Politechnika Śląska

PRACA MAGISTERSKA

„Ocena jakości obrazu bez obrazu odniesienia”

Paweł MISZTAL
297712

Kierunek: Informatyka
Specjalność: Internet i Technologie Sieciowe

PROWADZĄCY PRACĘ
Dr hab. inż. Henryk Palus, prof. PŚ
KATEDRA Inżynierii i Analizy Eksploracyjnej Danych
Wydział Automatyki, Elektroniki i Informatyki

GLIWICE 2025

Tytuł pracy:

Ocena jakości obrazu bez obrazu odniesienia

Streszczenie:

W niniejszej pracy poruszono problematykę bezreferencyjnej oceny jakości obrazów. W pracy dokonano analizy literatury oraz wybrano i zaimplementowano model o nazwie TReS, oparty na architekturze transformera. Przeanalizowano architekturę modelu oraz przeprowadzono szereg modyfikacji mających na celu poprawę wydajności implementowanego modelu. W ramach pracy przeanalizowano rodzaje ocen jakości obrazu, percepcję jakości obrazu oraz dokonano analizy dostępnych zbiorów danych wykorzystywanych w dziedzinie bezreferencyjnej jakości obrazów, takich jak LIVE, CISQ, TID2013, KADID-10k, CLIVE, BID i KonIQ-10k. W części implementacyjnej dokonano analizy modelu TReS identyfikując kluczowe różnice w implementacji transformera. Przeprowadzono modyfikację ekstraktora cech zastępując model ResNet 50, modelami EfficientNet B4 oraz ConvNeXt Tiny. Zaproponowano również alternatywną stratę rankingową z adaptacyjnym marginesem. W części badawczej przeprowadzono eksperymenty mające na celu zbadanie wpływu: modyfikacji transformera, zmiany ekstraktora cech, wartości dropoutu oraz nowej funkcji strat na wydajność modelu. Na końcu porównano zoptymalizowaną metodę z innymi metodami z aktualnego stanu wiedzy, wskazujące, że osiąga ona konkurencyjne lub najlepsze wyniki na większości testowanych zbiorów danych. Praca kończy się podsumowaniem wraz ze wskazaniem możliwych kierunków dalszego rozwoju.

Słowa kluczowe:

Bezreferencyjna ocena jakości obrazu, Model TReS, Transformer, Rankingowa funkcja strat, Ekstrakcja cech

Thesis title:

Image quality assessment without reference image

Abstract:

This thesis addresses the topic of no-reference image quality assessment. The thesis includes a literature analysis and the selection and implementation of a model called TReS, based on transformer architecture. The model's architecture was analyzed, and a series of modifications was introduced to improve its performance. The thesis also analyzes types of quality assessment, perception of image quality and available datasets used in the field of no-reference image quality assessment, such as LIVE, CISQ, TID2013, KADID-10k, CLIVE, BID and KonIQ-10k. In the implementation part, the TReS model was analyzed to identify key differences in the transformer. The feature extractor was modified by replacing original ResNet 50 with EfficientNet B4 and ConvNeXt Tiny. An

alternative ranking loss function with adaptive margin was also proposed. In the experimental section, tests were conducted to evaluate the impact of transformer modifications, changes in feature extractors, dropout values and the new loss function on model performance. Finally, the optimized method was compared with other state-of-the-art approaches, showing that it achieves competitive or top results on most of the tested datasets. The thesis concludes with a summary and suggestions for future research directions.

Keywords:

No-reference image quality assessment, TReS model, Transformer, Ranking loss function, Feature extraction

Spis treści

1. Wstęp

W czasach dynamicznego rozwoju technologii cyfrowych jakość obrazu odgrywa kluczową rolę w wielu dziedzinach życia od: medycyny, przez systemy wizyjne pojazdów autonomicznych oraz wizję komputerową w warunkach przemysłowych, a kończąc na wyznaczeniu najlepszego zdjęcia w galerii zdjęć telefonu konieczne jest jak najlepsze określenie jakości obrazu [1]. Tradycyjne metody oceny jakości obrazu opierają się na porównaniu z obrazem referencyjnym, jednak w wielu wypadkach taki obraz nie jest dostępny. W takich sytuacjach niezbędne stają się metody bezreferencyjnej oceny jakości obrazu. Ocena jakości obrazu, bez obrazu odniesienia stanowi jedną z bardziej wymagających dziedzin przetwarzania obrazu. Jest to spowodowane koniecznością oceny jakości obrazu wyłącznie na podstawie jego zawartości, bez jakiegokolwiek punktu odniesienia. Dodatkowo wyzwanie staje się jeszcze bardziej złożone przez charakter percepcji wizualnej człowieka, który do oceny jakości nie bierze wyłącznie ostrości czy kontrastu, ale uwzględniania też takie cechy obrazu jak jego subiektywna atrakcyjność estetyczna czy kontekst.

Celem niniejszej pracy jest analiza literatury oraz wybranie i zbadanie jednej z dostępnych metod oceny jakości obrazu bez obrazu odniesienia. Jako dodatkowy cel postawiono próbę zoptymalizowania metody pod kątem jej wydajności na testowanych zbiorach danych.

Przedmiotem niniejszej pracy jest analiza bezreferencyjnej oceny jakości obrazu. W szczególności skupiono się na analizie, implementacji oraz wpływie modyfikacji metody TReS na jej wydajność. Badania obejmują również porównanie wyników zoptymalizowanej metody do innych metod z aktualnego stanu wiedzy.

W kolejnych rozdziałach dokonano analizy tematu poprzez analizę rodzaju ocen jakości obrazu, przyjrzano się percepcji jakości obrazu oraz temu jak ludki mózg postrzega jakość obrazu. Przeanalizowano dostępne metody oceny jakości obrazu z uwzględnieniem ich ogólnych charakterystyk. Dokonano analizy zbiorów danych syntetycznych oraz autentycznych wykorzystywanych w dziedzinie bezreferencyjnej oceny jakości obrazów, poprzez zebranie cech charakterystycznych zbiorów takich jak: LIVE, CISQ, TID2013, KADID-10k, CLIVE, BID, KonIQ-10k.

W następnym rozdziale wybrano metodę bezreferencyjnej oceny jakości obrazu pod nazwą TReS. Dokonano dokładnej analizy architektury modelu, dzięki czemu udało się

znaleźć istotne różnice w implementacji transformera w implementowanym modelu, co posłużyło się do przeprowadzenia badań porównujących wydajność tych zmian. Przeprowadzono opis funkcji strat oryginalnie wykorzystywanej przez ten model. Dodatkowo przeprowadzono modyfikację ekstraktora cech z obrazu poprzez zamianę modelu wyciągającego cechy z obrazu z oryginalnie wykorzystywanego modelu ResNet 50 na modele EfficientNet B4 oraz ConvNeXt Tiny. W dalszych badaniach zaproponowano również alternatywną stratę rankingową z adaptacyjnym marginesem.

Ostatni rozdział skupia się na przeprowadzonych badaniach. Na początku przybliżono miary wykorzystywane w badaniach oraz dokładnie opisano metodologię przeprowadzanych badań. Później przeprowadzono cztery eksperymenty, które miały na celu zbadanie wpływu: modyfikacji architektury transformera, wpływ zmiany ekstraktora cech, optymalizację wartości dropout oraz proponowaną funkcję strat na wydajność modelu. Na końcu tego rozdziału zbadano wydajność zoptymalizowanego modelu na wcześniej przytoczonych zbiorach danych oraz porównano go do innych metod z aktualnego stanu wiedzy.

W ostatnim rozdziale przedstawiono podsumowanie dokonanych prac wraz z wnioskami oraz zaproponowano potencjalne kierunki dalszych badań.

2. Analiza tematu

W niniejszym rozdziale omówione zostaną kluczowe aspekty oceny jakości obrazu bez obrazu odniesienia. Na początku opisane zostały rodzaje oceny jakości obrazu w tym ocena z pełnym odniesieniem, ocena z częściowym odniesieniem oraz ocena bez odniesienia. Następnie opisano percepcje jakości obrazu, która jest złożonym zagadnieniem związanym z tym, jak ludzki mózg interpretuje widziane obrazy. Zrozumienie tego procesu jest kluczowe dla skutecznej oceny jakości, ponieważ subiektywne odczucia mogą znacząco różnić się od obiektywnych miar. W dalszej części przeanalizowano metody bezreferencyjnej oceny jakości obrazu. Przedstawiono również miary do oceny jakości obrazów, takie jak MOS i DMOS, które są powszechnie stosowane w badaniach nad oceną jakości obrazów. Na końcu omówiono dostępne zbiory danych wykorzystywane w tej pracy.

2.1. Rodzaje oceny jakości obrazu

Ocenę jakości obrazu można rozpatrywać z różnych punktów widzenia, z czego w literaturze wyróżnia się następujące trzy sposoby.

Ocena jakości obrazu z pełnym odniesieniem FR-IQA (ang. *Full Reference Image Quality Assessment*) jest to ocena jakości obrazu z wykorzystaniem obrazu odniesienia, dzięki posiadaniu obrazu oryginalnego możliwe jest porównanie obrazu zdegradowanego w celu określenia jego jakości lub stopnia zdegradowania [2].

