Raport z kursu Probabilistyczne Modele Grafowe

Wykorzystanie modeli Generatywnych dla zadania Inpainting

Dawid Kopeć, Dawid Krutul, Maciej Wizerkaniuk

260332, 263413, 260329

Spis treści

Wprowadzenie

Projekt ten skupia się na wypełnianiu luk w obrazie, aby uzyskać spójny i realistyczny efekt. Stosowana technika nosi nazwę Inpainting i znajduje szerokie zastosowanie w dziedzinach takich jak konserwacja zabytków, retuszowanie fotografii oraz usuwanie niechcianych obiektów ze zdjęć. Brakujące fragmenty mogą wynikać z uszkodzeń, cenzury lub innych czynników, które wpływają na integralność obrazu. Naszym celem jest rekonstrukcja brakujących informacji, zachowanie kontekstu obrazu oraz uzyskanie naturalnego wyglądu ostatecznej rekonstrukcji.

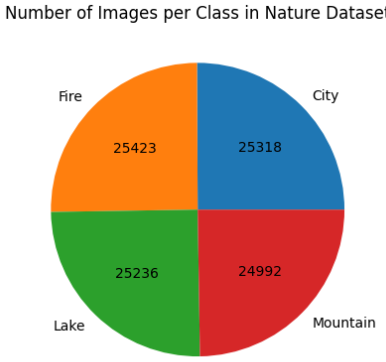
1. Eksploracyjna analiza danych

1.1 Podstawowe informacje

W celu realizacja zadania wybrany został zbiór danych Nature. Zbiór ten posiada po 100 969 zdjęć o rozdzielczościach 64x64 oraz 128x128 pikseli. Na potrzeby projektowe wybrane zostały tylko obrazy o rozdzielczości 64x64, z powodu mniejszego rozmiaru, który redukuje potrzebną złożoność obliczeniową i pamięciową. Przykładowe zdjęcia pokazane zostały na rysunku Rys. 1.1.

Rys. 1.1: Przykładowe zdjęcia ze zbioru danych Nature.

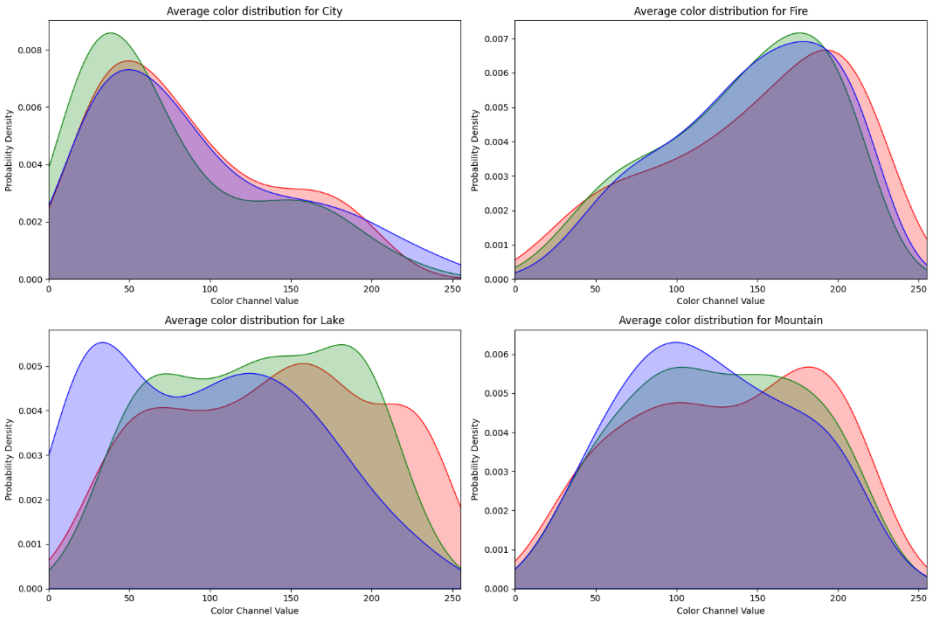
Zdjęcia stanowią klatki z czterech filmów, które dzielą je równomiernie na cztery klasy: Miasto, Góry, Ogień i Jezioro (Rys.1.2). Zbiór danych jest kompletny i nie zawiera brakujących ani uszkodzonych zdjęć. Wszystkie obrazy są w kolorze i zapisane w skali RGB.



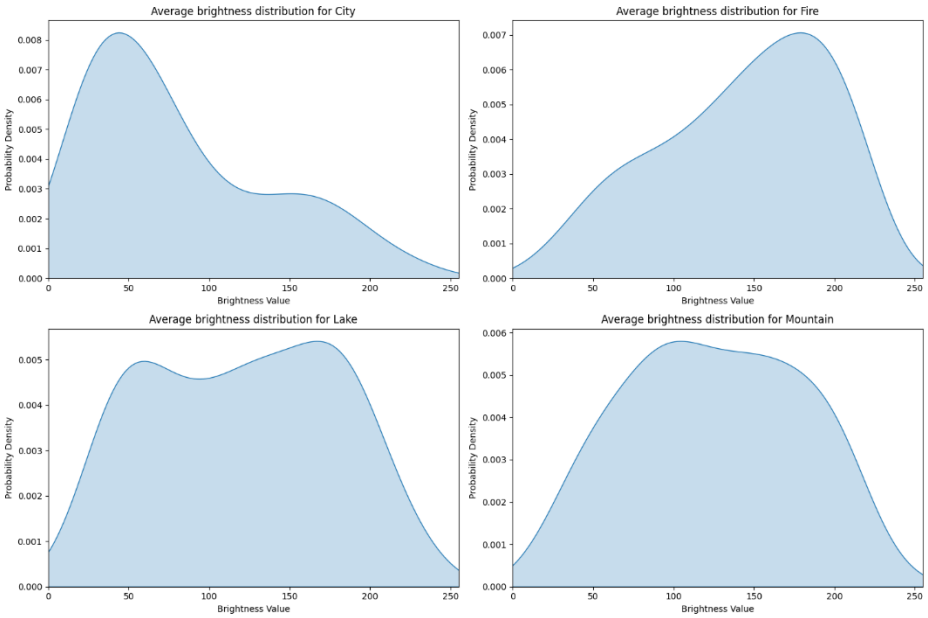
Rys. 1.2: Rozkład klasy w zbiorze danych Nature.

1.2 Pogłębiona Analiza

W celu analizy jednorodności zbioru pod względem kolorów obrazów został wykreślony rozkład kolorów zdjęć według klas, który pokazany został na rysunku Rys. 1.3. Jak można na nim zaobserwować wszystkie klasy zdjęć posiadają pełne spektrum kolorystyczne, od 0, koloru czarnego do wartości 255, koloru białego. Wszystkie kanały posiadają podobny rozkład, co sugeruje o tym, że zdjęcia są kolorowe i nie posiadają, żadnych skrajnych i dominujących barw, jest to dobra informacja w kontekście zadania Inpaintingu, ponieważ model będzie uczony lepszej generalizacji kolorów. Dodatkowo na rysunku Rys. 1.4 zaobserwować można rozkład jasności dla każdej z klas. Zaobserwować na nim można, że dla klasy miasta dominują jasne odcienie kolorów, natomiast dla ognia dominują odcienie ciemne. Klasy jezioro oraz góry posiadają równy rozkład jasności dla całego spektrum odcienie. Taki rozkład także stanowi dobre uwarunkowanie dla modeli, ponieważ będą one zmuszone nauczyć się pełnego spektrum.



Rys. 1.3: Rozkład kolorów zdjęć według klas.



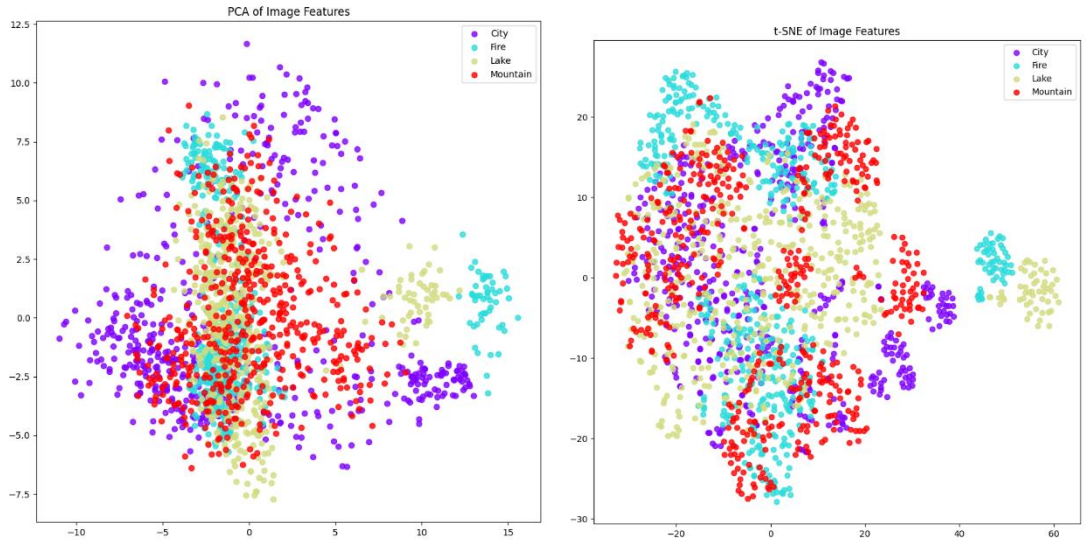
Rys. 1.4: Rozkład jasności zdjęć według klas.

Aby móc w pełni zrozumieć strukturę oraz różnorodność zbioru danych przeprowadzona analiza reprezentacji zdjęć. Ponieważ proces ten jest czasochłonny zdecydowano się ograniczyć badany zbiór do pierwszych 500 zdjęć dla każdej klasy. Analiza podzielona została na trzy etapy:

- Wykorzystanie wstępnie wytrenowanego modelu ResNet18 do wyodrębnienia wielowymiarowych wektorów cech ze zbioru danych obrazów.
- Zastosowanie technik redukcji wymiarowości, takich jak PCA i t-SNE, w celu wyodrębnienia kluczowych informacji wizualnych.
- Wygenerowanie wizualizacji rzutowanych podprzestrzeni danych.

