19/75
674/674 [==============================] - ETA: 0s - d_loss: -67.6846 - g_loss:
29.3733
```



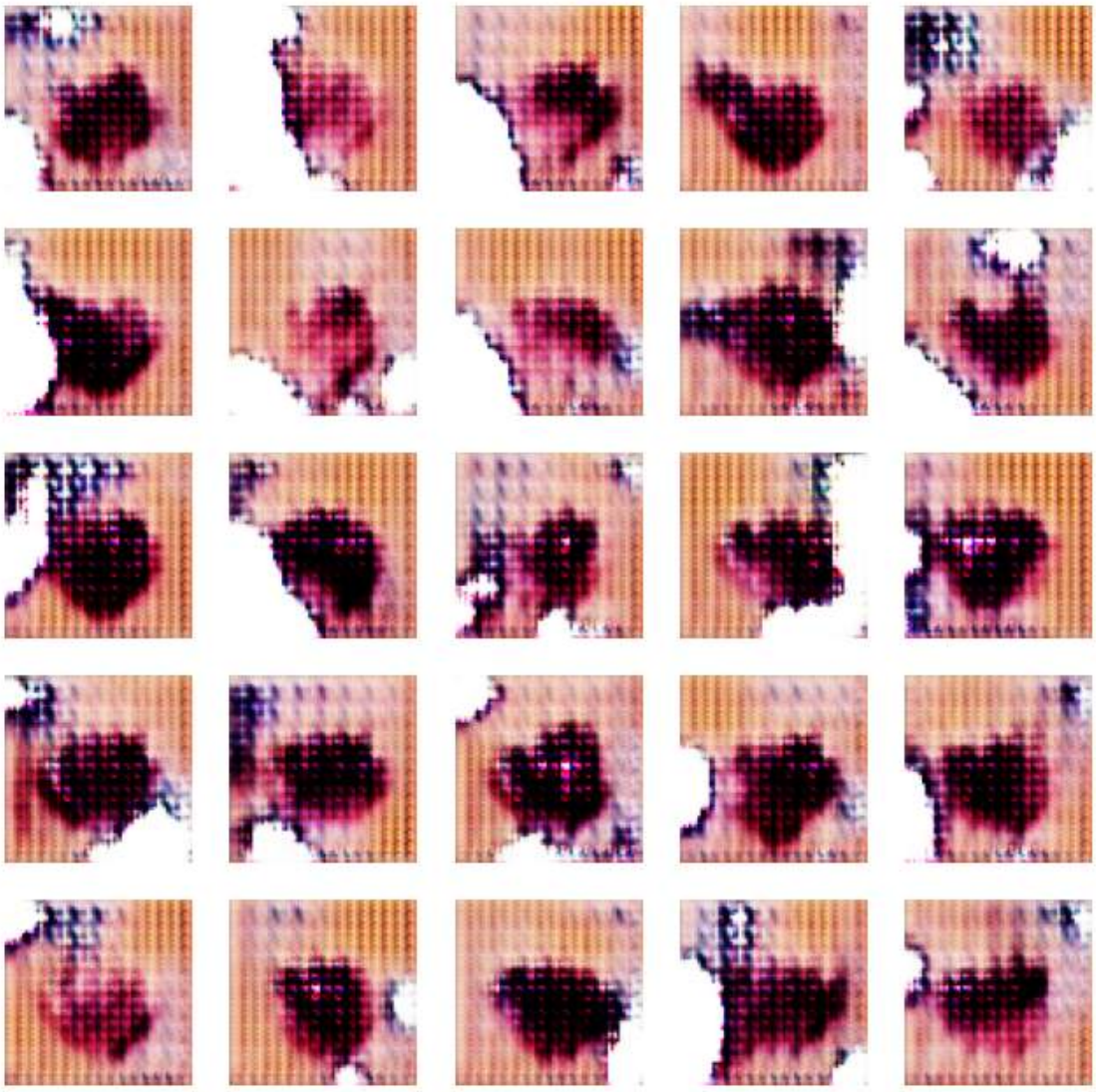
```
674/674 [==============================] - 72s 106ms/step - d_loss: -67.6846 -
g_loss: 29.3733
Epoch 20/75
674/674 [==============================] - ETA: 0s - d_loss: -76.3981 - g_loss:
24.6090
```



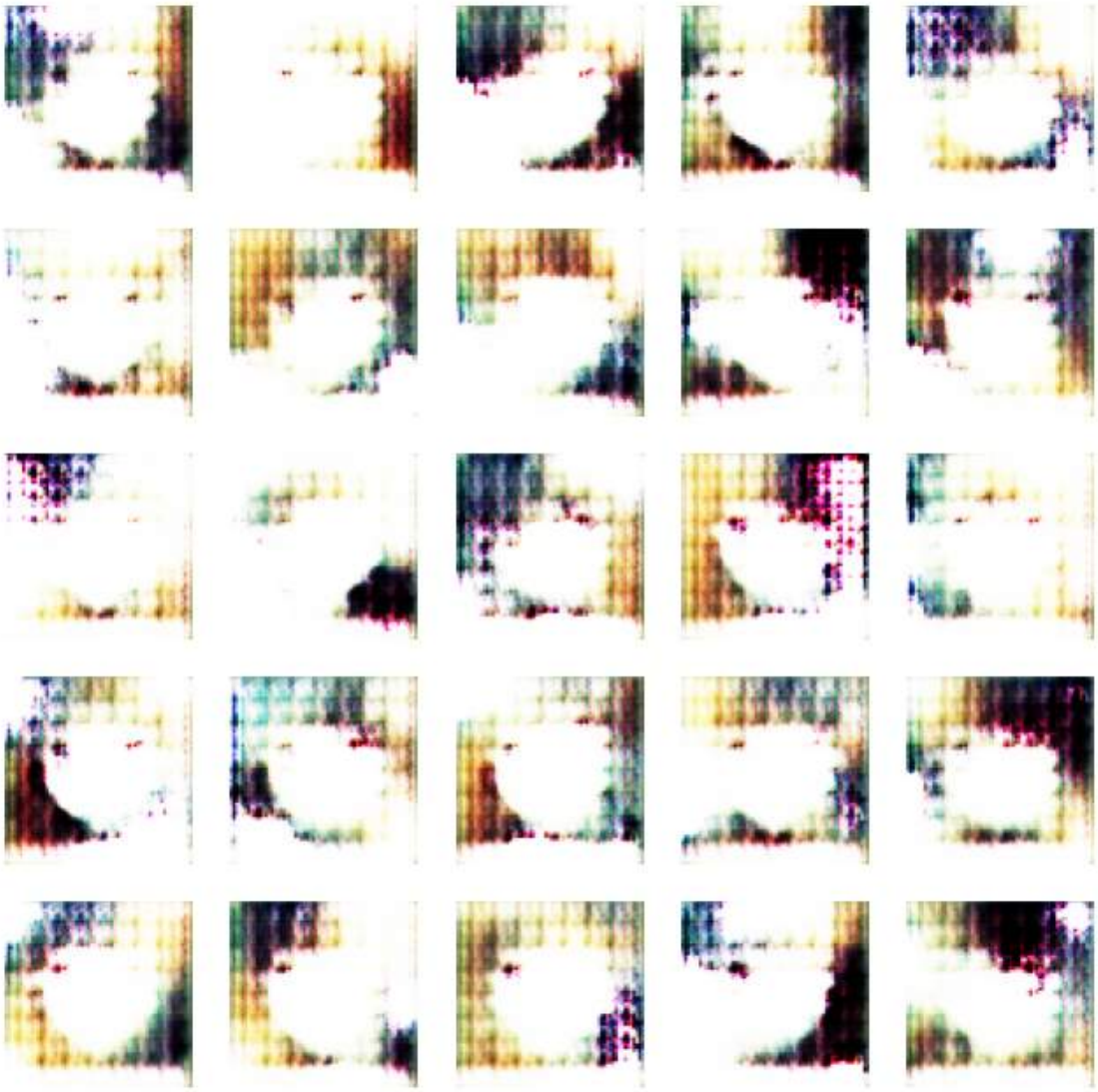
```
674/674 [==============================] - 72s 106ms/step - d_loss: -76.3981 -
g_loss: 24.6090
Epoch 21/75
674/674 [==============================] - ETA: 0s - d_loss: -82.9087 - g_loss:
59.3323
```



```
674/674 [==============================] - 72s 107ms/step - d_loss: -82.9087 -
g_loss: 59.3323
Epoch 22/75
674/674 [==============================] - ETA: 0s - d_loss: -81.2420 - g_loss:
48.7166
```

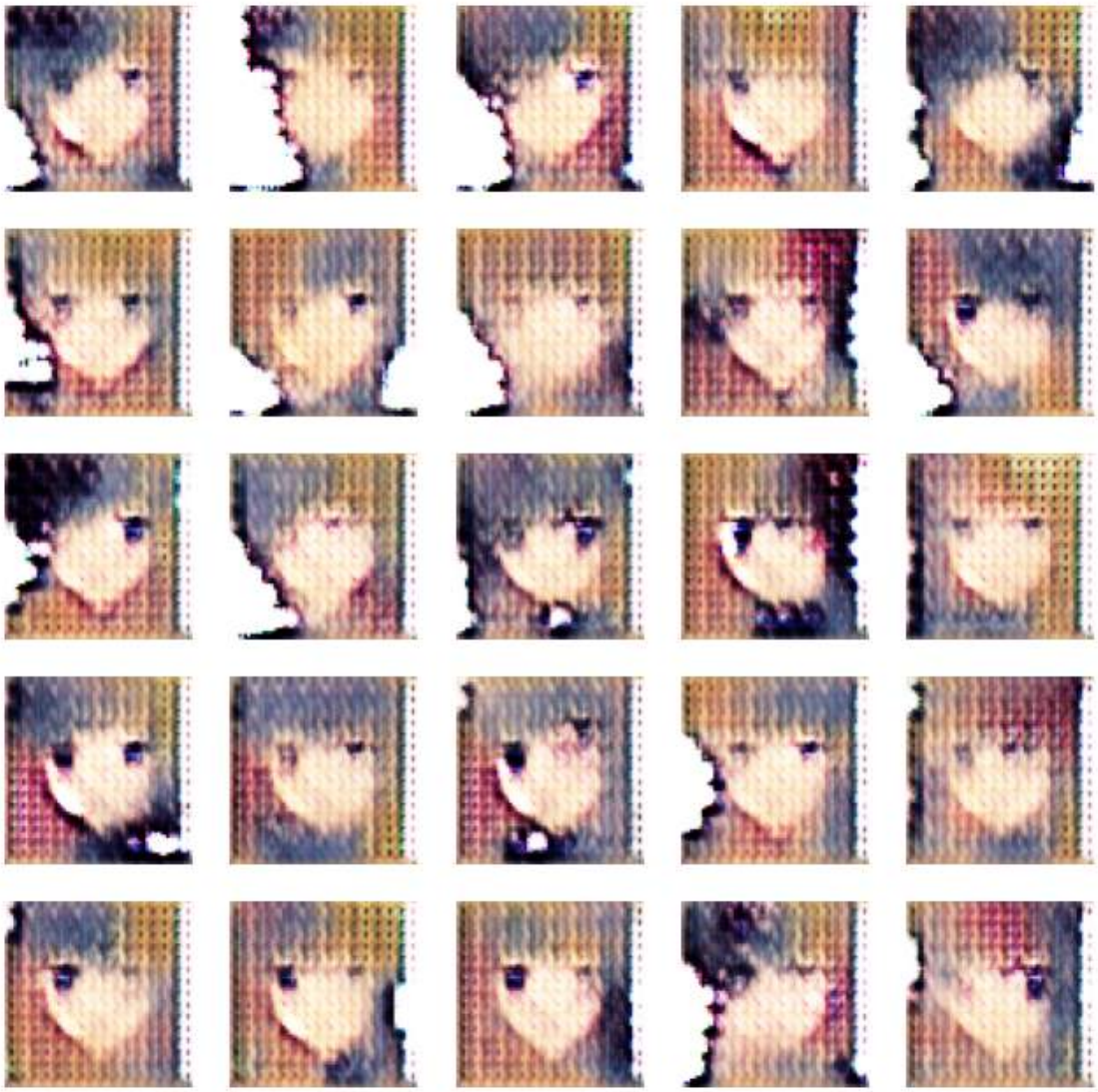