Ocena jakości obrazu z częściowym odniesieniem RR-IQA (ang. *Reduced Reference Image Quality Assessment*) jest to ocean jakości obrazu, gdzie posiadane są pewne informacje o obrazie odniesienia. Przykładowo można posiadać informacje o naturalnych statystykach sceny obrazu odniesienia i na podstawie wyznaczenia tych samych statystyk z obrazu zdegradowanego możliwe jest wyznaczenie jakości lub stopnia zdegradowania obrazu [3].

Ocena jakości obrazu bez obrazu odniesienia NR-IQA (ang. *No Reference Image Quality Assessment*) lub BIQA (ang. *Blind Image Quality Assessment*) jest to ocena jakości

obrazu, gdzie nie posiada się żadnych informacji o obrazie odniesienia, ponieważ w wielu przypadkach ten obraz odniesienia nie istnieje [2].

2.2. Percepcja jakości obrazu

Percepcja jakości obrazu jest skomplikowanym zagadnieniem, głownie przez to jak ludzki mózg postrzega jakość widzianego obrazu.

System wzrokowy człowieka HVS (ang. *Human Vision System*) jest to układ biologiczny człowieka rozpoczynający się od oczu a kończący na mózgu analizującym informacje pozyskane przez oczy. Dlatego oprócz posiadania odpowiedniej jakości zdjęcia konieczne jest też wzięcie pod uwagę tego jak ludzi mózg będzie to interpretować. Przykładowo możemy posiadać obraz, który nie jest rozmazany, ale jeśli pojawi się zniekształcenie rybiego oka, przykładowe zdjęcie na Rys. 1, to człowiek może ocenić takie zdjęcie jako słabej jakości. Innym problemem widzenia człowieka, że człowiek do jakości obrazu często zalicza wrażenia estetyczne, które są subiektywne i zależą od kontekstu. Jako przykład takiego zdjęcia można podać ilustracje na Rys. 2, która pokazuje oświetlony napis nocą. Pomimo słabego oświetlenia zdjęcia, obraz ten posiada walory artystyczne, które podnoszą ostatecznie postrzeganą jakość zdjęcia [4, 5].

Rys. 1 Obraz ze zniekształceniami typu „rybiego oka” [6, 7].

Rys. 2 Świecący napis w nocy [6, 7]

2.3. Analiza metod bezreferencyjnej oceny jakości obrazu

Pierwsze metody oceny jakości obrazu bez obrazu odniesienia polegały na ręcznym wyciąganiu cech z danego obrazu. Metody te opierały się o koncepcję naturalnych statystyk sceny, określanych w literaturze anglojęzycznej jako NSS (ang. Natural Scene Statistics). Koncepcja ta opiera się na tym, że typowe zdjęcie bez zniekształceń będzie posiadać pewne charakterystyczne właściwości statystyczne, może to być przykładowo odpowiedni rozkład jasności, ilość ostrych krawędzi na obrazie czy stopień rozmycia

obrazu. Następnie te cechy obrazu przy pomocy ręcznie zrobionych algorytmów lub metod uczenia maszynowego pozwalają określić jakość obrazu [8]. Metody te dzięki swojej prostocie i przejrzystości dobrze sobie radzą z wykrywaniem konkretnych zniekształceń obrazu, jednak nie radzą one sobie w warunkach rzeczywistych, gdzie zniekształcenia są zależne od kontekstu. Poza przykładem z poprzedniego rozdziału „2.2 Percepcja jakości obrazu", gdzie pokazano, że można mieć zniekształcenie spowodowane optyką kamery lub obraz o technicznie złym rozkładzie jasności, który będzie postrzegany jako dobrej jakości. Kolejnym przykładem zależnym od kontekstu obrazu może być przykładowe zdjęcie chmury jak na Rys. 3. W tym wypadku, gdyby wykrywać krawędzie na obrazie, to można by powiedzieć, że obraz ten jest słabej jakości, bo ma mało szczegółów.

Rys. 3 Zdjęcie chmury na niebie [6, 7]

Natomiast przykładem odwrotnym może być zdjęcie śniegu jak na Rys. 4, gdzie algorytm wykrywania krawędzi mógłby pomylić dużą ilość szczegółów na śniegu z szumem.

Rys. 4 Zdjęcie śniegu [6, 7]

Alternatywą, która w ostatnich czasach jest szeroko eksploatowaną ścieżką jest wykorzystanie głębokich sieci neuronowych DNN (ang. *Deep Neural Network*). W szczególności wykorzystywane są modele z wykorzystaniem konwolucyjnych sieci neuronowych CNN (ang. *Convolutional Neural Network*), dzięki swojej możliwości do wyciągania lokalnych cech z obrazu osiągają one wysokie wyniki w testach na autentycznych zbiorach danych. Kolejnym krokiem w rozwoju tych metod jest zastosowanie transformerów, dzięki swojej możliwości do uchwycenia globalnych zależności w obrazie pozwala to na osiągnięcie jeszcze wyższych wydajności niż w przypadku korzystania wyłącznie z konwolucyjnych sieci neuronowych [8].

2.4. Miary jakości w zbiorach danych

W zbiorach danych najczęściej występują dwie miary oceny jakości obrazu, jest to MOS oraz DMOS.

2.4.1. MOS

Pierwszą z nich jest MOS (ang. *Mean Opinion Score*) współczynnik ten określa średnią subiektywną ludzką ocenę jakości dla danego obrazu. W zależności od zbioru danych zakres wartości może być różny przykładowo od 1 do 5 lub od 1 do 100. Współczynnik MOS najczęściej jest wykorzystywany w autentycznych zbiorach danych. Wyższa wartość oznacza lepszą jakość. Współczynnik ten jest miarą jakości obrazu.

2.4.2. DMOS

Drugim współczynnikiem jest DMOS (ang. *Difference Mean Opinion Score*). Współczynnik ten określa różnicę w średniej subiektywnej ocenie jakości dla obrazu względem obrazu referencyjnego bez degradacji. W zależności od zbioru danych wartość tego współczynnika też może mieć różny zakres, przykładowo od 1 do 5 lub od 1 do 100. W przeciwieństwie do MOS, ten współczynnik jest wykorzystywany tylko w syntetycznych zbiorach danych, dodatkowo wyższa wartość oznacza gorszą jakość. Oznacza to, że ten współczynnik jest miarą degradacji obrazu.

2.5. Typy zbiorów danych

Aktualnie dostępne zbiory danych można podzielić głównie na dwie kategorie, zbiory danych określane jako syntetyczne oraz zbiory danych określane jako autentyczne.

2.5.1. Syntetyczne zbiory danych

Pierwszą z nich są zbiory danych syntetyczne. Kluczową ich cechą jest to, że tego typu zbiory danych degradują obraz poprzez nałożenie odpowiednich filtrów przykładowo biały szum, rozmycie gaussowskie czy kompresja obrazu. Dzięki temu możliwe jest generowanie dużej ilości danych ze stosunkowo małej ilości danych wejściowych oraz możliwe jest uczenie i testowanie na tych zbiorach danych wykrywania odpowiedniego typu degradacji jakości obrazu. Jednak nie licząc degradacji w postaci kompresji obrazu, filtry tego typu nie odzwierciedlają odpowiednio skomplikowania rzeczywistych zniekształceń, występujących na fotografiach.

2.5.2. Autentyczne zbiory danych

Drugim typem zbiorów danych są autentyczne zbiory danych. Charakteryzują się one tylko rzeczywistymi obrazami, bez dokonywania na nich sztucznej degradacji. Tego typu zbiory danych w większości posiadają mniejszą ogólną liczbę obrazów. Jednak dzięki posiadaniu autentycznych zniekształceń możliwe jest lepsze testowanie algorytmów w warunkach rzeczywistych.

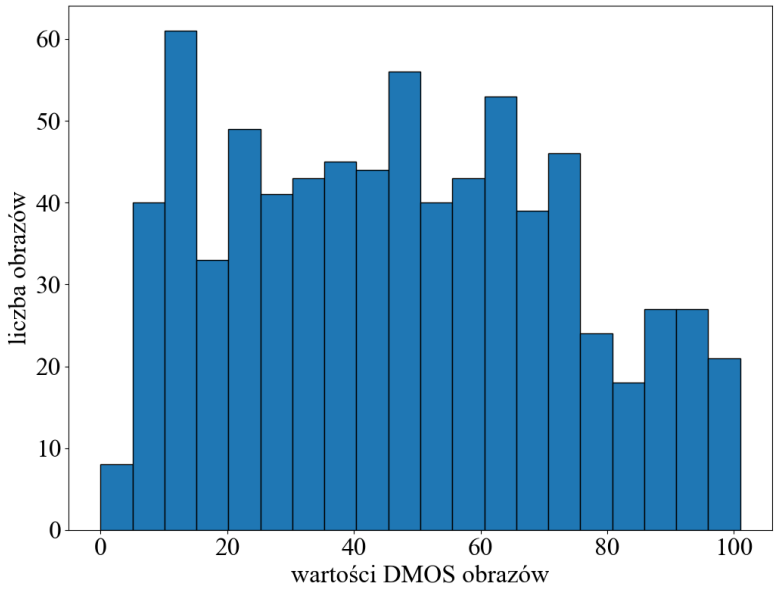
2.6. Zbiory danych

Jednym z kluczowych kroków potrzebnych do oceny działania danego algorytmu jest odpowiedni zbiór danych. Na przestrzeni lat powstało wiele dostępnych zbiorów danych wykorzystywanych do uczenia oraz testowania algorytmów i modeli oceny jakości obrazu bez obrazu odniesienia. W tym rozdziale dokonana zostanie ogólna charakterystyka istniejących zbiorów danych oraz zestawienie ich w Tabeli 1.