Na podstawie wykresów PCA i t-SNE reprezentacji obrazów pokazanych na rysunku Rys. 1.5 można zauważyć, że:

- Obrazy klas jezioro i góry są wyraźnie podzielone na grupy, co sugeruje, że reprezentacje są w stanie wychwycić ich unikalne właściwości wizualne,
- Obrazy klas miasto i ogień nakładają się na siebie, co wskazuje posiadają one podobne cechy wizualne,
- Klasa ogień posiada wartości odstające, które mogą stanowić nietypowe obrazy,
- Rozkład punktów różni się w zależności od klasy, odzwierciedlając różne poziomy różnorodności w ramach każdego typu obrazu.



Rys. 1.5: Prezentacja reprezentacji ukrytych zdjęć przy użyciu algorytmów PCA (po lewej) oraz t-SNE (po prawej).

Ostatecznie w ramach EDA przeprowadzona została analiza wewnątrz klasowej korelacji zdjęć, a jej wyniki przedstawia tabela Tab. 1.1. Zawarto w niej podstawowe miary korelacji oraz jako wysoce pozytywnie skorelowane obrazy uznano obrazy o korelacji powyżej 0.75, a jako wysoce negatywnie skorelowane, obrazy o korelacji poniżej -0.75. Na podstawie tabeli można wysunąć dwa kluczowe wnioski, po pierwsze cały zbiór danych nie jest zauważalnie skorelowany, co jest pozytywną informacją w kontekście dalszej pracy oraz treningu modeli. Drugim wnioskiem jest największa liczba wysoce dodatnio i wysoce ujemnie skorelowanych obrazów znajdujących się w klasie ogień, jest to sensowny wynik, który można łatwo uargumentować tym, że obrazy stanowią klatki z filmu w którym kamera stoi nieruchomo.

Tab. 1.1: Korelacja wewnątrzklasowa obrazów.

	Średnia Korelacja	**Maksymalna Korelacja**	**Minimalna Korelacja**	**Odchylenie Standardowe**	**Procent wysoce pozytywnie skorelowanych obrazów**	**Procent wysoce negatywnie skorelowanych obrazów**
Miasto	0.035340	0.841251	-0.693326	0.192562	0.010702	0.000000
Ogień	0.011917	0.925900	-0.861226	0.279317	0.245383	0.041279
Jezioro	0.030966	0.780223	-0.729818	0.190046	0.001529	0.000000
Góry	0.027057	0.686925	-0.549062	0.138047	0.000000	0.000000

1.3 Przygotowanie i obróbka danych

W celu przygotowania danych został przygotowany skrypt, który miał na celu podzielenie danych na zbiory treningowe, walidacyjne oraz testowe w skali 60:20:20 względem oryginalnej liczby danych oraz znormalizowanie obrazów do wartości [-1, 1], zgodnie z dobrą i powszechnie wykorzystywaną praktyką, która ma na celu polepszyć proces uczenia modeli. Aby zachować równowagę klas, podział został dokonany na każdej klasie osobno, a następnie wybrane obrazy z każdej klasy zostały połączone w kolejne zbiory. Obrazy te są zapisywane w folderze *divided_x64* i w dalszej części projektu, posłużą jako zbiór danych przechowujący dane oryginalne, do których chcemy dążyć.

Po rozdzieleniu obrazów na trzy zbiory, obrazy zostały sztucznie zniszczone poprzez naniesienie na obraz czarnego prostokąta, które modele będą miały za zadanie odtworzyć. Jest on nanoszony w losowym miejscu na obrazie, pod losowym kątem nachylenia, a jego wielkość nie wykracza poza jedną czwartą wielkości całego obrazu. Tak zaugmentowane obrazy zostały następnie zapisane do folderu *augmented_x64* i w dalszej części projektu, posłużą jako zbiór danych przechowujący dane zniszczone, do których chcemy naprawić.

Oprócz zapisania obrazów w formacie jpg, zbiory zostały zapisane w formacie listy i zapisane w pliku typu *pickle* w celu łatwego ładowania obrazów, zarówno lokalnie, jak i w trakcie korzystania z platform umożliwiających pisanie kodu typu *Google Colab* czy *Kaggle* C*ode*.

2. Modele

Inpainting zdjęć stanowi wymagające zadanie dla modeli uczenia maszynowego. Aby skutecznie zrekonstruować brakujące detale, model musi zrozumieć kontekst obrazu, zależności między elementami oraz posiadać zdolność generowania realistycznych fragmentów, które pasują do całości.

Do inpaintingu zdjęć najlepiej nadają się modele generatywne, które potrafią tworzyć nowe dane o podobnej charakterystyce do danych treningowych. Te modele uczą się wzorców i zależności w danych, a następnie wykorzystują tę wiedzę do generowania nowych obrazów, które są spójne i estetyczne.

2.1 Wariacyjny Autokoder

Model VAE (Variational Autoencoder), początkowo zaprojektowany do generowania nowych obrazów, okazał się niezwykle skuteczny także w zadaniach takich jak inpainting zdjęć. Dzięki unikalnej architekturze i właściwościom, jest idealnym narzędziem do rekonstrukcji brakujących fragmentów obrazów, zapewniając realistyczne i estetyczne rezultaty. Architektura typu koder-dekoder umożliwia modelowi skuteczne uchwycenie kontekstu obrazu poprzez reprezentację jego cech w przestrzeni ukrytej.

Jedną z kluczowych zalet VAE jest jego probabilistyczne podejście do reprezentacji danych. Zamiast mapować dane wejściowe na pojedynczy punkt, VAE koduje je jako rozkłady probabilistyczne w przestrzeni ukrytej. Pozwala to na bardziej elastyczne i złożone modelowanie struktury danych. Dekoder przekształca te rozkłady w obrazy, umożliwiając generowanie różnych realistycznych wariantów obrazu na podstawie tej samej reprezentacji ukrytej.

Największym atutem architektury VAE jest wprowadzenie rekuraryzacji poprzez dodanie terminu dywergencji Kullbacka Leiblera do funkcji straty. Ta rekuraryzacja pomaga modelowi nauczyć się bardziej uporządkowanej i ciągłej przestrzeni ukrytej, co jest kluczowe dla generowania realistycznych obrazów. Dzięki temu VAE może skutecznie odtworzyć uszkodzony obraz już po kilku epokach treningu.

Model VAE jest również prosty w budowie i posiada wiele wariantów. Dzięki swojej elastyczności może być dostosowany do różnych zadań poprzez zmianę liczby warstw kodera i dekodera oraz liczby kanałów. Funkcja straty w VAE składa się z dwóch części: błędu rekonstrukcji (często używany jest błąd średniokwadratowy) oraz odchylenia KL, co zapewnia efektywne uczenie modelu.

2.2 U-Net

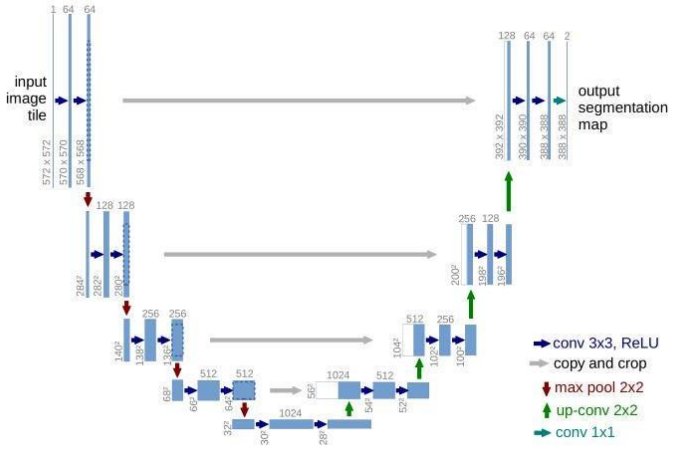
Model U-Net, opracowany pierwotnie do segmentacji obrazów medycznych, okazał się niezwykle skuteczny również dla zadań takich jak inpainting zdjęć. Jego unikalna architektura i właściwości czynią go idealnym narzędziem do rekonstrukcji brakujących fragmentów obrazów, zapewniając realistyczne i estetyczne rezultaty. Dzięki swojej architekturze koder-dekoder model ten jest w stanie skutecznie wydobyć kontekst obrazu ze stworzonej reprezentacji jego cech.

Kolejną zaletą modelu U-Net jest fakt, iż posiada on warstwy konwolucyjne w koderze oraz dekoderze. Rosnąca liczba filtrów w koderze pozwala mu na wychwycenie zarówno drobnych szczegółów jak i bardziej ogólnej struktury obrazu. Natomiast zwiększające się

rozmiary filtra w dekoderze pozwalają na stopniowe odbudowywanie obrazu i precyzyjne wypełnienie brakujących elementów.

Największą jednak zaletą architektury U-Net, która odróżnia go od prostego dekodera jest stosowanie przejść (ang. *skip connections*) między warstwami. Dzięki nim model jest w stanie zachować spójność kolorów oraz właściwości obraz, zapewniając naturalne uzupełnienie. Pozwala to także na odtworzenie uszkodzonego obrazu już po kilku epokach uczenia.

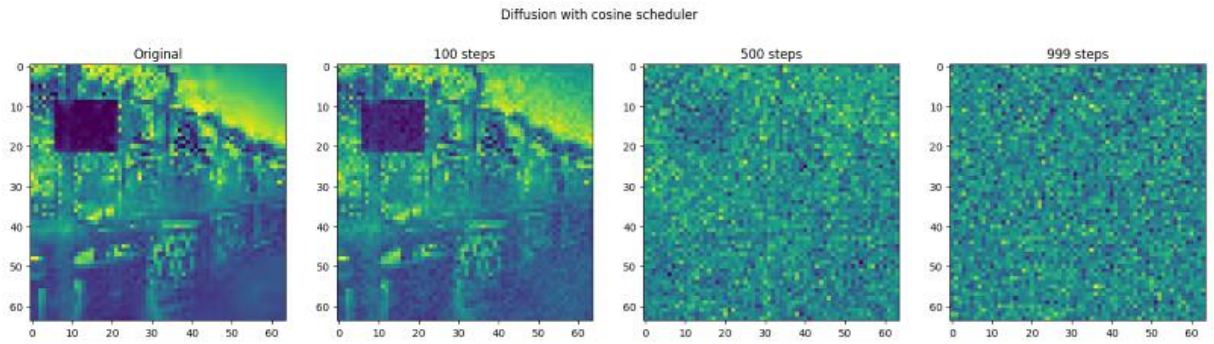
Sam model jest prosty w budowie oraz posiada wiele opracowań. Dodatkowo dzięki prostocie jego implementacji może on być elastyczny i posiadać sparametryzowaną liczbę bloków kodera oraz dekodera, a także liczbę kanałów. Funkcję starty w modelu stanowi błąd średniokwadratowy. Przykładową architekturę modele przedstawia rysunek Rys. 2.1.



