

Politechnika Wroclawska

Wydział Informatyki i Telekomunikacji

KIERUNEK: Informatyka Stosowana (IST)
SPECJALNOŚĆ: Projektowanie Systemów Informatycznych (PSI)

PRACA DYPLOMOWA MAGISTERSKA

TYTUŁ PRACY:

Wykorzystanie modelu generatywnego do
rozszerzenia zbioru danych

AUTOR:

Adam Jankowiak

PROMOTOR:

dr inż. Wojciech Domski

Dedykuję tą pracę rodzicom i przyjaciołom, którzy wspierali mnie i pomagali pokonywać wyzwania. Szczególnie chciałbym podziękować mojej babci Uli.

STRESZCZENIE

Głównym celem pracy było sprawdzenie wpływu rozszerzenia zbioru danych, za pomocą generatywnej sieci przeciwstawnej, na jakość uczenia sieci neuronowej. Aby zrealizować to zadanie, zdecydowano się wykorzystać model StyleGan2 oraz własną architekturę sieci GAN. Dodatkowo wykorzystany został zbiór danych ORL [5], który został wykorzystany do wytrenowania generatywnych modeli. Liczba próbek dostępna w pierwotnej bazie została zwiększona poprzez rozszerzenie zbioru, dzięki czemu można było uzyskać lepsze rezultaty. W kolejnym kroku wytrenowano model StyleGan2 z etykietami, bez etykiet oraz własnej architektury GAN, po czym przystąpiono do rozszerzenia zbioru danych. Otrzymane próbki zostały poddane selekcji w celu odrzucenia błędnie wygenerowanych zdjęć. Po otrzymaniu odpowiedniej bazy danych, przystąpiono do badania wpływu nowo wygenerowanych fotografii na jakość sieci neuronowej określającej podobieństwo pomiędzy dwoma twarzami osób. Do celów testowych użyto reprezentacji złożonej z przeplatanych wierszy zdjęć. W efekcie udało się stworzyć dziesięć modeli określających podobieństwo, które zawierały różne zbiory treningowe w zależności od przyjętej metodologii. Uzyskane wyniki wykazały, że zastosowanie modelu generatywnego do powiększenia zbioru danych przyczyniło się do poprawy jakości modelu.

SUMMARY

The main goal of the work was to check the impact of expanding the data set using a generative adversarial network on the quality of training the neural network. To accomplish this task, it was decided to use the StyleGan2 model and its own GAN network architecture. Additionally, the ORL [5] dataset was used to train generative models. The number of samples available in the original database was increased by expanding the set, which allowed for better results. In the next step, the StyleGan2 model was trained with and without labels, and then the data set was expanded. The received samples were filtered selected to reject incorrectly generated photos. After obtaining a well represented and balanced database, the study began on the impact of newly generated photographs on the quality of the neural network determining the similarity between two photos of people's faces. For testing purposes, a representation consisting of intertwined rows of photos was used. As a result, it was possible to create ten similarity models, which included different training sets depending on the adopted methodology. The obtained results showed that the use of the generative model to enlarge the data set contributed to improving the quality of the model.

Słowa kluczowe: sztuczna inteligencja, generatywna sieć przeciwstawna, rozszerzenie zbioru, techniki generatywne

Keywords: artificial intelligence, generative adversarial network, dataset augmentation, generative techniques

Spis treści

1	Wstęp	3
1.1	Teza	4
1.2	Podział pracy	4
2	Generatywne sieci przeciwstawne	5
2.1	Generator	5
2.2	Dyskryminator	6
2.3	Trening sieci	6
2.4	Ocena sieci	6
3	Wykorzystane technologie	7
3.1	Metody rozszerzenia zbioru	7
3.2	Platformy programistyczne wykorzystujące generatywne sieci przeciwstawne	7
4	Generowanie fikcyjnych zdjęć	9
4.1	Wykorzystany zbiór zdjęć	9
4.2	Dostosowanie formatu zdjęć	10
4.3	Rozszerzenie zbioru	10
4.4	Trenowanie modeli z wykorzystaniem własnej architektury GAN	11
4.5	Trenowanie modeli z wykorzystaniem sieci StyleGan2	14
5	Porównanie rezultatów	19
5.1	Zbiory danych	20
5.2	Rzeczywiste próbki	20
5.3	Łączenie zbiorów	22
5.3.1	StyleGan2 z etykietami	22
5.3.2	StyleGan2 bez etykiet	24
5.3.3	Własny model GAN	25
5.4	Wyuczenie modeli na fikcyjnych zdjęciach	27
5.4.1	StyleGan2 z etykietami	27
5.4.2	StyleGan2 bez etykiet	28
5.4.3	Własny model GAN	30
5.5	Dotrenowanie rzeczywistego modelu	31
5.5.1	StyleGan2 z etykietami	31
5.5.2	StyleGan2 bez etykiet	33
5.5.3	Własny model GAN	34
5.6	Porównanie uzyskanych rezultatów	36
6	Podsumowanie	39

Załącznik A

40

Bibliografia

41

Rozdział 1

Wstęp

Sztuczna inteligencja to ogólny termin, który opisuje zestaw metod i algorytmów umożliwiający maszyną na samodzielnie uczenie się, dostosowywanie do zmieniających warunków oraz naśladowanie ludzkiej inteligencji. Początki badań nad sztuczną inteligencją sięgają lat 50 XX wieku. Bardzo istotne dla rozwoju tej dziedziny było letnie seminarium naukowe Dartmouth workshop, podczas którego naukowcy dyskutowali na temat sztucznej inteligencji [3]. To wydarzenie jest obecnie uważane za kamień milowy, który zapoczątkował pierwsze etapy prac nad tą nową technologią.

Algorytmy uczenia maszynowego są powszechnie wykorzystywane do przetwarzania dużych ilości danych, wykrywania wzorców oraz podejmowania decyzji na podstawie zgromadzonej wiedzy. Systemy sztucznej inteligencji cechują się wysoką dokładnością oraz szybkością działania. Jedyną z głównych wad SI jest potrzeba wykorzystania odpowiednio dużego zbioru danych do skutecznego treningu modeli. W przypadku algorytmów do przetwarzania obrazów lub analizy języka naturalnego należy dostarczyć duże zbiory danych do wyuczenia sieci. Dodatkowo brak reprezentatywności próbek może doprowadzić do problemu związanego z jakością.

Generatywne sieci przeciwstawne (z ang. *generative adversarial networks, GAN*) znacząco wyróżniają się na tle innych metod sztucznej inteligencji. Nie tylko wykorzystują nowatorskie podejście do generowania danych, ale także dają możliwość tworzenia realistycznych fotografii. Generatywne sieci przeciwstawne po raz pierwszy zostały zaprezentowane w pracy autorstwa Iana Goodfellowa z 2014 roku [1]. Cechują się one możliwością replikowania zbioru danych poprzez generowanie próbek. Model GAN składa się z dwóch powiązanych ze sobą sieci, generatora i dyskryminatora. Pierwsza z nich, odpowiada za generowanie nowych danych, natomiast druga dokonuje oceny wytworzonych próbek. W trakcie nauczania modelu, generator próbuje wytworzyć na tyle dobre próbki, żeby dyskryminator nie był w stanie odróżnić ich od prawdziwych danych. W trakcie treningu oba modele na wzajem rywalizują, dzięki czemu nierzeczywiście rośnie.

Wraz z dynamicznym rozwojem SI, Unia Europejska zdecydowała się na wprowadzenie nowych przepisów mających na celu uregulować prawnie wykorzystanie algorytmach uczenia maszynowego w celu ochrony praw człowieka [7]. Europejska ustawa o sztucznej inteligencji (z ang. *EU Artificial Intelligence Act*) dzieli sztuczną inteligencję ze względu na zagrożenie jakie może wywierać na społeczeństwo. Pierwszą grupą są systemy niedopuszczalnego ryzyka, do których można zaliczyć klasyfikację punktową obywateli oraz kategoryzację ludzi. Przykładem jest system stosowany w Chinach do klasyfikacji i oceniania społeczeństwa na podstawie dokonywanych płatności, zachowań w sieci oraz lojalności do rządu. Drugą grupą są systemy wysokiego ryzyka, które mają negatywny wpływ na bezpieczeństwo oraz prawa podstawowe obywateli. Trzecią kategorią są sys-

temy niskiego ryzyka, które służą przykładowo do automatycznej klasyfikacji dokumentów. Dodatkowo, wprowadzenie nowych przepisów będzie obligowało firmy do rejestracji wszystkich tego typu systemów w bazie danych Unii Europejskiej.

Celem pracy było sprawdzenie wpływu sztucznie wygenerowanych próbek na jakość modelu sieci neuronowej. W celu zrealizowania tego zadania, wykorzystana została generatywna sieć przeciwstawna (z ang. *generative adversarial networks, GAN*), która jest w stanie wygenerować nowe, fikcyjne zdjęcia. Aby móc porównać wpływ sztucznie wytworzonych fotografii wykorzystany został model sztucznej inteligencji, który określa podobieństwo pomiędzy dwoma twarzami. Zbiór danych, wykorzystany do nauki pierwszej sieci, posiadał jedynie rzeczywiste zdjęcia, natomiast zbiór danych dla drugiego modelu zawierał także nowo wygenerowane obrazy.

1.1 Teza

Możliwe jest zwiększenie dokładności modelu sztucznej inteligencji poprzez wykorzystanie technik generatywnych do stworzenia zbioru uczącego.

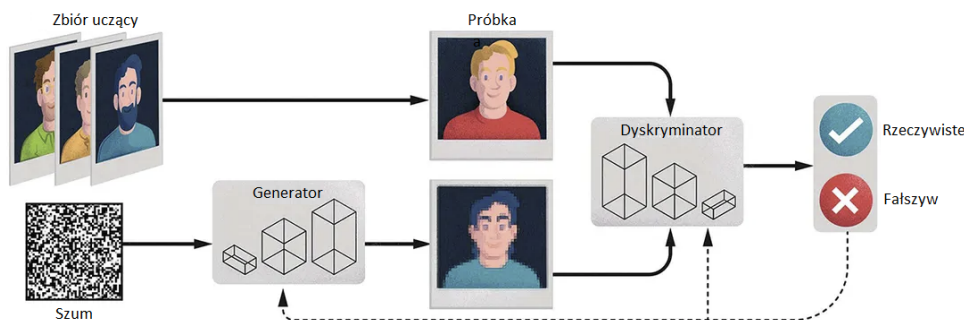
1.2 Podział pracy

W rozdziale drugim zamieszczono opis działania generatywnych sieci przeciwstawnych. Kolejny rozdział zawiera opis możliwych technologii, które można wykorzystać do rozszerzenia zbioru danych. Rozdział czwarty zawiera opis przetwarzania danych oraz sposób trenowania modeli GAN. W rozdziale piątym zamieszczono badania nad jakością uzyskanych sieci w zależności od zastosowanej techniki rozszerzenia zbioru danych. Zbadano wpływ trzech różnych wariantów sieci, StyleGan2 z etykietami, StyleGan2 bez etykiet oraz własną sieć GAN. Ostatni rozdział zawiera podsumowanie wykonanych wcześniej badań wraz z wnioskami.

Rozdział 2

Generatywne sieci przeciwstawne

Generatywne sieci przeciwstawne należą do klasy modeli sztucznej inteligencji, które rewolucjonizowały podejście do generowania danych syntetycznych. Algorytm został zaprezentowany w 2014 roku przez Iana Goodfellowa oraz jego współpracowników. GAN można być wykorzystany do generowania fikcyjnych obrazów, dźwięków, tekstów czy nawet sekwencji wideo, za pomocą dostarczonych wzorców. Model składa się z dwóch niezależnych sieci neuronowych, które wzajemnie się trenują. Pierwsza z nich, zwana generatorem, odpowiada za tworzenie fikcyjnych próbek. Druga, nazywana dyskryminatorem, określa czy stworzone próbki są zbliżone do danych uczących. Na rysunku 2.1, przedstawiona została graficzna reprezentacja architektury sieci GAN.



Rysunek 2.1: Graficzna reprezentacja architektury sieci GAN [6]

2.1 Generator

Generator jest siecią neuronową zdolną do tworzenia obrazów, dźwięków lub innych danych za pomocą dostarczonych wzorców. Proces uczenia polega na tworzeniu coraz to dokładniejszych próbek, które są następnie oceniane przez dyskryminator. Na początku treningu generator otrzymuje losowy wektor, który następnie, za pomocą rozkładu Gaussa, zostaje przekształcony w obraz. Podczas treningu generator stara się oszukać dyskryminator, tworząc próbki, które są coraz bardziej zbliżone do prawdziwych danych. Poprawa jakości generowanych obrazów odbywa się poprzez minimalizację funkcji straty, która mierzy, jak dobrze generator oszukuje dyskryminator. Głównym celem generatora jest wytworzenie sztucznych próbek, które są na tyle realistyczne, że dyskryminator nie był w stanie określić czy dana próbka jest fałszywa.

2.2 Dyskryminator

Dyskryminator to model sztucznej inteligencji, którego rolą jest określenie, czy wygenerowana próbka jest prawdziwa lub fałszywa. Danymi wejściowymi dla algorytmu są zarówno prawdziwe próbki z zbioru danych, jak i próbki wygenerowane przez generator. Jako wyjście, dyskryminator zwraca wartość od 0 do 1, która określa prawdopodobieństwo, że ocenione dane są rzeczywiste. Wartość 1 określa, że dane wejściowe są prawdziwe, natomiast 0 wskazuje, że dana próbka jest fałszywa. Dyskryminator jest trenowany tak, aby maksymalizować swoje umiejętności rozróżniania między prawdziwymi a fałszywymi danymi. Proces ten jest kluczowy, ponieważ determinuje on, jak dobrze generator potrafi tworzyć realistyczne próbki.

2.3 Trening sieci

Trening sieci GAN polega na iteracyjnym procesie współzawodnictwa pomiędzy generatorem a dyskryminatorem. Na każdym kroku treningu, generator tworzy nowe próbki, które są następnie oceniane przez dyskryminator. Dyskryminator stara się poprawić swoje umiejętności w wykrywaniu fałszywych danych, podczas gdy generator dąży do tworzenia coraz bardziej przekonujących próbek. Dyskryminator jest trenowany na dwóch zestawach danych, prawdziwych i fałszywych, wygenerowanych przez generator. Celem jest maksymalizacja funkcji straty dla próbek prawdziwych i minimalizacja dla próbek fałszywych. Generator jest trenowany na podstawie informacji zwrotnej od dyskryminatora. Generator stara się minimalizować funkcję straty dyskryminatora, która informuje, jak dobrze dyskryminator rozróżnia między prawdziwymi a fałszywymi próbkami. Proces ten kontynuuje się do momentu, gdy generator jest w stanie tworzyć próbki tak realistyczne, że dyskryminator nie może ich odróżnić od prawdziwych danych.

