

Cel laboratorium

Celem laboratorium było zbudowanie własnej architektury sieci GAN przy pomocy pakietów TensorFlow i Keras, a następnie przeprowadzenie eksperymentów badających jakość klasyfikacji dla wykorzystując obrazy z bazy danych CIFAR. Wyniki należało porównać obiektywnie i subiektywnie.

Własna architektura sieci

Poniżej zaproponowana własna architektura sieci GAN. Najlepiej pokazać kod, który mówi sam za siebie.

Generator

```
def make_generator_model(resolution: int, noise_dim: int) -> Sequential:
    r2 = int(resolution / 2)
    r4 = int(resolution / 4)

    model = Sequential()
    model.add(Dense(r4 * r4 * 256, use_bias=False, input_shape=(noise_dim,)))
    model.add(BatchNormalization())
    model.add(LeakyReLU())

    model.add(Reshape((r4, r4, 256)))
    assert model.output_shape == (None, r4, r4, 256) # Note: None is the batch size

    model.add(Conv2DTranspose(128, (5, 5), strides=(1, 1), padding='same',
use_bias=False))
    assert model.output_shape == (None, r4, r4, 128)
    model.add(BatchNormalization())
    model.add(LeakyReLU())

    model.add(Conv2DTranspose(64, (5, 5), strides=(2, 2), padding='same',
use_bias=False))
    assert model.output_shape == (None, r2, r2, 64)
    model.add(BatchNormalization())
    model.add(LeakyReLU())

    model.add(Conv2DTranspose(3, (5, 5), strides=(2, 2), padding='same',
use_bias=False, activation='tanh'))
    assert model.output_shape == (None, resolution, resolution, 3)

    return model
```

Dyskryminator

```
def make_discriminator_model(resolution: int) -> Sequential:
    """
    Discriminator returns positives values for real images and negatives values for
    fake images,
    """
    model = Sequential()
    model.add(Conv2D(64, (5, 5), strides=(2, 2), padding='same',
input_shape=(resolution, resolution, 3)))
```

```
model.add(LeakyReLU())
model.add(Dropout(0.3))

model.add(Conv2D(128, (5, 5), strides=(2, 2), padding='same'))
model.add(LeakyReLU())
model.add(Dropout(0.3))

model.add(Flatten())
model.add(Dense(1))

return model
```

Przebieg laboratorium

Zadaniem było przy pomocy własnej architektury sieci wykonanie eksperymentów wykorzystując wspomnianą powyżej bazę danych. Wyniki należało przedstawić w formie tabel lub wykresów.

Do eksperymentów wykorzystano wersję bazy danych CIFAR-10. Wybrana klasa do generowania obrazów to pies. Bazowe ustawienia parametrów:

- liczba epok: 100
- wielkość wektora inicjalizującego generator: 100
- rozdzielczość obrazu: oryginalna, 32x32 piksele

Jako optymalizator zarówno dla generatora jak i dyskriminatora został wykorzystany optymalizator 'Adam' z parametrem 'learning_rate=0.001'.

Dla każdej eksperymentalnej konfiguracji dokonywano pomiarów czasu uczenia. Nie było potrzeby na podział danych na część uczącą oraz część testową ze względu na specyfikę sieci GAN - to generowane obrazy były częścią porównywaną z oryginalnymi obrazami.

Wpływ wielkości wektora inicjującego generator na jakość i szybkość uczenia

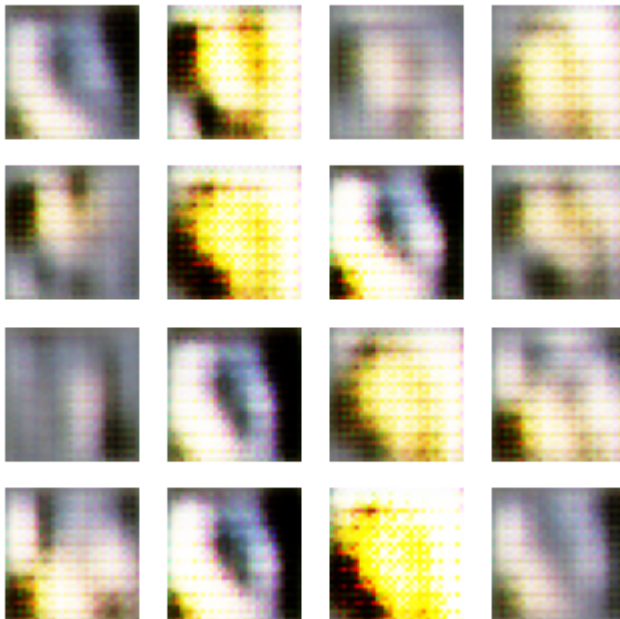
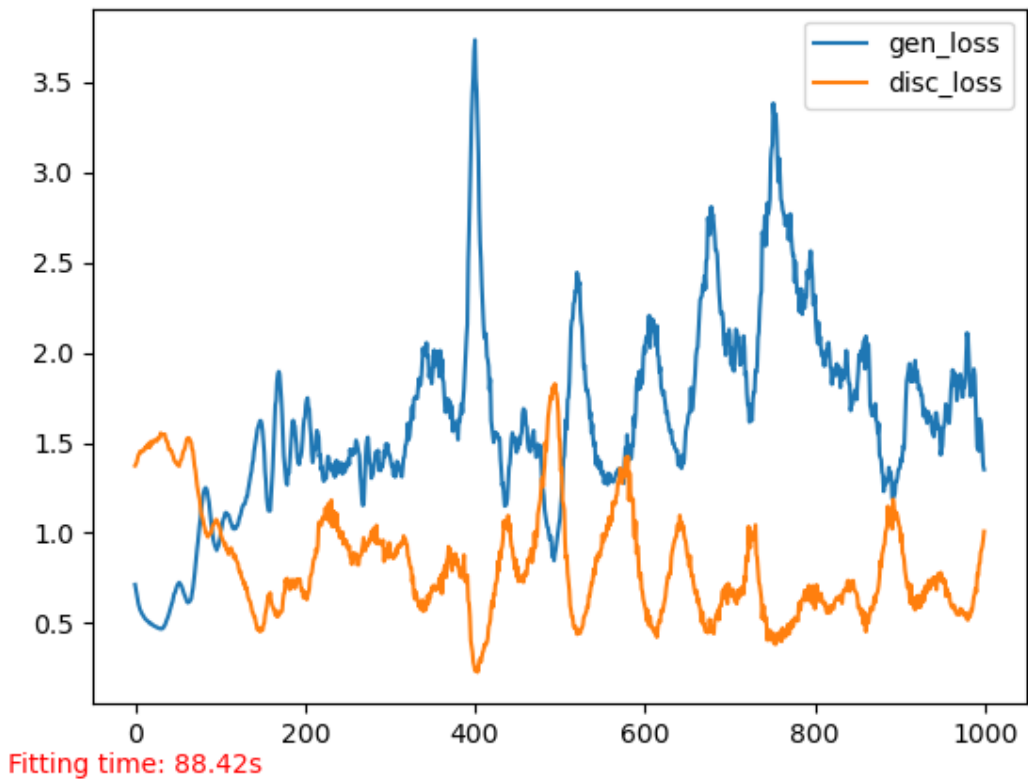
Eksperyment przeprowadzono na 3 wielkościach:

- 50,
- 100,
- 200.