```
674/674 [==============================] - 72s 107ms/step - d_loss: -81.2420 -
g_loss: 48.7166
Epoch 23/75
674/674 [==============================] - ETA: 0s - d_loss: -79.7168 - g_loss:
74.6912
```

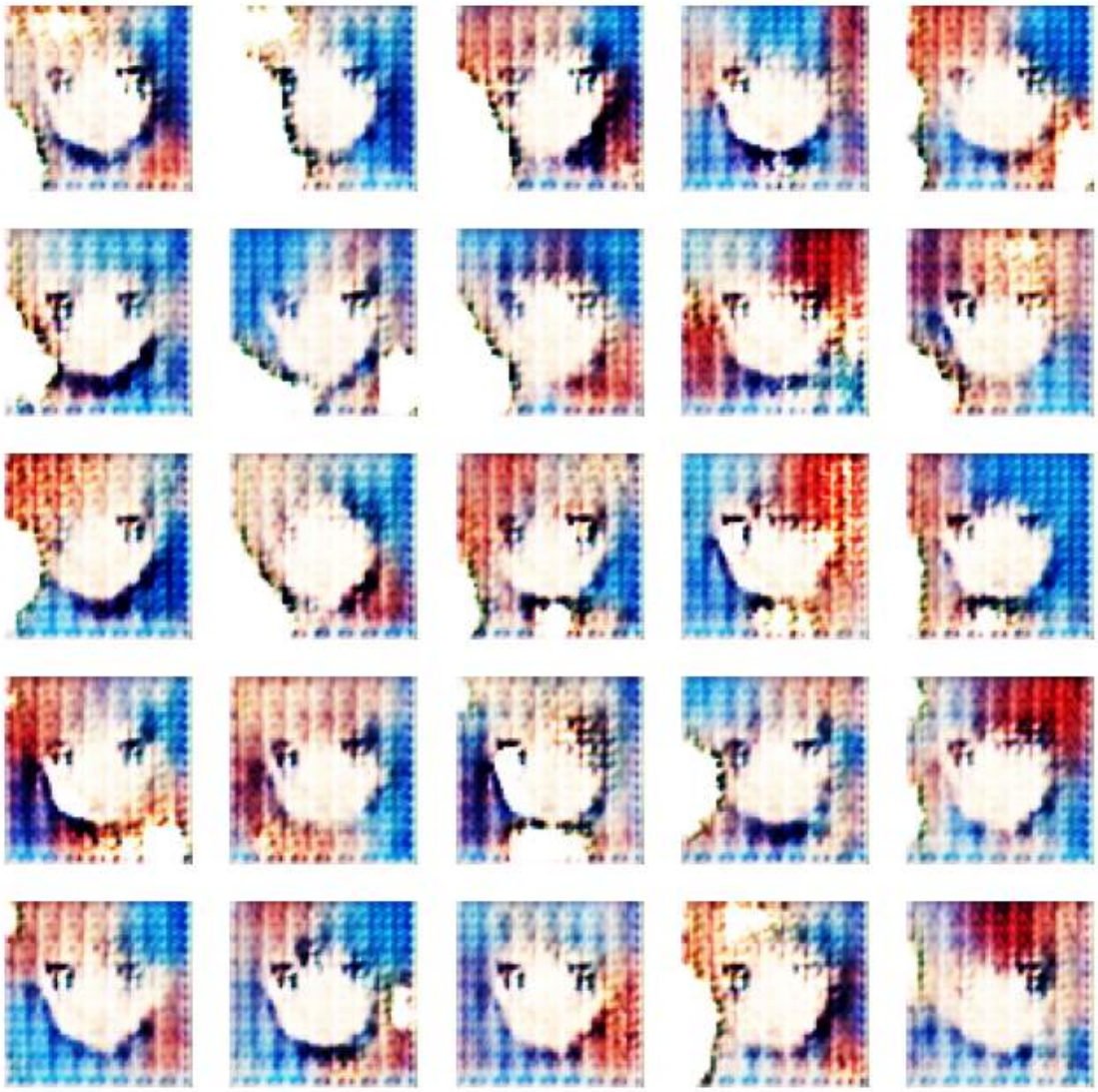


```
674/674 [==============================] - 72s 107ms/step - d_loss: -79.7168 -
g_loss: 74.6912
Epoch 24/75
674/674 [==============================] - ETA: 0s - d_loss: -83.0436 - g_loss:
35.3425
```

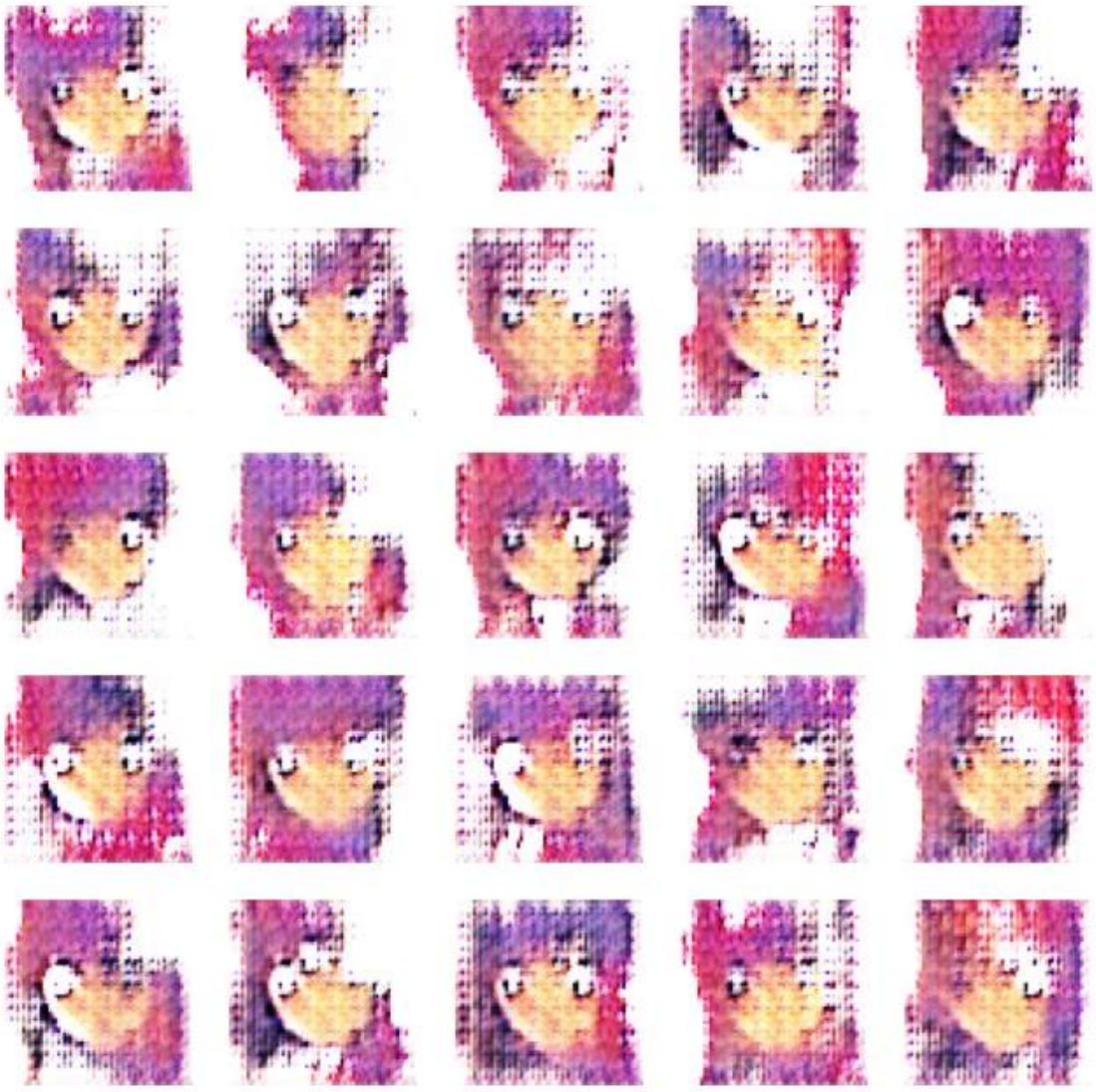
```
674/674 [==============================] - 72s 107ms/step - d_loss: -83.0436 -
g_loss: 35.3425
Epoch 25/75
674/674 [==============================] - ETA: 0s - d_loss: -85.4488 - g_loss:
50.1068
```