Rys. 2.1: Architektura modelu U-Net [1].

2.3 DDPM

Modele DDPM (Denoising Diffusion Probabilistic Models) zostały stworzone na potrzeby generowania danych w inny sposób niż wcześniej wykorzystywane modele generatywne np. modele GAN (Generative Adversarial Network) w celu uniknięcia ich błędów oraz problemów, które pojawiają się zarówno w trakcie uczenia jak i generowania danych. Modele Dyfuzyjne działają poprzez stopniowe dodawanie szumu do danych, a następnie uczenie procesu odszumiania i rozkładu danych. Warto również wspomnieć, że w procesie dyfuzji, często wykorzystywane są łańcuchy Markova w celu symulacji procesu oraz predykcji oryginalnych, niezaszumionych danych.



Rys. 2.2: Proces zaszumiania obrazu ze zbioru treningowego.

W trakcie procesu odszumiania, obecny jest również model U-Net, który ma za zadanie przewidzieć jak zachowuje się szum na każdym kroku dyfuzji. Model ten bierze za wejście zaszumiony obraz oraz informację o kroku czasowym i przewiduje oryginalny szum, który został dodany do obrazu. Usunięcie tego szumu z zaszumionego obrazu pozwala na stopniowe odtwarzanie oryginalnych danych z ich zaszumionej wersji lub wygenerowanie danych na podstawie losowego szumu.

W przypadku tradycyjnego zastosowania modelu dyfuzyjnego, to znaczy w przypadku generowania obrazów podobnych do tych na których podstawie model został wyuczony, w celu uzyskania obrazu podaje się losowy szum lub zaszumioną wcześniej wersję obrazu. Jednakże w zadaniu Inpaintingu, znana jest już część obrazu, którą chcemy zachować. Z tego powodu w trakcie uczenia modelu DDPM, do modelu U-Net zostało podane połączenie zaszumionej wersji oryginalnego obrazu wraz z wybrakowanym zdjęciem, a w trakcie inferencji połączenia losowego szumu z wybrakowanym zdjęciem, które chcemy uzupełnić. Dzięki temu rozwiązaniu model generuje od zera jedynie brakujące informacje, przy jednoczesnym zachowaniu niezmienionej, oryginalnej części obrazu.

Należy jednak pamiętać, że modele dyfuzyjne wymagają znacznie więcej czasu oraz zasobów w celu nauki, jak również generowania danych, niż prostsze modele. Oferują one jednak wysoki poziom jakości danych, stabilniejsze działanie oraz poprawę przy danych, które zdarzają się być zbyt trudne do nauki dla innych, tradycyjnych modeli generatywnych.

3. Eksperymenty

Każdy z modeli trenowany był na całości zbioru treningowego oraz walidowany co określoną ilość epok na zbiorze walidacyjnym. W celu otrzymania miarodajnych wyników każdy z wyuczonych modeli poddawany był ewaluacji na zbiorze testowym. Z powodu małych zasobów obliczeniowych oraz złożoności modeli zdecydowano się, na wykorzystywanie możliwie najmniejszych architektur, które zapewniały zadowalające wyniki. Dodatkowo w ramach ostatecznego porównania każdy z modeli uczony był 30 epok.

3.1 Trening modeli

3.1.1 Metryki jakości

W celu ewaluacji oraz walidacji modeli wykorzystane zostały cztery metryki jakości, które są odpowiednie dla zadnia inpainting'u obrazów. Warto dodać, że klasyczne metryki jakości takie jak dokładność lub miara f1 były nieadekwatne dla tego zadnia. Metryki, na które się zdecydowano to:

- MSE (Mean Squared Error) – błąd średniokwadratowy, mierzy średnią różnicę między pikselami obrazu oryginalnego a wygenerowanego. Jest to prosta i łatwa do obliczenia metryka, jednakże nie jest w pełni miarodajna,
- NRMSE (Normalized Root Mean Square Error) – znormalizowany błąd średniokwadratowy jest to udoskonalona wersja klasycznego błędu średniokwadratowego, która normalizuje wartości błędu do zakresu [0, 1], ułatwiając interpretację i porównywanie wyników.
- PSNR (Peak Signal-to-Noise Ratio) – mierzy stosunek mocy sygnału do mocy szumu w obrazie wygenerowanym. Jest to logarytmiczna skala, a wyższe wartości PSNR oznaczają lepszą jakość obrazu.
- SSIM (Structural Similarity Index Measure) – mierzy podobieństwo strukturalne między obrazem oryginalnym a wygenerowanym. Bierze pod uwagę nie tylko różnice w jasności pikseli, ale także ich lokalną strukturę i teksturę.

3.1.2 Wyniki VAE

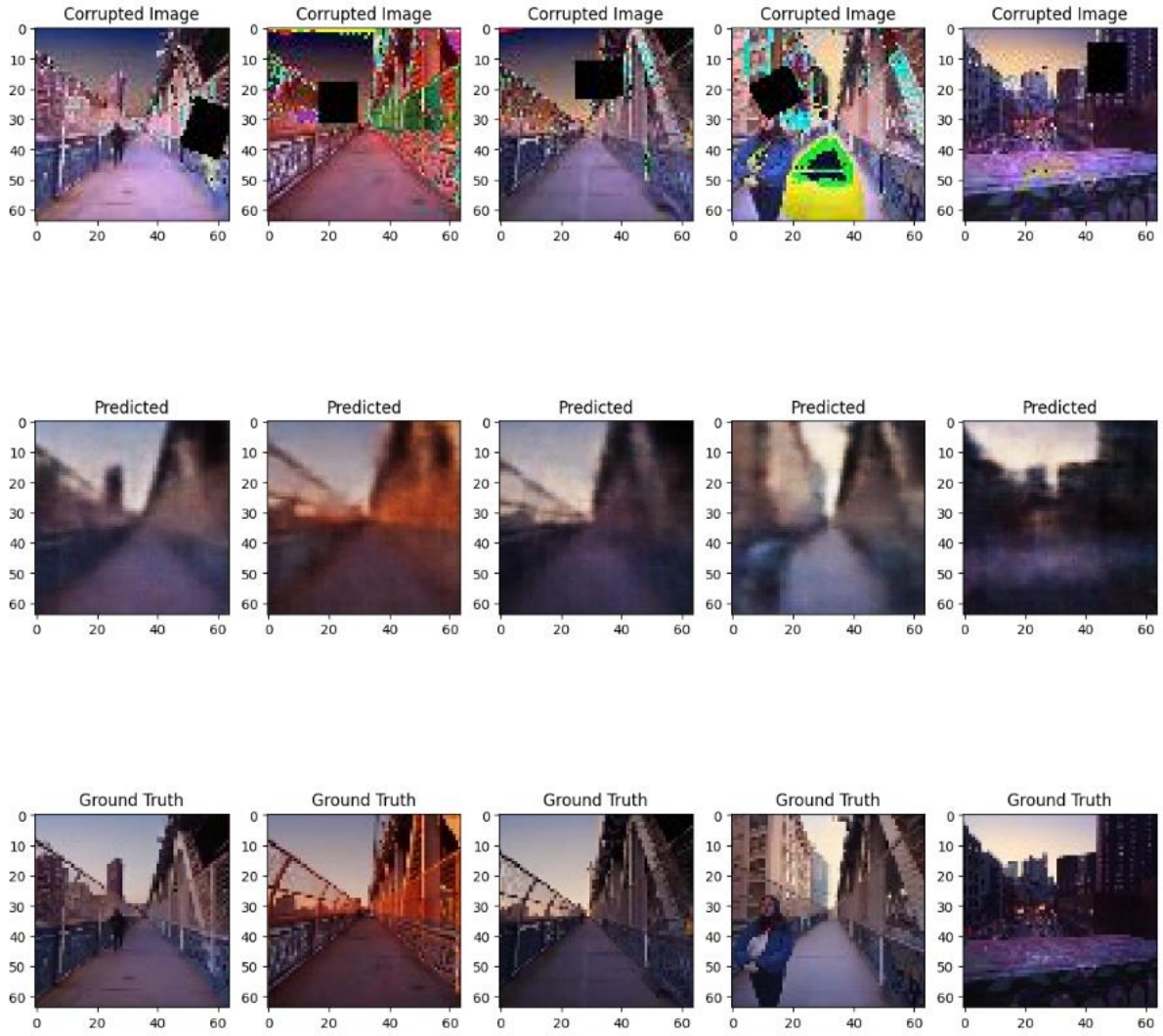
Model VAE bardzo dobrze się trenował szczególnie w pierwszych epokach, w późniejszych momentach różnice były coraz trudniej zauważalne ludzkim okiem, jedynym odniesieniem w późniejszych epokach była wartość funkcji straty, która co raz mniej malała z każdym krokiem nauki. Podsumowując zwiększenie liczby epok ponad 30 dałoby jeszcze lepsze rezultaty.

Poniżej znajdują się wizualizacje z ostatniej epoki nauki jak i same wartości metryk jakie model osiągnął pod koniec treningu (Tab. 3.1 / Rys. 3.1). Model VAE ma jak najbardziej spory potencjał w zadaniu odtwarzania zdjęć. Zważając na ograniczone zasoby obliczeniowe, model poradził sobie z zadaniem przyzwoicie.

Tab. 3.1: Wyniki ewaluacji wytrenowanego modelu VAE.