Ilość epok uczenia ustawiono na 100, a rozdzielczości obrazów nie były zmieniane. Wyniki zostały zebrane w formie wykresów obrazujących wartości funkcji straty dla generatora oraz dyskriminatora w zależności od wielkości wektora inicjującego generator zależne od czasu, a także zbiorcze porównanie w postaci histogramu wpływu wielkości wektora inicjującego generator na szybkość uczenia.

Rozmiar wektora inicjującego generator = 50

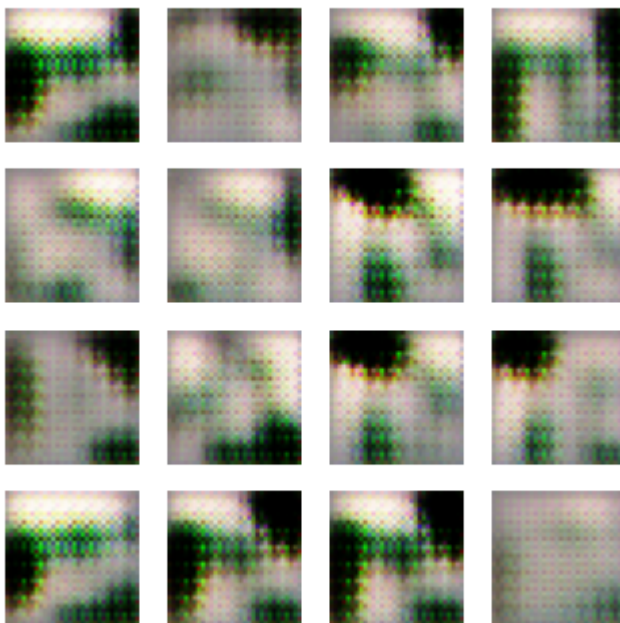
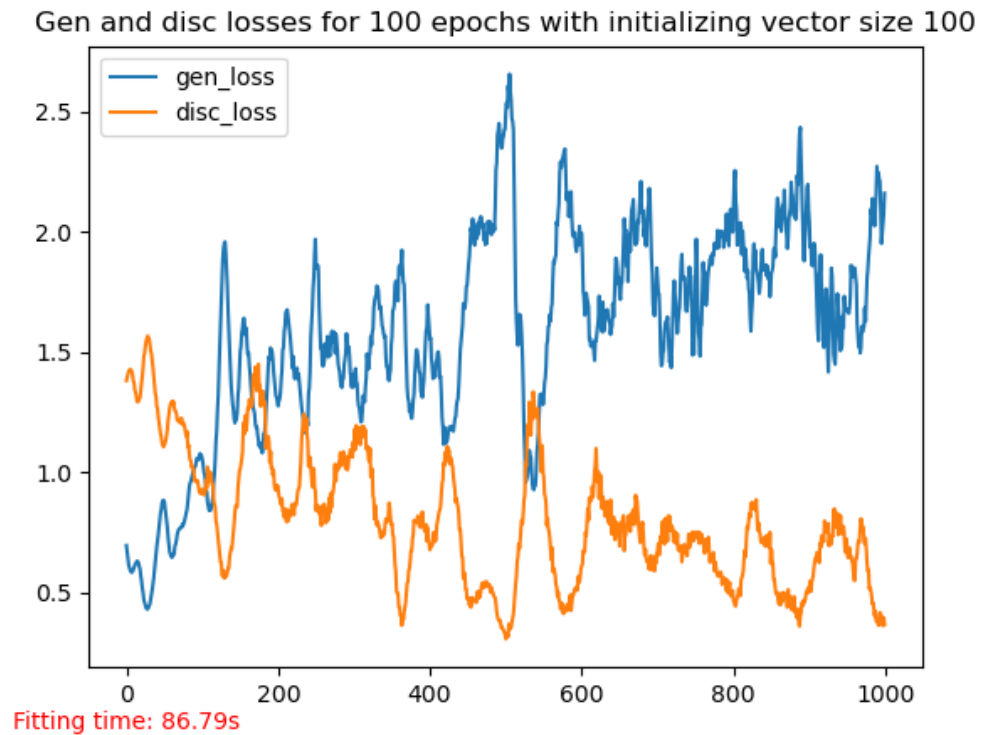
Gen and disc losses for 100 epochs with initializing vector size 50



Wygenerowane obrazy nie przypominają psa w żadnym przypadku. Generator w końcowych epokach stara się minimalizować funkcję straty, natomiast dyskryminator nie maksymalizuje

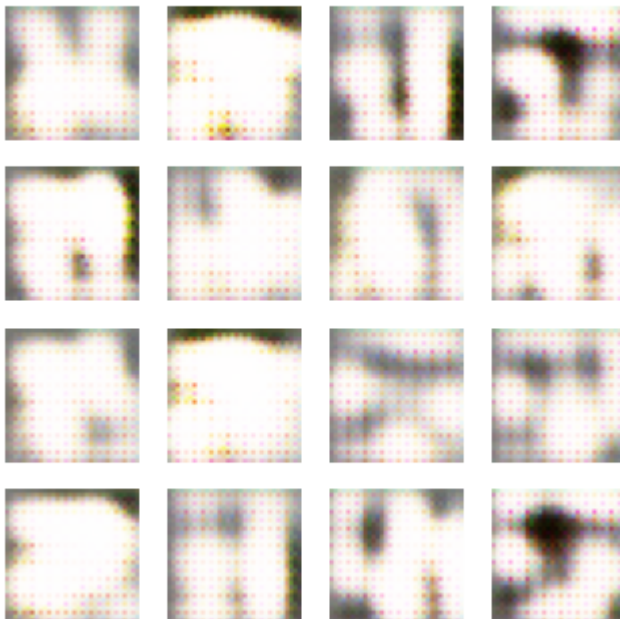
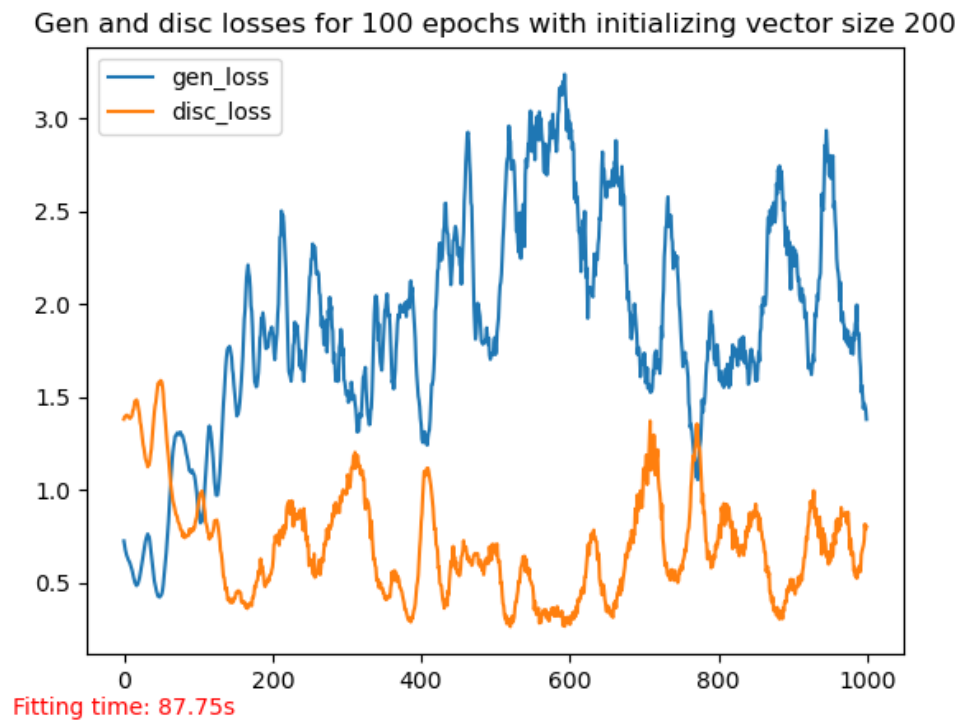
swojej funkcji straty (patrzac na wykres). Wartości funkcji straty obu mają duże przeskoki i są niestabilizowane.