```
674/674 [==============================] - 72s 107ms/step - d_loss: -85.4488 -
g_loss: 50.1068
Epoch 26/75
674/674 [==============================] - ETA: 0s - d_loss: -90.5754 - g_loss:
65.0850
```



```
674/674 [==============================] - 72s 107ms/step - d_loss: -90.5754 -
g_loss: 65.0850
Epoch 27/75
674/674 [==============================] - ETA: 0s - d_loss: -98.5745 - g_loss:
38.6338
```

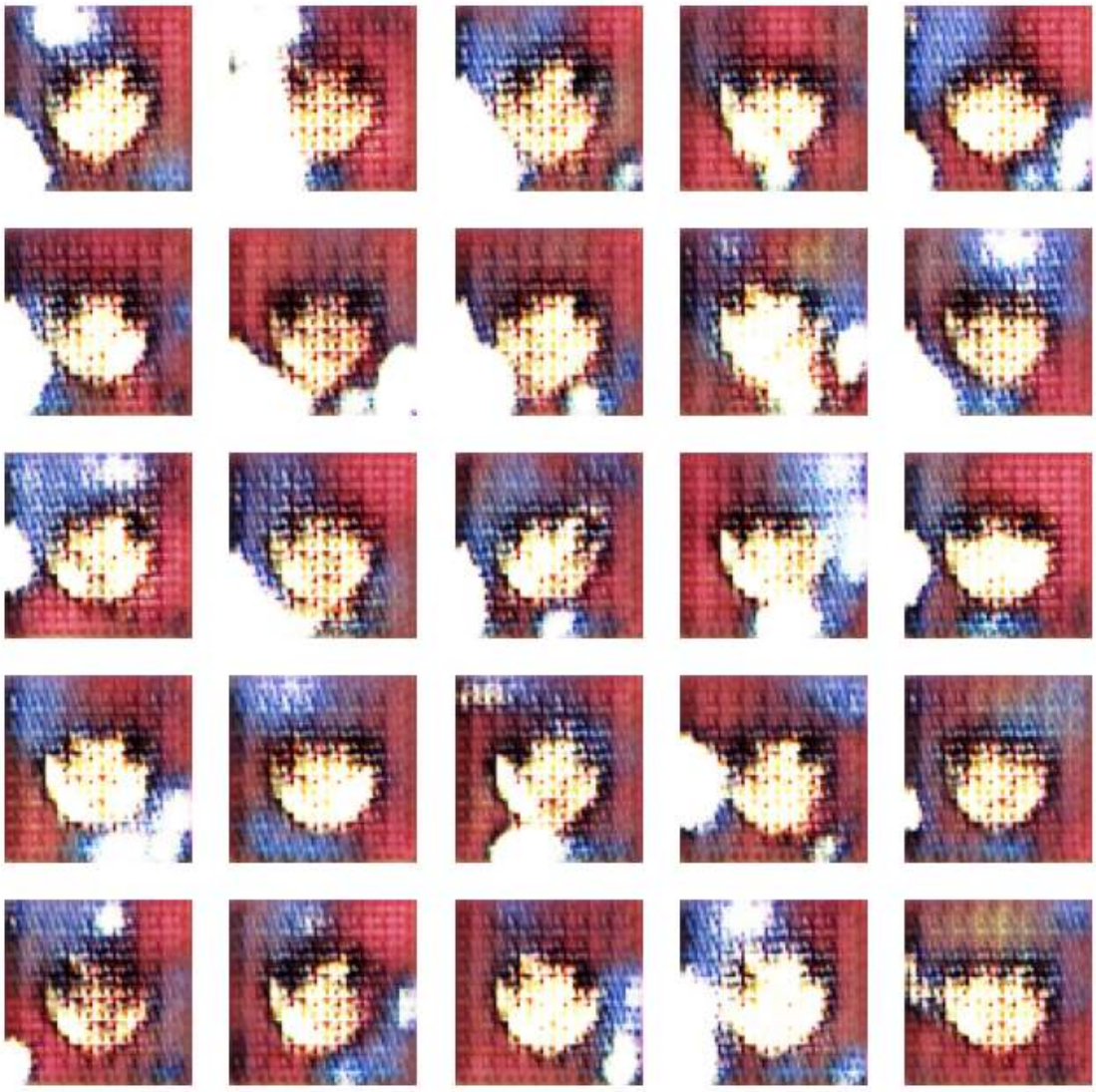


```
674/674 [==============================] - 72s 107ms/step - d_loss: -98.5745 -
g_loss: 38.6338
Epoch 28/75
674/674 [==============================] - ETA: 0s - d_loss: -69.6181 - g_loss:
51.9661
```

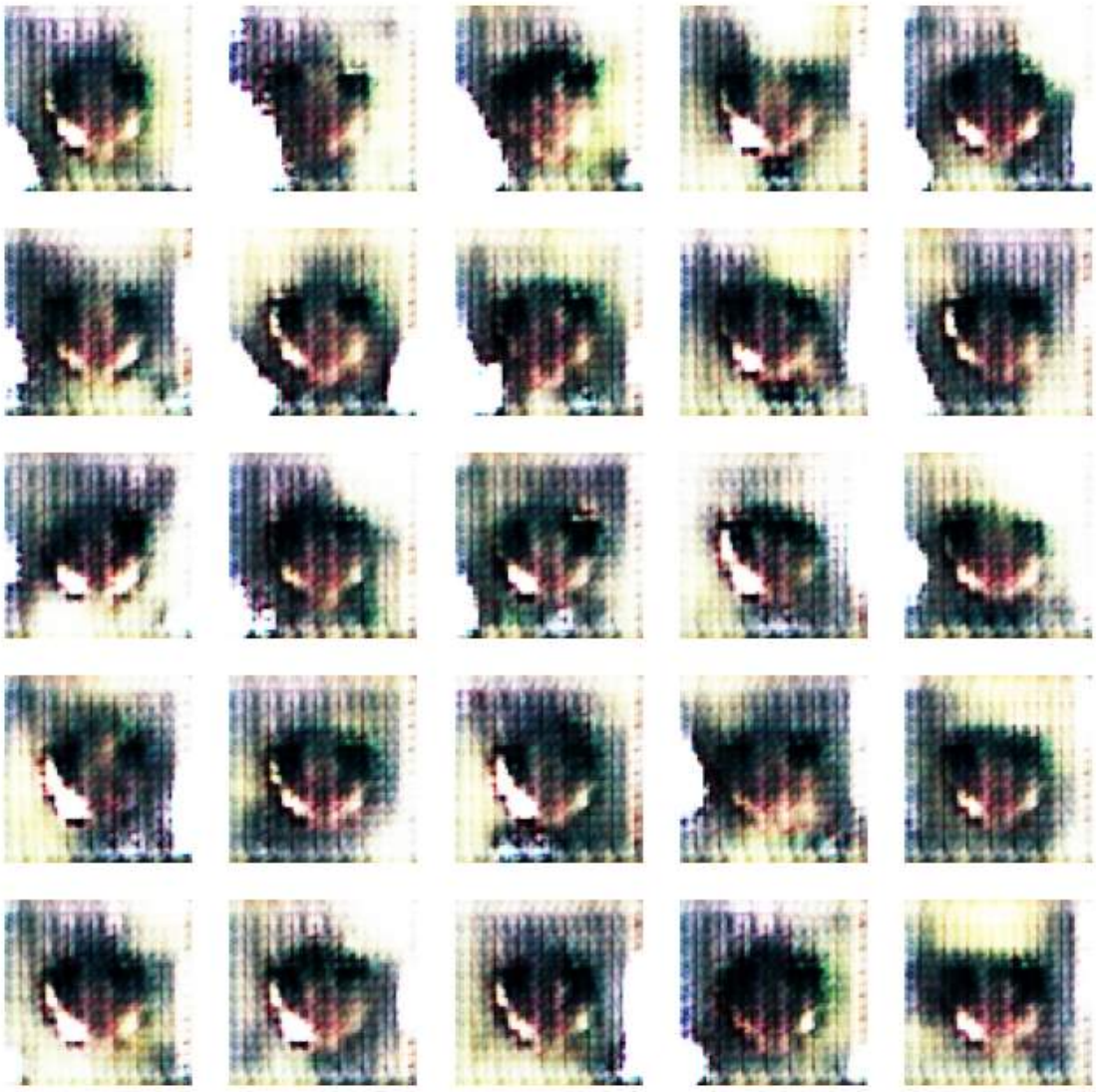
```
674/674 [==============================] - 72s 107ms/step - d_loss: -69.6181 -
g_loss: 51.9661
Epoch 29/75
674/674 [==============================] - ETA: 0s - d_loss: -90.8580 - g_loss:
34.6157
```