Opisane w tym rozdziale zbiory danych nie wyczerpują wszystkich dostępnych zbiorów danych wykorzystywanych w bezreferencyjnej ocenie jakości obrazów, opisano jedynie zbiory danych występujące w pracach „Progress in Blind Image Quality Assessment: A Brief Review" oraz „No-Reference Image Quality Assessment via Transformers, Relative Ranking, and Self-Consistency" ze względu na możliwość łatwego porównania z innymi rozwiązaniami w dziedzinie bezreferencyjnej oceny jakosci obrazu [8, 9].

2.6.1. LIVE

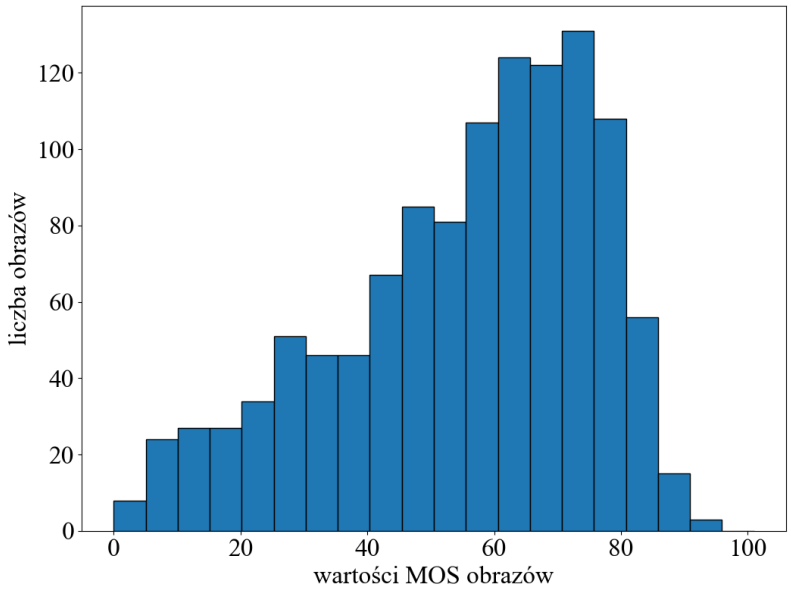
Jest to syntetyczny zbiór danych. W drugiej edycji posiada on 779 obrazów powstałych na skutek degradowania 29 obrazów bazowych przy pomocy 5 różnych filtrów m.in.: kompresji JPEG, kompresji JPEG2000, rozmycia Gaussa, białego szumu oraz błędu bitów w strumieniu bitów JPEG2000. Wszystkie obrazy posiadają ocenę DMOS w zakresie od 0 do 100 gdzie niższe wartości oznaczają lepszą jakość obrazu. Na Rys. 5 przedstawiono histogram rozkładu wartości DMOS dla tego zbioru danych. Dodatkowo można zauważyć, że istnieje stosunkowo równomierny rozkład wartości w przedziale od 20 do 60. Widoczne jest również, że liczba obrazów maleje wraz ze wzrostem wartości DMOS, co świadczy o mniejszej ilości obrazów o niskiej jakości. [10, 11, 12].



Rys. 5 Histogram wartości DMOS dla zbioru danych LIVE

2.6.2. CLIVE

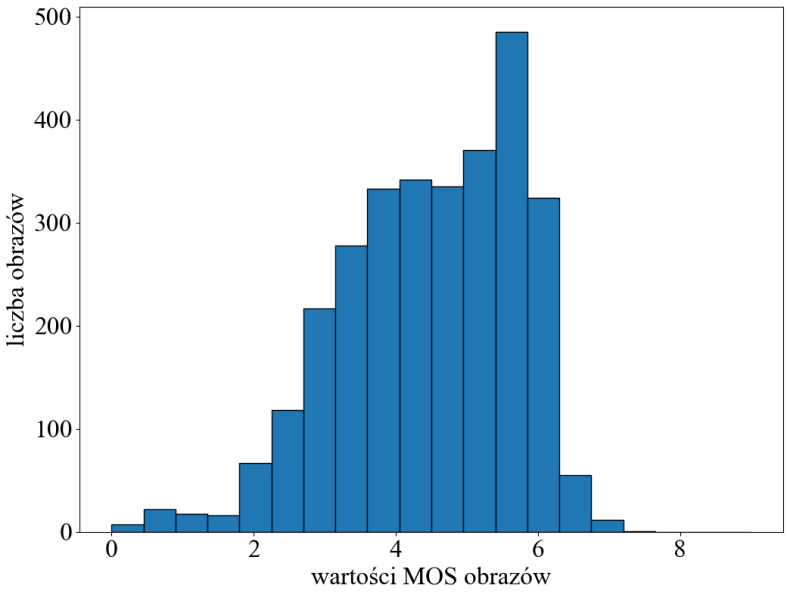
Jest to autentyczny zbiór danych składający się z 1162 obrazów wykonanych przez wiele różnych urządzeń mobilnych. Wszystkie obrazy posiadają ocenę MOS w zakresie od 0 do 100, gdzie większe wartości oznaczają lepszą jakość. Na Rys. 6 przedstawiono histogram rozkładu wartości oceny obrazów w zbiorze danych. Na tym histogramie widoczna jest przewaga obrazów dobrej jakości, w zakresie od 60 do 80, względem innych wartości. Może to powodować niedouczenie modeli w dolnym i górnym zakresie wartości, ze względu na brak wystarczających danych do nauki [6, 7].



Rys. 6 Histogram wartości MOS dla zbioru danych CLIVE

2.6.3. TID2013

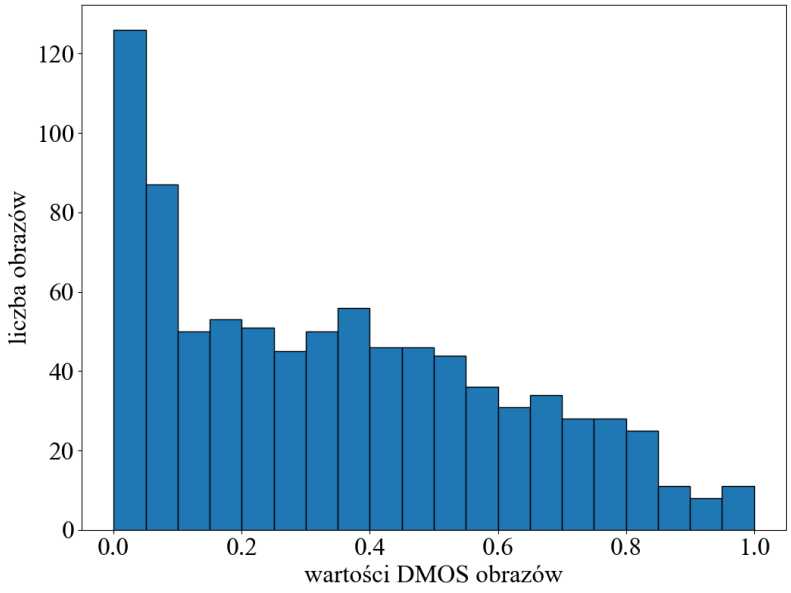
Jest to syntetyczny zbiór danych składający się z 3000 obrazów powstałych na skutek degradacji 25 zdjęć referencyjnych przy użyciu 24 filtrów z 5 stopniami degradacji. Obrazy w tym zbiorze danych posiadają ocenę MOS w zakresie od 0 do 9. Na Rys. 7 przedstawiono histogram rozkładu wartości ocen obrazów w tym zbiorze danych. Na tej ilustracji widoczne jest, że rozkład wartości zbliżony jest kształtem do rozkładu normalnego. Dlatego można się spodziewać, że modele uczone na tym zbiorze danych będą najczęściej przewidywać najlepiej jakość obrazów, która jest w przedziale od 3 do 6 [13, 14, 15].



Rys. 7 Histogram wartości MOS dla zbioru danych TID2013

2.6.4. CISQ

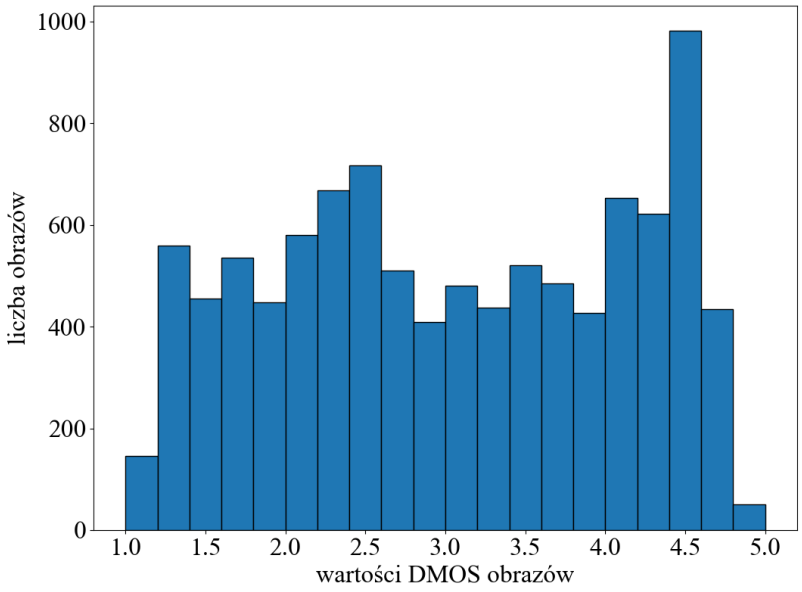
Jest to syntetyczny zbiór danych posiadający 866 obrazów powstałych poprzez nałożenie sześciu różnych typów zniekształceń, od 4 do 5 poziomów natężenia. Obrazy w tym zbiorze danych posiadają DMOS w zakresie od 0 do 1. Na Rys. 8 przedstawiono histogram wartości zbioru danych CISQ. Na ilustracji widoczne jest, że największa liczba obrazów jest w wartość DMOS 0, następnie widać stopniowy spadek liczby obrazów wraz ze wzrostem wartości DMOS [16].