Rozmiar wektora inicjującego generator = 100



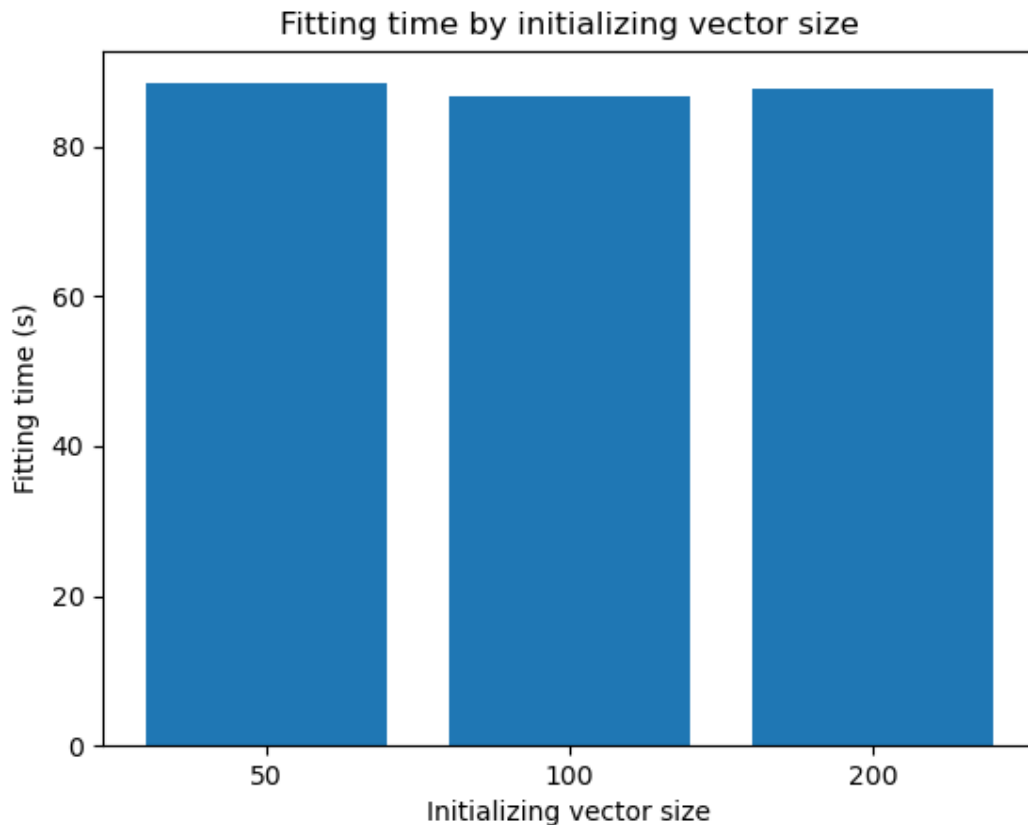
Generator zaczął stabilizować skoki pomiędzy wartościami swojej funkcji straty. Podobnie zachowują się wartości funkcji straty dyskriminatora. Zbyt mała liczba epok do maksymalizacji oraz minimalizacji wartości funkcji strat. Obrazy nie przypominają psa w żaden sposób.

Rozmiar wektora inicjującego generator = 200



Obrazy nie przypominają psa w żaden sposób. Funkcje strat generatora i dyskryminatora zaczynają się stabilizować w końcowych epokach.

Porównanie wielkości wektora inicjującego generator do czasu uczenia



Zmiana wielkości wektora inicjującego generator nie ma wpływu na szybkość uczenia.

Wpływ rozdzielczości obrazu na jakość i szybkość uczenia

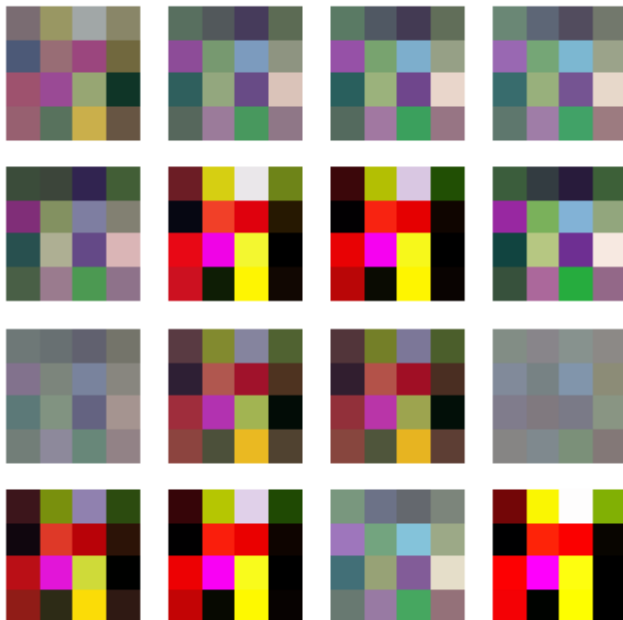
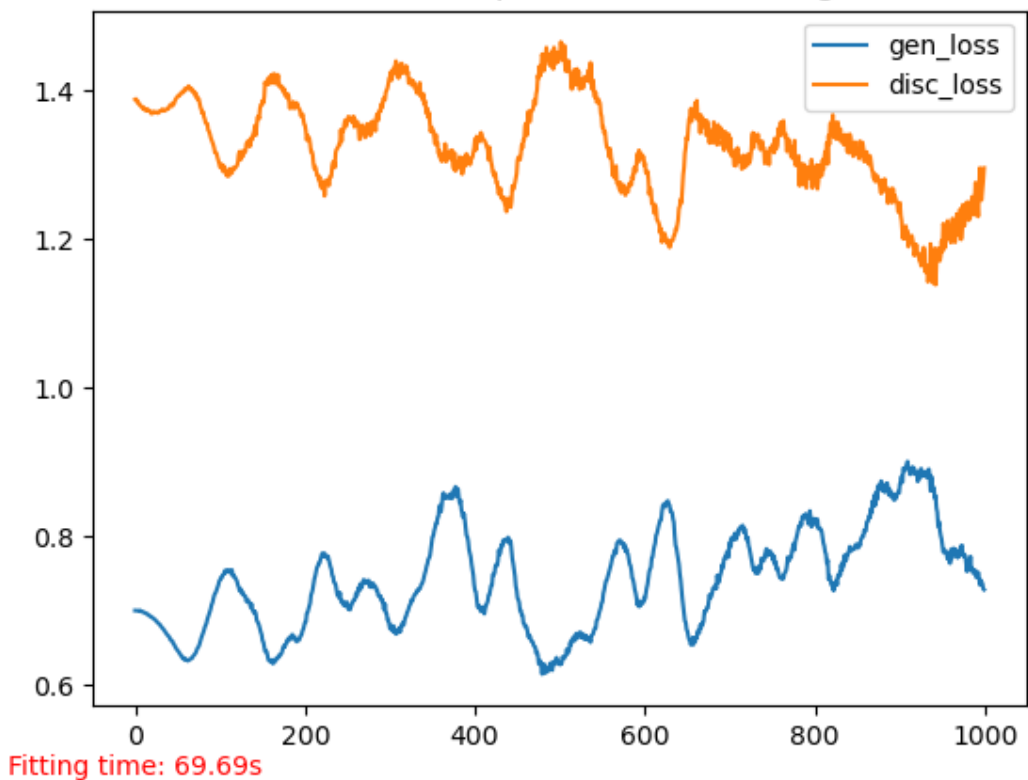
Eksperyment przeprowadzono na 3 rozdzielczościach obrazu:

- 4x4,
- 16x16,
- 32x32 (oryginalna).

Ilość epok uczenia ustawiono na 100, wielkość wektora inicjującego generator została ustawiona na 100. Wyniki zostały zebrane w formie wykresów obrazujących wartości funkcji straty dla generatora oraz dyskryminatora w zależności od wielkości wektora inicjującego generator zależne od czasu, a także zbiorcze porównanie w postaci histogramu wpływu wielkości wektora inicjującego generator na szybkość uczenia.

Rozdzielczość obrazu 4x4

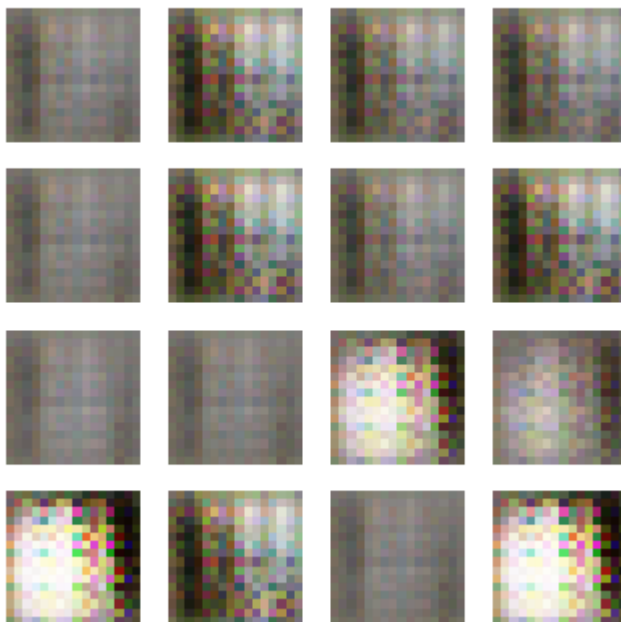
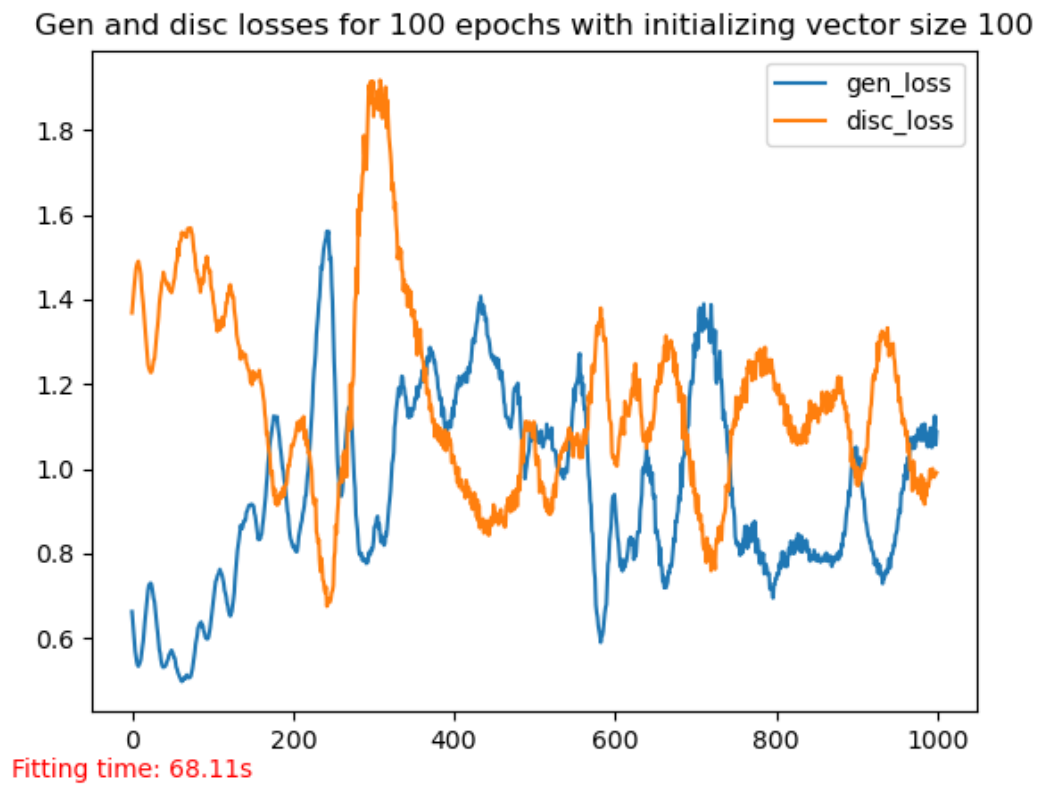
Gen and disc losses for 100 epochs with initializing vector size 100



Bardzo mała rozdzielczość generowanych obrazów wypada bardzo słabo. Wygenerowane obrazy w żadnym stopniu nie przypominają psa, ich rozdzielczość jest zbyt mała. Ponadto