```
674/674 [==============================] - 72s 107ms/step - d_loss: -90.8580 -
g_loss: 34.6157
Epoch 30/75
674/674 [==============================] - ETA: 0s - d_loss: -84.0308 - g_loss:
49.5511
```

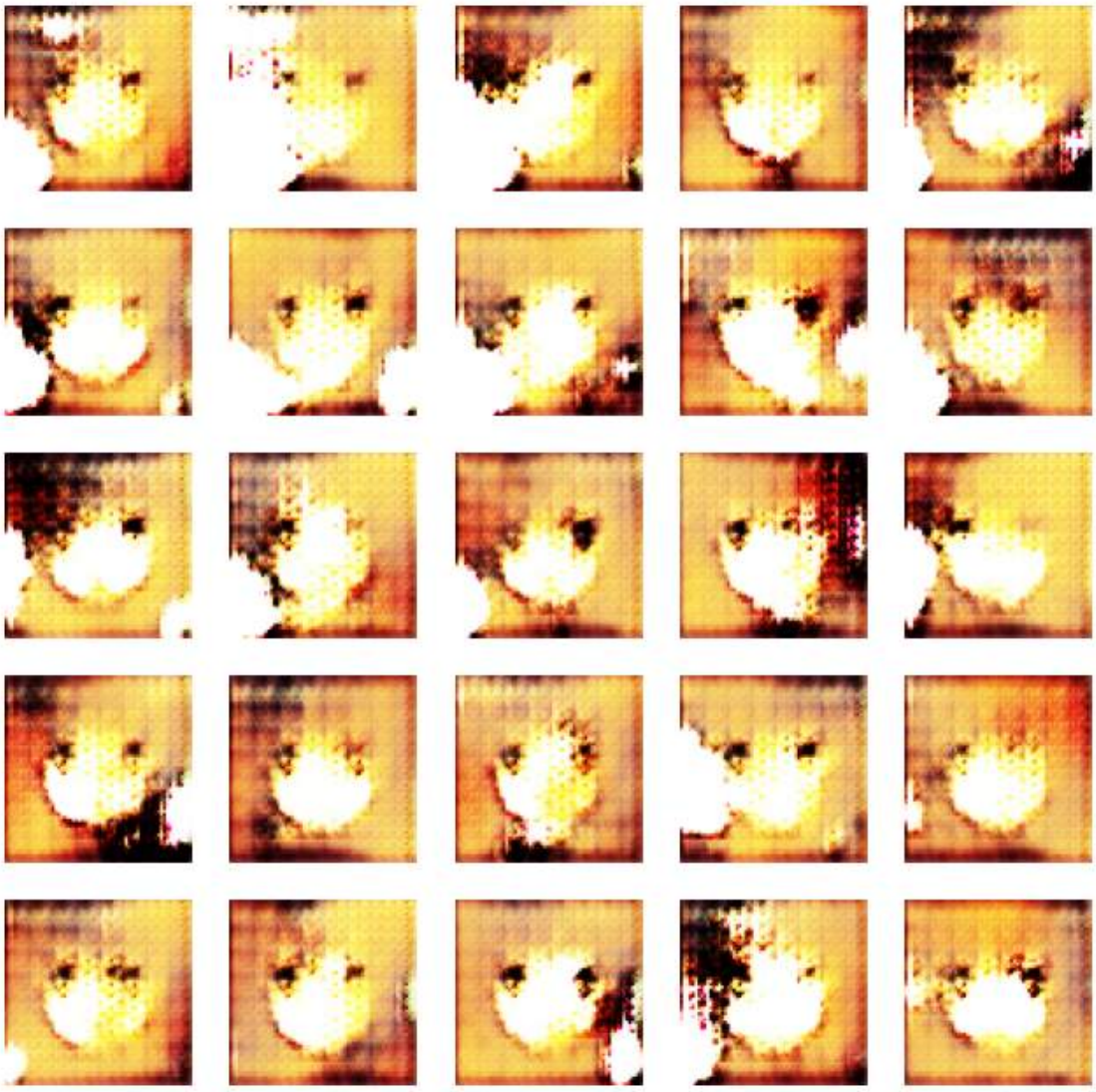
```
674/674 [==============================] - 72s 107ms/step - d_loss: -84.0308 -
g_loss: 49.5511
Epoch 31/75
674/674 [==============================] - ETA: 0s - d_loss: -115.9056 - g_loss:
43.4240
```



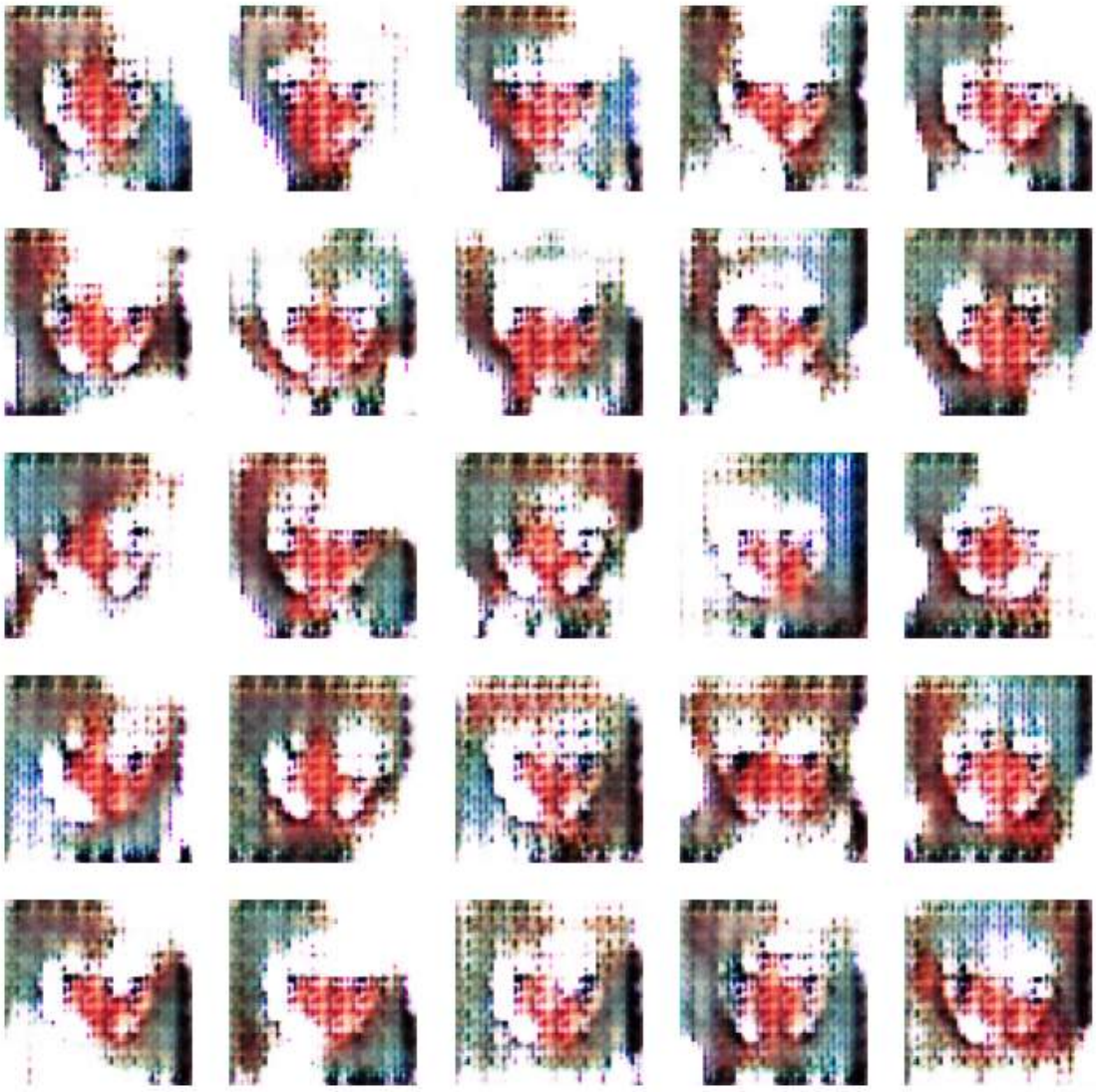
```
674/674 [==============================] - 72s 107ms/step - d_loss: -115.9056 -
g_loss: 43.4240
Epoch 32/75
674/674 [==============================] - ETA: 0s - d_loss: -114.8881 - g_loss:
34.0537
```



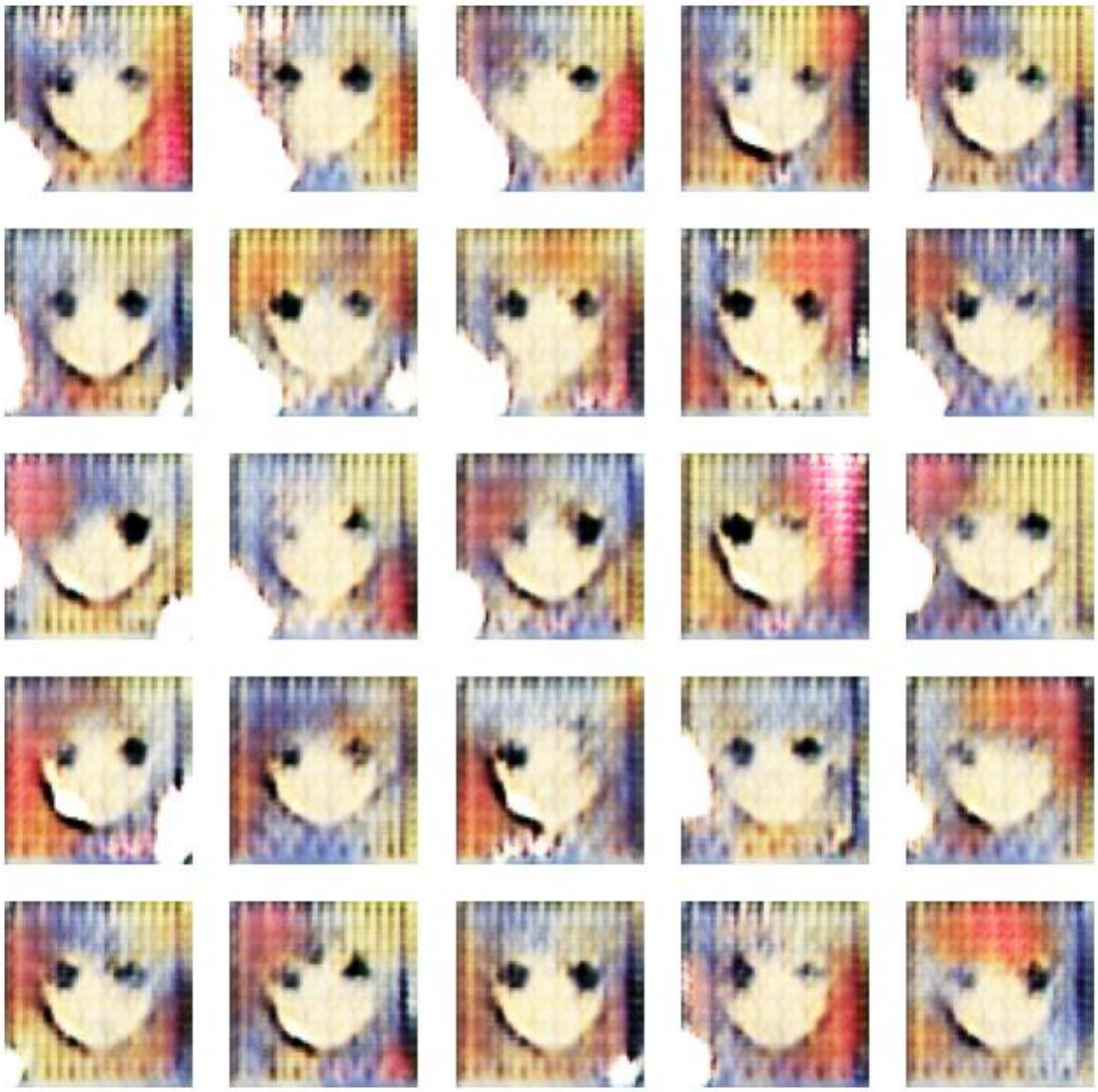
```
674/674 [==============================] - 72s 107ms/step - d_loss: -114.8881 -
g_loss: 34.0537
Epoch 33/75
674/674 [==============================] - ETA: 0s - d_loss: -139.0019 - g_loss:
56.8829
```