	MSE	NRMSE	PSNR	SSIM
max	0.026251	0.162021	28.520228	0.929877
średnia	0.007474	0.087733	21.622611	0.623723
min	0.001298	0.037496	15.808594	0.313435

Rys 3.1: Wyniki ewaluacji wytrenowanego modelu VAE.



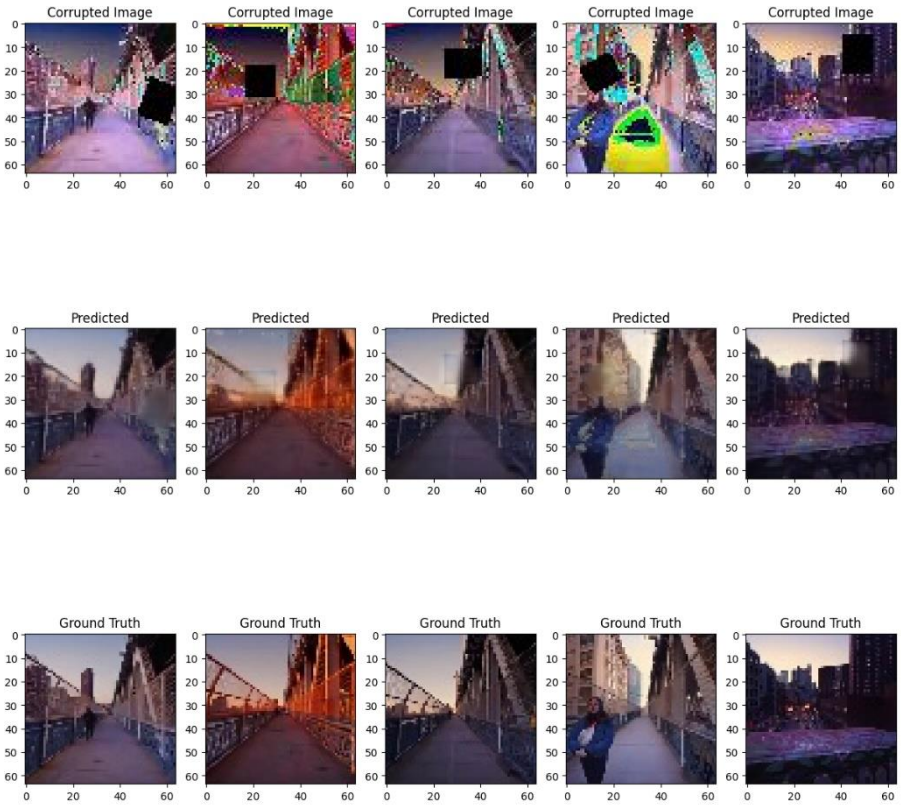
3.1.3 Wyniki U-Net

Model U-Net w trakcie całego treningu zaliczał progres, a większa liczba epok mogłaby poprawić skutecznie jego wyniki. Wyniki ewaluacji modelu na danych testowych zostały przedstawione w tabeli Tab. 3.2 oraz na rysunku Rys. 3.2. W tabeli przedstawione zostały opisane w podpunkcie 3.1 pracy, metryki jakości, które są w trzech wariantach: max – największa wartość metryki, średnia – średnia wartość metryki oraz min – minimalna wartość metryki. Najlepsze wartości metryk (min dla MSE i NRMSE oraz max dla PSNR i SSIM) mówią, że najlepiej wygenerowane przez model obrazy były niemal identyczne z nieuszkodzonymi oryginałami. Jeżeli natomiast chodzi o wartości najgorsze, są one wciąż zadowalające i dobre. Ostatecznie najważniejsza z miar, średnia, prezentuje satysfakcjonujące

wyniki. PSNR wynoszące prawie 24 oraz SSIM wynoszące ponad 0.83 to wyniki bardzo dobrej jakości obrazu. Efekt wizualny wyników modelu przedstawiony na rysunku Rys. 3.2, który przedstawia losowe trójki zdjęć ze zbioru testowego: *zdjęcie uszkodzone*, *zdjęcie wygenerowane* oraz *oryginalne zdjęcie nie uszkodzone*. Jak widać na rysunku zamalowany kwadrat jest widoczny w postaci łaty oraz nie wszystkie wypełnienia są poprawne, jednakże w większości przypadków uszkodzony obszar został skutecznie degenerowany i można uznać go za zadowalający.

Tab. 3.2: Wyniki ewaluacji wytrenowanego modelu U-Net.

	MSE	**NRMSE**	**PSNR**	**SSIM**
max	0.010118	0.114511	28.867353	0.934879
średnia	0.004065	0.065525	23.969002	0.830932
min	0.000812	0.036027	18.823051	0.661782



Rys. 3.2: Wyniki ewaluacji wytrenowanego modelu U-Net.

3.1.4 Wyniki DDPM

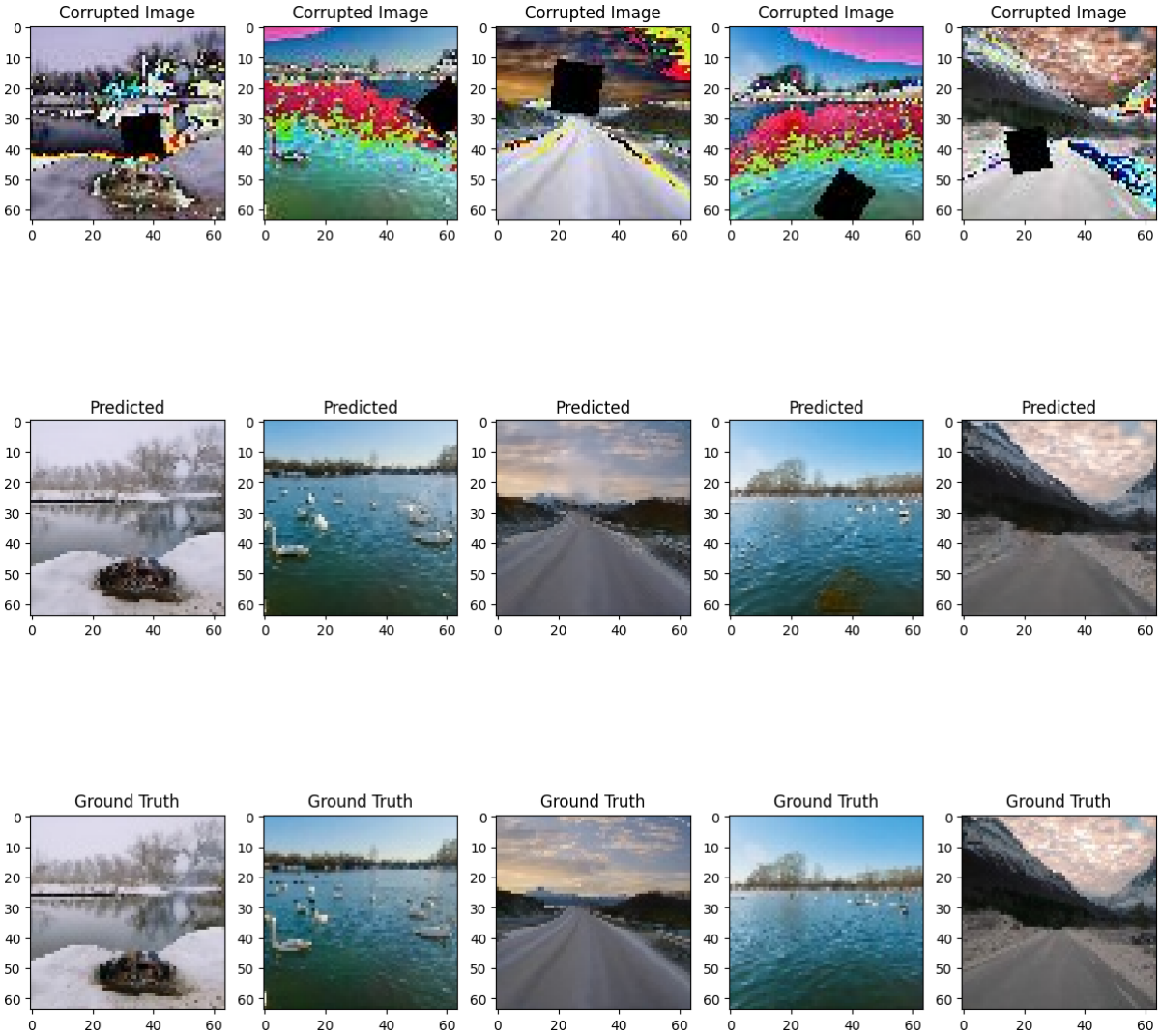
Model DDPM dobrze poradził sobie z zadaniem inpaintigu. Już na fazie pierwszych epok, odwzorowywał on w poprawny sposób obraz, jednakże w celu usunięcia halucynacji w wybrakowanych miejscach, potrzebuje on większej ilości epok, jednakże należy uważać na problem przeuczenia modelu. W trakcie testów subiektywnych na kolejnych epokach, wybrakowane miejsce było coraz trudniejsze do zlokalizowania bez obrazu referencyjnego, zwłaszcza na obrazach przedstawiających ognisko czy jezioro powyżej 5 epoki.

Tabela Tab. 3.3 oraz rysunek Rys 3.3 przedstawiają wyniki ewaluacji metryk. Analizując wynik można zauważyć, bardzo wysoki wynik dla wszystkich badanych metryk, co wskazuje

na wysoki poziom rekonstrukcji obrazów. Posiada on jednak wadę w postaci czasu potrzebnego na uczenie, który rośnie wraz ze zwiększaniem liczby parametrów modelu oraz liczby epok.

Tab. 3.3: Wyniki ewaluacji wytrenowanego modelu DDPM.

	MSE	NRMSE	PSNR	SSIM
max	0.003464	0.059797	30.974572	0.963996
średnia	0.001839	0.043395	27.529949	0.889963
min	0.000702	0.028266	24.466451	0.840695



Rys. 3.3: Wyniki ewaluacji wytrenowanego modelu DDPM.

3.2 Badanie Hiper parametrów

Najważniejsze oraz wspólne hiper parametry modeli stanowią parametry architektury, liczba bloków oraz liczba kanałów. Z tego powodu dla każdego z modeli dokonano przeglądu przykładowych ilości bloków kodera oraz dekodera, a także początkową liczbę kanałów kodera, która ma bezpośredni wpływa na wielkość oraz liczbę parametrów w sieci.