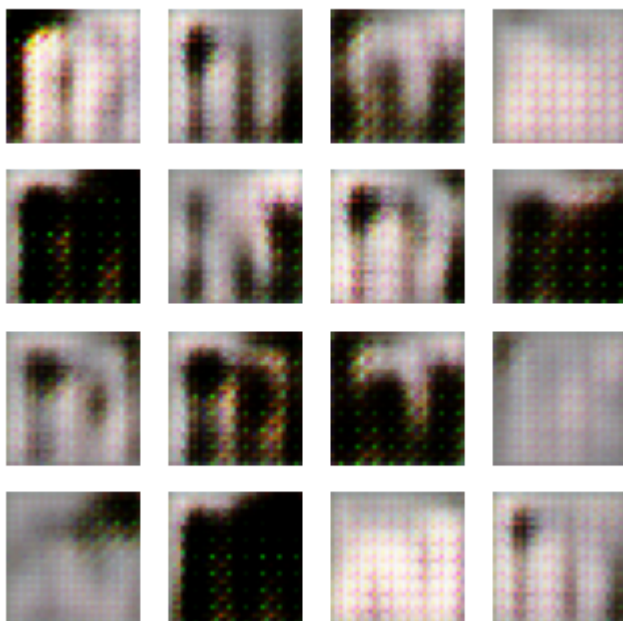
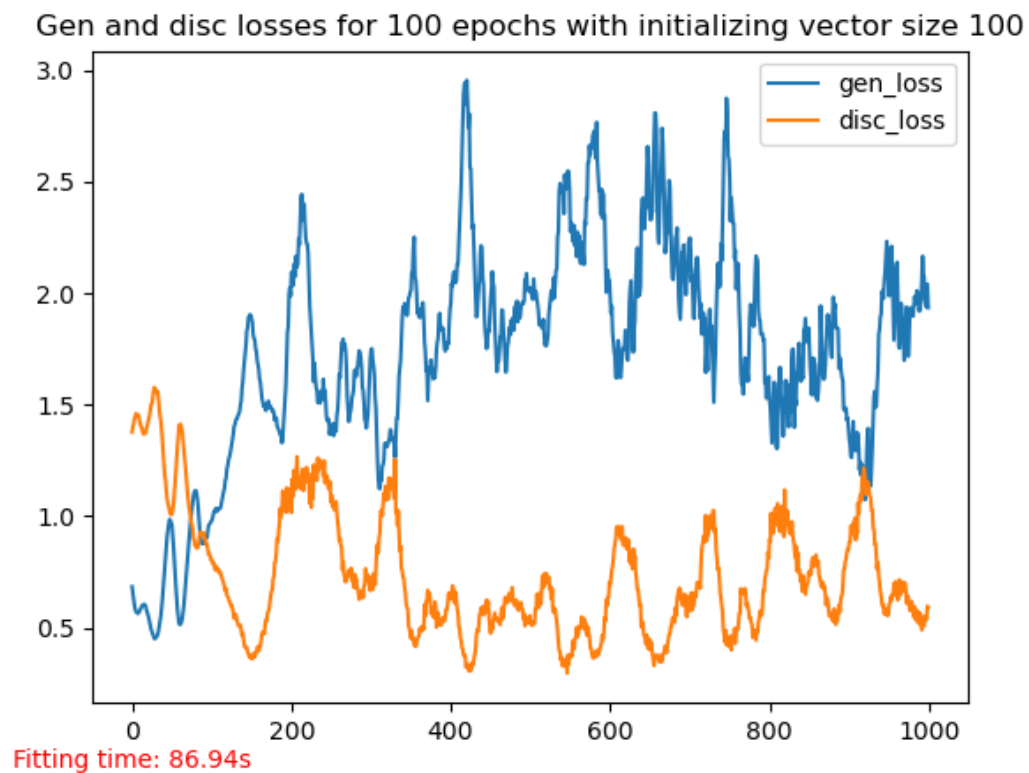
wartości funkcji straty generatora nie są w stanie przebić wartości funkcji straty dyskriminatora.

Rozdzielczość obrazu 16x16



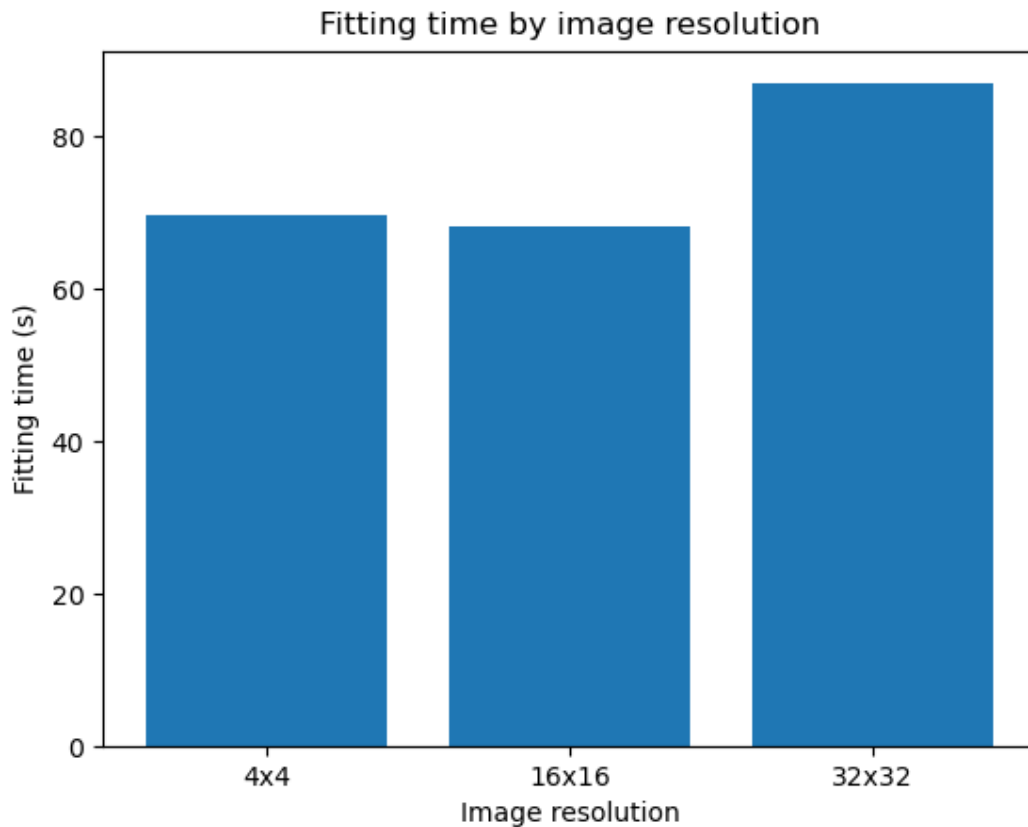
Wykres funkcji strat już bardziej przypomina działanie sieci GAN, chociaż jest to wciąż zbyt mało, aby wygenerować obrazy przypominające obiekt.

Rozdzielczość obrazu 32x32



Funkcje strat generatora i dyskryminatora przypominają już proces uczenia, z którego można wygenerować rozpoznawalne dla człowieka obrazy. Zarys wygenerowanych obrazów zaczyna kreować się w kierunku generowania obiektów.