```
674/674 [==============================] - 72s 107ms/step - d_loss: -139.0019 -
g_loss: 56.8829
Epoch 34/75
674/674 [==============================] - ETA: 0s - d_loss: -114.4027 - g_loss:
55.5309
```



```
674/674 [==============================] - 72s 107ms/step - d_loss: -114.4027 -
g_loss: 55.5309
Epoch 35/75
674/674 [==============================] - ETA: 0s - d_loss: -117.8698 - g_loss:
63.5535
```

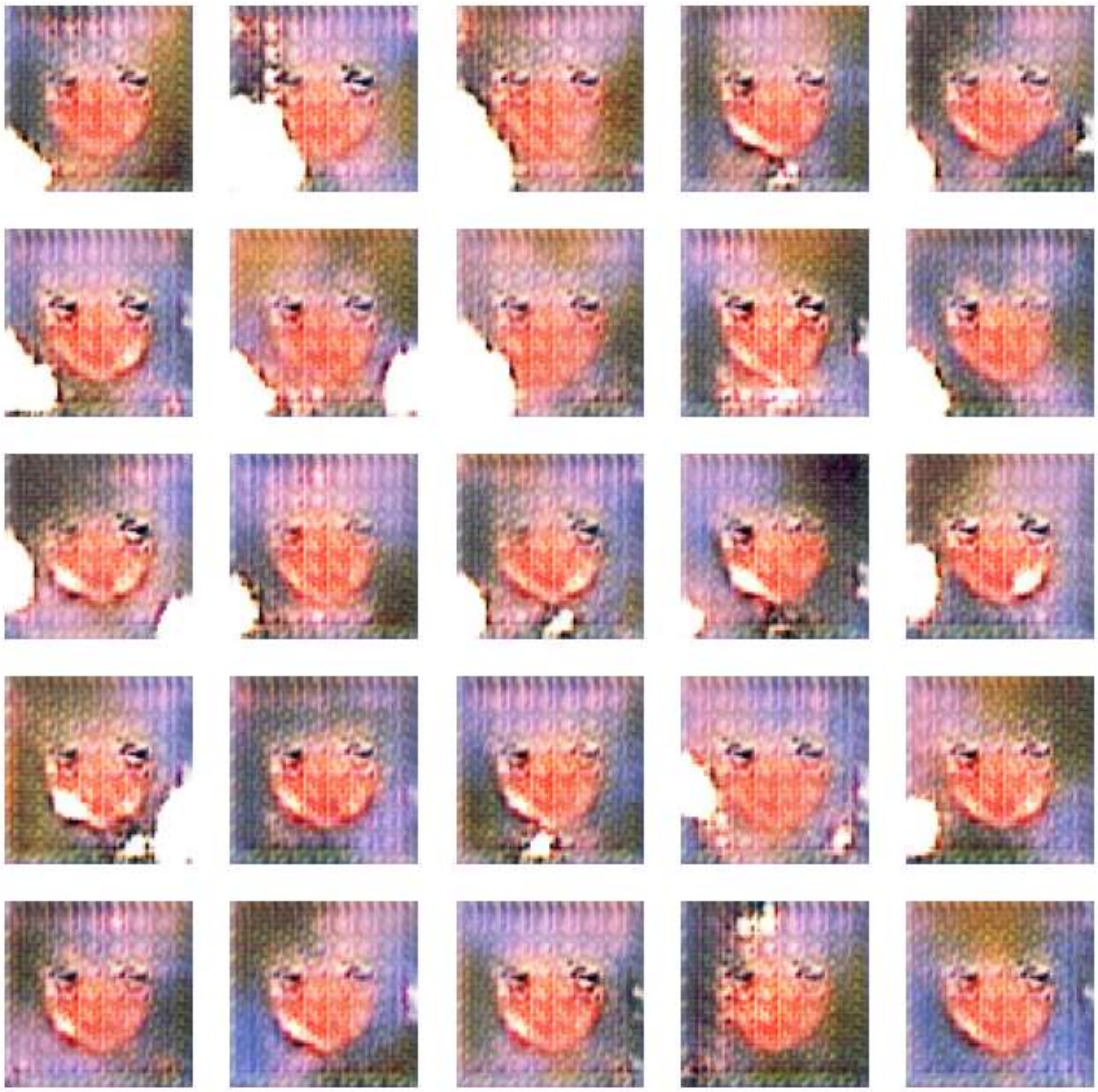


```
674/674 [==============================] - 72s 107ms/step - d_loss: -117.8698 -
g_loss: 63.5535
Epoch 36/75
674/674 [==============================] - ETA: 0s - d_loss: -115.0989 - g_loss:
90.0934
```

```
674/674 [==============================] - 72s 107ms/step - d_loss: -115.0989 -
g_loss: 90.0934
Epoch 37/75
674/674 [==============================] - ETA: 0s - d_loss: -67.1854 - g_loss:
55.4376
```

```
674/674 [==============================] - 72s 107ms/step - d_loss: -67.1854 -
g_loss: 55.4376
Epoch 38/75
674/674 [==============================] - ETA: 0s - d_loss: -113.1154 - g_loss:
46.0145
```

```
674/674 [==============================] - 72s 107ms/step - d_loss: -113.1154 -
g_loss: 46.0145
Epoch 39/75
674/674 [==============================] - ETA: 0s - d_loss: -127.1840 - g_loss:
102.1515
```

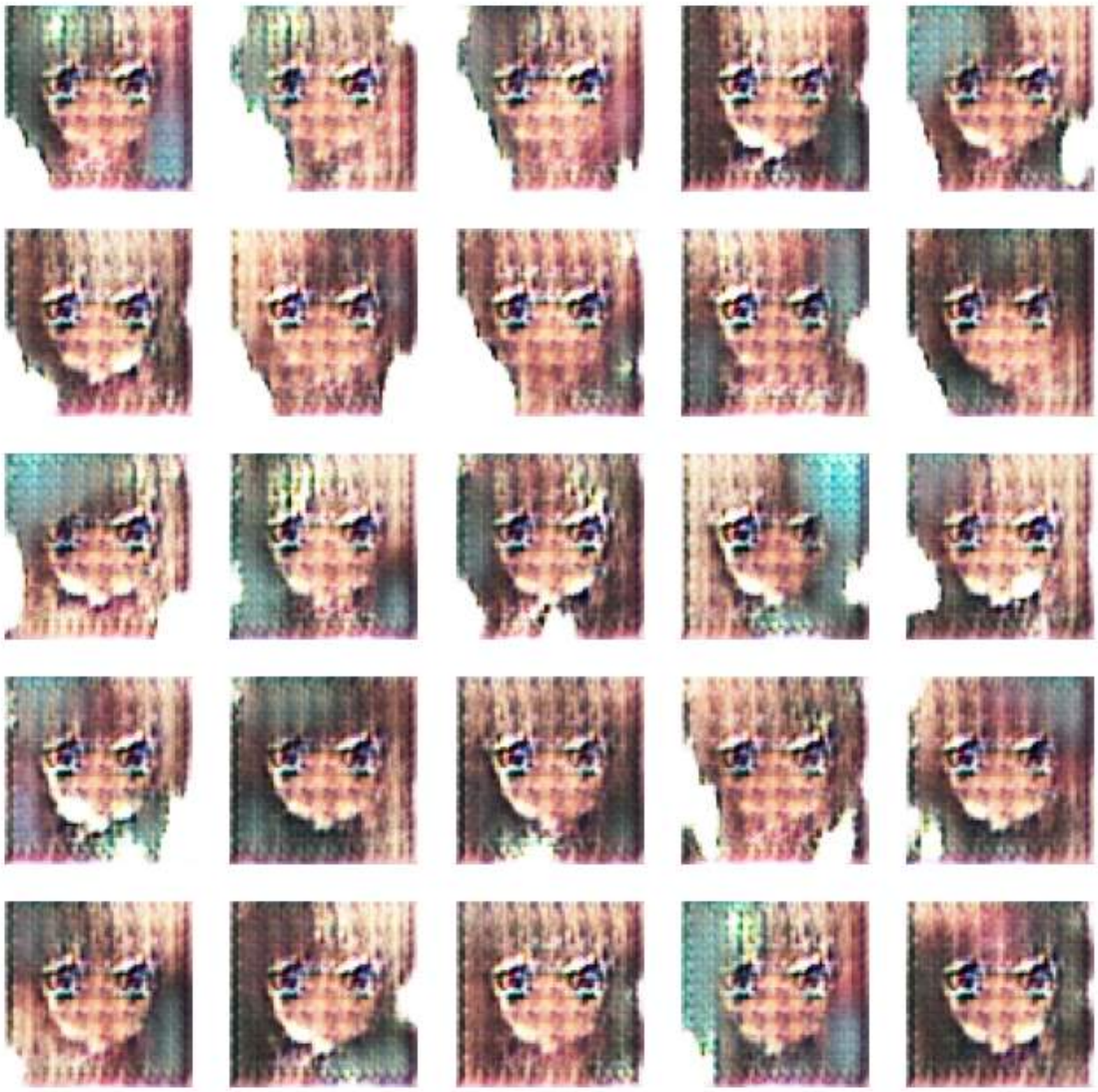


```
674/674 [==============================] - 72s 107ms/step - d_loss: -127.1840 -
g_loss: 102.1515
Epoch 40/75
674/674 [==============================] - ETA: 0s - d_loss: -157.2643 - g_loss:
62.2345
```

```
674/674 [==============================] - 72s 107ms/step - d_loss: -157.2643 -
g_loss: 62.2345
Epoch 41/75
674/674 [==============================] - ETA: 0s - d_loss: -130.5483 - g_loss:
76.4238
```

```
674/674 [==============================] - 72s 107ms/step - d_loss: -130.5483 -
g_loss: 76.4238
Epoch 42/75
674/674 [==============================] - ETA: 0s - d_loss: -129.4716 - g_loss:
79.0026
```