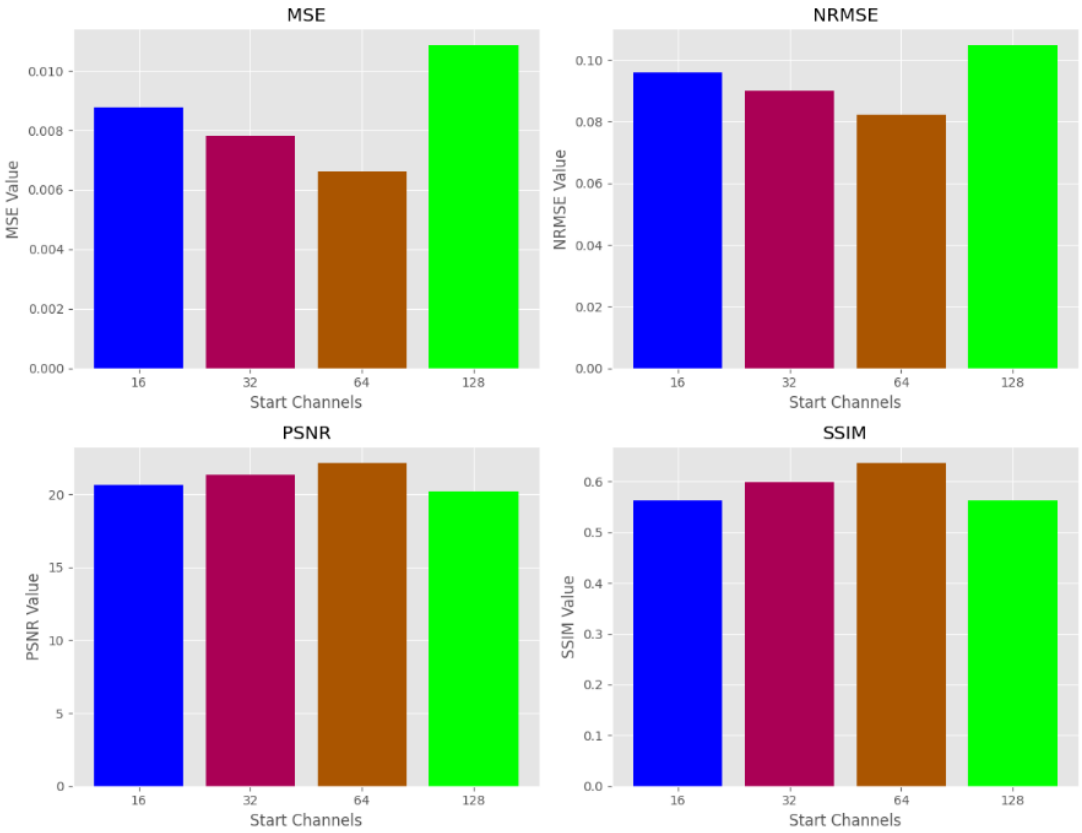
3.2.1 VAE

Proponowana architektura VAE została oceniona pod kątem wpływu dwóch głównych hiper parametrów: liczby początkowych kanałów (start channels) oraz liczby bloków (num blocks) na jakość rekonstrukcji zdjęć. Wyniki tych testów prezentują rysunki Rys. 3.4 i Rys. 3.5 oraz tabele Tab. 3.4 i Tab. 3.5, które zawierają średnie, minimalne i maksymalne wyniki miar uzyskanych w wyniku ewaluacji modeli na danych testowych.

Każdy model był uczony przez 30 epok. W przypadku liczby początkowych kanałów, najlepsze wyniki osiągnął model posiadający 64 kanały, choć wyniki modelu z 32 kanałami były zbliżone. Model z 128 kanałami uzyskał najgorsze rezultaty, co sugeruje, że zbyt mała liczba parametrów ogranicza zdolność modelu do uchwycenia skomplikowanych zależności w obrazach.

Podobne testy przeprowadzono dla liczby bloków. Wyniki tych testów prezentują rysunki Rys. 3.5 oraz tabela Tab. 3.5. Najlepsze rezultaty uzyskał model z 5 blokami, chociaż model z 4 blokami osiągnął wyniki zbliżone do najlepszego. Model z 2 blokami osiągnął najgorsze wyniki, co wskazuje na zbyt małą głębokość modelu, która nie pozwala na odpowiednie uchwycenie złożoności danych.

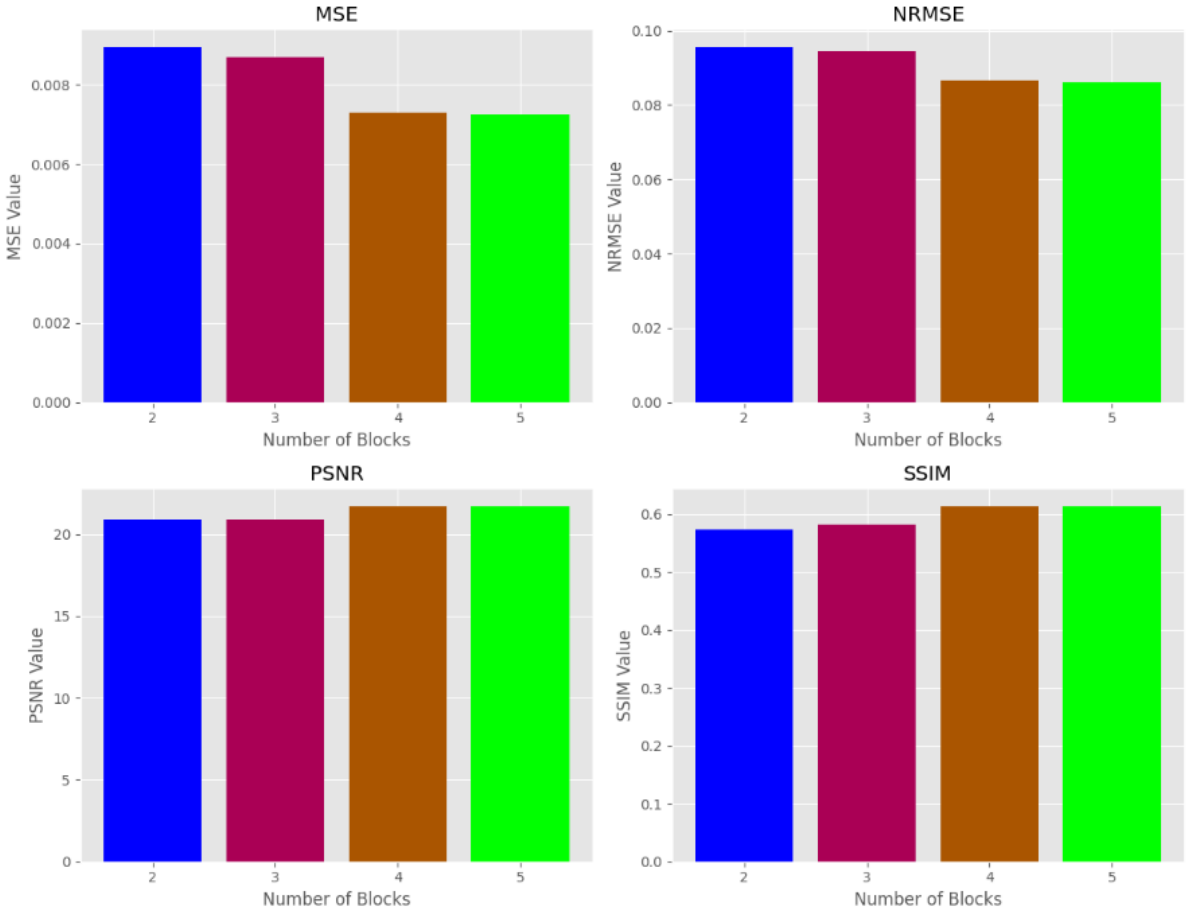
Podsumowując, wyniki testów pokazują, że zarówno liczba początkowych kanałów, jak i liczba bloków mają istotny wpływ na jakość rekonstrukcji zdjęć przez model VAE, przy czym optymalne wartości dla tych parametrów to 64 początkowych kanałów i 5 bloków.



Rys. 3.4: Wyniki badania liczby start channels dla architektury VAE.

Tab. 3.4: Wyniki badania liczby start channels dla architektury VAE.

Liczba kanałów	MSE	NRMSE	PSNR	SSIM
16	0.008776	0.095969	20.657512	0.562164
32	0.007827	0.090037	21.314198	0.599257
64	0.006617	0.082151	22.139926	0.637043
128	0.010866	0.104804	20.177577	0.562489



Rys. 3.5: Wyniki badania liczby bloków dla architektury VAE.

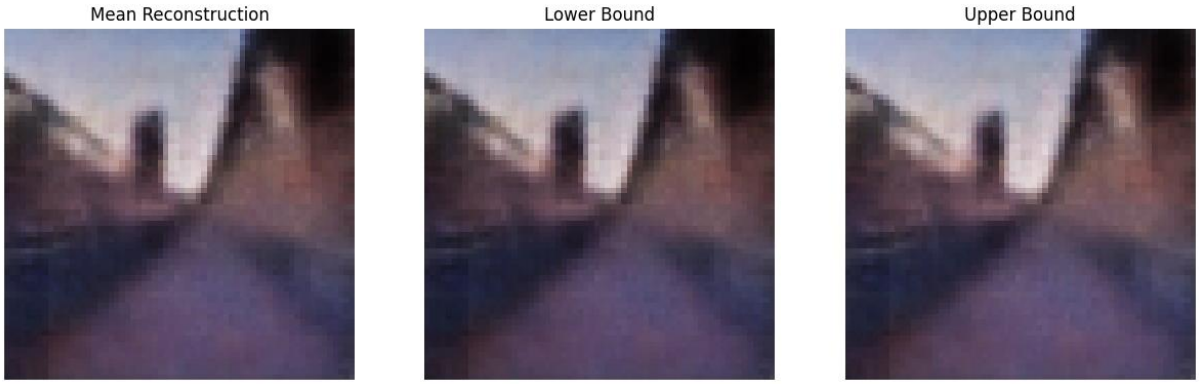
Tab. 3.5: Wyniki badania liczby bloków dla architektury VAE.