Rys. 8 Histogram wartości DMOS dla zbioru danych CISQ

2.6.5. KADID-10k

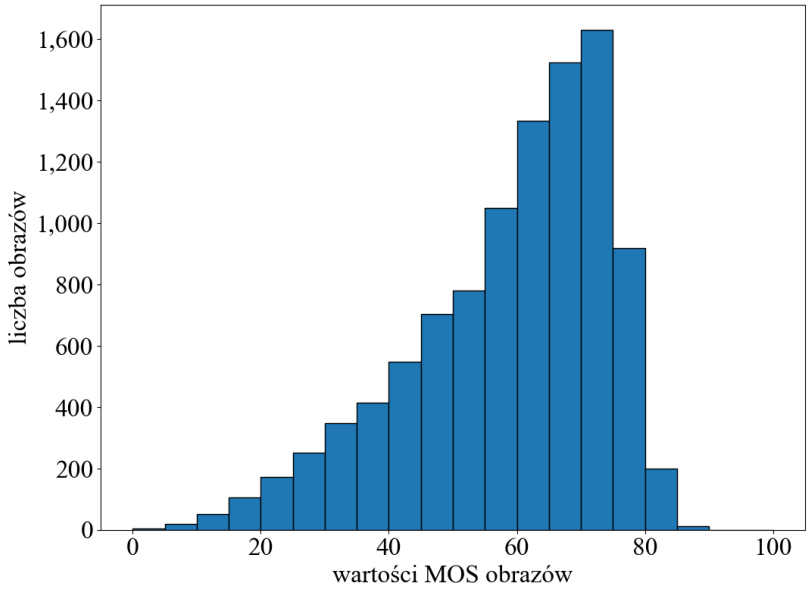
Jest to syntetyczny zbiór danych posiadający 10125 obrazy powstałe wskutek nałożenia 25 różnych filtrów w 5 poziomach natężenia na 81 zdjęć bazowych. Obrazy w tym zbiorze danych posiadaj DMOS w zakresie od 1 do 5. Na Rys. 9 przedstawiono histogram wartości zbioru danych KADID-10k, ten zbiór danych posiada względnie równomierny rozkład wartości, względem innych wymienionych zbiorów danych [17, 18].



Rys. 9 Histogram wartości MOS dla zbioru danych KADID-10k

2.6.6. KonIQ-10k

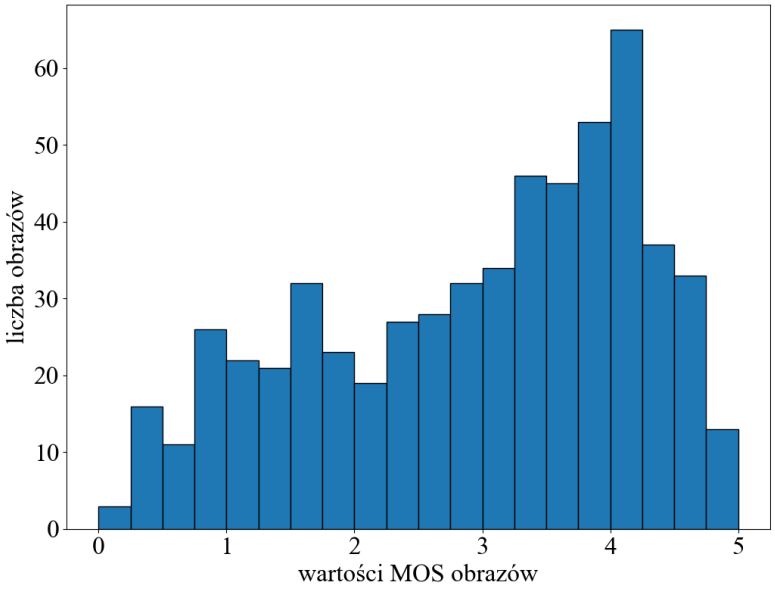
Jest to autentyczny zbiór danych zawierający 10073 obrazy. Każdy z obrazów posiada ocenę MOS w zakresie od 0 do 100. Na Rys. 10 przedstawiono histogram wartości zbioru danych KonIQ-10k. Na ilustracji widoczna jest znacząca przewaga w liczbie obrazów o ocenie w okolicach 3,5 oraz stopniowy spadek liczby obrazów w kierunku obrazów z oceną 1. Dodatkowo w kierunku obrazów z oceną 5 spadek jest bardziej gwałtowny oraz jest bardzo mała liczba obrazów z oceną powyżej 4,5 [19].



Rys. 10 Histogram wartości MOS dla zbioru danych KonIQ-10k

2.6.1. BID

Jest to autentyczny zbiór danych zawierający 586 obrazów, gdzie każdy z obrazów posiada MOS w zakresie od 0 do 5. Na Rys. 11 przedstawiono histogram wartości zbioru danych BID, na wykresie widoczna jest delikatna przewaga obrazów o wysokiej jakości. Zbiór danych udało się znaleźć w serwisie *github* [20, 21, 22]



Rys. 11 Histogram wartości zbioru danych BID

W Tabeli 1 przedstawiono zestawienie opisanych zbiorów danych, w pierwszej kolumnie umieszczono nazwę zbioru danych, w drugiej kolumnie liczbę obrazów w zbiorze danych, w trzeciej kolumnie zostało określone, czy jest to syntetyczny, czy autentyczny zbiór danych, w czwartej kolumnie określono, czy w zbiorze danych użyto metryki MOS, czy DMOS oraz w ostatniej kolumnie przedstawiono zakres wartości jakości obrazów w zbiorach danych.

Tabela 1. Zestawienie przedstawionych zbiorów danych

Nazwa zbioru danych	**Liczba obrazów**	**Typ zniekształceń**	**Typ oceny jakości**	**Zakres wartości**
LIVE	779	Syntetyczny	DMOS	0-100
CISQ	866	Syntetyczny	DMOS	0-1
TID2013	3000	Syntetyczny	MOS	0-8
KADID10K	10125	Syntetyczny	DMOS	1-5
CLIVE	1162	Autentyczny	MOS	0-100
BID	586	Autentyczny	MOS	0-5
KONIQ10K	10073	Autentyczny	MOS	0-100

3. Implementacja wybranej metody

W tym rozdziale opisany zostanie wybrany i zaimplementowany model pod nazwą TReS (ang. ***T****ransformers,* ***Re****lative ranking, and* ***S****elf consistency*) [9]. Autorzy tej metody umieścili swój kod źródłowy w serwisie Github [23]. Z tego względu po przeprowadzeniu autorskiej implementacji, porównano implementację z oryginałem oraz przedstawiono różnice jakie zostały wprowadzone przez autorów, a nie zostały one opisane w ich pracy [9].

Dodatkowo zostały w tym rozdziale opisano modyfikację ekstraktora cech modelu TReS oraz zaproponowano nową funkcję strat biorącą pod uwagę ranking obrazów, co miało na celu zwiększenie współczynników SROCC oraz PLCC.

3.1. Uzasadnienie wyboru

Wybór modelu TReS do badań w niniejszej pracy wynika z zastosowania w jego architekturze transformera, który pozwala na wzięcie pod uwagę globalnych cech obrazu oraz zbieranie lokalnych cech obrazu z wykorzystaniem wstępnie trenowanej sieci ResNet 50. W literaturze przedmiotu zauważono, że modele oparte na architekturze transformera, takie jak TReS osiągają lepsze wyniki w porównaniu do tradycyjnych metod.