```
674/674 [==============================] - 72s 107ms/step - d_loss: -129.4716 -
g_loss: 79.0026
Epoch 43/75
674/674 [==============================] - ETA: 0s - d_loss: -87.1903 - g_loss:
75.0035
```



```
674/674 [==============================] - 72s 107ms/step - d_loss: -87.1903 -
g_loss: 75.0035
Epoch 44/75
674/674 [==============================] - ETA: 0s - d_loss: -115.1673 - g_loss:
88.9074
```

```
674/674 [==============================] - 72s 106ms/step - d_loss: -115.1673 -
g_loss: 88.9074
Epoch 45/75
674/674 [==============================] - ETA: 0s - d_loss: -64.9345 - g_loss:
76.2938
```

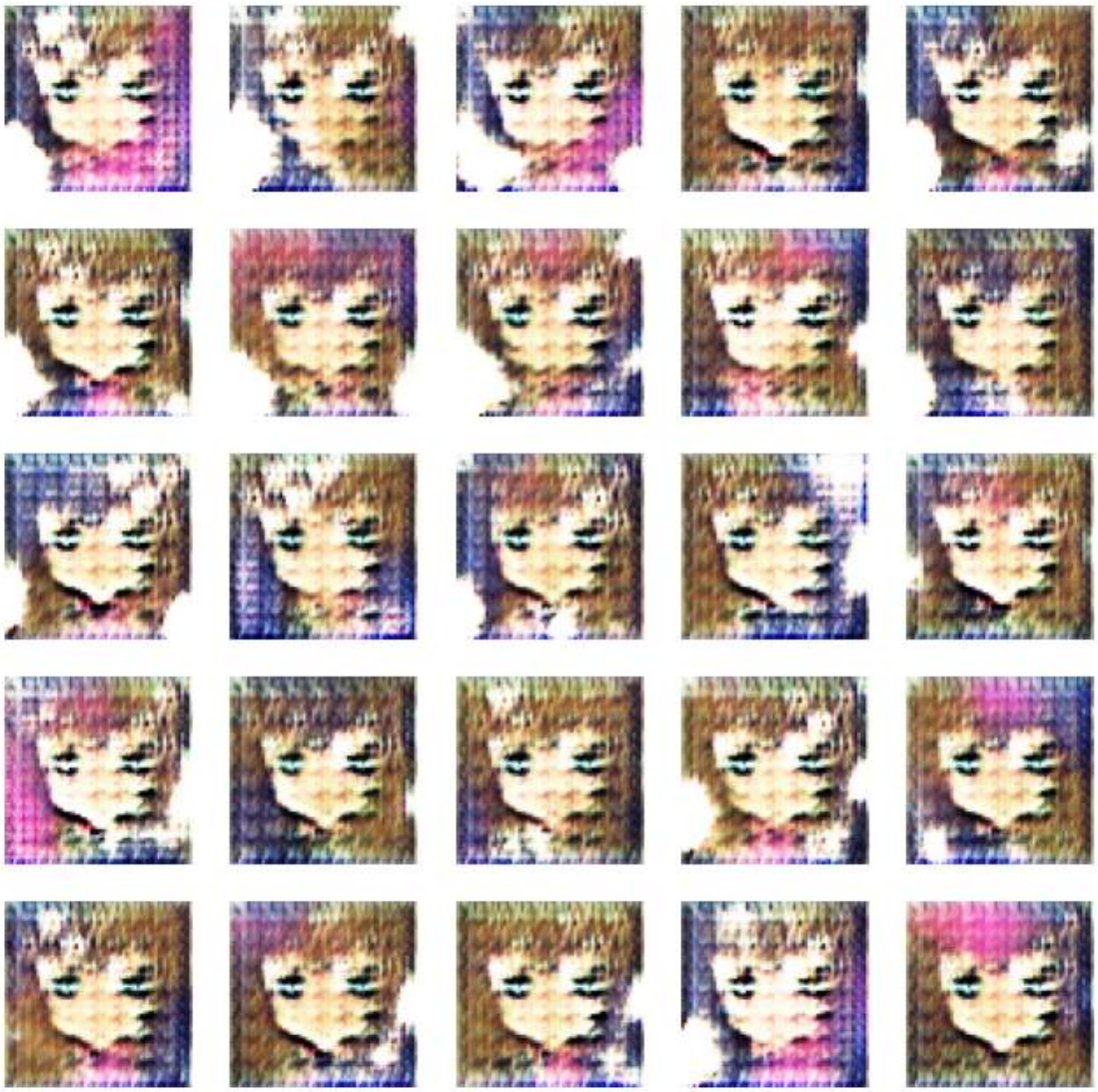
```
674/674 [==============================] - 72s 107ms/step - d_loss: -64.9345 -
g_loss: 76.2938
Epoch 46/75
674/674 [==============================] - ETA: 0s - d_loss: -65.1819 - g_loss:
42.0368
```

```
674/674 [==============================] - 72s 107ms/step - d_loss: -65.1819 -
g_loss: 42.0368
Epoch 47/75
674/674 [==============================] - ETA: 0s - d_loss: -79.1877 - g_loss:
48.1109
```

```
674/674 [==============================] - 72s 107ms/step - d_loss: -79.1877 -
g_loss: 48.1109
Epoch 48/75
674/674 [==============================] - ETA: 0s - d_loss: -96.5779 - g_loss:
93.5472
```

```
674/674 [==============================] - 72s 107ms/step - d_loss: -96.5779 -
g_loss: 93.5472
Epoch 49/75
674/674 [==============================] - ETA: 0s - d_loss: -114.5992 - g_loss:
86.4651
```

```
674/674 [==============================] - 72s 107ms/step - d_loss: -114.5992 -
g_loss: 86.4651
Epoch 50/75
674/674 [==============================] - ETA: 0s - d_loss: -99.5894 - g_loss:
70.2929
```



```
674/674 [==============================] - 72s 107ms/step - d_loss: -99.5894 -
g_loss: 70.2929
Epoch 51/75
674/674 [==============================] - ETA: 0s - d_loss: -134.8259 - g_loss:
72.6984
```

```
674/674 [==============================] - 72s 107ms/step - d_loss: -134.8259 -
g_loss: 72.6984
Epoch 52/75
674/674 [==============================] - ETA: 0s - d_loss: -90.8337 - g_loss:
123.1768
```

```
674/674 [==============================] - 72s 107ms/step - d_loss: -90.8337 -
g_loss: 123.1768
Epoch 53/75
674/674 [==============================] - ETA: 0s - d_loss: -123.3144 - g_loss:
121.7697
```

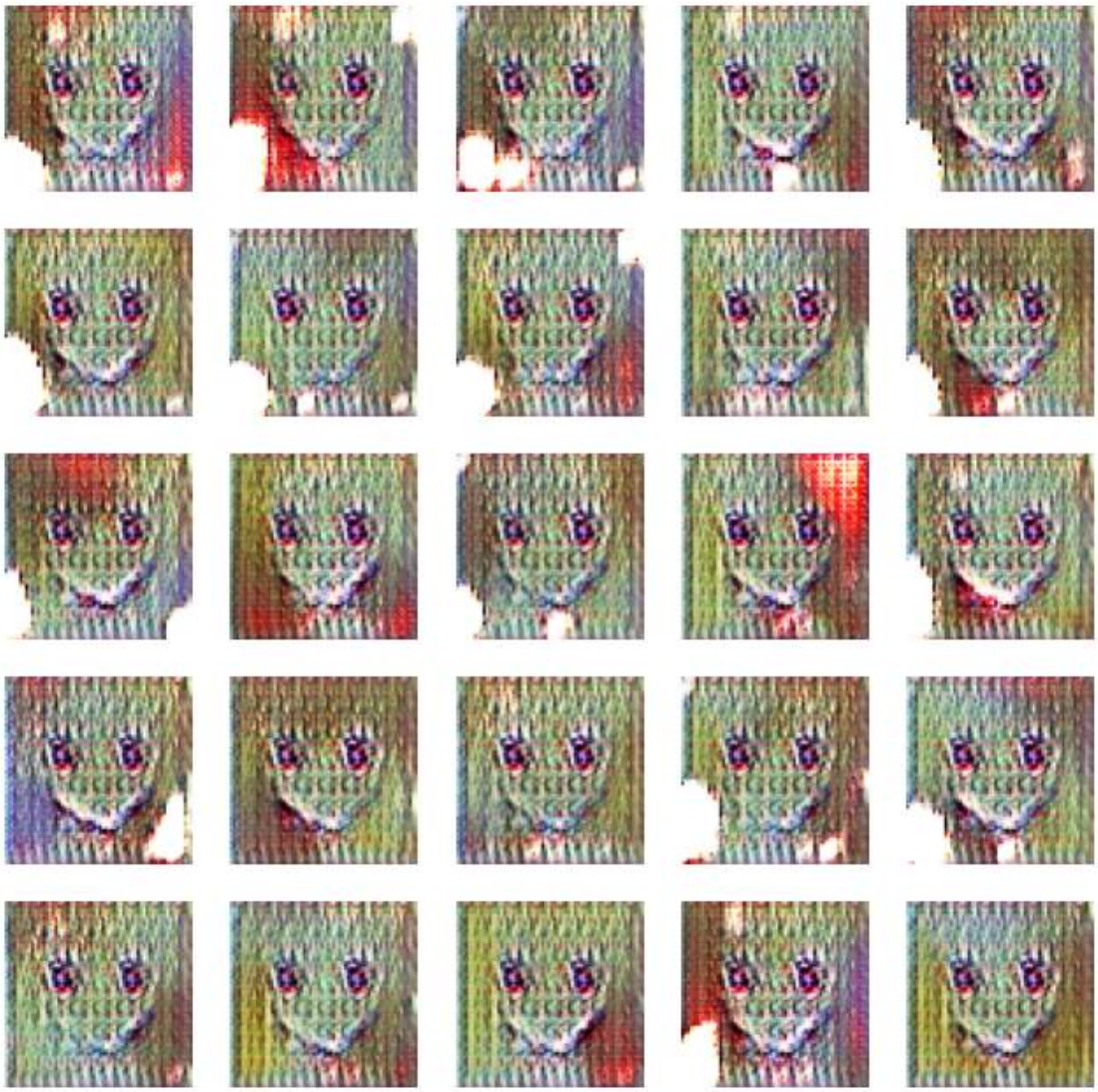
```
674/674 [==============================] - 72s 107ms/step - d_loss: -123.3144 -
g_loss: 121.7697
Epoch 54/75
674/674 [==============================] - ETA: 0s - d_loss: -143.4628 - g_loss:
73.6266
```