Liczba kanałów	MSE	NRMSE	PSNR	SSIM
2	0.008941	0.095561	20.887827	0.574276
3	0.008675	0.094571	20.923562	0.581538
4	0.007304	0.086513	21.698724	0.613535
5	0.007230	0.086167	21.686541	0.613644

3.2.2 Credible Intervals

Przedziały wiarygodności (credible intervals) to przedziały, w których z określonym prawdopodobieństwem znajduje się rzeczywista wartość parametru, zgodnie z analizą bayesowską. W kontekście Wariacyjnych Autokoderów (VAE), przedziały te służą do oceny niepewności generowanych próbek. Aby sprawdzić przedziały wiarygodności w VAE,

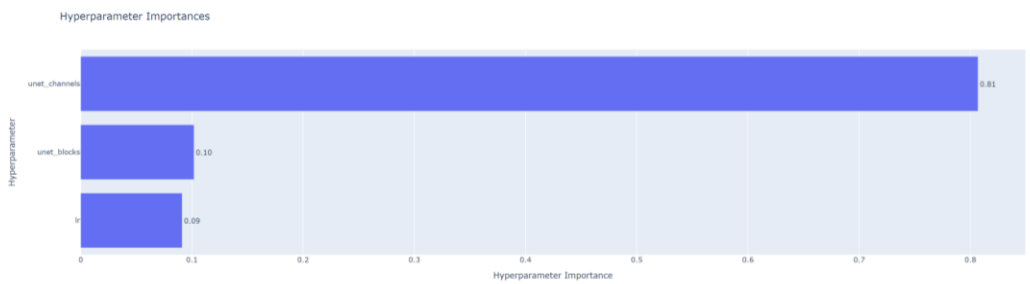
wygenerowano wiele próbek z modelu dla danego wejścia, wykorzystując kodowanie wejścia do rozkładu normalnego, a następnie dekodowanie z próbkowanej przestrzeni ukrytej. Dla uzyskanych próbek obliczono odpowiednie percentyle, tworząc przedziały wiarygodności na poziomie ufności 0.9. Taki proces umożliwia ocenę, że 90% generowanych wartości pikseli mieści się w wyznaczonych przedziałach, zapewniając zgodność z rozkładem treningowym. Wyniki badania zobrazowane zostały na rysunku Rys. 3.6. Pokazano na nim średnią rekonstrukcje z 10_000 generacji, dolną granicę, która mówi nam, że 95% pikseli będzie jaśniejsza oraz górną granice, która mówi nam, że 95% pikseli będzie ciemniejsza. MSE między górną a dolną granicą wyniosło $1.58e^{-3}$, co jest bardzo małym i zadowalającym wynikiem.



Rys. 3.6: Credible Intervals dla modelu VAE.

3.2.3 U-Net

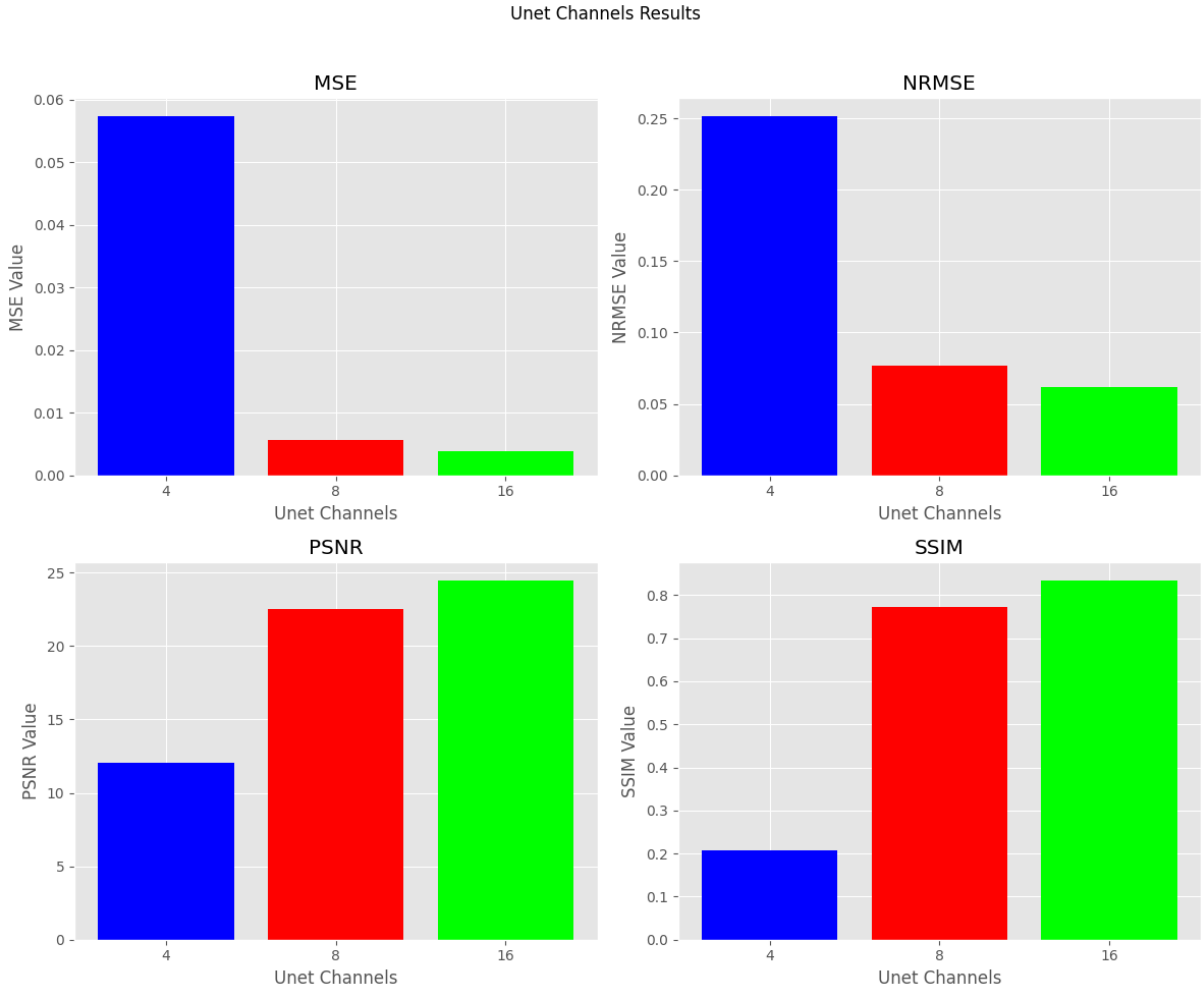
Proponowana architektura U-Net posiada jedynie trzy hiper parametry: liczbę bloków, liczbę kanałów początkowych oraz współczynnik uczenia. W ramach ich przeglądu sprawdzony został wpływ każdego z parametrów modelu, na jego wyniki. Wpływ ten prezentuje rysunek Rys. 3.7. Jak można na nim zaobserwować zdecydowanie najbardziej istotnym parametrem jest liczba kanałów, z liczbą bloków na drugim miejscu. Z tego powodu zdecydowano się je zbadać. Wynik porównania liczby kanałów można zobaczyć na rysunku Rys. 3.8 oraz w tabeli Tab. 3.6, które zawierają średnie wyniki miar uzyskanych w wyniku ewaluacji modelów na danych testowych. W porównaniu każdy z modeli posiadał 3 bloki, współczynnik uczenia równy $1e^{-3}$ oraz uczony był przez 20 epok. Zdecydowanie najgorszy okazał się model posiadający 4 kanały początkowe, powodem tego może być zbyt mała liczba parametrów oraz prostata modelu, z powodu której nie był on w stanie wyłapać skomplikowanych zależności w obrazach. Najlepszy okazał się model posiadający 16 kanałów początkowych, jednakże jego wyniki były zbliżone do modelu posiadającego 8 kanałów.



Rys. 3.7: Badanie wpływu hiper parametrów na model U-Net.

Tab. 3.6: Wyniki badania liczby kanałów dla architektury U-Net.

Liczba kanałów	MSE	NRMSE	PSNR	SSIM
4	0.057372	0.251557	12.035172	0.206664
8	0.005600	0.077106	22.483057	0.773563
16	0.003794	0.062195	24.461186	0.834091