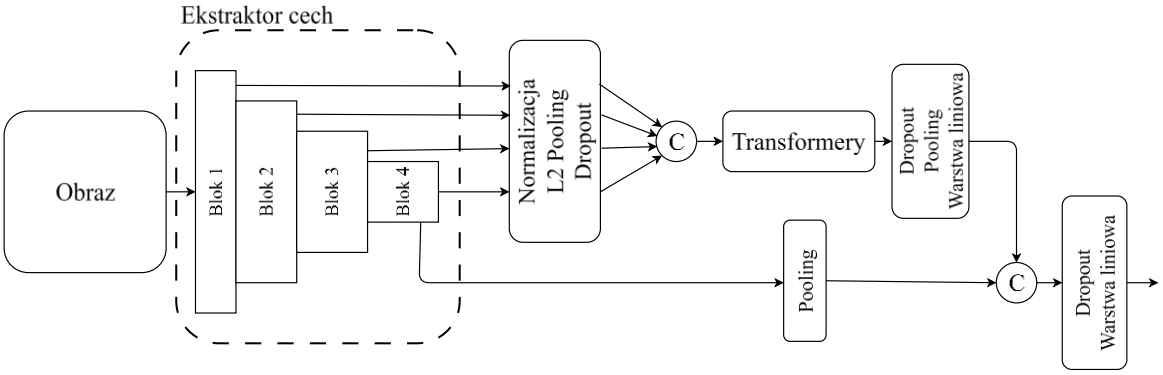
3.2. Wybór środowiska

Do realizacji niniejszej pracy zdecydowano się na wybór języka programowania Python wraz z biblioteką PyTorch jako środowisko do implementacji wybranej metody. Python jest obecnie jednym z najczęściej wykorzystywanych języków programowania w dziedzinie uczenia maszynowego i przetwarzania obrazów, głównie ze względu na swoją prostotę oraz dużą liczbę bibliotek. Biblioteka PyTorch została wybrana ze względu na swoją elastyczność w definiowaniu architektur sieci nieuronowych, możliwość trenowania modeli z wykorzystaniem kart graficznych, co znacząco przyspiesza czas nauki modelu. Dodatkowym atutem PyTorch jest dostępność gotowych wstępnie trenowanych

modeli w ramach biblioteki *torchvision*, które znacząco ułatwiły pracę w późniejszych etapach badań.

3.3. Opis modelu TReS

Na Rys. 12 przedstawiono schemat architektury modelu TReS. Model ten składa się w pierwszej jego części z ekstraktora cech w postaci wstępnie trenowanej sieci konwolucyjnej ResNet 50, sieć ta była wstępnie trenowana na zbiorze ImageNet 1k [24].



Rys. 12 Architektura modelu TReS [9]

Sieć ResNet 50 składa się z 4 bloków. Z końcowej warstwy każdego z bloków wyciągnięto tensor cech, którego rozmiar można określić wzorem (1), gdzie $i \in \{1,2,3,4\}$ oraz i odnosi się do numeru bloku z sieci ResNet 50, c jest to ilość kanałów, w oraz h jest to odpowiednio szerokość oraz wysokość kanału.

$$s_i = b \times c_i \times w_i \times h_i \tag{1}$$

Następnie cechy z każdej warstwy osobno przechodzą przez normalizację, autorzy wykorzystali normalizację Euklidesową, która można zdefiniować wzorem (2), gdzie $\boldsymbol{X}$ jest tensorem z odpowiedniego bloku ekstraktora cech, ε jest stałą będącą bardzo małą liczbą większą od zera.

$$n(\boldsymbol{X}) = \frac{\boldsymbol{X}}{\max\left(\|\boldsymbol{X}\|_2, \varepsilon\right)} \tag{2}$$

W następnym każdy tensor przechodzi przez warstwę L2 Pooling, której wzór można przedstawiono w równaniu (3), gdzie g jest rozmywającym jądrem o rozmiarze 3x3, zaimplementowanym przy pomocy okna Hanninga, gdzie wszystkie elementy tego jądra sumują się do 1.

$$P_{L2}(\boldsymbol{X}) = \sqrt{g * \boldsymbol{X}^2} \tag{3}$$

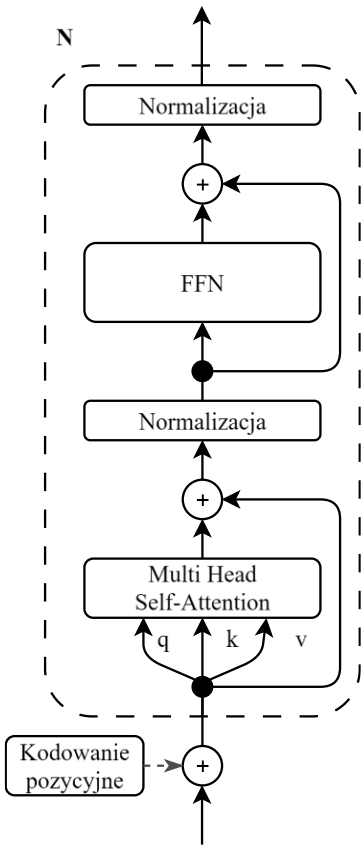
Następnie tensory przechodzą przez funkcję dropout, która losowo wyłącza połączenia w modelu. Przed konkatenacją oznaczoną na Rys. 12 jako C w okręgu, konieczna jest zmiana rozmiaru tensorów, z bloków ekstraktora cech, poprzez ujednolicenie rozmiaru dwóch ostatnich wymiarów *w* i *h*, przy pomocy uśredniającego poolingu 2D. Docelowy rozmiar tensorów można opisać wzorem (4), wymiary *w* i *h* posiadają rozmiar z ostatniego bloku ekstraktora cech.

$$s'_i = b \times c_i \times w_4 \times h_4 \tag{4}$$

Przed przekazaniem tensora do transformera, przeprowadzono konkatenację tensorów wzdłuż wymiaru *c*, co daje nam rozmiar tensora, który można opisać równaniem (5).

$$s''_i = b \times \sum_{i=1}^{4} c_i \times w_4 \times h_4 \tag{5}$$

Autorzy modelu TReS opisując model, pokazali, że ich transformer jest taki sam jak opisany w pracy „Attention Is All You Need" [25]. Schemat enkodera transformera przedstawiono na Rys. 13. Symbol N oznacza liczbę transformatorów w sekwencji jeden po drugim.



Rys. 13 Schemat enkodera transformera [25]

Blok FFN można zdefiniować wzorem (6), gdzie **W** jest optymalizowaną macierzą wag, a $\bar{b}$ optymalizowanym wektorem przesunięć.

$$FFN(\boldsymbol{X}) = \boldsymbol{W}_2 \times \max\left(\boldsymbol{W}_1 \times \boldsymbol{X} + \overline{b_1}, 0\right) + \overline{b_2} \tag{6}$$

Sygnał po wyjściu z transformera, przechodzi przez dropout, a następnie zostaje zmniejszony rozmiar sygnału, przy pomocy ujednolicającego poolingu, a następnie zostaje spłaszczony do wymiaru, który można opisać wzorem (7).

$$s''_i = b \times c_4 \tag{7}$$

Ostatnim elementem w tym bloku jest warstwa liniowa której rozmiar wyjściowy wynosi c_4, co odpowiada liczbie kanałów w ostatniej warstwy z ekstraktora cech.

Następnie przeprowadzana jest konkatenacja z ostatniego bloku ekstraktora cech, który został zmniejszony przy pomocy ujednolicającego poolingu i spłaszczony do wymiaru, który można opisać wzorem (7), oraz z wyjścia transformera. Potem sygnał przechodzi przez dropout i ostatnią warstwę liniową, której wyjściem jest pojedyncza wartość oznaczająca przewidywaną jakość obrazu.

3.3.1. Funkcja strat

Autorzy modelu TReS zastosowali ciekawe podejście do funkcji strat, zamiast ograniczać się do używania tylko średniego błędu bezwzględnego jako funkcji strat zaproponowali oni dodatkowo połączenie tego błędu ze stratą tripletu oraz stratą nazwaną przez autorów stratą spójności (ang. *Consistency Loss*) [9].

Strata tripletu polega na tym, że brana jest pierwsza, druga i ostatnia wartość przewidywana z posortowanej partii, następnie przy pomocy wzoru opisanego w równaniu (8) [26], gdzie literą *a, p* oraz *n* oznaczono odpowiednio pierwszą, drugą oraz ostatnią wartość a literą *m* oznaczono margines. Model jest karany, jeśli ostatnia wartość jest bliżej pierwszej niż drugiej. Analogicznie postępuje się dla ostatniej, przedostatniej i pierwszej wartości. Dodatkowo obliczany jest dynamicznie margines jako różnica ostatniej i drugiej wartości docelowej oraz przedostatniej i pierwszej wartości docelowej [9]:

$$L_t(a,p,n) = max(\|a-p\|_2 - \|a-n\|_2) + m, 0) \tag{8}$$

Strata spójności polega na wzięciu wartości z ostatniego bloku ekstraktora cech oraz wartości z wyjścia transformerów oraz straty triplet i porównanie go z lustrzanym odbiciem poprzez zastosowanie średniego błędu bezwzględnego na każdej z wartości. Gdzie średni błąd bezwzględny dla strat tripletu był mnożony przez 0,5. Operacje te można opisać równaniem (9), gdzie y_{conv} to wartości z ostatniego bloku ekstraktora cech, y_{trans} to wartości z wyjścia transformera, symbolem ‘ oznaczono straty dla lustrzanego odbicia [9].

$$L_{\text{C}} = \left\| y_{\text{conv}} - y'_{\text{conv}} \right\| + \left\| y_{\text{trans}} - y'_{\text{trans}} \right\| + 0{,}5 \cdot \| L_{\text{t}} - L'_{\text{t}} \| \tag{9}$$