2.4 Ocena sieci

Głównym celem generatywnej sieci przeciwstawnej jest stworzenie na tyle precyzyjnego modelu generatora, który będzie w stanie oszukać dyskryminator na tyle skutecznie, że generowane próbki będą nieodróżnialne od prawdziwych danych dla minimum połowy przypadków. W trakcie nauki model dyskryminatora i generatora uzyskuje coraz to większą dokładność dzięki czemu generowane próbki stają się coraz to lepsze. Ocena jakości generowanych danych może odbywać się na kilka sposobów. Pierwszym z nich jest przeglądanie wygenerowanych obrazów i porównywanie nich z rzeczywistymi danymi. Jest to dosyć skuteczna metoda pod względem precyzji, jednak pochłania bardzo dużą ilość czasu i jest mało skuteczna dla dużych zbiorów danych. Drugą metodą jest wykorzystanie różnych miar jakości, takich jak odległość początkowa Frechta (z ang. *Frechet Inception Distance, FID*), czy też wskaźnik iniepcji (z ang. *Inception Score, IS*). Wymienione miary są w stanie kwalifikować różnice pomiędzy prawdziwymi a wygenerowanymi próbkami. Wykorzystanie metody opartej na miarach jakości dobrze sprawdza się przy ocenianiu jakości dużych zbiorów danych.

Rozdział 3

Wykorzystane technologie

Wśród dostępnych sieci GAN można wyróżnić wiele gotowych rozwiązań, które umożliwiają łatwe i szybkie stworzenie modelu generującego fikcyjne zdjęcia. Dzięki gotowym rozwiązaniom do tworzenia sieci GAN proces wdrażania i uczenia się jest znacznie uproszczony. Bez względu na wybraną platformę ważne jest, aby wybrać odpowiednie narzędzie zgodnie ze szczegółami projektu i wymaganiami użytkownika. Dzięki różnorodności dostępnych metod można znaleźć najlepsze rozwiązanie zarówno dla początkujących, jak i zaawansowanych użytkowników.

3.1 Metody rozszerzenia zbioru

Po za wykorzystaniem sieci GAN istnieją także inne możliwości rozszerzenia zbioru danych. Wśród nich można wyróżnić tradycyjne techniki, polegające na dokonywaniu różnego rodzaju transformacji danych wejściowych. Do tych technik można zaliczyć dodanie delikatnej rotacji do zdjęć, zmiana jasności, przybliżenie lub oddalenie zdjęcia. W tym przypadku rozszerzenie zbioru umożliwia zwiększenie różnorodności zbioru treningowego bez konieczności tworzenia nowych danych od podstaw. Kolejnym sposobem rozszerzenia zbioru danych jest transfer początkowych wartości wag (z ang. *Transfer Learning*). Metoda ta polega na przeniesieniu wiedzy zdobytej przez model uczony na jednym zadaniu do innego [8]. Dzięki transfer learning możliwe jest wykorzystanie dużych, już istniejących modeli do rozwiązywania innych problemów przy ograniczonych zbiorach danych. Kolejnym sposobem na rozszerzenie zbioru danych jest technika generująca syntetyczne dane dla klasy mniejszości (z ang. *Synthetic Minority Over-sampling Technique, SMOTE*). Technika ta polega na wytworzeniu syntetycznych danych dla klasy mniejszościowej poprzez interpolację między istniejącymi próbkami. Wykorzystuje się ją głównie w nierównoważonych zbiorach danych.

3.2 Platformy programistyczne wykorzystujące generatywne sieci przeciwstawne

Istnieje wiele gotowych rozwiązań i bibliotek, które umożliwiają łatwe trenowanie sieci GAN. Wśród nich można wyróżnić TensorFlow, który jest jedną z najpopularniejszych bibliotek do uczenia maszynowego. Keras, będący wysokopoziomowym interfejsem programowania aplikacji (z ang. *application programming pnterface, API*) dla TensorFlow, umożliwia szybkie i łatwe trenowanie sieci GAN. Wśród zalet można wyróżnić wsparcie

dla uczenia sieci za pomocą GPU, dużą społeczność użytkowników oraz rozległą dokumentację. Wadą tej biblioteki jest wysoka złożoność w porównaniu do innych dostępnych rozwiązań. Drugą dostępną platformą jest PyTorch, opracowany przez Facebook's AI Research lab. Zyskuje on coraz to większą popularność dzięki swojej elastyczności i intuicyjnemu interfejsowi. Jest on szeroko stosowany do badań i rozwoju sztucznej inteligencji. Jako zalety można wymienić dynamiczne zarządzanie grafem obliczeniowym oraz dużą społeczność naukową. Największą wadą tego rozwiązania jest znacznie mniejsza liczba gotowych narzędzi w porównaniu do TensorFlow. Kolejną biblioteką jest GAN Lab. Jest to interaktywne narzędzie do wizualizacji działania GAN, opracowane przez Google. Umożliwia użytkownikom interaktywne eksperymentowanie z różnymi architekturami i ustawieniami GAN.

Rozdział 4

Generowanie fikcyjnych zdjęć

W celu stworzenia modelu umożliwiającego wygenerowanie fikcyjnych zdjęć twarzy wykorzystany został model generatywny StyleGAN2. Umożliwia on wygenerowanie nowych zdjęć na podstawie dostarczonej wcześniej bazy danych. Model został wybrany ze względu na wysoką dokładność tworzonych zdjęć oraz krótki czas trenowania modelu. StyleGAN2 został stworzony przez naukowców z firmy NVIDIA i jest jednym z bardziej zaawansowanych modeli generatywnych wykorzystywanych do tworzenia realistycznych zdjęć. Daje on także możliwość wydobycia odpowiednich cech na wygenerowanych obrazach poprzez dodanie odpowiednich etykiet (z ang. *labels*) do zbioru uczącego.

4.1 Wykorzystany zbiór zdjęć

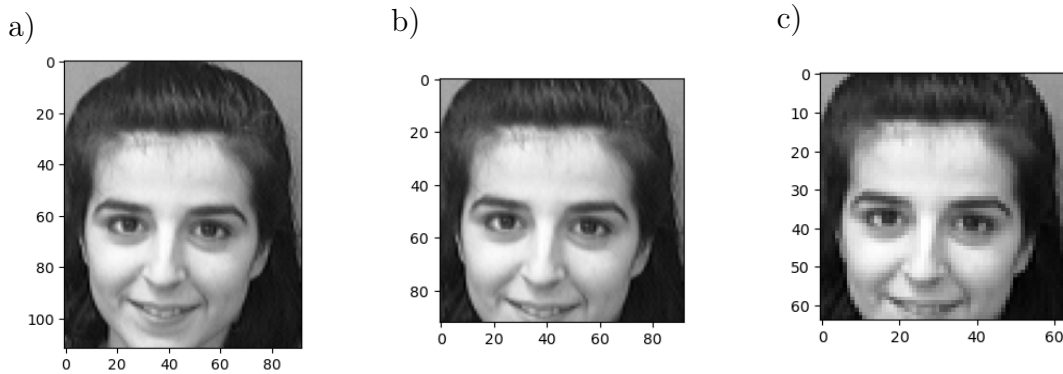
Spośród dostępnych zbiorów, wybrana została baza zdjęć twarzy osób ORL [5]. Została ona udostępniona przez laboratorium badawcze Olivetti z Wielkiej Brytanii (z ang. *Olivetti Research Laboratory, ORL*), od którego pochodzi jego nazwa. Katalog zawiera fotografie twarzy wykonane 40 różnym osobom, z których każda miała wykonane 10 różnych zdjęć. Na załączonym rysunku 4.1 można zobaczyć, że zdjęcia zostały zrobione w różnych chwilach, przy zróżnicowanym oświetleniu. Przedstawiają one różne wyrazy twarzy, a niektórzy ludzie mają nałożone lub zdjęte okulary. Dodatkowo wszystkie zdjęcia zostały wykonane na jednolitym czarnym tle, postacie są wyprostowane, ustawione frontalnie z dozwolonym minimalnym obrotem. Obrazy są zapisane w formacie ośmiobitowym w skali szarości i mają rozmiar 92 na 112 piksele.



Rysunek 4.1: Przykładowe fotografie z bazy ORL

4.2 Dostosowanie formatu zdjęć

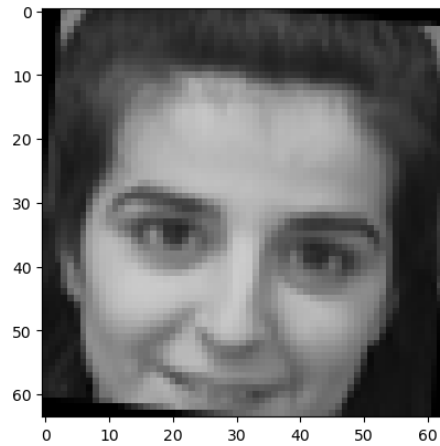
W celu ujednoczenia bazy wykonano konwersję formatu zdjęć. Przykład oryginalnego zdjęcia przed modyfikacjami znajduje się po lewej stronie rysunku 4.2. Fotografie zostały edytowane poprzez usunięcie górnych i dolnych wierszy zdjęcia. W wyniku tej operacji uzyskano próbki w rozmiarze 92 na 92 piksele, co zostało przedstawione na środku rysunku 4.2. Wykorzystanie obrazów o proporcjach kwadratu było podyktowane możliwością uchwycenia nieregularności ludzkiej twarzy. Ostatni etap formatowania zdjęć polegał na przeskalowaniu zbioru do rozmiaru 64 na 64 piksele, co zostało przedstawione po prawej stronie rysunku 4.2. Zdjęcia o takich wymiarach zawierają wystarczającą ilość informacji potrzebnych do wytrenowania sieci GAN.



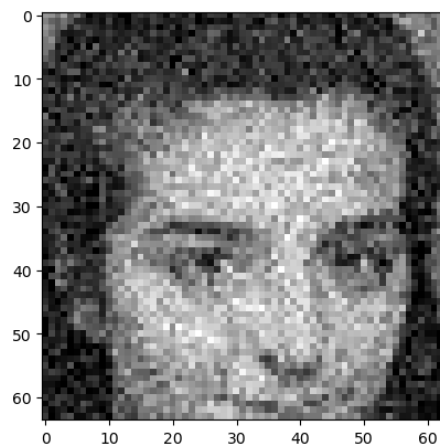
Rysunek 4.2: Fotografie o rozmiarze 92x112, 92x92 oraz 64x64 piksele

4.3 Rozszerzenie zbioru

Baza ORL zawiera 400 zdjęć twarzy, co uniemożliwia wyuczenie sieci i uzyskanie wysokiej precyzji. Aby zwiększyć liczbę przykładów, zastosowano rozszerzenie zbioru poprzez obracanie i zaszumianie zdjęć. Dodatkowo fotografie zostały obrócone o kąty 2° , 4° i 6° w obu kierunkach, co ilustruje rysunek 4.3, przez co liczba obrazów przedstawiających jedną osobę wzrosła z 10 do 70. Nowo powstałe przykłady są traktowane przez sieć jako odmienne od oryginalnych. Dzieje się tak, ponieważ algorytm wczytuje zdjęcia wierszami, a wprowadzenie rotacji zmienia układ pikseli w macierzy, co pozwala uzyskać większą liczbę danych.

Rysunek 4.3: Zdjęcie po dodaniu rotacji o kąt 6°

Drugą metodą zastosowaną w celu rozszerzenia zbioru było dodanie szumu. W tym celu do wszystkich zdjęć dodano szum Gaussa, co wpłynęło na wartości poszczególnych pikseli, rysunek 4.4. Dzięki tej operacji liczba dostępnych próbek wzrosła z 70 do 140. W tym przypadku sieć neuronowa traktuje nowe przykłady także jako różne od oryginałów.



Rysunek 4.4: Zdjęcie po dodaniu szumu Gaussa

Wykonanie rotacji oraz dodaniu zaszumienia do zdjęć, umożliwiło w łatwy i szybki sposób znacząco zwiększyć liczbę dostępnych fotografii. Po wykonaniu obu operacji uzyskano łącznie 5600 próbek.

4.4 Trenowanie modeli z wykorzystaniem własnej architektury GAN

W celu zbadania wpływu rozszerzenia zbioru na jakość modelu, zdecydowano się na stworzenie własnej sieci GAN. Pierwszy krok polegał na dobraniu odpowiedniej architektury dla generatora i dyskryminatora. Tabela 4.1 przedstawia parametry opisujące architekturę dla modelu generatora. Tabela 4.2 zawiera liczbę parametrów, które podlegają zmianie w procesie nauki dla każdej z warstw. W tabeli 4.3 przedstawione zostało podsumowanie liczby wszystkich parametrów.

Rodzaj warstwy	Liczba filtów	Rozmiar jądra	Rozmiar łączenia	Aktywacja
Dense	-	-	-	-
BatchNormalization	-	-	-	-
LeakyReLU	-	-	-	-
Reshape	-	-	16 x 16 x 256	-
Conv2DTranspose	128	5 x 5	1 x 1	-
BatchNormalization	-	-	-	-
LeakyReLU	-	-	-	-
Conv2DTranspose	128	5 x 5	2 x 2	-
BatchNormalization	-	-	-	-
LeakyReLU	-	-	-	-
Conv2DTranspose	64	5 x 5	2 x 2	-
BatchNormalization	-	-	-	-
LeakyReLU	-	-	-	-
Conv2DTranspose	1	5 x 5	1 x 1	Tanh

Tabela 4.1: Architektura modelu generatora

Rodzaj warstwy (typ)	Rozmiar wyjściowy	Parametry
Dense	(Brak, 65536)	6,553,600
BatchNormalization	(Brak, 65536)	262,144
LeakyReLU	(Brak, 65536)	0
Reshape	(Brak, 16, 16, 256)	0
Conv2DTranspose	(Brak, 16, 16, 128)	819,200
BatchNormalization	(Brak, 16, 16, 128)	512
LeakyReLU	(Brak, 16, 16, 128)	0
Conv2DTranspose	(Brak, 32, 32, 128)	409,600
BatchNormalization	(Brak, 32, 32, 128)	512
LeakyReLU	(Brak, 32, 32, 128)	0
Conv2DTranspose	(Brak, 64, 64, 64)	204,800
BatchNormalization	(Brak, 64, 64, 64)	256
LeakyReLU	(Brak, 64, 64, 64)	0
Conv2DTranspose	(Brak, 64, 64, 1)	1,600

Tabela 4.2: Parametry wykorzystane w modelu generatora

Całkowita liczba parametrów: 8 252 224
Trenowane parametry: 8 120 512
Nie trenowane parametry: 131 712

Tabela 4.3: Podsumowanie liczby parametrów generatora

Następnie stworzony został model dyskriminatora, którego zadaniem była ocena wygenerowanych zdjęć przez generator. W tabeli 4.4 przedstawiono architekturę dyskriminatora. Tabela 4.5 zawiera liczbę parametrów, które podlegają zmianie w procesie nauki dla każdej z warstw. W tabeli 4.6 przedstawione zostało podsumowanie liczby wszystkich parametrów.