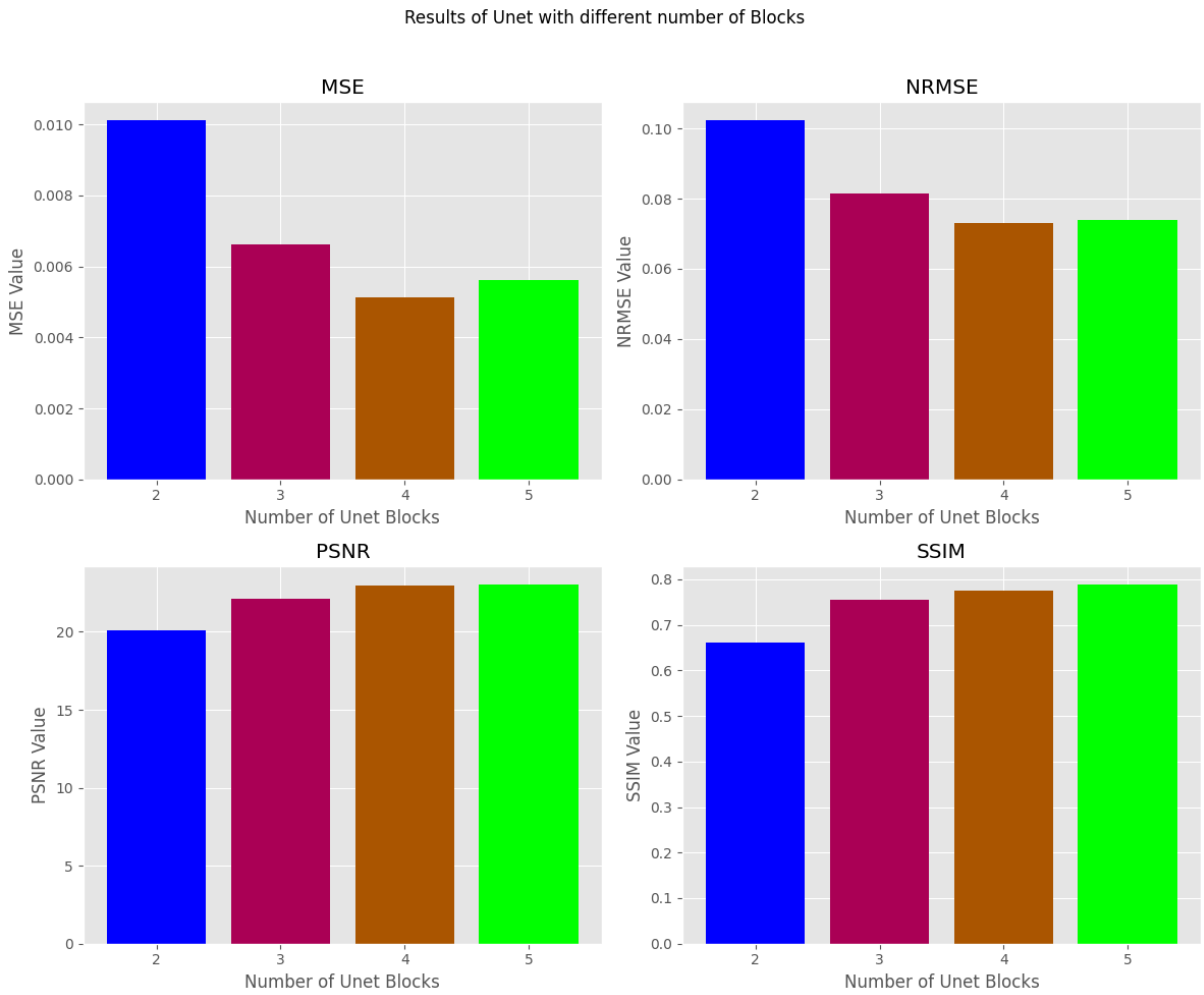
Rys. 3.7: Wyniki badania liczby kanałów dla architektury U-Net.

Kolejne przeprowadzone badanie stanowi porównanie liczby bloków kodera i dekodera (ponieważ sieć musi być symetryczna) na wyniki model. Średnie wyniki metryk uzyskanych w wyniku ewaluacji modeli na danych testowych zobaczyć można na rysunku Rys. 3.8 oraz w tabeli Tab. 3.7. Wszystkie modele posiadały 8 kanałów początkowych, współczynnik uczenia równy $1e^{-3}$ oraz uczone były przez 20 epok. Zauważalnie najgorsze wyniki uzyskał model posiadający jedynie 2 bloki, spowodowane może być to zbyt małą ilością parametrów, które nie były w stanie wygenerować zadowalającego obrazu. Warto zauważyć, że wyniki dla 3, 4 oraz 5 kanałów są do siebie zbliżone, co pokrywa się z analizą wpływu parametrów na wyniki sieci. Trudno jednoznacznie określić, która z liczb jest najlepsza na podstawie wyników, jednakże ponieważ wyniki modeli są zbliżone, a wraz ze wzrostem liczby bloków znacząco

wzrasta także liczba parametrów sieci, zdecydowano się uznać za najlepszą najprostszą z opcji, sieć o 3 blokach.

Tab. 3.7: Wyniki badania liczby bloków dla architektury U-Net.

Liczba bloków	MSE	NRMSE	PSNR	SSIM
2	0.010119	0.102368	20.068010	0.662012
3	0.006604	0.081421	22.124633	0.755946
4	0.005115	0.073131	22.974632	0.776160
5	0.005624	0.073909	23.033660	0.788653



Rys. 3.8: Wyniki badania liczby bloków dla architektury U-Net.

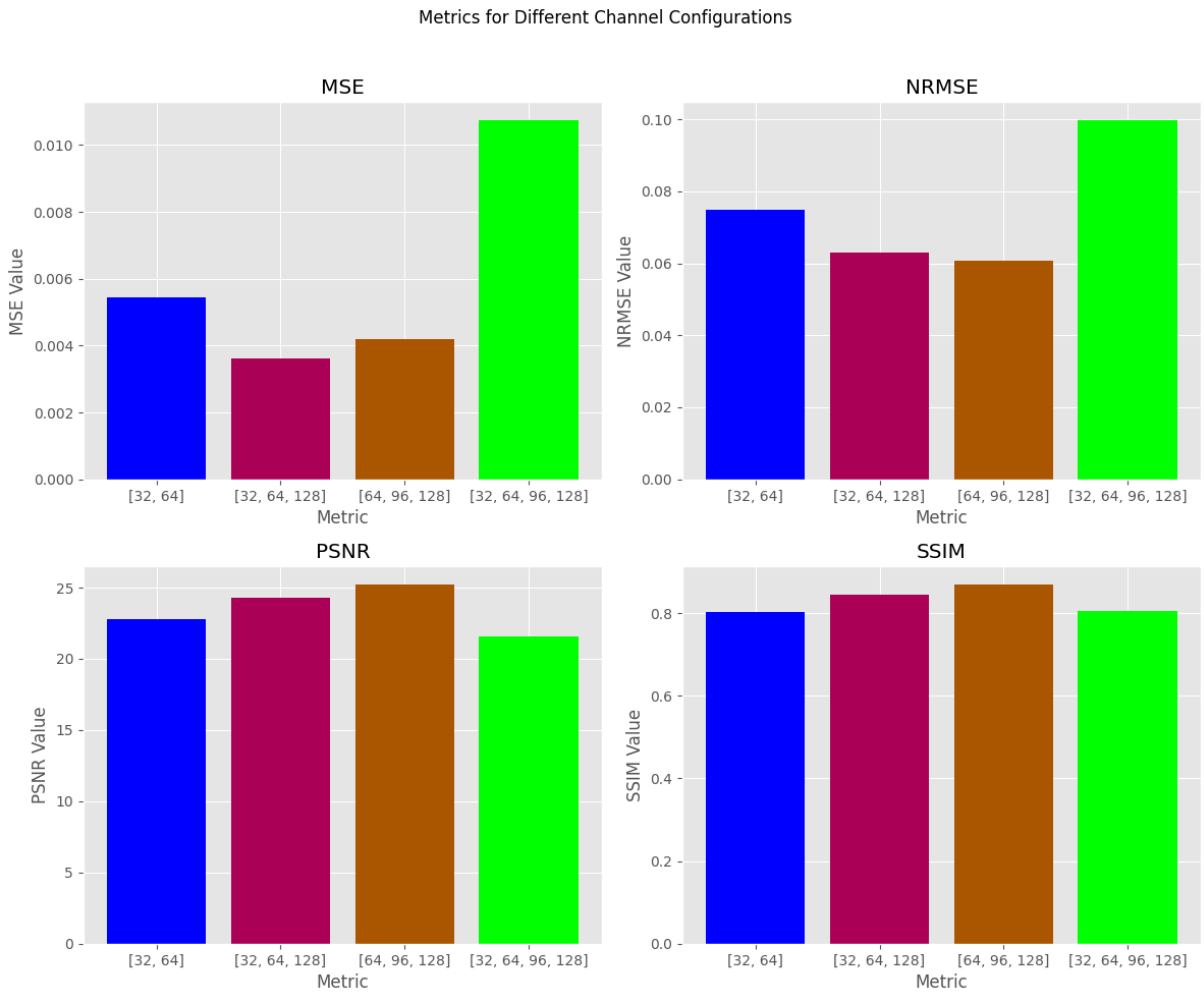
3.2.4 DDPM

Dla proponowanego modelu dyfuzji, badane były dwa hiper parametry: architektura sieci U-Net oraz sposób generowania szumu obrazu. W tym celu zostały wyuczone modele z różnymi ustawienia i przedstawione na rysunkach Rys. 3.9, Rys.3.10 oraz w tabelach Tab. 3.8, Tab. 3.9 i Tab. 3.10. Każdy model był uczony na tej samej liczbie epok również 15, liczbie kroków zaszumiania równej 1000, natomiast ze względu na bardzo długi czas ewaluacji pełnego zbioru testowego, ewaluacja odbywała się jedynie na części tego zbioru.

W przypadku zastosowania różnej architektury systemu, a co za tym idzie, rożnej liczby parametrów, najlepsze wyniki zostały otrzymane dla modelu z trzema blokami. Posiadają one zbliżone do siebie wyniki, jednakże model o architekturze [64, 96, 128] posiada lepsze wyniki dla każdego testu z wyjątkiem metryki MSE. Najgorsze wyniki otrzymał model [32, 64, 96, 128], jednakże wciąż są to zadowalające wyniki nie tylko pod względem wartości przedstawionych przez metryki, ale również przez testy subiektywne polegające na wizualnym porównaniu losowych obrazów.

Tab. 3.8: Zależność liczby filtrów i liczby parametrów

Architektura modelu U-Net	**Liczba parametrów**
[32, 64]	978851
[32, 64, 128]	3878883
[64, 96, 128]	5580131
[32, 64, 96, 128]	5475843



Rys. 3.9: Wyniki badania architektury modelu U-Net w modelu DDPM.