Porównanie rozdzielczości obrazu do czasu uczenia



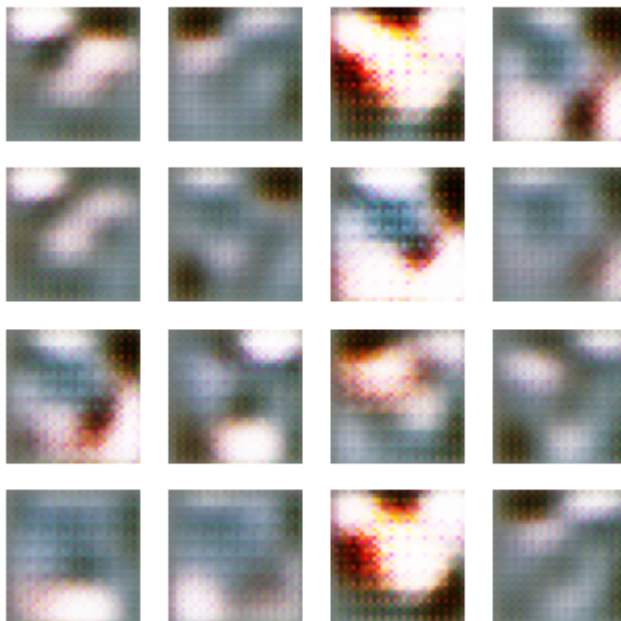
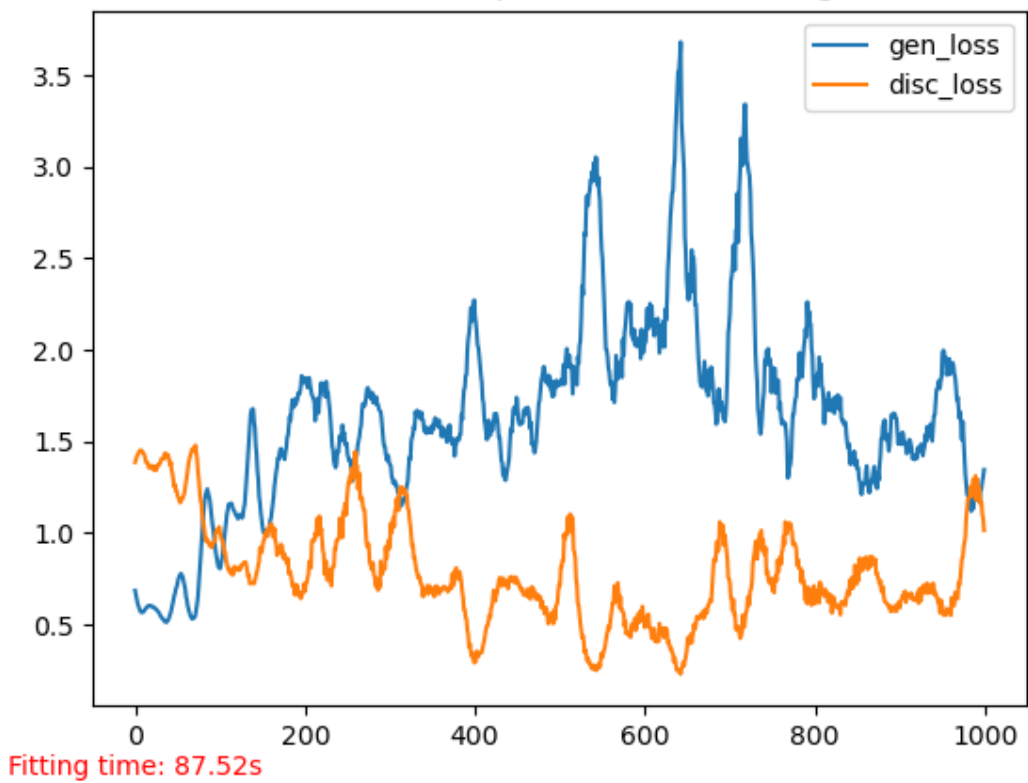
Zmiana rozdzielczości obrazu wpływa na czas uczenia, jednak nieznacznie. Przy tak niskiej rozdzielczości nie sposób ich porównać. Jednak zauważalny jest przeskok z rozdzielczości oryginalnej 32x32 do dwukrotnie niższej 16x16. Może to oznaczać, że przy większych rozdzielczościach zmniejszenie czasu uczenia byłoby znacznie większe.

Wpływ ilości epok uczenia na jakość i szybkość uczenia

Eksperyment przeprowadzono na wielkości wektora inicjującego generator równej 100, na wyniki zebrano i przedstawiono w formie wykresów obrazujących zmiany wartości funkcji strat generatora oraz dyskryminatora. Na koniec wykonano zbiorcze porównanie wpływu ilości epok uczenia na czas uczenia w postaci histogramu.

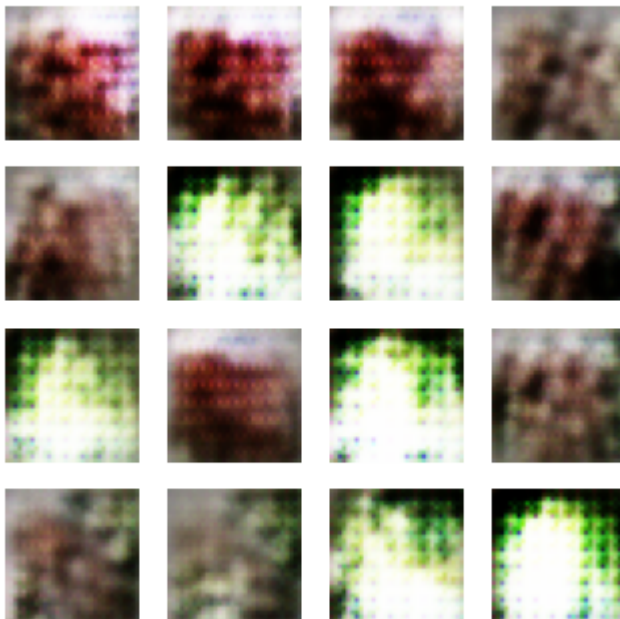
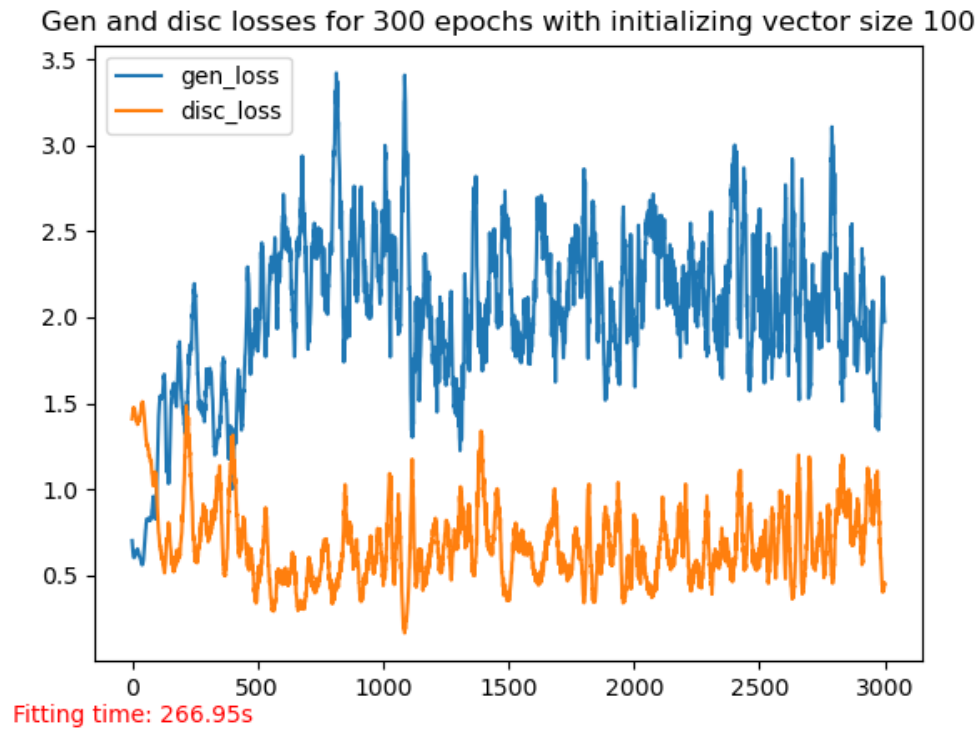
Ilość epok uczenia = 100