Rodzaj warstwy	Liczba filtów	Rozmiar jądra	Rozmiar łączenia	Aktywacja
Conv2D	64	5 x 5	2 x 2	-
LeakyReLU	-	-	-	LeakyReLU
Dropout	-	-	-	-
Conv2D	128	5 x 5	2 x 2	-
LeakyReLU	-	-	-	LeakyReLU
Dropout	-	-	-	-
Conv2D	256	5 x 5	2 x 2	-
LeakyReLU	-	-	-	LeakyReLU
Dropout	-	-	-	-
Flatten	-	-	-	-
Dense	1	-	-	-

Tabela 4.4: Architektura modelu dyskryminatora

Rodzaj warstwy (typ)	Rozmiar wyjściowy	Parametry
Conv2D	(Brak, 32, 32, 64)	1664
LeakyReLU	(Brak, 32, 32, 64)	0
Dropout	(Brak, 32, 32, 64)	0
Conv2D	(Brak, 16, 16, 128)	204928
LeakyReLU	(Brak, 16, 16, 128)	0
Dropout	(Brak, 16, 16, 128)	0
Conv2D	(Brak, 8, 8, 256)	819456
LeakyReLU	(Brak, 8, 8, 256)	0
Dropout	(Brak, 8, 8, 256)	0
Flatten	(Brak, 16384)	0
Dense	(Brak, 1)	16385

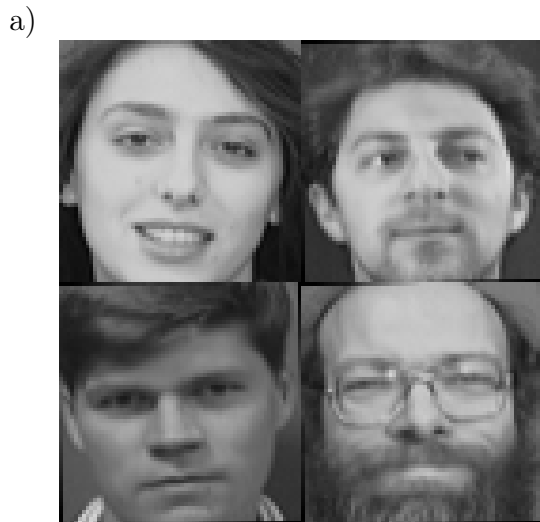
Tabela 4.5: Parametry wykorzystane w modelu dyskryminatora

Całkowita liczba parametrów: 1 042 433
Trenowane parametry: 1 042 433
Nie trenowane parametry: 0

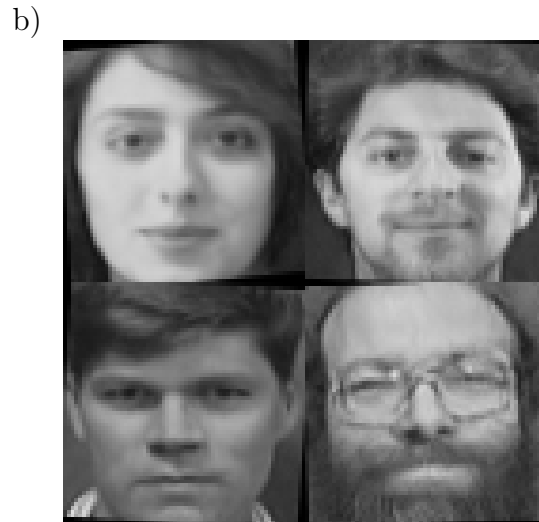
Tabela 4.6: Podsumowanie liczby parametrów dyskryminatora

Pierwszym etapem nauki sieci GAN była inicjalizacja wag obu sieci. Na podstawie dostarczonego wektora zawierającego losowe wartości, generator tworzył nowe, fikcyjne zdjęcia, które następnie były oceniane przez dyskryminator. Nauka dyskryminatora polegała na dostarczaniu mu zarówno rzeczywistych, jak i syntetycznych danych, aby nauczył się odróżniać fikcyjne zdjęcia od prawdziwych. Kolejnym krokiem było trenowanie generatora, który z każdą iteracją próbuje tworzyć coraz lepsze próbki, zbliżone do oryginalnych. Następnie wygenerowane próbki były dostarczane na wejście dyskryminatora, po czym obliczana była funkcja kosztu dla obu sieci. Celem generatora jest zmniejszenie tej funkcji, dzięki czemu generowane próbki będą lepszej jakości. Proces trenowania generatora i dyskryminatora był powtarzany naprzemiennie, aż oba modele osiągnęły satysfakcjonującą dokładność i stabilność w generowaniu realistycznych danych.

Wykorzystując własną architekturę GAN udało się stworzyć 24 modele, gdzie każdy z nich był w stanie generować fikcyjne zdjęcia dla jednej z osób. Jako wejście generator przyjmuje wektor liczb losowych, natomiast jako wyjście jest zwracany obraz w formacie 64 na 64 piksele. W celu wytrenowania modelu wykonano 20 tysięcy iteracji dla każdej z sieci. Na rysunku 4.6 przedstawiono zdjęcia wygenerowane za pomocą własnej sieci GAN. W porównaniu, rysunek 4.5 przedstawia rzeczywiste zdjęcia.



Rysunek 4.5: Rzeczywiste zdjęcia



Rysunek 4.6: Fikcyjne zdjęcia

4.5 Trenowanie modeli z wykorzystaniem sieci StyleGan2

Kolejnym krokiem generowania fikcyjnych zdjęć twarzy była nauka modelu GAN za pomocą wcześniej rozszerzonego zbioru rzeczywistych próbek. StyleGan2 daje możliwość dodania dodatkowej etykiety, która określa jakiego typu obraz ma zostać wygenerowany. W tym przypadku zdecydowano się na przebadaniu dwóch rodzajów sieci. Pierwszy z nich zawierał jedynie zdjęcia tylko jednej osoby. Było to spowodowane możliwością mieszania się cudzych cech twarzy na jednym zdjęciu. Parametry wykorzystane podczas nauki sieci przedstawiono w tabeli 4.7.

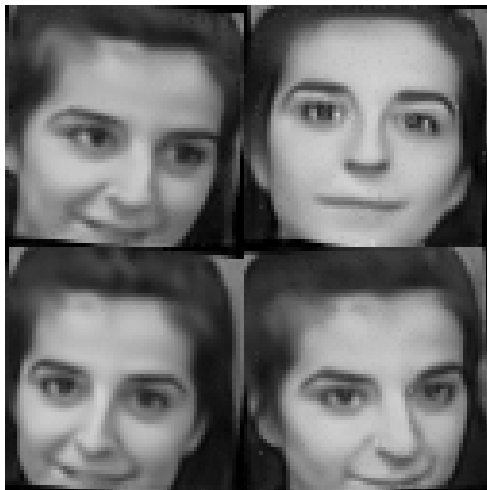
Parametr	wartość
Rozmiar partii	16
Gamma	3.2768
Całkowity czas treningu	100 epok

Tabela 4.7: Przedstawia parametry wykorzystane do uczenia modelu StyleGan2 bez etykiet.

Rozmiar partii (z ang. *batch size*) oznacza liczbę próbek danych przetwarzanych jednocześnie podczas jednej iteracji treningowej. Gamma (z ang. *batch size*) określa, w jakim stopniu statystyki stylu są skalowane. Im większa wartość parametru gamma, tym intensywniejsza jest zmiana stylu, podczas gdy mniejsza wartość gamma prowadzi do bardziej subtelnych zmian. Całkowity czas treningu (z ang. *total training duration*) oznacza całkowity czas treningu modelu, liczony jest tysiącach iteracji.

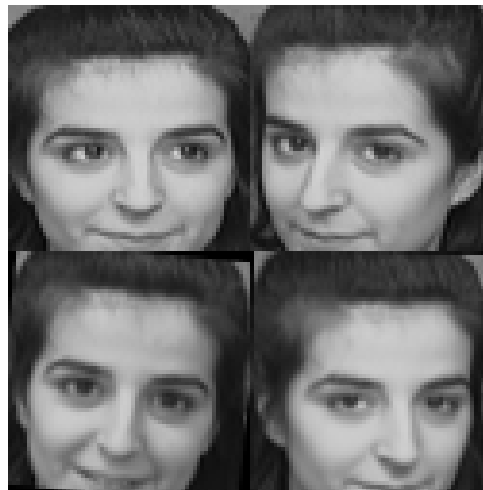
Głównymi wadami rozwiązania związanego z tworzeniem osobnego modelu dla każdej osoby jest znacznie dłuższy czas generowania modeli. Wytrenowanie jednego modelu dla 100 tysięcy iteracji zajmowało około 18 godzin. Modele były trenowane przy wykorzystaniu GPU Nvidia Quadro P4000. Na rysunku 4.7 przedstawione zostały rezultaty. Dla porównania po lewej stronie znajdują się rzeczywiste zdjęcia wybranej osoby 4.7, natomiast po prawej przedstawione zostały fikcyjne zdjęcia, wygenerowane przez model StyleGan2, rysunek 4.8.

a)



Rysunek 4.7: Rzeczywiste zdjęcia

b)



Rysunek 4.8: Fikcyjne zdjęcia

Drugim sposobem generowania fikcyjnych zdjęć było wykorzystanie dodatkowych etykiet, które określały jaka osoba została przedstawiona na fotografii. W tym przypadku został stworzony tylko jeden model, który był w stanie generować zdjęcia różnych osób w zależności od podanej etykiety. Czas trenowania modelu, zawierającego zdjęcia wszystkich osób, wyniósł 120 godzin. Parametry wykorzystane podczas nauki sieci przedstawiono w tabeli 4.8.

Parametr	wartość
Rozmiar partii	16
Gamma	3.2768
Całkowity czas treningu	1000 epok
Warunek	true

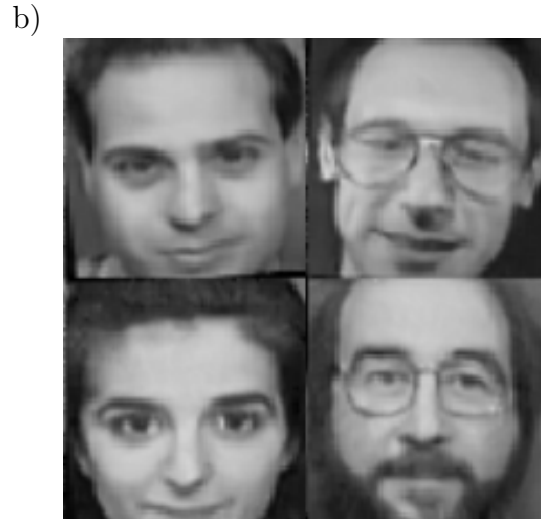
Tabela 4.8: Przedstawia parametry wykorzystane do uczenia modelu stylegan2 z etykietami.

Warunek (z ang. *condition*) umożliwia generowanie obrazów na podstawie dodatkowych informacji wejściowych.

Zaletą podejścia związanego z wykorzystaniem dodatkowych etykiet jest znacząco zmniejszony czas nauki modelu. Na rysunku 4.9 przedstawione zostało porównanie uzyskanych rezultatów z rzeczywistymi zdjęciami. Po lewej stronie pokazane zostały rzeczywiste zdjęcia 4.9, natomiast po prawej znajdują się fikcyjne fotografie 4.10, wygenerowane przez model StyleGan2 z dodatkowym wejściem określającym daną osobę.



Rysunek 4.9: Rzeczywiste zdjęcia



Rysunek 4.10: Fikcyjne zdjęcia

Jak można zauważyć na rysunkach 4.10 oraz 4.8, wygenerowane zdjęcia są wysokiej jakości, zawierają także dużą ilość szczegółów, a co najważniejsze, są one podobne do pierwowzorów.

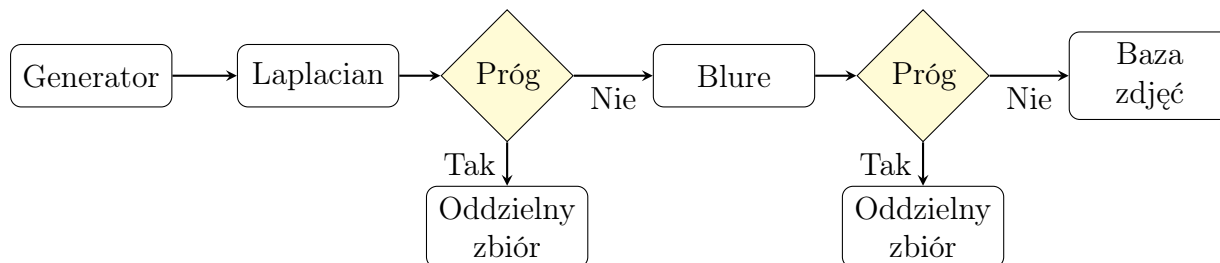
Wśród wygenerowanych fikcyjnych zdjęć twarzy można znaleźć także wiele różnych, niedoskonałych fotografii. Jak można zauważyć na rysunku 4.11, model GAN wygenerował zdjęcia, które posiadają różnego rodzaju wady. Przykładowo zdjęcie po lewej stronie przedstawia osobę z dwoma nosami, natomiast osoba z prawej strony ma zniekształcone oczy.



Rysunek 4.11: Błędnie wygenerowane próbki

W celu wyeliminowania tych wad zdecydowano się wykorzystać dodatkowy model, który jest w stanie wykrywać twarze osób na zdjęciach. Zastosowanie tego modelu umożliwiło odrzucenie próbek, na których znajdowały się mało wyraźne osoby. Dodatkowo wszystkie zdjęcia zostały przetworzone przez funkcje Laplacian oraz Blure, które dostępne są w bibliotece OpenCV. Laplacian wykorzystany był do wykrywania krawędzi w obrazie, dzięki czemu określono miarę ostrości obrazu. Blure wykorzystuje analizę częstotliwościową obrazu, przy użyciu szybkiej transformaty Furiera (z ang. *Fast Fourier Transform, FFT*),

gdzie wyższa wartość wskaźnika rozmycia oznacza bardziej ostry obraz. Na rysunku 4.12 przedstawiony został schemat detekcji wadliwych zdjęć.