```
674/674 [==============================] - 73s 108ms/step - d_loss: -143.4628 -
g_loss: 73.6266
Epoch 55/75
674/674 [==============================] - ETA: 0s - d_loss: -148.0184 - g_loss:
61.3909
```

```
674/674 [==============================] - 72s 107ms/step - d_loss: -148.0184 -
g_loss: 61.3909
Epoch 56/75
674/674 [==============================] - ETA: 0s - d_loss: -90.0993 - g_loss:
62.1136
```

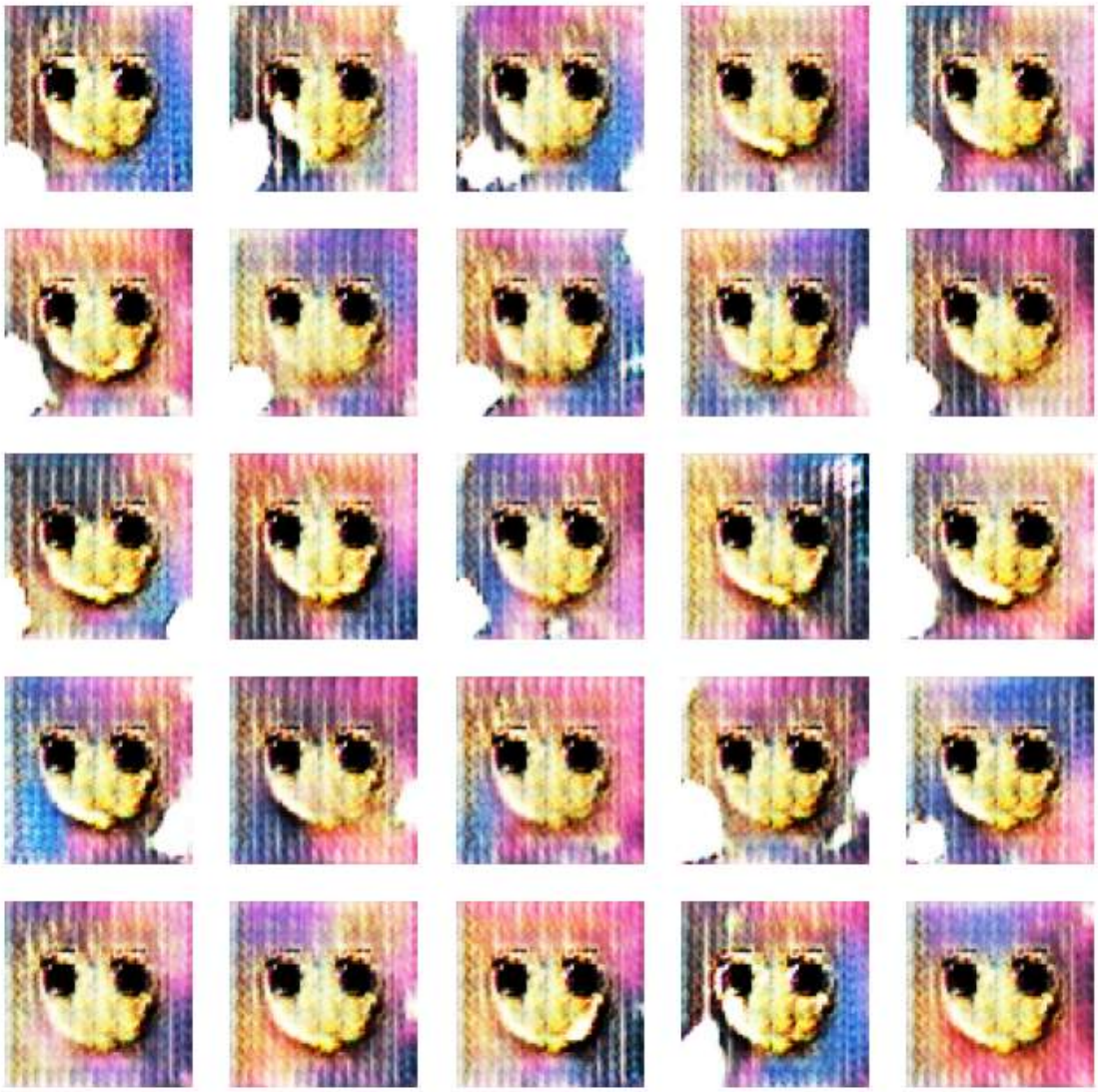
```
674/674 [==============================] - 72s 107ms/step - d_loss: -90.0993 -
g_loss: 62.1136
Epoch 57/75
674/674 [==============================] - ETA: 0s - d_loss: -119.0370 - g_loss:
91.0556
```

```
674/674 [==============================] - 73s 108ms/step - d_loss: -119.0370 -
g_loss: 91.0556
Epoch 58/75
674/674 [==============================] - ETA: 0s - d_loss: -160.3362 - g_loss:
50.4637
```



```
674/674 [==============================] - 72s 107ms/step - d_loss: -160.3362 -
g_loss: 50.4637
Epoch 59/75
674/674 [==============================] - ETA: 0s - d_loss: -134.8551 - g_loss:
119.6001
```

```
674/674 [==============================] - 72s 107ms/step - d_loss: -134.8551 -
g_loss: 119.6001
Epoch 60/75
674/674 [==============================] - ETA: 0s - d_loss: -172.4345 - g_loss:
125.1580
```



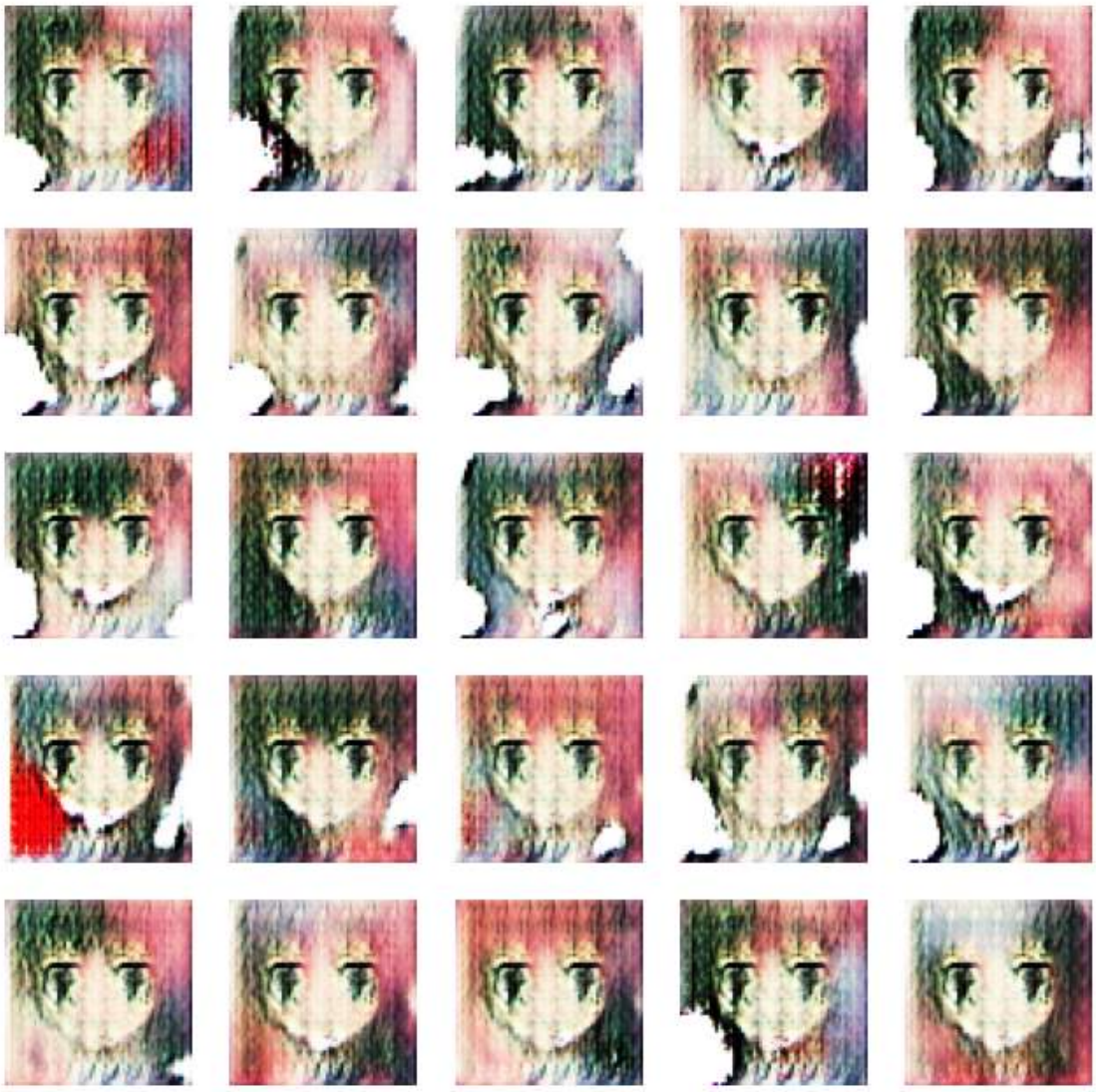
```
674/674 [==============================] - 72s 108ms/step - d_loss: -172.4345 -
g_loss: 125.1580
Epoch 61/75
674/674 [==============================] - ETA: 0s - d_loss: -174.4781 - g_loss:
101.6688
```