Gen and disc losses for 100 epochs with initializing vector size 100



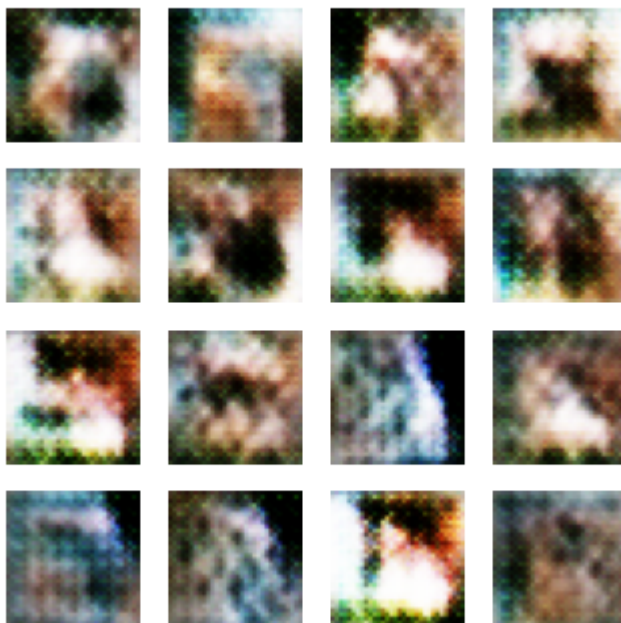
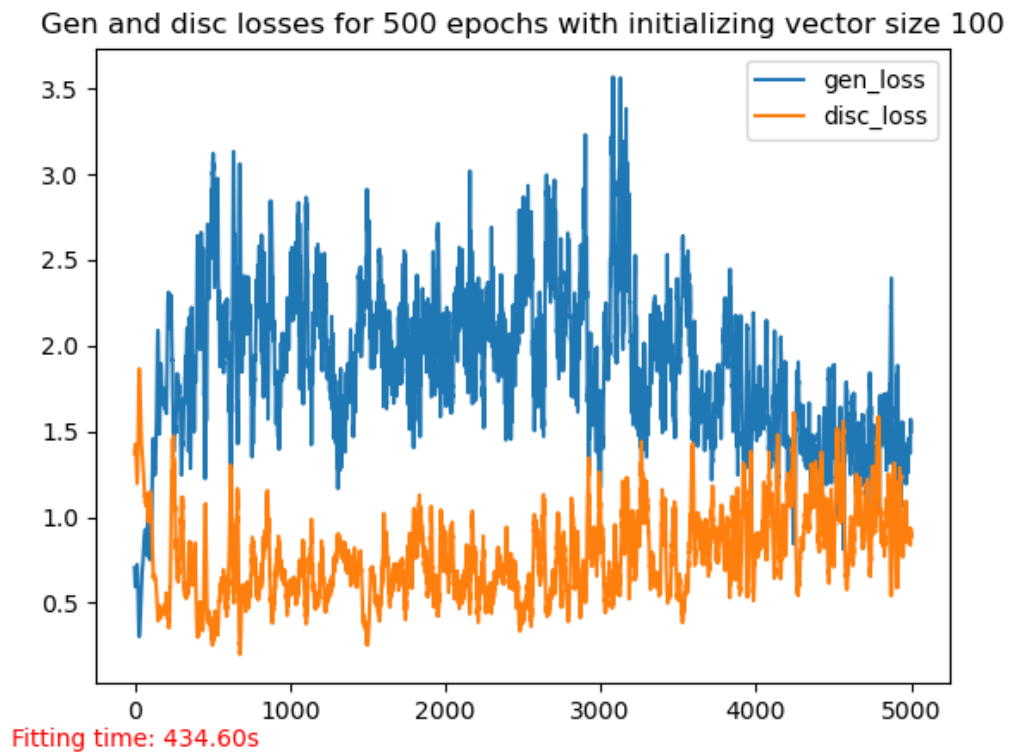
Widoczny jest zarys obiektów, jednak są one zbyt niewyraźne, aby mogły zostać rozpoznane przez człowieka. Funkcja straty generatora w końcowych epokach zaczyna się minimalizować, a funkcja straty dyskriminatora zaczyna się maksymalizować.

Ilość epok uczenia = 300



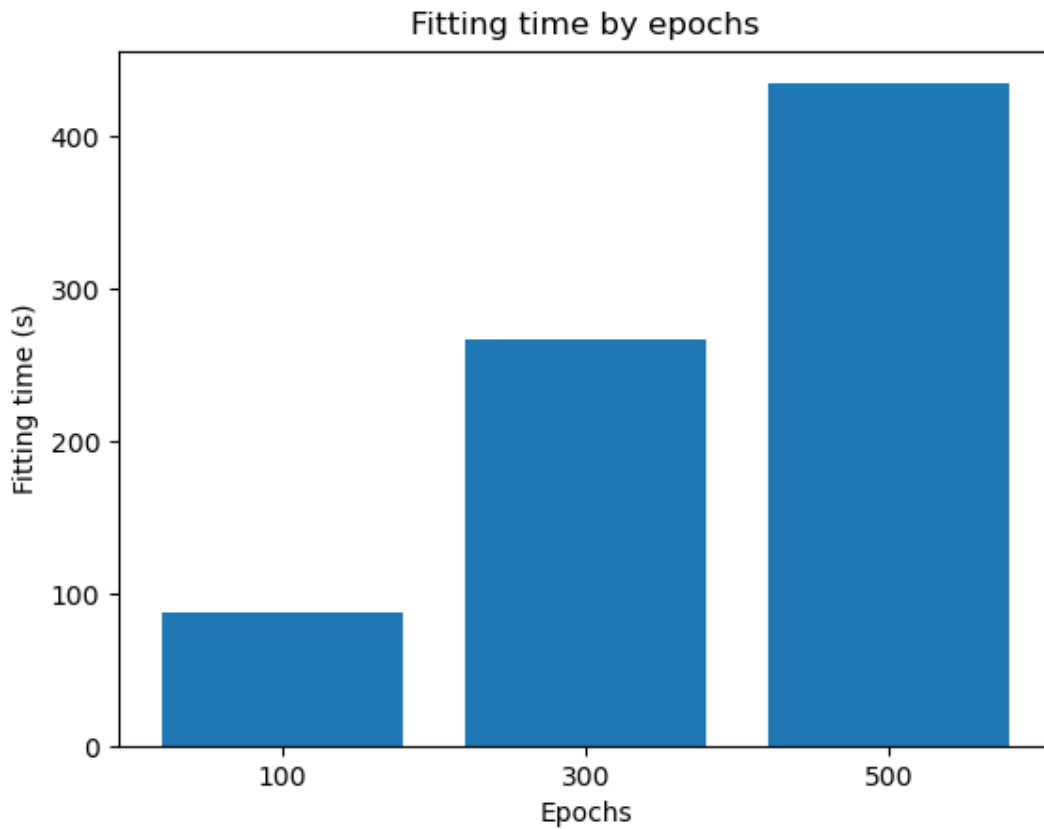
Z obrazów możliwe jest zauważenie zarysów psa. Wartości funkcji strat generatora i dyskryminatora ustabilizowały się.