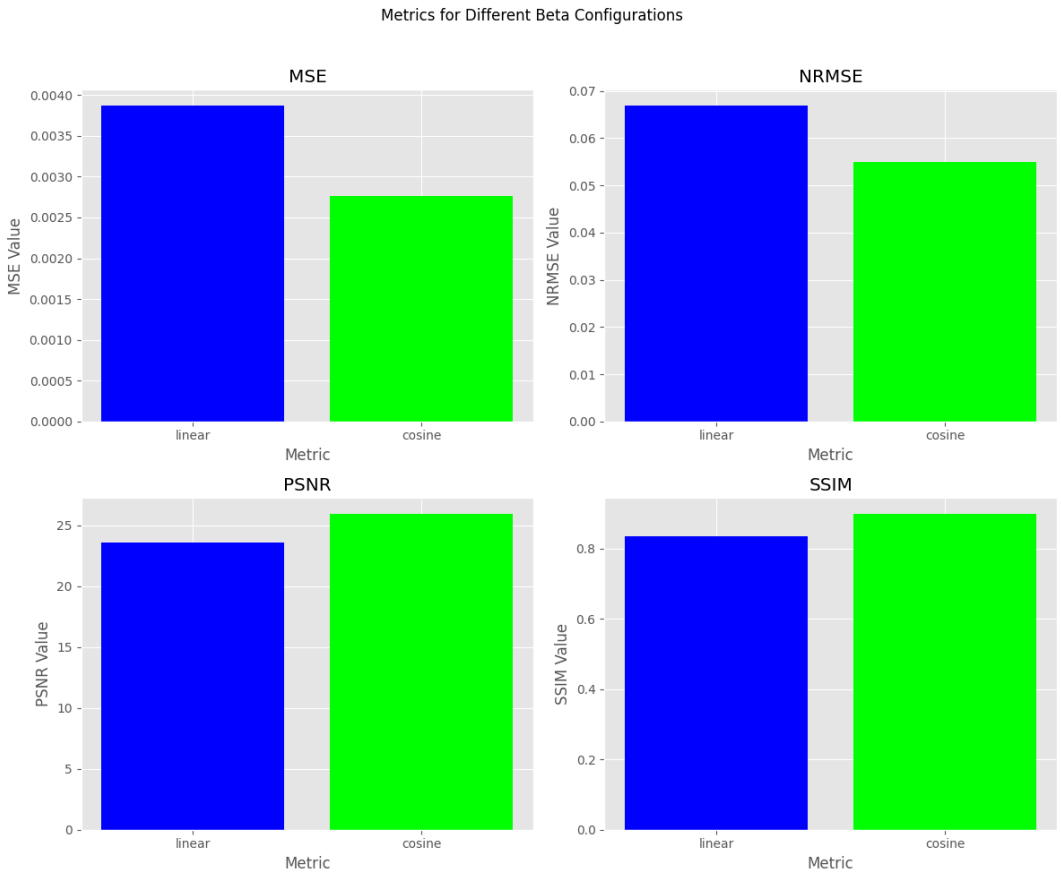
Tab. 3.9: Wyniki badania architektury modelu U-Net w modelu DDPM.

Liczba bloków	MSE	NRMSE	PSNR	SSIM
[32, 64]	0.005453	0.074738	22.755860	0.803339
[32, 64, 128]	0.003621	0.062869	24.279933	0.844038
[64, 96, 128]	0.004195	0.060709	25.220394	0.869721
[32, 64, 96, 128]	0.010742	0.099748	21.562272	0.806010

Kolejnym testem modelu dyfuzji było porównanie wyników dla dwóch sposobów zaszumiania obrazu: podejścia liniowego oraz cosinusowego. Test ten został wykonany dla 15 epok, 1000-ca kroków zaszumiania i architektury modelu U-Net [64, 98, 128], a jego wyniki przedstawione zostały w tabeli Tab.10 oraz na rysunku Rys. 3.10. Analizując Rys. 3.10 można zauważyć, że zaszumienie cosinusowe osiąga lepsze wyniki co oznacza, lepsze odwzorowanie obrazów. Dzieje się tak poprzez fakt, że szum jest dodawany w bardziej subtelny i zróżnicowany sposób.

Tab. 3.10: Wyniki badania sposobu generowania szumu w modelu DDPM

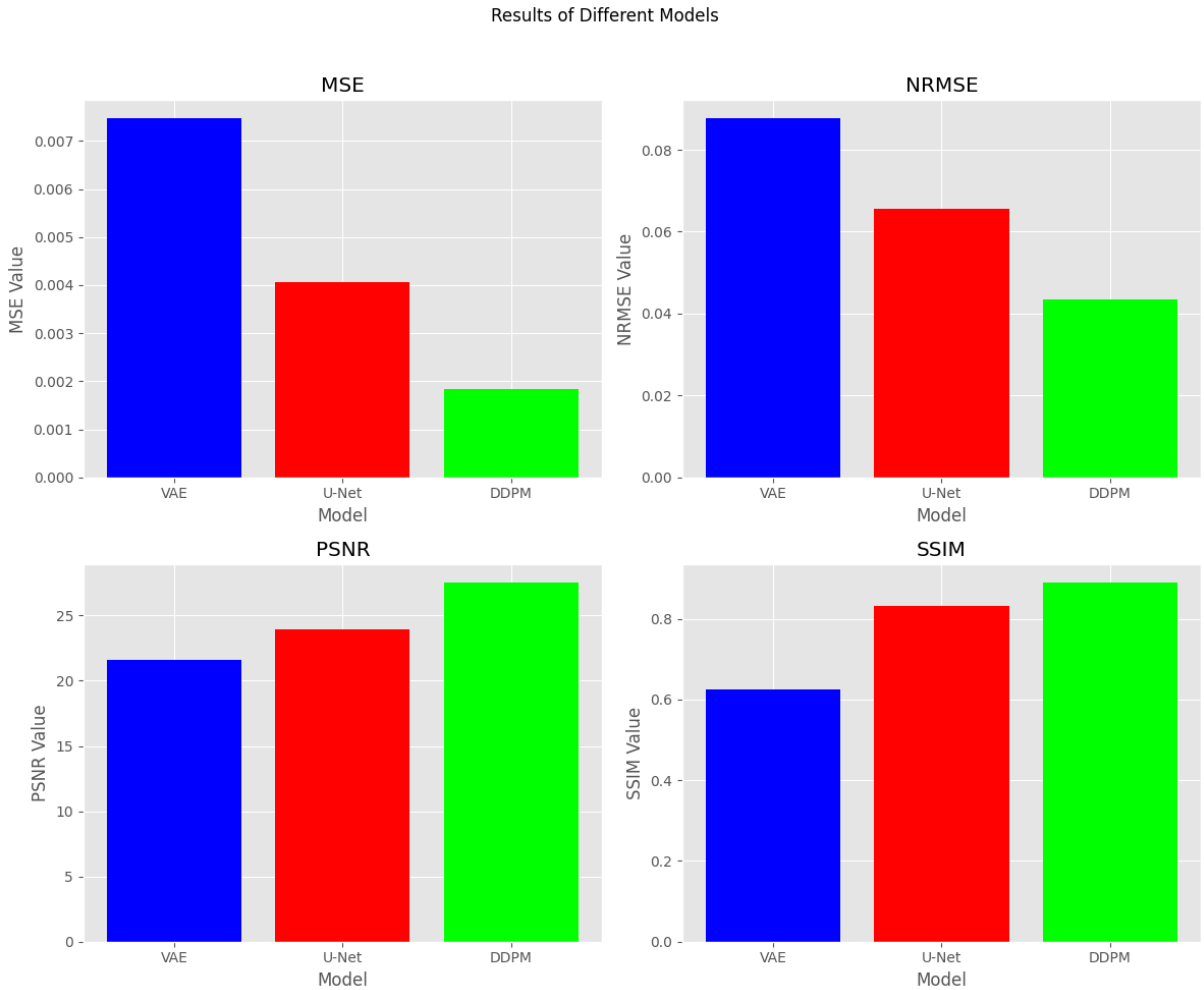
Metoda dodawania szumu	MSE	NRMSE	PSNR	SSIM
Linear	0.003872	0.066914	23.584375	0.834741
Cosine	0.002762	0.054888	25.958137	0.899143



Rys. 3.10: Wyniki badania sposobu generowania szumu w modelu DDPM

3.3 Porównanie wyników

Wszystkie wyniki trenowanych modeli: VAE, U-Net oraz DDPM, zostały porównane i przedstawione na rysunku Rys. 3.11 oraz w tabeli Tab. 3.11. Wyniki przedstawiają jednoznacznie przewagę modelu DDPM. Na drugim miejscu uplasował się U-Net, którego wyniki także są zadowalające, na ostatnim miejscu stanął najprostszy z modeli VAE, którego wyniki są jedynie dobre.



Rys. 3.11: Porównanie wyników modeli: VAE, U-Net i DDPM.

Tab. 3.11: Porównanie wyników modeli: VAE, U-Net i DDPM.

Model	MSE	NRMSE	PSNR	SSIM
VAE	0.007474	0.087733	21.622611	0.623723
U-Net	0.004065	0.065525	23.969002	0.830932
DDPM	0.001839	0.043395	27.529949	0.889963

4. Wnioski

Celem projektu było wypełnienie luk w obrazie za pomocą techniki Inpainting, aby zachować kontekst obrazu oraz uzyskać naturalny wygląd ostatecznej rekonstrukcji. W tym celu wykorzystane zostały trzy modele generatywne: VAE, U-Net oraz DDPM. W ramach projektu przeprowadzone zostały badania hiper parametrów mające na celu sprecyzowanie najlepszej wersji modelu do zadania. Z tak wybranymi parametrami wszystkie z modele uzyskały zadowalające rezultaty, a najlepszy okazał się model DDPM.

W trakcie projektu znalazło się kilka trudności. Największą z nich była niewątpliwie złożoność obliczeniowa wybranych modeli, która stanowiła duże ograniczenie. Udało się je ominąć poprzez użycie platformy *Google Colab*. Kolejnym problemem była duża ilość danych. Zbiór posiadał ponad 100 tys. zdjęć i mimo, że były one niskiej rozdzielczości i nie zajmowały dużo przestrzeni, z powodu ich ogromnej ilości przenoszenie oraz wczytywanie było długotrwałe. Z tego powodu zdecydowano się także użyć formatu przechowywania plików *pickle*.

W zadaniu zastosowano podstawowe wersje modeli co pozostawia wiele przestrzeni na modyfikacje oraz poprawę wyników. Choć model dyfuzji uzyskał zdecydowanie najlepsze wyniki, proces jego trenowania i wykorzystywania był bardzo czasochłonny. Dlatego, w przypadku prostszych zadań, gdzie dostępny czas i moc obliczeniowa są ograniczone, zaleca się korzystanie z modelu U-Net.

Bibliografia

[1] U-Net: Convolutional Networks for Biomedical Image Segmentation by Olaf Ronneberger, Philipp Fischer, and Thomas Brox (2015)

[2] M. Wael, Cook your First U-Net in PyTorch. A magic recipe to empower your image segmentation projects, Towards Data Science, https://towardsdatascience.com/cook-your-first-u-net-in-pytorch-b3297a844cf3