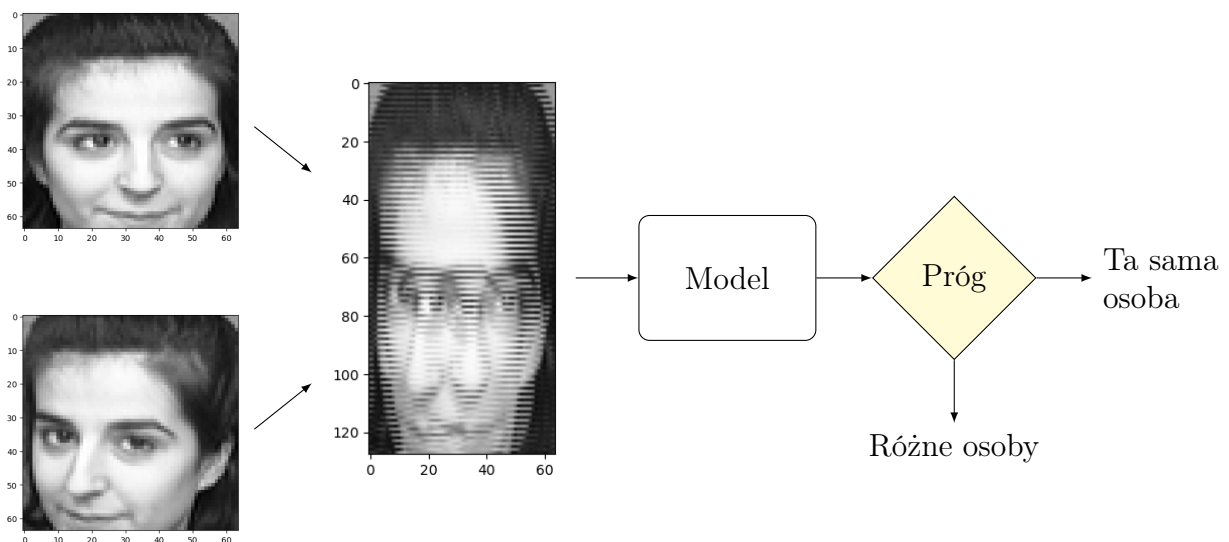


Rysunek 4.12: Schemat detekcji wadliwych zdjęć

Rozdział 5

Porównanie rezultatów

Decydując się zbadać, czy próbki generowane przez model GAN mogą rozszerzyć zbiór uczący, wykorzystano model określający podobieństwo między zdjęciami twarzy [2]. Na rysunku 5.1, przedstawiony został schemat ilustrujący zasadę działania modelu. Na samym początku przyjmowane są dwa zdjęcia twarzy, które następnie zostały połączone w jeden obraz poprzez wzajemne przeplatanie wierszy. Zgodnie z wynikami uzyskanymi w pracy [2], tego rodzaju sposób przedstawienia danych daje najlepsze rezultaty. Jako wejście, model przyjmuje zdjęcie, które przedstawia dwie twarze osób, natomiast jako wynik określa podobieństwo pomiędzy dwoma twarzami. Następnie zdjęcia są kwalifikowane jako pozytywne lub negatywne w zależności od wartości progu, którą ustalono na 0,5. Jako dane walidacyjne i dane treningowe, podczas nauki modelu, zdecydowano się wykorzystać próbki zawierające rzeczywiste zdjęcia. Jako metrykę oceny jakości modelu wybrano wartość dokładności (z ang. *accuracy*) oraz wartość straty (z ang. *loss*) dla zbioru testowego.



Rysunek 5.1: Schemat działania modelu

5.1 Zbiory danych

Do wyuczenia modelu wykorzystano zbiór treningowy oraz walidacyjny. Natomiast do badania jakości modelu zastosowano zestaw danych testowych. Zbiór treningowy zawierał zdjęcia przedstawiające 24 osoby, natomiast zbiór testowy i walidacyjny po 4 postacie. Każda baza zawierała zdjęcia innych osób w celu ograniczenia przecieku danych (ang. *data leakage*). Zbiory danych zawierały także etykiety, które określały czy dana próbka jest pozytywna lub też negatywna. Pozytywna próbka była przedstawiana jako wartość 1, natomiast negatywna jako 0.

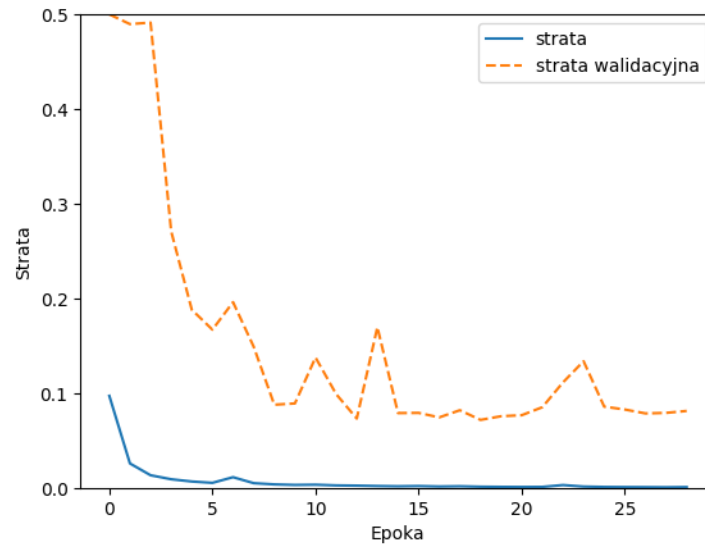
Kolejnym etapem pracy było wybranie odpowiedniego sposobu nauki modelu. Wśród wielu technik można wyróżnić kilka strategii wykorzystania wygenerowanych danych do nauki sieci neuronowej. Pierwszym sposobem trenowania modelu określającego podobieństwo jest wykorzystanie rzeczywistych danych do treningu oraz prawdziwych zdjęć do testowania [4]. Wykorzystując ten sposób można poddać ocenie rzeczywistą wydajność modelu na realnych zbiorach. Wadą tego podejścia jest ograniczona wielkość oraz różnorodność zbioru treningowego i testowego. Drugim sposobem nauki modelu jest wykorzystanie danych fikcyjnych do treningu oraz rzeczywistego zbioru do walidacji i testowania. To podejście pozwala na wykorzystanie syntetycznych danych do rozszerzenia zbioru treningowego. Stosuje się je głównie w ograniczonych zbiorach, bądź gdy pozyskanie nowych danych jest kosztowne. Wykorzystując to rozwiązanie można wygenerować duży i zróżnicowany zbiór danych oraz zaoszczędzić czas związany z pozyskaniem nowych próbek. Jednakże istnieje duże ryzyko wystąpienia nadmiernego dopasowania (z ang. *overfitting*). Spowodowane to jest wykorzystaniem danych syntetycznych oraz możliwość nieadekwatnego przeniesienia nauki na rzeczywiste dane (z ang. *transfer learning*). Trzecim sposobem trenowania modelu jest wykorzystanie danych fikcyjnych do treningu, walidacji i testowania. Jest to podejście bardziej eksperymentalne, używane głównie w badaniach nad samymi modelami GAN. Wykorzystanie tego podejścia daje pełną kontrolę nad jakością i różnorodnością danych oraz możliwością łatwego i szybkiego tworzenia dużych zbiorów danych. Głównymi wadami tego podejścia jest brak bezpośredniego odniesienia do rzeczywistych danych i ryzyko, że model nie będzie działał poprawnie na rzeczywistych danych. Ostatnim sposobem trenowania modelu jest wykorzystanie rzeczywistego zbioru do trenowania oraz fikcyjnych zdjęć do walidacji i testowania wyuczonego modelu. Tego typu podejście jest wykorzystywane głównie przy walidacji jakości danych generowanych przez model GAN. Główną zaletą jest możliwość bezpośredniego oceniania jakości generowanych danych oraz możliwość szybkiego testowania modeli. Z drugiej strony wykorzystanie tego podejścia może nie dostarczać pełnej informacji o rzeczywistej wydajności modelu na prawdziwych danych.

W badaniach nad wpływem rozszerzenia zbioru za pomocą sieci GAN, zdecydowano się wybrać drugie podejście, które polegało na trenowaniu modelu za pomocą danych fikcyjnych i testowaniu na danych rzeczywistych. Wybrano to podejście, ponieważ daje ono możliwość przetestowania, czy model trenowany na syntetycznych danych jest w stanie poprawnie określać podobieństwo dla rzeczywistych próbek.

5.2 Rzeczywiste próbki

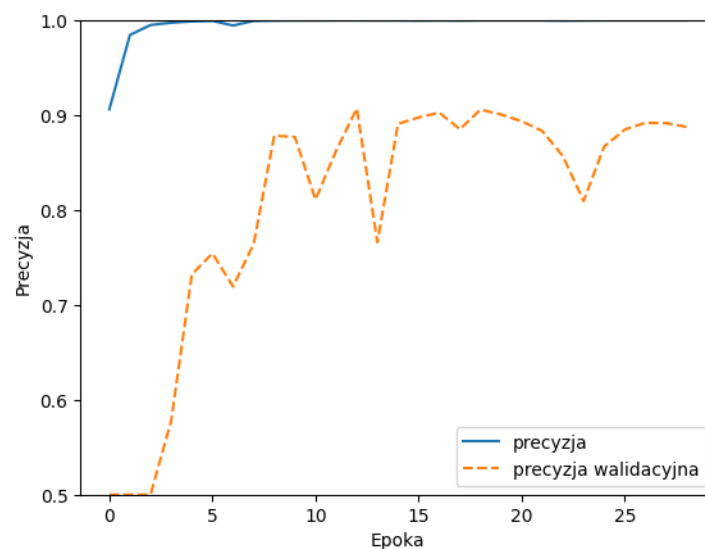
Jako pierwszy wyuczony został model zawierający jedynie rzeczywiste zdjęcia osób. Zbiór treningowy zawierał łącznie 15120 próbek, z czego baza danych walidacyjnych i testowych posiadała po 2520 próbek, przy czym stosunek liczby próbek pozytywnych i negatywnych

był w wszystkich zbiorach taki sam. Rysunek 5.2 prezentuje wykres zawierający dwa przebiegi. Pierwszy z nich, oznaczony jako strata, reprezentuje wartość funkcji strat dla zbioru treningowego, podczas gdy drugi, oznaczony jako strata walidacyjna, określa wartość tej funkcji dla danych walidacyjnych.



Rysunek 5.2: Wykres wartości funkcji strat dla rzeczywistych zdjęć

Następnie wygenerowany został drugi wykres, przedstawiony na rysunku 5.3, który przedstawiał funkcje precyzji. Pierwsza linia, oznaczana jako precyzja, reprezentuje wartość funkcji precyzji sieci dla zestawu treningowego. Druga linia, oznaczana jako precyzja walidacyjna, przedstawia wartość funkcji precyzji uzyskaną dla próbek walidacyjnych.



Rysunek 5.3: Wykres wartości funkcji precyzji dla rzeczywistych zdjęć

Po wygenerowaniu wykresów przystąpiono do oceny efektywności modelu. W tym celu skorzystano z metody `model.evaluate()`, co pozwoliło na ocenę stworzonego modelu przy użyciu zbioru testowego. W wyniku uzyskano precyzję równą 86,75% oraz wartość straty wynoszącą 12,49%.

Kolejny krok w ocenie jakości sieci obejmował użycie macierzy pomyłek. W tym celu użyto funkcję *probability_model()*, która dla zbioru testowego zwróciła etykiety z prawdopodobieństwem trafności klasyfikacji. Przyjęto, że wartości powyżej lub równych 0,5 oznaczają, że osoby na zdjęciach są tej samej tożsamości, podczas gdy wartości poniżej 0,5 wskazują, że osoby te nie są podobne. Następnie otrzymane wyniki wraz z etykietami zostały przedstawione w postaci macierzy pomyłek (z ang. confusion matrix).

$$\begin{bmatrix} T_n & N_n \\ N_p & T_p \end{bmatrix} = \begin{bmatrix} 1144 & 116 \\ 281 & 979 \end{bmatrix}, \quad (5.1)$$

gdzie:

- T_n (z ang. *True negative*) oznacza próbki negatywne, które zostały poprawnie ocenione,
- N_n (z ang. *Negative negative*) oznacza próbki negatywne, które zostały błędnie ocenione,
- N_p (z ang. *Negative positive*) oznacza próbki pozytywne, które zostały błędnie ocenione,
- T_p (z ang. *True positive*) oznacza próbki pozytywne, które zostały poprawnie ocenione.

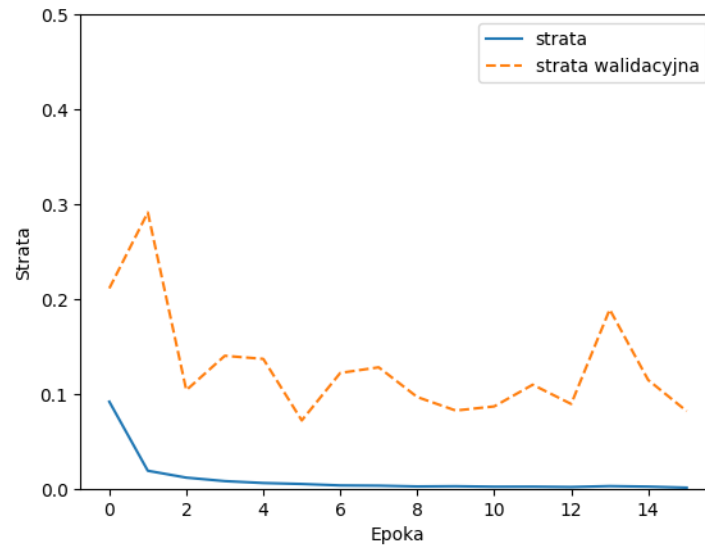
Wyuczona sieć poprawnie sklasyfikowała 2263 próbek, co stanowi 84,2% wszystkich próbek w zbiorze testowym. Warto zauważyć, że błędnie sklasyfikowane pozytywne próbki stanowią 11,2% wszystkich próbek testowych. Sieć, która wykorzystuje rzeczywiste zdjęcia w reprezentacji, radzi sobie bardzo dobrze z identyfikacją negatywnych próbek. W sumie algorytm popełnił 116 błędów w identyfikacji negatywnych próbek, co stanowi 4,6% zbioru testowego.

5.3 Łączenie zbiorów

Kolejnym etapem było wyuczenie modelu określającego podobieństwo przy wykorzystaniu danych zawierających rzeczywiste oraz fikcyjne zdjęcia. W tym celu do zbioru treningowego, posiadającego prawdziwe fotografie, dodano próbki wygenerowane przez model StyleGan2.