Ilość epok uczenia = 500



Przy większej ilości epok uczenia widoczne są na obrazach psy, chociaż niezbyt wyraźnie. Funkcje strat generatora i dyskryminatora zbliżają się ku sobie, o czym świadczą generowane obrazy.

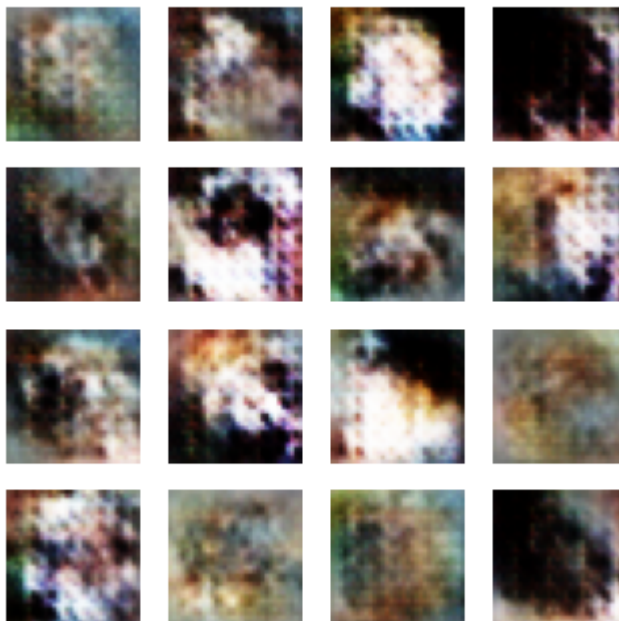
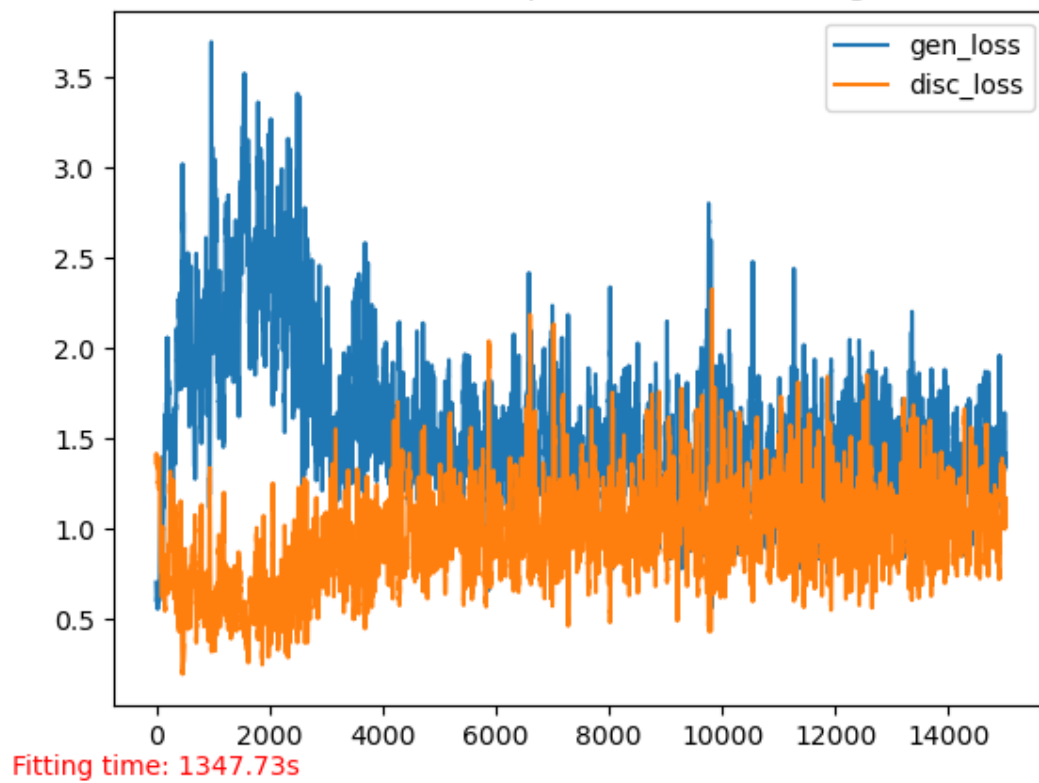
Porównanie ilości epok do czasu uczenia



Ilość epok uczenia ma znaczny wpływ na czas uczenia. Jednak wraz z ich zwiększaniem generowane obrazy są coraz lepsze, a wartości funkcji strat są coraz bliżej siebie.

Bonus! Ilość epok = 1500

Gen and disc losses for 1500 epochs with initializing vector size 100



Zwiększona liczba epok przyczynia się znacznie na korzyść lepszego generowania obrazów. Pomimo rozmazań wyraźnie widać na nich sylwetki psów. Wartości funkcji strat zaczynają być bardzo blisko siebie.

Jednak czas uczenia również znacznie się wydłużył. Również jest to wciąż zbyt mała liczba epok do generowania obrazów nierozróżnialnych dla człowieka od oryginałów.

Wnioski

Im większa wielkość wektora inicjującego generator tym szybciej wartości funkcji strat generatora oraz dyskryminatora zaczynają się stabilizować. Jednak zbyt duża wielkość zdaje się działać jak przeuczenie. Wolniej stabilizujące się wartości funkcji strat. Wielkość ta powinna zostać dobrana zależnie od rozmiaru obrazu.

Zbyt niskie rozdzielczości wypadają bardzo słabo, jeżeli chodzi o maksymalizację funkcji straty dyskryminatora i minimalizację funkcji straty generatora. Generowane obrazy także nie są rozpoznawalne dla człowieka. Niosą one zbyt mało informacji.

Zmniejsza to jednak czas potrzebny na przeprowadzenie uczenia modelu. Być może przy obrazach rozdzielczości 512x512 zmniejszenie ich rozdzielczości dwukrotnie do 256x256 dałoby bardzo podobne rezultaty jeżeli chodzi o generowane obrazy i wartości funkcji strat, znacznie redukując przy tym czas potrzebny na uczenie modelu.

Im większa ilość epok uczenia, tym generowane obrazy bardziej przypominają rzeczywisty obiekt. Jednak ich zwiększanie powoduje także proporcjonalny wzrost potrzebny na nauczenie modelu. Również potrzebna ich ilość jest ogromna, aby móc generować realistyczne obrazy nie do odróżnienia dla człowieka - podczas gdy 1500 epok wciąż generuje słabo rozpoznawalne obrazy.