```
674/674 [==============================] - 72s 107ms/step - d_loss: -174.4781 -
g_loss: 101.6688
Epoch 62/75
674/674 [==============================] - ETA: 0s - d_loss: -167.9291 - g_loss:
100.3695
```

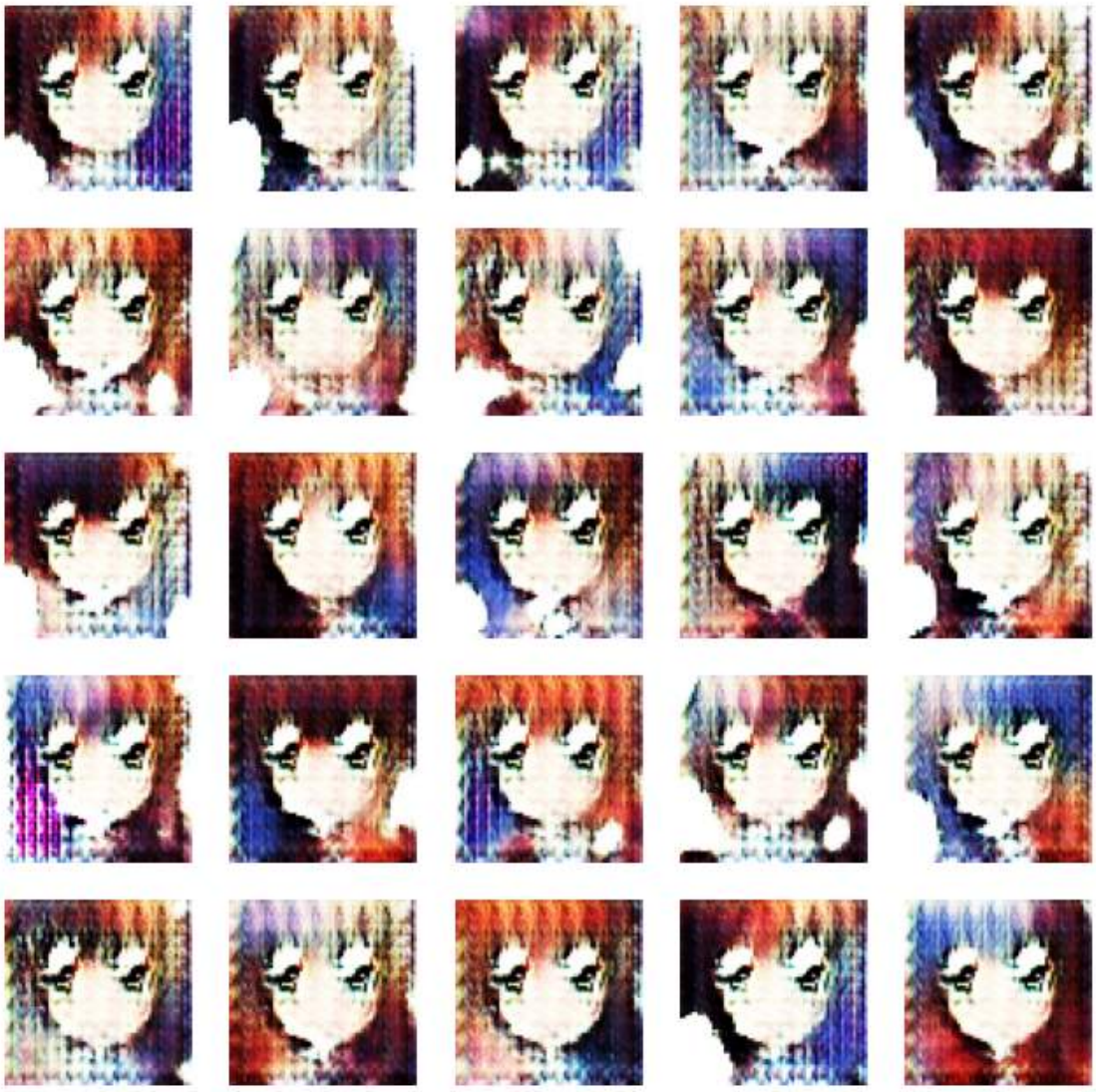
```
674/674 [==============================] - 72s 107ms/step - d_loss: -167.9291 -
g_loss: 100.3695
Epoch 63/75
674/674 [==============================] - ETA: 0s - d_loss: -143.9668 - g_loss:
122.3489
```

```
674/674 [==============================] - 73s 108ms/step - d_loss: -143.9668 -
g_loss: 122.3489
Epoch 64/75
674/674 [==============================] - ETA: 0s - d_loss: -174.4255 - g_loss:
78.2100
```

```
674/674 [==============================] - 72s 107ms/step - d_loss: -174.4255 -
g_loss: 78.2100
Epoch 65/75
674/674 [==============================] - ETA: 0s - d_loss: -166.1821 - g_loss:
138.0456
```



```
674/674 [==============================] - 72s 107ms/step - d_loss: -166.1821 -
g_loss: 138.0456
Epoch 66/75
674/674 [==============================] - ETA: 0s - d_loss: -115.6962 - g_loss:
76.7395
```



```
674/674 [==============================] - 72s 107ms/step - d_loss: -115.6962 -
g_loss: 76.7395
Epoch 67/75
674/674 [==============================] - ETA: 0s - d_loss: -124.4037 - g_loss:
84.7016
```

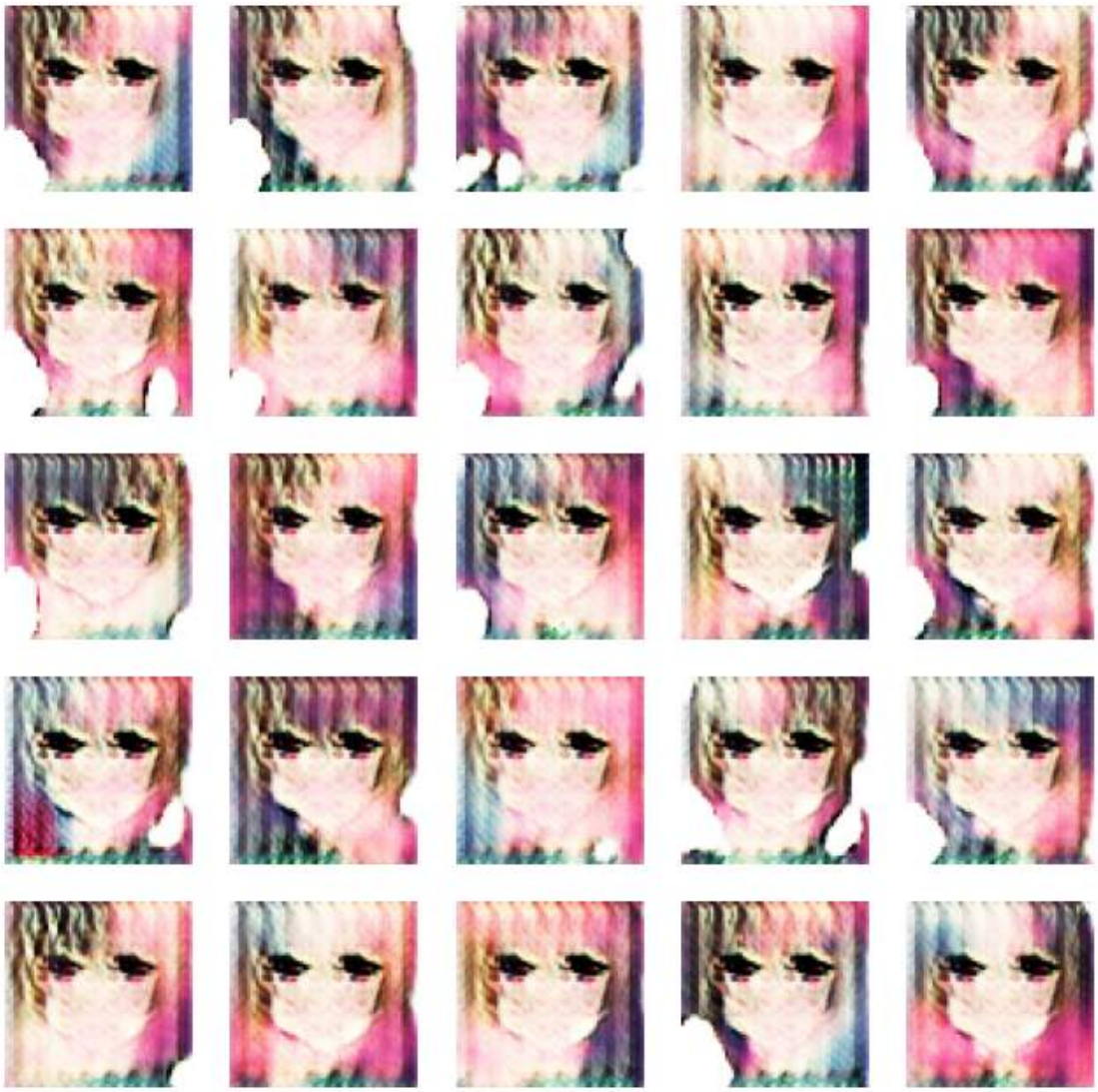
```
674/674 [==============================] - 72s 107ms/step - d_loss: -124.4037 -
g_loss: 84.7016
Epoch 68/75
674/674 [==============================] - ETA: 0s - d_loss: -137.0121 - g_loss:
104.9457
```

```
674/674 [==============================] - 72s 107ms/step - d_loss: -137.0121 -
g_loss: 104.9457
Epoch 69/75
674/674 [==============================] - ETA: 0s - d_loss: -212.3623 - g_loss:
175.7818
```

```
674/674 [==============================] - 72s 107ms/step - d_loss: -212.3623 -
g_loss: 175.7818
Epoch 70/75
674/674 [==============================] - ETA: 0s - d_loss: -107.8471 - g_loss:
71.3133
```

```
674/674 [==============================] - 72s 107ms/step - d_loss: -107.8471 -
g_loss: 71.3133
Epoch 71/75
674/674 [==============================] - ETA: 0s - d_loss: -193.2453 - g_loss:
57.7298
```

```
674/674 [==============================] - 72s 107ms/step - d_loss: -193.2453 -
g_loss: 57.7298
Epoch 72/75
674/674 [==============================] - ETA: 0s - d_loss: -226.4283 - g_loss:
107.6847
```



```
674/674 [==============================] - 72s 107ms/step - d_loss: -226.4283 -
g_loss: 107.6847
Epoch 73/75
674/674 [==============================] - ETA: 0s - d_loss: -211.5102 - g_loss:
96.6939
```

```
674/674 [==============================] - 72s 106ms/step - d_loss: -211.5102 -
g_loss: 96.6939
Epoch 74/75
674/674 [==============================] - ETA: 0s - d_loss: -199.7720 - g_loss:
116.8044
```

8 Generate New Anime Image

[22]:
```
noise = tf.random.normal([1, 100])
fig = plt.figure(figsize=(3, 3))
# generate the image from noise
g_img = dcgan.generator(noise)
```

```
# denormalize the image
g_img = (g_img * 127.5) + 127.5
g_img.numpy()
img = array_to_img(g_img[0])
plt.imshow(img)
plt.axis('off')
# plt.savefig('epoch_{:03d}.png'.format(epoch))
plt.show()
```