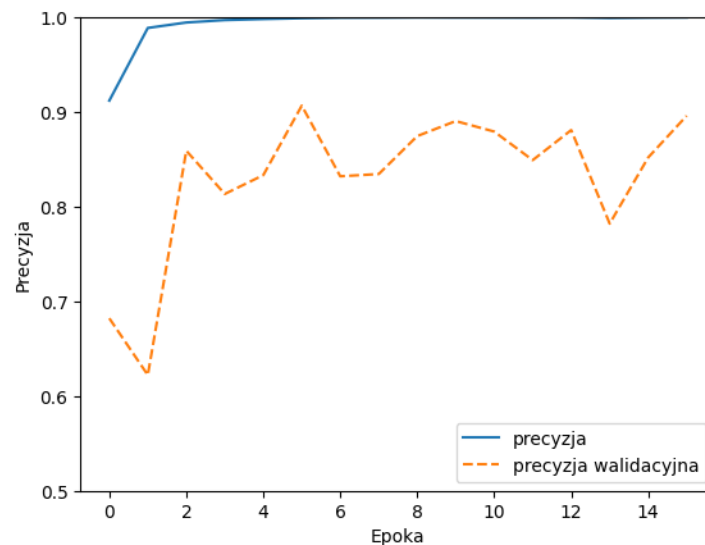
5.3.1 StyleGan2 z etykietami

Zbiór treningowy zawierał łącznie 30240 próbek wśród których było 15120 próbek rzeczywistych oraz 15120 próbek fikcyjnych. Dane fikcyjne zostały wygenerowane poprzez model StyleGan2, który zawierał dodatkowe etykiety określające wybraną osobę. Zbiór danych walidacyjnych i testowych posiadał po 2520 próbek rzeczywistych. Rysunek 5.4 prezentuje wykres zawierający dwa przebiegi. Pierwszy z nich, oznaczony jako strata, reprezentuje wartość funkcji strat dla zestawu treningowego, podczas gdy drugi, oznaczony jako strata walidacyjna, określa wartość tej funkcji dla zbioru walidacyjnego.



Rysunek 5.4: Wykres wartości funkcji strat dla zbioru zawierającego rzeczywiste i fikcyjne zdjęć

Następnie wygenerowany został drugi wykres, przedstawiony na rysunku 5.3, który przedstawiał funkcje precyzji. Pierwsza linia, oznaczana jako precyzja, reprezentuje wartość funkcji precyzji sieci dla zestawu treningowego. Druga linia, oznaczana jako precyzja walidacyjna, przedstawia wartość funkcji precyzji uzyskaną dla danych walidacyjnych.



Rysunek 5.5: Wykres wartości funkcji precyzji dla zbioru zawierającego rzeczywiste i fikcyjne zdjęć

Po wygenerowaniu wykresów przystąpiono do oceny efektywności modelu. W tym celu skorzystano z metody *model.evaluate()*, co pozwoliło na ocenę stworzonego modelu przy użyciu zbioru testowego. W wyniku uzyskano precyzję równą 82,94% oraz wartość straty wynoszącą 13,46%.

Kolejny krok w ocenie jakości sieci obejmował użycie macierzy pomyłek. W tym celu użyto funkcję *probability_model()*, która dla zbioru testowego zwróciła etykiety z prawdopodobieństwem trafności klasyfikacji. Przyjęto, że wartości powyżej lub równych 0,5

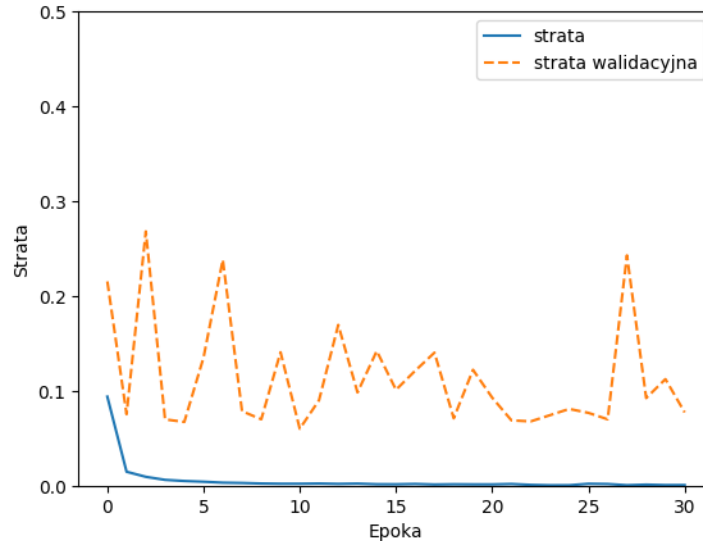
oznaczają, że osoby na zdjęciach są tej samej tożsamości, podczas gdy wartości poniżej 0,5 wskazują, że osoby te nie są podobne. Następnie otrzymane wyniki wraz z etykietami zostały przedstawione w postaci macierzy pomyłek (z ang. confusion matrix).

$$\begin{bmatrix} T_n & N_n \\ N_p & T_p \end{bmatrix} = \begin{bmatrix} 1134 & 126 \\ 304 & 956 \end{bmatrix}, \quad (5.2)$$

gdzie wyuczona sieć poprawnie sklasyfikowała 2090 próbek, co stanowi 82,9% wszystkich próbek w zbiorze testowym. Model błędnie sklasyfikował 304 próbki jako pozytywne co stanowi 12,1% wszystkich próbek testowych. Dodatkowo algorytm określił błędnie 126 negatywnych próbek, co stanowi 5% zbioru testowego.

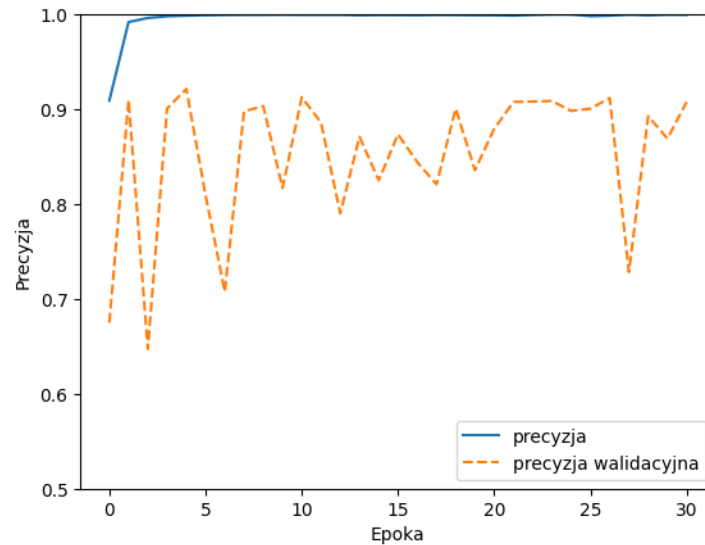
5.3.2 StyleGan2 bez etykiet

Zbiór treningowy zawierał łącznie 30240 próbek wśród których było 15120 próbek rzeczywistych oraz 15120 próbek fikcyjnych. Dane fikcyjne zostały wygenerowane poprzez 24 modele StyleGan2, które były w stanie wygenerować zdjęcia tylko dla jednej postaci. W tym przypadku modele nie wykorzystywały dodatkowych etykiet. Zbiór danych walidacyjnych i testowych posiadał po 2520 próbek rzeczywistych. Rysunek 5.6 prezentuje wykres zawierający dwa przebiegi. Pierwszy z nich, oznaczony jako strata, reprezentuje wartość funkcji strat dla zbioru treningowego, podczas gdy drugi, oznaczony jako strata walidacyjna, określa wartość tej funkcji dla danych walidacyjnych.



Rysunek 5.6: Wykres wartości funkcji strat dla zbioru zawierającego rzeczywiste i fikcyjne zdjęć

Następnie wygenerowany został drugi wykres, przedstawiony na rysunku 5.7, który przedstawiał funkcje precyzji. Pierwsza linia, oznaczana jako precyzja, reprezentuje wartość funkcji precyzji sieci dla zestawu treningowego. Druga linia, oznaczana jako precyzja walidacyjna, przedstawia wartość funkcji precyzji uzyskaną dla zestawu walidacyjnego.



Rysunek 5.7: Wykres wartości funkcji precyzji dla rzeczywistych i fikcyjnych zdjęć

Po wygenerowaniu wykresów przystąpiono do oceny efektywności modelu. W tym celu skorzystano z metody *model.evaluate()*, co pozwoliło na ocenę stworzonego modelu przy użyciu zbioru testowego. W wyniku uzyskano precyzję równą 88,77 % oraz wartość straty wynoszącą 8,72%.

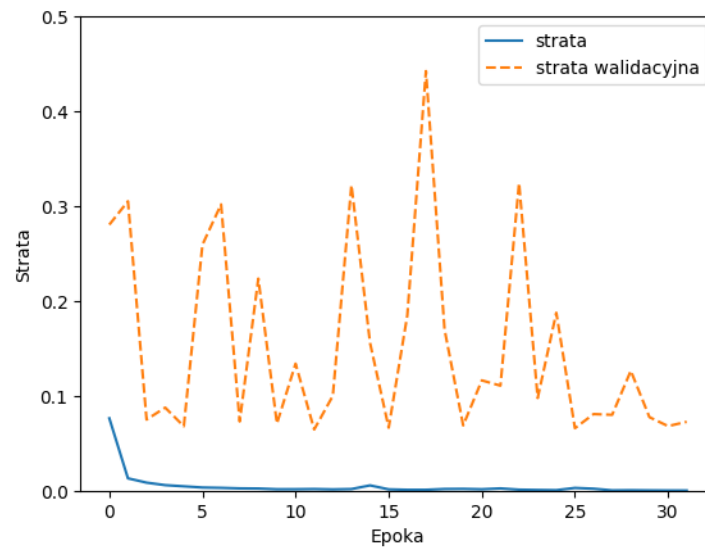
Kolejny krok w ocenie jakości sieci obejmował użycie macierzy pomyłek. W tym celu użyto funkcję *probability_model()*, która dla zbioru testowego zwróciła etykiety z prawdopodobieństwem trafności klasyfikacji. Przyjęto, że wartości powyżej lub równych 0,5 oznaczają, że osoby na zdjęciach są tej samej tożsamości, podczas gdy wartości poniżej 0,5 wskazują, że osoby te nie są podobne. Następnie otrzymane wyniki wraz z etykietami zostały przedstawione w postaci macierzy pomyłek.

$$\begin{bmatrix} T_n & N_n \\ N_p & T_p \end{bmatrix} = \begin{bmatrix} 1107 & 153 \\ 129 & 1131 \end{bmatrix}, \quad (5.3)$$

gdzie wyuczona sieć poprawnie sklasyfikowała 2238 próbek, co stanowi 88,8% wszystkich próbek w zbiorze testowym. Warto zauważyć, że błędnie sklasyfikowane pozytywne próbki stanowią 5,1% wszystkich próbek testowych. Model błędnie sklasyfikował 153 negatywnych próbek, co stanowi 6,1% zbioru testowego.

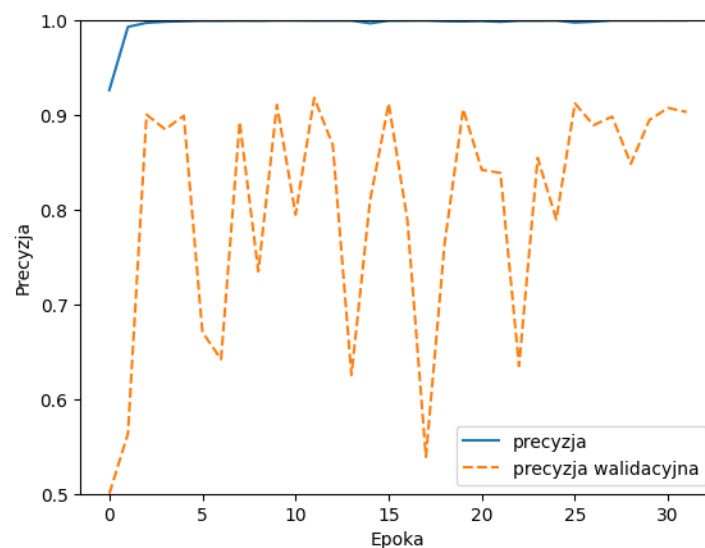
5.3.3 Własny model GAN

Zbiór treningowy zawierał łącznie 30240 próbek wśród których było 15120 próbek rzeczywistych oraz 15120 próbek fikcyjnych. Dane fikcyjne zostały wygenerowane poprzez 24 modele GAN, które były w stanie wygenerować zdjęcia tylko dla jednej postaci. Zbiór danych walidacyjnych i testowych posiadał po 2520 próbek rzeczywistych. Rysunek 5.8 prezentuje wykres zawierający dwa przebiegi. Pierwszy z nich, oznaczony jako strata, reprezentuje wartość funkcji strat dla zbioru treningowego, podczas gdy drugi, oznaczony jako strata walidacyjna, określa wartość tej funkcji dla zestawu walidacyjnego.



Rysunek 5.8: Wykres wartości funkcji strat dla zbioru zawierającego rzeczywiste i fikcyjne zdjęć

W kolejnym kroku wygenerowany został drugi wykres, rysunek 5.9, który przedstawiał funkcje precyzji. Pierwsza linia, oznaczana jako precyzja, reprezentuje wartość funkcji precyzji sieci dla zestawu treningowego. Druga linia, oznaczana jako precyzja walidacyjna, przedstawia wartość funkcji precyzji uzyskaną dla zestawu walidacyjnego.



Rysunek 5.9: Wykres wartości funkcji precyzji dla rzeczywistych i fikcyjnych zdjęć

Po wygenerowaniu wykresów przystąpiono do oceny efektywności modelu. W tym celu skorzystano z metody *model.evaluate()*, co pozwoliło na ocenę stworzonego modelu przy użyciu zbioru testowego. W wyniku uzyskano precyzję równą 82,10% oraz wartość straty wynoszącą 8,72%.