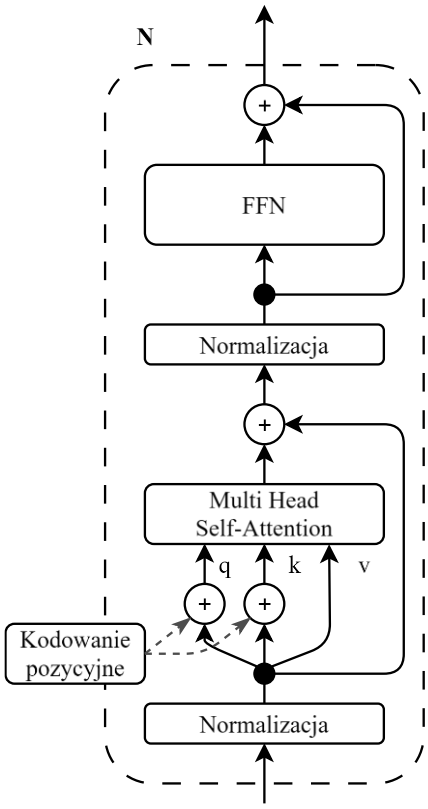
W równaniu (10) przedstawiono ostateczny wzór na funkcję strat, gdzie L_{mae} to średni błąd bezwzględny, L_c to strata spójności a L_t to strata tripletu [9].

$$L = L_{mae} + 0{,}05 \cdot L_c + L_t \tag{10}$$

3.3.2. Analiza implementacji - rozbieżność w implementacji transformera

Podczas analizy oryginalnej implementacji modelu TReS, zauważono rozbieżność w implementacji transformera. Schemat tej implementacji przedstawiono na rysunku 14. Pierwszą różnicą jest to, że najpierw zastosowano normalizację przed wejściem do transformera, zamiast normalizacji za wyjściem z transformera. Drugą różnicą jest zmiana miejsca, w którym do sygnału dodawane jest kodowanie pozycji. W typowej implementacji transformera, kodowanie pozycji dodawane jest do sygnału przed wejściem do transformera i dodawane jest tylko raz. Natomiast w tej implementacji kodowanie

pozycyjne jest dodawane w każdym bloku transformatora oraz dodawane jest tylko do elementów *q* i *k*.



Rys. 14 Zmodyfikowany transformer z modelu TReS

3.4. Przeprowadzone modyfikacje

W tym podrozdziale opisano modyfikację ekstraktora cech modelu TReS oraz zaproponowano nową funkcję strat biorącą pod uwagę ranking obrazów.

3.4.1. Zmiana ekstraktora cech

Oprócz zmian, które zostały wprowadzone w implementacji autorów, a nie zostały opisane w ich pracy, postanowiono sprawdzić jak na wydajność modelu będzie wpływać na zastosowanie różnego rodzaju ekstraktorów cech. W oryginale ekstraktorem cech była wstępnie trenowana konwolucyjna sieć neuronowa Resnet-50. W ramach badań zdecydowano się na sprawdzenie wydajności modelu TReS z innymi wstępnie trenowanymi sieciami takimi jak EfficientNet B4 oraz ConvNeXt Tiny.

Podczas wyboru alternatywnych modeli do ekstrakcji cech z obrazu wykorzystano zestawienie modeli z strony pytorch [27]. Przy wyborze kierowano się tym, żeby wybrane sieci posiadały podobną złożoność obliczeniową, posiadały większą wydajność na zbiorze danych ImageNet-1k oraz nie posiadały w sobie mechanizmu uwagi w formie Transformera (ang. *Transformer*), ze względu na to, że TReS wykorzystuje później w swojej architekturze mechanizm uwagi w formie transformatora, mechanizm ten byłby redundantny.

W Tabeli 2 przedstawiono zestawienie wstępnie trenowanych modeli wizyjnych na zbiorze danych ImageNet 1k. W pierwszej kolumnie przedstawiono nazwy modelu z wagami, w drugiej kolumnie przedstawiono wydajność w postaci dokładności przewidywania etykiet ze zbioru ImageNet 1k, w trzeciej kolumnie przedstawiono liczbę trenowalnych parametrów modeli oraz w ostatniej kolumnie przedstawiono liczbę operacji zmiennoprzecinkowych koniecznych do jednego przejścia obrazu przez model. Modele w tabeli są posortowane malejąco po wydajności. Zgodnie z wcześniej wymienionymi wymaganiami do alternatywnych modeli ekstraktora cech z obrazu zdecydowano się na modele EfficientNet B4 oraz ConvNeXt Tiny. Dodatkowo są to modele, które posiadają najwyższą wydajność w przy ograniczeniu liczbę operacji zmiennoprzecinkowych do $4{,}5 \cdot 10^9$ FLOPS. Zakres ten został wybrany jako maksymalna wartość jaką może mieć model, ze względu na podobną liczbę parametrów do modelu bazowego ResNet50, którego liczba operacji zmiennoprzecinkowych wynosi $4{,}09 \cdot 10^9$ FLOPS.

Tabela 2 Zestawienie wstępnie trenowanych modeli na zbiorze ImageNet 1k [27]

Nazwa wag	**Wydajność** [%]	**Parametry** [10^6]	**FLOPS** [10^9]
EfficientNet_B4_Weights.IMAGENET1K_V1	83,384	19,3	4,39
ConvNeXt_Tiny_Weights.IMAGENET1K_V1	82,520	28,6	4,46
EfficientNet_B3_Weights.IMAGENET1K_V1	82,008	12,2	1,83
RegNet_Y_3_2GF_Weights.IMAGENET1K_V2	81,982	19,4	3,18
Swin_T_Weights.IMAGENET1K_V1	81,474	28,3	4,49
ResNeXt50_32X4D_Weights.IMAGENET1K_V2	81,198	25,0	4,23
RegNet_X_3_2GF_Weights.IMAGENET1K_V2	81,196	15,3	3,18
RegNet_Y_1_6GF_Weights.IMAGENET1K_V2	80,876	11,2	1,61
ResNet50_Weights.IMAGENET1K_V2	80,858	25,6	4,09

Przy adaptacji modelu TReS do wykorzystania innych ekstraktorów cech istnieje niedogodność tego typu, że modele o innej architekturze niż ResNet nie posiadają

identycznej liczby bloków co użyty ResNet 50. Dodatkowo liczba kanałów w blokach innych wskazanych modeli również nie jest podobna. Dlatego poniżej znajduje się opis implementacji modeli EfficientNet B4 oraz ConvNeXt Tiny do modelu TReS, a w szczególności z których warstw brano cechy jako wejście do transformatora.

Przy wykorzystaniu ResNet 50 model dzieli się na 4 bloki o rozmiarach wyjściowych z każdego bloku opisanych w Tabeli Tabela 3. Co w efekcie po zastosowaniu wzoru z równania (5) daje rozmiar tensora $b \times 3840 \times 7 \times 7$.

Tabela 3 Rozmiary bloków modelu ResNet 50

Numer bloku	**Rozmiar bloku**
1	$b \times 256 \times 56 \times 56$
2	$b \times 512 \times 28 \times 28$
3	$b \times 1024 \times 14 \times 14$
4	$b \times 2048 \times 7 \times 7$

W przypadku wykorzystania modelu EfficientNet B4 można zauważyć, że posiada on 7 bloków o rozmiarach wyjściowych opisanych w Tabeli 4. Co w efekcie po zastosowaniu wzoru z równania (5) daje rozmiar tensora $b \times 1104 \times 7 \times 7$.

Tabela 4 Rozmiary bloków modelu EfficientNet B4

Numer bloku	**Rozmiar bloku**
1	$b \times 24 \times 112 \times 112$
2	$b \times 32 \times 56 \times 56$
3	$b \times 56 \times 28 \times 28$
4	$b \times 112 \times 14 \times 14$
5	$b \times 160 \times 14 \times 14$
6	$b \times 272 \times 7 \times 7$
7	$b \times 448 \times 7 \times 7$

W przypadku wykorzystania modelu ConvNeXt Tiny konieczne jest inne podejście, które jest to spowodowane tym, że bloku odpowiednio 2,4, i 6 są blokami redukującymi rozmiar. W Tabeli 5 przedstawiono rozmiary bloków modelu ConvNeXt Tiny. Po zastosowaniu wzoru z równania (5) daje rozmiar tensora $b \times 1458 \times 7 \times 7$.

Tabela 5 Rozmiary bloków modelu ConvNeXt Tiny

Numer bloku	Rozmiar bloku
1	$b \times 96 \times 56 \times 56$
2	$redukcja\ rozmiaru$
3	$b \times 192 \times 28 \times 28$
4	$redukcja\ rozmiaru$
5	$b \times 384 \times 14 \times 14$
6	$redukcja\ rozmiaru$
7	$b \times 786 \times 7 \times 7$

3.4.2. Strata rankingowa z adaptacyjnym marginesem

W dalszych badaniach zainspirowano się funkcją strat wykorzystaną przez autorów TReS, w postaci straty tripletu z adaptacyjnym marginesem. Zaproponowano alternatywne rozwiązanie wykorzystujące stratę rankingową z adaptacyjnym marginesem. Autorzy pracy [9] twierdzą, że wzięcie pod uwagę wszystkich próbek w partii nauki jest skomplikowane obliczeniowo. Wykorzystanie straty tripletu jako funkcji strat biorącej pod uwagę ranking jest skomplikowane. W porównaniu do straty tripletu, strata marginesu pozwala w nieskomplikowany sposób wziąć pod uwagę całą partię podczas nauki, dzięki czemu możliwe jest lepsze odwzorowanie relacyjnego rankingu obrazów podczas nauki w partii.