Kolejny krok w ocenie jakości sieci obejmował użycie macierzy pomyłek. W tym celu użyto funkcję *probability_model()*, która dla zbioru testowego zwróciła etykiety z prawdopodobieństwem trafności klasyfikacji. Przyjęto, że wartości powyżej lub równych 0,5 oznaczają, że osoby na zdjęciach są tej samej tożsamości, podczas gdy wartości poniżej

0,5 wskazują, że osoby te nie są podobne. Następnie otrzymane wyniki wraz z etykietami zostały przedstawione w postaci macierzy pomyłek.

$$\begin{bmatrix} T_n & N_n \\ N_p & T_p \end{bmatrix} = \begin{bmatrix} 1138 & 122 \\ 302 & 958 \end{bmatrix}, \quad (5.4)$$

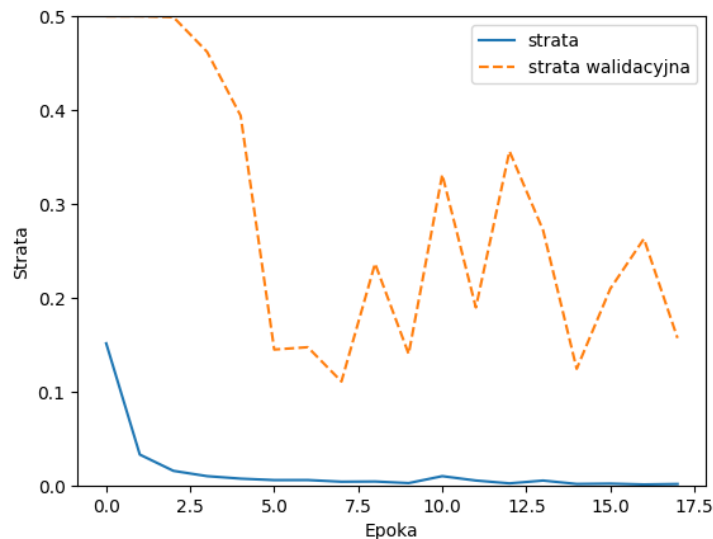
gdzie wyuczona sieć poprawnie sklasyfikowała 2096 próbek, co stanowi 83,2% wszystkich próbek w zbiorze testowym. Warto zauważyć, że błędnie sklasyfikowane pozytywne próbki stanowią 12,0% wszystkich próbek testowych. Model błędnie sklasyfikował 122 negatywnych próbek, co stanowi 4,8% zbioru testowego.

5.4 Wyuczenie modeli na fikcyjnych zdjęciach

Następnie wytrenowano trzy kolejne modele, określające podobieństwo pomiędzy dwoma zdjęciami, przy wykorzystaniu zbioru treningowego zawierającego jedynie fikcyjne fotografie twarzy.

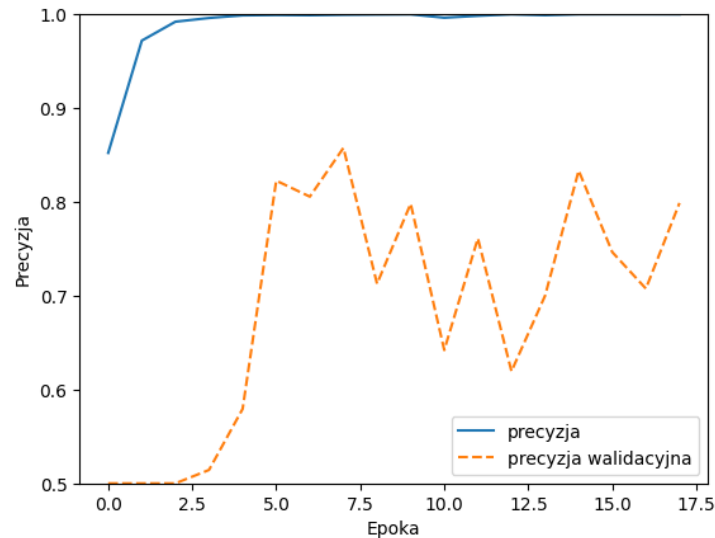
5.4.1 StyleGan2 z etykietami

Zbiór treningowy zawierał łącznie 15120 fikcyjnych próbek. Dane zostały wygenerowane poprzez model StyleGan2, który zawierał dodatkowe etykiety określające wybraną osobę. Zbiór danych walidacyjnych i testowych posiadał łącznie 2520 próbek rzeczywistych. Rysunek 5.10 prezentuje wykres zawierający dwa przebiegi. Pierwszy z nich, oznaczony jako strata, reprezentuje wartość funkcji strat dla zbioru treningowego, podczas gdy drugi, oznaczony jako strata walidacyjna, określa wartość tej funkcji dla zbioru walidacyjnego.



Rysunek 5.10: Wykres wartości funkcji strat dla zbioru zawierającego jedynie fikcyjne zdjęć

Następnie wygenerowany został drugi wykres, przedstawiony na rysunku 5.11, który przedstawiał funkcje precyzji. Pierwsza linia, oznaczana jako precyzja, reprezentuje wartość funkcji precyzji sieci dla zestawu treningowego. Druga linia, oznaczana jako precyzja walidacyjna, przedstawia wartość funkcji precyzji uzyskaną dla zestawu walidacyjnego.



Rysunek 5.11: Wykres wartości funkcji precyzji dla fikcyjnych zdjęć

Po wygenerowaniu wykresów przystąpiono do oceny efektywności modelu. W tym celu skorzystano z metody *model.evaluate()*, co pozwoliło na ocenę stworzonego modelu przy użyciu zbioru testowego. W wyniku uzyskano precyzję równą 82,38% oraz wartość straty wynoszącą 14,60%.

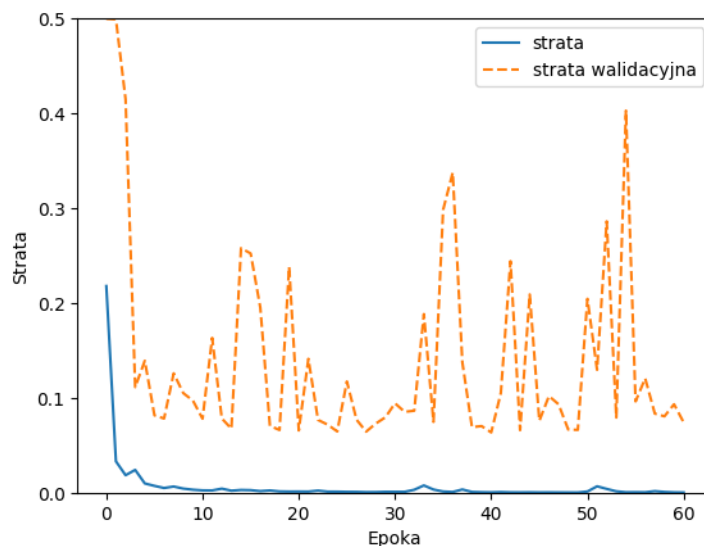
Kolejny krok w ocenie jakości sieci obejmował użycie macierzy pomyłek. W tym celu użyto funkcję *probability_model()*, która dla zbioru testowego zwróciła etykiety z prawdopodobieństwem trafności klasyfikacji. Przyjęto, że wartości powyżej lub równych 0,5 oznaczają, że osoby na zdjęciach są tej samej tożsamości, podczas gdy wartości poniżej 0,5 wskazują, że osoby te nie są podobne. Następnie otrzymane wyniki wraz z etykietami zostały przedstawione w postaci macierzy pomyłek.

$$\begin{bmatrix} T_n & N_n \\ N_p & T_p \end{bmatrix} = \begin{bmatrix} 1185 & 75 \\ 369 & 891 \end{bmatrix}, \quad (5.5)$$

gdzie wyuczona sieć poprawnie sklasyfikowała 2076 próbek, co stanowi 82,4% wszystkich próbek w zbiorze testowym. Warto zauważyć, że błędnie sklasyfikowane pozytywne próbki stanowią 14,6% wszystkich próbek testowych. Model błędnie sklasyfikował 75 negatywnych próbek, co stanowi 3,0% zbioru testowego.

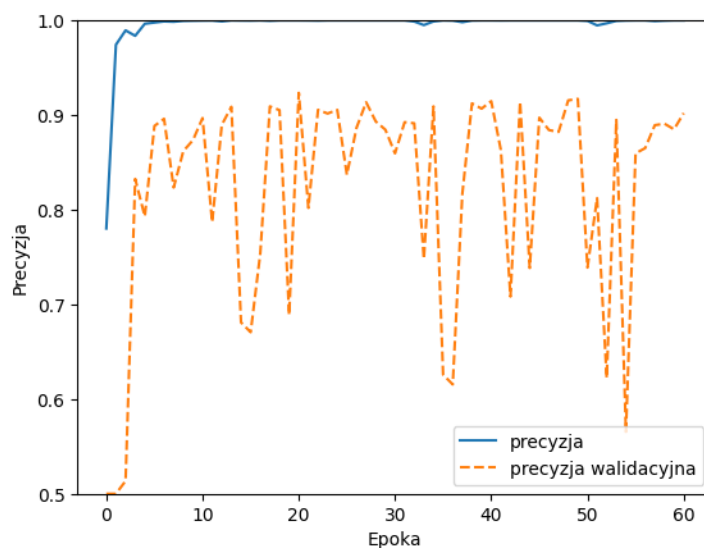
5.4.2 StyleGan2 bez etykiet

Zbiór treningowy zawierał łącznie 15120 próbek wśród których wszystkie zdjęcia nie były rzeczywiste. Fikcyjne dane zostały wygenerowane poprzez 24 modele StyleGan2, które są w stanie stworzyć zdjęcia tylko dla jednej postaci. W tym przypadku modele nie wykorzystywały dodatkowych etykiet. Zbiór danych walidacyjnych i testowych posiadał po 2520 próbek rzeczywistych. Rysunek 5.6 prezentuje wykres zawierający dwa przebiegi. Pierwszy z nich, oznaczony jako strata, reprezentuje wartość funkcji strat dla zbioru treningowego, podczas gdy drugi, oznaczony jako strata walidacyjna, określa wartość tej funkcji dla zbioru walidacyjnego.



Rysunek 5.12: Wykres wartości funkcji strat dla zbioru zawierającego jedynie fikcyjne zdjęcia

Kolejnym krokiem był stworzenie drugiego wykresu, rysunek 5.13, który przedstawiał funkcje precyzji. Pierwsza linia, oznaczana jako precyzja, reprezentuje wartość funkcji precyzji sieci dla zestawu treningowego. Druga linia, oznaczana jako precyzja walidacyjna, przedstawia wartość funkcji precyzji uzyskaną dla zestawu walidacyjnego.



Rysunek 5.13: Wykres wartości funkcji precyzji dla zbioru zawierającego jedynie fikcyjne zdjęcia

Po wygenerowaniu wykresów przystąpiono do oceny efektywności modelu. W tym celu skorzystano z metody *model.evaluate()*, co pozwoliło na ocenę stworzonego modelu przy użyciu zbioru testowego. W wyniku uzyskano precyzję równą 87,06% oraz wartość straty wynoszącą 10,18%.

Kolejny krok w ocenie jakości sieci obejmował użycie macierzy pomyłek. W tym celu użyto funkcję *probability_model()*, która dla zbioru testowego zwróciła etykiety z prawdopodobieństwem trafności klasyfikacji. Przyjęto, że wartości powyżej lub równych 0,5

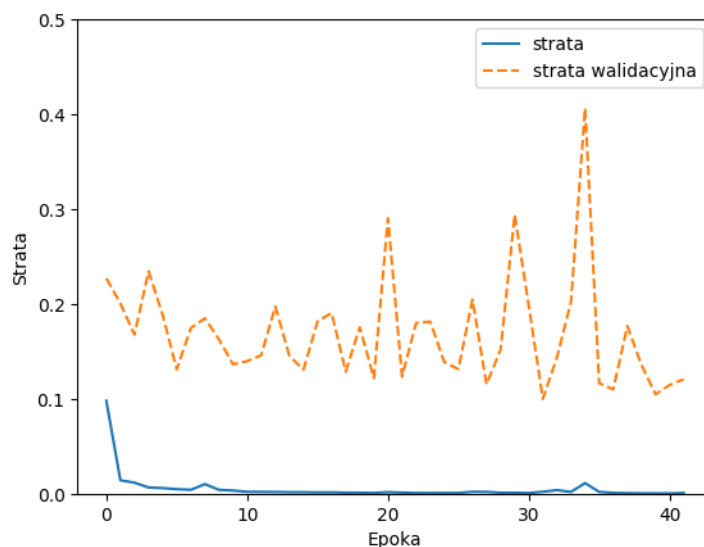
oznaczają, że osoby na zdjęciach są tej samej tożsamości, podczas gdy wartości poniżej 0,5 wskazują, że osoby te nie są podobne. Następnie otrzymane wyniki wraz z etykietami zostały przedstawione w postaci macierzy pomyłek.

$$\begin{bmatrix} T_n & N_n \\ N_p & T_p \end{bmatrix} = \begin{bmatrix} 1181 & 79 \\ 247 & 1013 \end{bmatrix}, \quad (5.6)$$

gdzie wyuczona sieć poprawnie sklasyfikowała 2194 próbek, co stanowi 87,1% wszystkich próbek w zbiorze testowym. Warto zauważyć, że błędnie sklasyfikowane pozytywne próbki stanowią 9,8% wszystkich próbek testowych. Model błędnie sklasyfikował 79 negatywnych próbek, co stanowi 3,1% zbioru testowego.

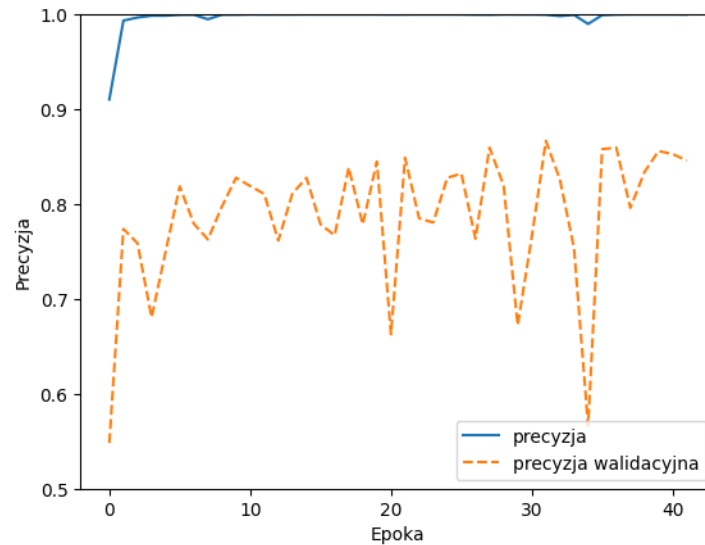
5.4.3 Własny model GAN

Zbiór treningowy zawierał łącznie 15120 próbek wśród których wszystkie zdjęcia były syntetyczne. Fikcyjne dane zostały wygenerowane poprzez 24 modele GAN, które były w stanie stworzyć zdjęcia tylko dla jednej postaci. Zbiór danych walidacyjnych i testowych posiadał po 2520 próbek rzeczywistych. Rysunek 5.8 prezentuje wykres zawierający dwa przebiegi. Pierwszy z nich, oznaczony jako strata, reprezentuje wartość funkcji strat dla zbioru treningowego, podczas gdy drugi, oznaczony jako strata walidacyjna, określa wartość tej funkcji dla zbioru walidacyjnego.