Stratę rankingową z adaptacyjnym marginesem można w prostych słowach opisać jako funkcję straty, która bierze pod uwagę względną różnicę przewidywanych jakości pomiędzy posortowanymi obrazami w danej partii uczenia, gdzie obrazy są sortowane rosnąco według docelowych wartości jakości obrazów. Przy czym adaptacyjny margines obliczany jest ze względnej różnicy pomiędzy docelowymi wartościami.

Założeniem tej funkcji strat jest to, że rozmiar partii uczenia musi być parzysty oraz jego rozmiar musi wynosić minimum 4 obrazy na partię. Gdy obrazów jest nieparzysta liczba, to ignorowany jest ostatni obraz w partii. Natomiast gdy obrazów w partii jest mniej niż 4 to ignorowana jest ta strata i zwracana jest wartość 0. Sytuacje te mogą się zdarzyć tylko, gdy liczba obrazów w zbiorze danych nie jest podzielna bez reszty przez rozmiar partii uczenia.

Wzór na rankingową stratę z adaptacyjnym marginesem został przedstawiony w równaniu (11), gdzie L_{amr} – jest rankingową stratą z adaptacyjnym marginesem, $\bar{p}$ – jest tablicą przewidywanych jakości obrazów, $\bar{t}$ – jest tablicą docelowych jakości obrazów, m – jest adaptacyjnym marginesem, α – jest wzmocnieniem adaptacyjnego marginesu. Przy przedstawianiu wzorów przyjęto założenie, że pozioma kreska nad symbolem oznacza

tablicę wartości. L_{mr} jest rankingową stratą z marginesem, której wzór opisano równaniem (12) [28].

$$L_{amr}(\bar{p}, \bar{t}, \alpha) = L_{mr}(\bar{p}_{02}, \bar{p}_{13}, \bar{t}_{02}, m_{01} \cdot \alpha) + L_{mr}(\bar{p}_{13}, \bar{p}_{02}, \bar{t}_{12}, m_{12} \cdot \alpha) \tag{11}$$

$$L_{mr}(\bar{p}_1, \bar{p}_2, \bar{t}, m) = \max(0, -\bar{t} \cdot (\bar{p}_1 - \bar{p}_2) + m) \tag{12}$$

Konieczne jest odpowiednie podzielenie elementów, dlatego w równaniach (13), (14), (15), (16) przedstawiono sposób podziału tablicy predykcji jakości obrazu na odpowiednie wycinki oraz w równaniach (17), (18), (19), (20) przedstawiono sposób podziału tablicy docelowych wartości jakości obrazów na odpowiednie wycinki:

$$\bar{p}_{02} = \bar{p}_i \, , dla \; i \in \{0,1, \dots, n-2\} \tag{13}$$

$$\bar{p}_{13} = \bar{p}_i \, , dla \; i \in \{1,3, \dots, n-1\} \tag{14}$$

$$\bar{p}'_{13} = \bar{p}_i \, , dla \; i \in \{1,3, \dots, n-3\} \tag{15}$$

$$\bar{p}'_{02} = \bar{p}_i \, , dla \; i \in \{2,4, \dots, n-2\} \tag{16}$$

$$\bar{t}_{02} = \bar{t}_i \, , dla \; i \in \{0,1, \dots, n-2\} \tag{17}$$

$$\bar{t}_{13} = \bar{t}_i \, , dla \; i \in \{1,3, \dots, n-1\} \tag{18}$$

$$\bar{t}'_{13} = \bar{t}_i \, , dla \; i \in \{1,3, \dots, n-3\} \tag{19}$$

$$\bar{t}'_{02} = \bar{t}_i \, , dla \; i \in \{2,4, \dots, n-2\} \tag{20}$$

W równiach (21), (22) przedstawiono sposób obliczania różnic pomiędzy sąsiadującymi, docelowymi wartościami, gdzie wynik z obliczania różnicy przekazano następnie do funkcji z równania (23), które ma na celu wyciągnięcie znaku, czy dana liczba w tablicy jest mniejsza, większa czy równa zero.

$$\bar{t}_{01} = s(\bar{t}_{02} - \bar{t}_{13}) \tag{21}$$

$$\bar{t}_{12} = s(\bar{t}'_{13} - \bar{t}'_{02}) \tag{22}$$

$$s(x) = \begin{cases} 1 & , gdy \; x > 0 \\ -1 & , gdy \; x < 0 \\ 0 & , gdy \; x = 0 \end{cases} \tag{23}$$

Ostatnimi dwoma wzorami są równania (24), (25), które obliczają adaptacyjny margines z różnic pomiędzy sąsiadującymi docelowymi wartościami jakości obrazów, następnie brana jest wartość bezwzględna dla każdego elementu w tablicy i obliczana średnia.

$$m_{01} = \frac{1}{n}\sum |\bar{t}_{02} - \bar{t}_{13}| \tag{24}$$

$$m_{12} = \frac{1}{n}\sum |\bar{t}'_{13} - \bar{t}'_{02}| \tag{25}$$

4. Badania wybranej metody

W tym rozdziale przybliżono miary, takie jak SROCC, PLCC oraz MAE, które były wykorzystywane w eksperymentach nad badaniem modelu TReS oraz badaniu wydajności zoptymalizowanej metody. Następnie opisano metodologię przeprowadzanych badań. W kolejnej części przedstawiono szereg eksperymentów mających na celu znalezienie czynników, które pozwoliłyby na optymalizację wydajności modelu. W ostatniej części rozdziału przedstawiono ostateczne wyniki zoptymalizowanej metody przetestowanej na powszechnie stosowanych zbiorach danych oraz porównano zoptymalizowany model do innych metod w tej dziedzinie.

4.1. Miary oceny wydajności modelu

Zgodnie z powszechną praktyką w dziedzinie bezreferencyjnej oceny jakości obrazu stosowano współczynnik korelacji rangowej Spearmana oraz współczynnik liniowej korelacji Pearsona i dodatkowo wykorzystano średni błąd bezwzględny.

4.1.1. Współczynnik korelacji rangowej Spearmana

Współczynnik korelacji rangowej Spearmana SROCC (ang. *Spearman Rank Order Correlation Coefficient*) – jest to pierwszy z dwóch najczęściej używanych współczynników przy ocenie wydajności modelu w dziedzinie bezreferencyjnej oceny jakości obrazu. Współczynnik ten opisuje się wzorem (26) [8], gdzie *d* oznacza różnicę pomiędzy rangą przewidywaną a docelową, *n* to jest rozmiar zbioru. Wyższa wartość tego współczynnika oznacza lepszą skuteczność modelu:

$$SROCC = 1 - \frac{6 \cdot \sum_i d_i^2}{n(n^2 - 1)} \tag{26}$$

4.1.2. Współczynnik liniowej korelacji Pearsona

Współczynnik liniowej korelacji Pearsona PLCC (ang. *Pearson Linear Correlation Coefficient*) – jest to drugi z dwóch najczęściej używanych współczynników przy ocenie

wydajności modelu w dziedzinie bezreferencyjnej oceny jakości obrazu. Współczynnik ten opisuje się wzorem (27) [8], gdzie $\bar{p}_i$ oraz $\bar{t}_i$ oznaczają odpowiednio wektor przewidywanych i docelowych wartości, a p_m oraz t_m oznaczają odpowiednio średnią wartość przewidywanych i docelowych wartości. Wyższa wartość tego współczynnika oznacza lepszą skuteczność modelu:

$$PLCC = \frac{\sum_i (\bar{t}_i - t_m)(\bar{p}_i - p_m)}{\sqrt{\sum_i (\bar{t}_i - t_m)^2 \sum_i (\bar{p}_i - p_m)^2}} \tag{27}$$

4.1.3. Średni błąd bezwzględny

Średni błąd bezwzględny MAE (ang. *Mean Absolute Error*) – ze względu na brak informacji o względnej relacji jakości obrazów, nie jest to powszechnie stosowany współczynnik przy ocenie wydajności modeli bezreferencyjnej oceny jakości obrazów. Jednak jeśli współczynnik ten zostanie pominięty możliwe jest otrzymanie modelu, który będzie posiadać wysokie poprzednie dwa współczynniki, jednak ze względu na duży błąd wyniki będą trudniejsze do użycia w rzeczywistych warunkach. W równaniu (28) [8] przedstawiono wzór na średni błąd bezwzględny, gdzie $\bar{p}_i$ jest wektorem przewidywanych wartości, a $\bar{t}_i$ jest wektorem docelowych wartości, gdzie i jest indeksem tego wektora. Niższa wartość tego współczynnika oznacza lepszą skuteczność modelu.

$$MAE = \frac{1}{n} \sum_i |\bar{p}_i - \bar{t}_i| \tag{28}$$

4.2. Metodologia przeprowadzonych badań

Podczas badania wpływu zmian na wydajność modelu postępowano zgodnie z powszechną praktyką w dziedzinie bezreferencyjnej oceny jakości obrazu, w ten sposób, że zbiór danych dzielono na zbiór treningowy i testowy w stosunku 80/20 z wykorzystaniem różnych ziaren przy losowaniu, do którego zbioru trafi obraz [9]. Przy czym zbiory syntetyczne były dzielone tak, żeby zdjęcia referencyjne znajdowały się w tym samym zbiorze co zdegradowane zdjęcia z tego konkretnego zdjęcia referencyjnego. Ma to na celu zapobiec wyciekowi danych do zbioru testowego. Zbiory testowe nie były wykorzystywane do uczenia modelu.