Rysunek 5.14: Wykres wartości funkcji strat dla zbioru zawierającego jedynie fikcyjne zdjęcia

Kolejnym krokiem był stworzenie drugiego wykresu, rysunek 5.15, który przedstawiał funkcje precyzji. Pierwsza linia, oznaczana jako precyzja, reprezentuje wartość funkcji precyzji sieci dla zestawu treningowego. Druga linia, oznaczana jako precyzja walidacyjna, przedstawia wartość funkcji precyzji uzyskaną dla zestawu walidacyjnego.



Rysunek 5.15: Wykres wartości funkcji precyzji dla zbioru zawierającego jedynie fikcyjne zdjęcia

Po wygenerowaniu wykresów przystąpiono do oceny efektywności modelu. W tym celu skorzystano z metody `model.evaluate()`, co pozwoliło na ocenę stworzonego modelu przy użyciu zbioru testowego. W wyniku uzyskano precyzję równą 83,73% oraz wartość straty wynoszącą 13,20%.

Kolejny krok w ocenie jakości sieci obejmował użycie macierzy pomyłek. W tym celu użyto funkcję `probability_model()`, która dla zbioru testowego zwróciła etykiety z prawdopodobieństwem trafności klasyfikacji. Przyjęto, że wartości powyżej lub równych 0,5 oznaczają, że osoby na zdjęciach są tej samej tożsamości, podczas gdy wartości poniżej 0,5 wskazują, że osoby te nie są podobne. Następnie otrzymane wyniki wraz z etykietami zostały przedstawione w postaci macierzy pomyłek.

$$\begin{bmatrix} T_n & N_n \\ N_p & T_p \end{bmatrix} = \begin{bmatrix} 1079 & 184 \\ 226 & 1034 \end{bmatrix}, \quad (5.7)$$

gdzie wyuczona sieć poprawnie sklasyfikowała 2113 próbek, co stanowi 83,4% wszystkich próbek w zbiorze testowym. Warto zauważyć, że błędnie sklasyfikowane pozytywne próbki stanowią 9,0% wszystkich próbek testowych. Model błędnie sklasyfikował 184 negatywnych próbek, co stanowi 7,3% zbioru testowego.

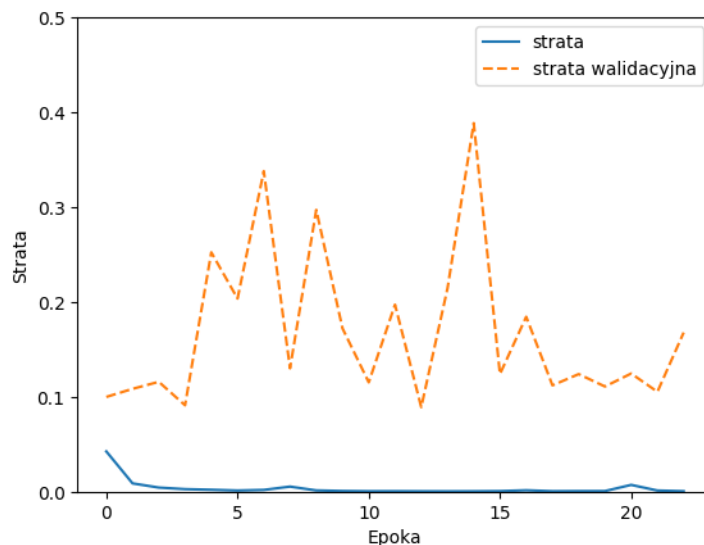
5.5 Dotrenowanie rzeczywistego modelu

Następnie zdecydowano się na zbadanie wpływu syntetycznych danych na już wcześniej wyuczoną sieć neuronową. W tym celu wykorzystany został model wytrenowany na rzeczywistych zdjęciach, który następnie został douczony przez dodanie fikcyjnych próbek.

5.5.1 StyleGan2 z etykietami

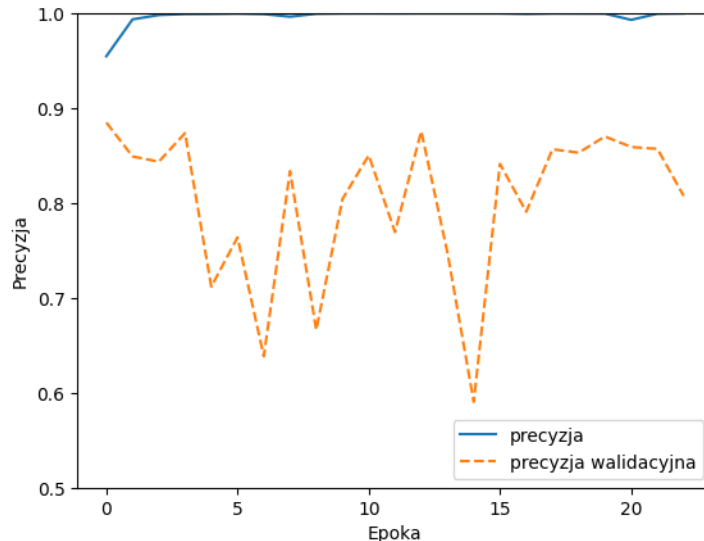
Zbiór treningowy zawierał łącznie 15120 fikcyjnych próbek. Dane zostały wygenerowane poprzez model StyleGan2, który zawierał dodatkowe etykiety określające wybraną osobę. Zbiór danych walidacyjnych i testowych posiadał łącznie 2520 próbek rzeczywistych. Rysunek 5.16 prezentuje wykres zawierający dwa przebiegi. Pierwszy z nich, oznaczony

jako strata, reprezentuje wartość funkcji strat dla zbioru treningowego, podczas gdy drugi, oznaczony jako strata walidacyjna, określa wartość tej funkcji dla zbioru walidacyjnego.



Rysunek 5.16: Wykres wartości funkcji strat dla zbioru zawierającego jedynie fikcyjne zdjęć

Następnie wygenerowany został drugi wykres, przedstawiony na rysunku 5.17, który przedstawiał funkcje precyzji. Pierwsza linia, oznaczana jako precyzja, reprezentuje wartość funkcji precyzji sieci dla zestawu treningowego. Druga linia, oznaczana jako precyzja walidacyjna, przedstawia wartość funkcji precyzji uzyskaną dla zestawu walidacyjnego.



Rysunek 5.17: Wykres wartości funkcji precyzji dla fikcyjnych zdjęć

Po wygenerowaniu wykresów przystąpiono do oceny efektywności modelu. W tym celu skorzystano z metody *model.evaluate()*, co pozwoliło na ocenę stworzonego modelu przy użyciu zbioru testowego. W wyniku uzyskano precyzję równą 82,38% oraz wartość straty wynoszącą 14,60%.

Kolejny krok w ocenie jakości sieci obejmował użycie macierzy pomyłek. W tym celu użyto funkcję *probability_model()*, która dla zbioru testowego zwróciła etykiety z praw-

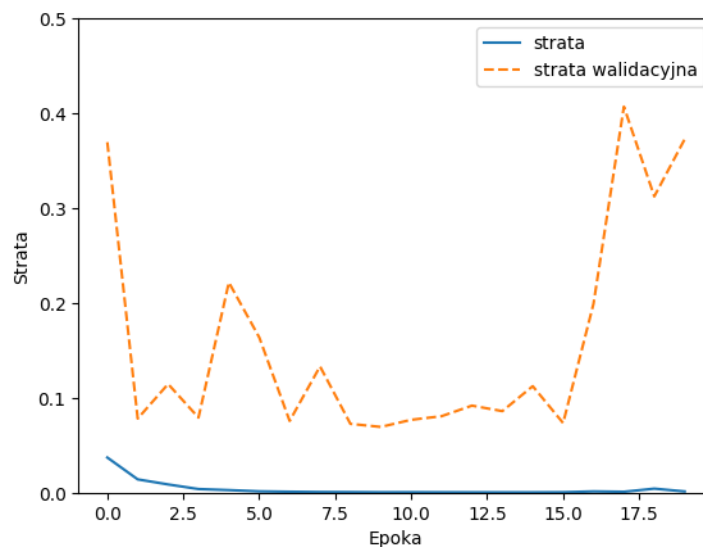
dopodobieństwem trafności klasyfikacji. Przyjęto, że wartości powyżej lub równych 0,5 oznaczają, że osoby na zdjęciach są tej samej tożsamości, podczas gdy wartości poniżej 0,5 wskazują, że osoby te nie są podobne. Następnie otrzymane wyniki wraz z etykietami zostały przedstawione w postaci macierzy pomyłek.

$$\begin{bmatrix} T_n & N_n \\ N_p & T_p \end{bmatrix} = \begin{bmatrix} 1139 & 121 \\ 313 & 947 \end{bmatrix}, \quad (5.8)$$

gdzie wyuczona sieć poprawnie sklasyfikowała 2076 próbek, co stanowi 82,4% wszystkich próbek w zbiorze testowym. Warto zauważyć, że błędnie sklasyfikowane pozytywne próbki stanowią 14,6% wszystkich próbek testowych. Model błędnie sklasyfikował 75 negatywnych próbek, co stanowi 3,0% zbioru testowego.

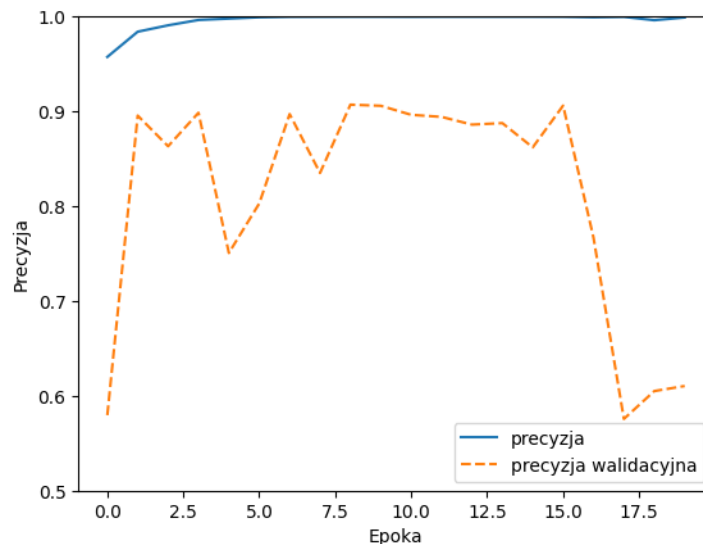
5.5.2 StyleGan2 bez etykiet

Zbiór treningowy zawierał łącznie 15120 próbek, z których wszystkie były syntetyczne. Dane fikcyjne zostały wygenerowane przez 24 modele StyleGan2, które mogły tworzyć zdjęcia tylko jednej postaci. Zbiory walidacyjne i testowe zawierały po 2520 próbek rzeczywistych. Rysunek 5.18 przedstawia wykres z dwoma przebiegami. Pierwszy, oznaczony jako strata, przedstawia wartość funkcji strat dla zbioru treningowego, natomiast drugi, oznaczony jako strata walidacyjna, pokazuje wartość tej funkcji dla zbioru walidacyjnego



Rysunek 5.18: Wykres wartości funkcji strat dla zbioru zawierającego fikcyjne zdjęć

Kolejnym krokiem było stworzenie drugiego wykresu, rysunek 5.19, który przedstawiał funkcje precyzji. Pierwsza linia, oznaczana jako precyzja, reprezentuje wartość funkcji precyzji sieci dla zestawu treningowego. Druga linia, oznaczana jako precyzja walidacyjna, przedstawia wartość funkcji precyzji uzyskaną dla zestawu walidacyjnego.



Rysunek 5.19: Wykres wartości funkcji precyzji dla fikcyjnych zdjęć

Po wygenerowaniu wykresów przystąpiono do oceny efektywności modelu. W tym celu skorzystano z metody *model.evaluate()*, co pozwoliło na ocenę stworzonego modelu przy użyciu zbioru testowego. W wyniku uzyskano precyzję równą 81,63% oraz wartość straty wynoszącą 14,13%.

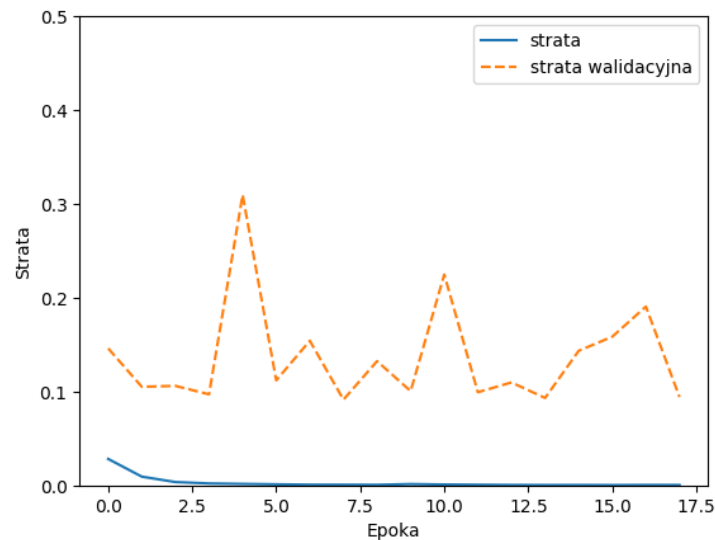
Kolejny krok w ocenie jakości sieci obejmował użycie macierzy pomyłek. W tym celu użyto funkcję *probability_model()*, która dla zbioru testowego zwróciła etykiety z prawdopodobieństwem trafności klasyfikacji. Przyjęto, że wartości powyżej lub równych 0,5 oznaczają, że osoby na zdjęciach są tej samej tożsamości, podczas gdy wartości poniżej 0,5 wskazują, że osoby te nie są podobne. Następnie otrzymane wyniki wraz z etykietami zostały przedstawione w postaci macierzy pomyłek.

$$\begin{bmatrix} T_n & N_n \\ N_p & T_p \end{bmatrix} = \begin{bmatrix} 1052 & 208 \\ 255 & 1005 \end{bmatrix}, \quad (5.9)$$

gdzie wyuczona sieć poprawnie sklasyfikowała 2057 próbek, co stanowi 81,6% wszystkich próbek w zbiorze testowym. Warto zauważyć, że błędnie sklasyfikowane pozytywne próbki stanowią 10,1% wszystkich próbek testowych. Model błędnie sklasyfikował 208 negatywnych próbek, co stanowi 8,3% zbioru testowego.

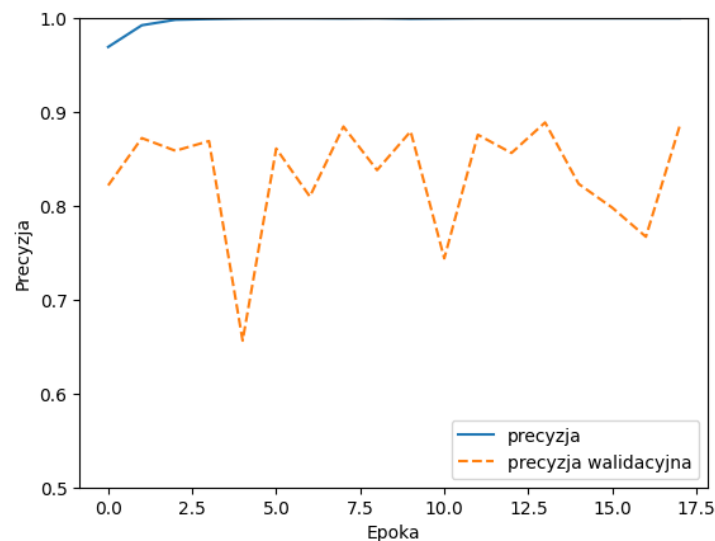
5.5.3 Własny model GAN

Zbiór treningowy zawierał łącznie 15120 próbek, z których wszystkie były sztuczne. Dane fikcyjne zostały wygenerowane przez 24 modele GAN, które mogły tworzyć zdjęcia tylko jednej postaci. Zbiory walidacyjne i testowe zawierały po 2520 rzeczywistych próbek. Rysunek 5.20 przedstawia wykres z dwoma przebiegami. Pierwszy, oznaczony jako strata, przedstawia wartość funkcji strat dla zbioru treningowego, natomiast drugi, oznaczony jako strata walidacyjna, pokazuje wartość tej funkcji dla zbioru walidacyjnego



Rysunek 5.20: Wykres wartości funkcji strat dla zbioru zawierającego fikcyjne zdjęcia

Kolejnym krokiem było stworzenie drugiego wykresu, rysunek 5.21, który przedstawiał funkcje precyzji. Pierwsza linia, oznaczana jako precyzja, reprezentuje wartość funkcji precyzji sieci dla zestawu treningowego. Druga linia, oznaczana jako precyzja walidacyjna, przedstawia wartość funkcji precyzji uzyskaną dla zestawu walidacyjnego.



Rysunek 5.21: Wykres wartości funkcji precyzji dla fikcyjnych zdjęć

Po wygenerowaniu wykresów przystąpiono do oceny efektywności modelu. W tym celu skorzystano z metody `model.evaluate()`, co pozwoliło na ocenę stworzonego modelu przy użyciu zbioru testowego. W wyniku uzyskano precyzję równą 81,63% oraz wartość straty wynoszącą 14,13%.

Kolejny krok w ocenie jakości sieci obejmował użycie macierzy pomyłek. W tym celu użyto funkcję `probability_model()`, która dla zbioru testowego zwróciła etykiety z prawdopodobieństwem trafności klasyfikacji. Przyjęto, że wartości powyżej lub równych 0,5 oznaczają, że osoby na zdjęciach są tej samej tożsamości, podczas gdy wartości poniżej 0,5 wskazują, że osoby te nie są podobne. Następnie otrzymane wyniki wraz z etykietami zostały przedstawione w postaci macierzy pomyłek.

$$\begin{bmatrix} T_n & N_n \\ N_p & T_p \end{bmatrix} = \begin{bmatrix} 1052 & 208 \\ 255 & 1005 \end{bmatrix}, \quad (5.10)$$

gdzie wyuczona sieć poprawnie sklasyfikowała 2057 próbek, co stanowi 81,6% wszystkich próbek w zbiorze testowym. Warto zauważyć, że błędnie sklasyfikowane pozytywne próbki stanowią 10,1% wszystkich próbek testowych. Model błędnie sklasyfikował 75 negatywnych próbek, co stanowi 8,3% zbioru testowego.

5.6 Porównanie uzyskanych rezultatów

W celu porównania wyników dla wszystkich wyuczonych modeli, należało wybrać odpowiedni parametr umożliwiający ocenę jakości sieci. Jako wskaźnik jakości modelu wykorzystano wartość precyzji dla zbioru testowego.

Porównanie wyników przeprowadzono na zbiorze testowym, który zawiera zdjęcia osób nieobecnych w zbiorze treningowym i walidacyjnym. Zbiór testowy składa się z 2520 zdjęć, z czego 1260 to próbki pozytywne przedstawiające te same osoby, a 1260 to próbki negatywne przedstawiające różne osoby. W pierwszym etapie porównano wyniki ewaluacji na zbiorze testowym. Model wykorzystujący jedynie rzeczywiste próbki osiągnął wartość straty wynoszącą 12,49% i precyzję 86,75%. Jak można zauważyć w tabeli 5.1, model wykorzystujący połączenie zbioru dla danych wygenerowanych przez model StyleGan2 bez etykiet uzyskał precyzję wynoszącą 88,77% oraz wartość straty 8,72% co jest lepszym wynikiem od modelu wykorzystującego jedynie rzeczywiste zdjęcia. Analogiczna sytuacja występuje dla modelu zawierającego jedynie fikcyjne zdjęcia, wygenerowane przez StyleGan2 bez etykiet. W tym przypadku wartość straty wynosi 10,18% oraz precyzja 87,06%. W pozostałych modelach wartość precyzji i straty była gorsza od pierwotnego modelu. Porównując uzyskane rezultaty można wywnioskować, że najlepsze wyniki uzyskano dla modelu wykorzystującego złączony zbiór danych rzeczywistych i fikcyjnych, wygenerowanych przez StyleGan2 bez etykiet.

Tabela 5.1: Uzyskane wyniki walidacji dla różnych typów danych

Rodzaj danych	Wartość precyzji [%]
Rzeczywiste zdjęcia	86,75
StyleGan2 z etykietami (mieszane)	82,94
StyleGan2 bez etykiet (mieszane)	88,77
Własny model GAN (mieszane)	82,10
StyleGan2 z etykietami (tylko fikcyjne)	82,38
StyleGan2 bez etykiet (tylko fikcyjne)	87,06
Własny model GAN (tylko fikcyjne)	83,73
StyleGan2 z etykietami (dotrenowanie sieci)	82,38
StyleGan2 bez etykiet (dotrenowanie sieci)	81,63
Własny model GAN (dotrenowanie sieci)	82,10

Kolejnym analizowanym parametrem była macierz pomyłek. Jak widać w macierzy (5.1), sieć używająca rzeczywiste zdjęcia poprawnie oceniła 2123 próbek, co stanowi 84,2% całego zbioru testowego. Dodatkowo, błędnie oceniono 421 próbek, co daje 15,8%. W porównaniu model wykorzystujący złączony zbiór danych rzeczywistych i próbek syntetycznych wygenerowanych przez model StyleGan2 bez etykiet poprawnie ocenił 2238 próbek,

co stanowi 88,8% całego zbioru, macierz 5.3. W tym przypadku model błędnie ocenił 282 próbek, co daje 11,2% całego zbioru. Kolejny model, wykorzystujący jedynie fikcyjne zdjęcia, wygenerowane przez StyleGan2 bez etykiet, poprawnie ocenił 2194 próbki, co stanowi 87,1% całego zbioru. Dodatkowo model błędnie ocenił 326 próbek, macierz 5.6. W pozostałych przypadkach stosunek liczby poprawnie ocenionych próbek do całego zbioru był niższy niż w modelu wykorzystującym jedynie rzeczywiste zdjęcia. Wśród wszystkich modeli, najlepsze wyniki uzyskała sieć wytrenowana na połączonym zbiorze zdjęć rzeczywistych i fikcyjnych fotografii, wygenerowanych przez StyleGan2 bez etykiet.

Z wykonanej analizy danych można wyciągnąć odpowiednie wnioski. Jak można zauważyć, model wspomagany fikcyjnymi próbkami osiąga wyższą precyzję niż model wytrenowany jedynie na rzeczywistych danych. Lepsze rezultaty uzyskano dla danych stworzonych przez model StyleGan2 bez etykiet dla syntetycznej oraz mieszanej reprezentacji danych. Analizując wyniki przedstawione w tabeli 5.1, można zauważyć, że sieć neuronowa, wytrenowana przy użyciu danych wygenerowanych przez model StyleGAN2 z etykietami, osiągnęła dokładność zbliżoną do 82%, niezależnie od zastosowanego rodzaju reprezentacji danych. Dzięki tym obserwacjom można wywnioskować, że zastosowanie StyleGAN2 z etykietami do rozszerzenia zbioru danych obniża precyzję sieci neuronowej. Kolejną zauważoną zależnością jest wpływ własnej architektury GAN na jakość sieci oceniającej podobieństwo. W tym przypadku precyzja utrzymuje się na poziomie około 82%, niezależnie od rodzaju reprezentacji danych. W tabeli 5.1 występuje anomalia, którą można zauważyć wśród modeli wytrenowanych przy wykorzystaniu StyleGan2 bez etykiet. Mianowicie, sieć neuronowa, która wykorzystuje dotrenowanie modelu osiągnęła znacznie mniejszą wartość precyzji niż pozostałe reprezentacje danych. W pozostałych modelach wartość precyzji była zależna od wykorzystanych danych do rozszerzenia zbioru, a nie samej reprezentacji. Jedną w tym przypadku tą regułą nie została spełniona.

Rozdział 6

Podsumowanie

Teza, przyjęta na początku tej pracy, została spełniona. W efekcie udało się stworzyć sieć neuronową, która po rozszerzeniu zbioru, przy wykorzystaniu modelu GAN, była w stanie uzyskać lepsze wyniki w porównaniu do sieci zawierającej jedynie rzeczywiste zdjęcia.

Model, wykorzystujący połączenie rzeczywistego zbioru treningowego z syntetycznymi próbkami wygenerowanym przez StyleGan2 bez etykiet, uzyskał precyzję o wartości 88,77%, dla porównania model wyuczony jedynie na rzeczywistych próbkach uzyskał dokładność wynoszącą 86,75%. Istotnym elementem w trakcie tworzenia nowych próbek było odpowiednie odrzucenie błędnie wygenerowanych fotografii. W celu stworzenia modelu z wyższą precyzją i niższą stratą, należałoby zwiększyć liczbę próbek poprzez wygenerowanie większej liczby zdjęć oraz poprawienie algorytmu do klasyfikacji stworzonych fotografii. Podczas rozszerzania zbioru należy pamiętać o zachowaniu odpowiedniej proporcji, tak aby cechy przedstawiane na zdjęciach się nie powtarzały.

Praca dowiodła, że rozszerzenie zbioru danych za pomocą modelu GAN ma wpływ na polepszenie się jakości uzyskanych rezultatów. Wśród testowanych modeli, najlepsze rezultaty osiągnięto dla zbioru treningowego, składającego się z rzeczywistych zdjęć oraz fotografii wygenerowanych przez StyleGan2 bez etykiet. Istotny wpływ na otrzymane rezultaty miała także odpowiednia klasyfikacja wygenerowanych zdjęć.

Jednym z bardziej istotnych problemów podczas tworzenia sieci neuronowej było wybranie odpowiedniej bazy zdjęć, która posiadała odpowiednio dużą liczbę reprezentatywnych próbek. Wśród dużej ilości zbiorów można znaleźć wiele baz niskiej jakości, które zawierały niereprezentatywne zdjęcia twarzy. Wśród nich, można było zlokalizować zdjęcia o niskiej jakości, których obraz był rozmazany. Drugim ważnym elementem podczas wybierania odpowiedniej bazy zdjęć był jej rozmiar. Większa liczba próbek umożliwiła polepszenie jakości sieci, a co za tym idzie zwiększenie jej precyzji.

Praca mogłaby być dalej rozwijana w kierunku przebadania wpływu rozszerzenia zbioru za pomocą sieci GAN na inne typy danych, takie jak rodzaj audio. Jest to stosunkowo nietrywialny problem ze względu na dużą złożoność tego typu danych. W celu rozwiązania tego problemu należałoby odszukać lub stworzyć nową bazę zawierającą pliki dźwiękowe oraz wybrać odpowiedni model, który umożliwiłby ocenę wpływu rozszerzenia zbioru. Drugim możliwym kierunkiem byłoby zbadanie wpływu różnych architektur własnych modeli GAN i porównanie rezultatów z najlepszym wynikiem uzyskanym przy użyciu StyleGAN2. Taka analiza mogłaby obejmować ocenę jakości generowanych danych, ich różnorodności oraz efektywności w treningu modeli na różnych zbiorach danych. Porównanie tych rezultatów pozwoliłoby na zidentyfikowanie najskuteczniejszych podejść i ewentualne usprawnienie procesów generacyjnych.

Załącznik A

Do pracy załączono płytę DVD zawierającą w poszczególnych katalogach:

/W4_252919_2024_praca_magisterska.pdf — wersja cyfrowa pracy,

/Rozpoznawanie_twarzy.zip — archiwum zip zawierające skrypty wykorzystane w trakcie opracowania projektu wraz z bazą zdjęć.

Literatura

- [1] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.
- [2] A. Jankowiak. Wykorzystanie sieci neuronowej do określenia podobieństwa między dwoma twarzami. Praca inżynierska, Politechnika Wrocławska, 2022.
- [3] J. McCarthy, M. L. Minsky, N. Rochester, C. E. Shannon. A proposal for the dartmouth summer research project on artificial intelligence August 31, 1955. *AI magazine*, 27(4):12–12, 2006.
- [4] A. Radford, L. Metz, S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint*, 2015.
- [5] F. Samaria, A. Harter. Parameterisation of a stochastic model for human face identification. *Applications of Computer Vision*, strony 138 – 142. Workshop on Applications of Computer Vision - Proceedings, Sarasota, FL, USA, 1995.
- [6] G. Vamsin. Wizualizacja koncepcji budowy sieci GAN. <https://c02.purpledshub.com/uploads/sites/41/2022/10/GANspreview-tb-f1fb529.jpg?w=1175&webp=1/>, 2022.
- [7] M. Veale, F. Zuiderveen Borgesius. Demystifying the draft eu artificial intelligence act—analysing the good, the bad, and the unclear elements of the proposed approach. *Computer Law Review International*, strony 97–112, 2021.
- [8] M. Xu, S. Yoon, A. Fuentes, D. S. Park. A comprehensive survey of image augmentation techniques for deep learning. *Pattern Recognition*, strony 109–347, 2023.