Podczas testowania wpływu zmian na wydajność modelu postępowano w ten sposób, że dzielono zbiór danych przy wykorzystaniu 5 różnych ziaren. Podczas treningu

zastosowano techniki sztucznego zwiększenia zbioru danych poprzez augmentację danych poprzez odbicie losowe, odbicie lustrzane obrazu w pionie oraz poziomie oraz losowe wycinanie kawałka obrazu o wymiarach 224x224 pikseli, przed wycięciem kawałka obrazu upewniono się, że rozmiar obrazu jest nie większy niż 512x512 pikseli [9].

Zgodnie z tym co robili autorzy modelu TReS, co 10 epok, testowano model w ten sposób, że każde zdjęcie w zbiorze danych było testowane 50 razy z losowym wycięciem. Następnie uśredniano wyniki dla każdego obrazu i z wykorzystaniem średniej przewidywanej wartości z każdego obrazu obliczano współczynniki SROCC, PLCC oraz MAE. Z każdego treningu brano najlepszy wynik na podstawie współczynnika SROCC. Ostateczną wydajność modelu obliczano jako średnią ze współczynników SROCC, PLCC oraz MAE z 5 treningów z różnymi ziarnami [9].

Podczas ostatecznej ewaluacji wydajności modelu, postępowano podobnie co podczas testowania wpływu zmian na wydajność modelu, z tą różnicą, że każdy zbiór danych był dzielony przy wykorzystaniu 10 różnych ziaren.

W obu przypadkach model za każdym razem był trenowany od tych samych wag początkowych, co znaczy, że model uczony na jednym zbiorze nie był potem uczony na drugim zbiorze, oraz model uczony na jednym podziale z ziarna, nie był uczony na tym samym zbiorze z innym ziarnem podziału.

4.3. Stanowisko badawcze

W ramach przeprowadzanych badań modele były trenowane oraz testowane na trzech komputerach z następującymi podzespołami:

Komputer 1:

- Procesor: Intel Core I5-4460, 3,20 GHz
- Pamięć operacyjna: 16 GB
- Karta graficzna: AMD Radeon RX 6600, 8GB VRAM
- System operacyjny: Ubuntu 22.04

Komputer 2:

- Procesor: Intel Core I9-14900KF, 3,20 GHz
- Pamięć operacyjna: 64 GB
- Karta graficzna: Nvidia GeForce RTX 4070 Ti Super, 16 GB VRAM
- System operacyjny: Windows 11 Pro

Komputer 3:

- Procesor: AMD Ryzen 5 3600, 3,60 GHz
- Pamięć operacyjna: 16 GB
- Karta graficzna: AMD Radeon Rx 6800 xt, 16 GB VRAM
- System operacyjny: Ubuntu 22.04

4.4. Eksperyment 1

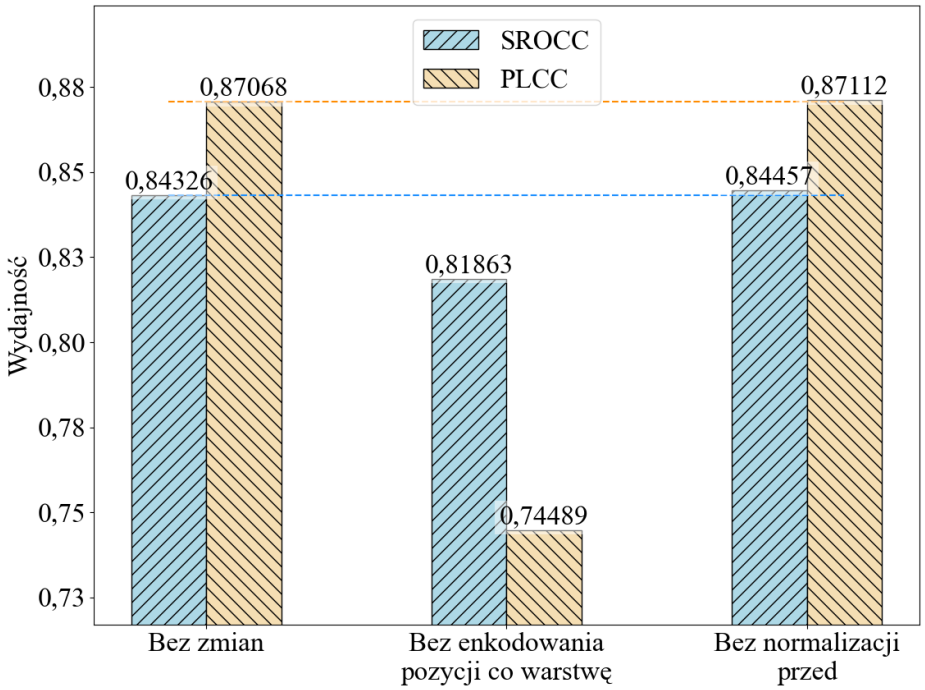
W pierwszym eksperymencie postanowiono sprawdzić wpływ zmian opisanych w rozdziale „3.3.2 Analiza implementacji - rozbieżność w implementacji transformera". Test ten został przeprowadzany na zbiorze danych CLIVE pięć razy tak jak opisano to rozdziale „4.2 Metodologia przeprowadzonych badań". Na Rys. 15 przedstawiono wykres słupkowy porównujący wydajność modelu po zastosowaniu zmian. Kolorem niebieskim oznaczono wydajność według współczynnika SROCC, a kolorem pomarańczowym wydajność modelu według współczynnika PLCC.

Pierwsze od lewej słupki reprezentują model, „bez zmian", czyli z zastosowaniem normalizacji na początku bloku transformera oraz z zastosowaniem dodawania enkodowania pozycji co warstwę transformera.

Następnie widoczne są słupki reprezentujące wydajność modelu „bez enkodowania pozycji co warstwę" transformatora. Tylko raz na początku przed transformerem.

Ostatnie słupki są dla modelu „bez normalizacji przed", co oznacza zastosowanie standardowej normalizacją na końcu bloku transformera.

Z przeprowadzonego badania wynika, że zastosowanie enkodowania pozycji co warstwę transformera polepszyło współczynnik SROCC o 3% a współczynnik PLCC o 15%. Z tego wynika, że ta modyfikacja ma znaczący wpływ na ostateczną wydajność modelu. Dodatkowo wpływ normalizacji na początku bloku względem normalizacji na końcu bloku transformera nie ma znaczącego wpływu na wydajność modelu.

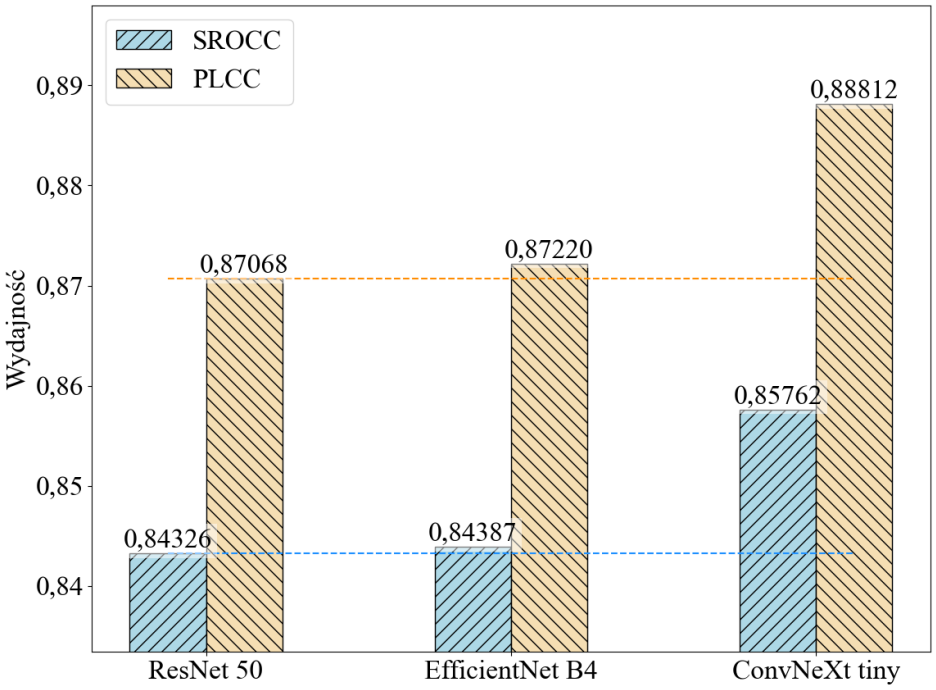


Rys. 15 Wpływ modyfikacji transformera, na wydajność modelu

4.5. Eksperyment 2

W drugim eksperymencie przeprowadzono testy jak na wydajność modelu wpływa zmiana ekstraktora cech. Szczegóły wyboru alternatywnych modeli ekstraktora cech opisano w rozdziale *„3.4.1 Zmiana ekstraktora cech”*. Warunki testu są takie same jak w poprzednim podrozdziale.

Na Rys. 16 przedstawiono wykres porównujący wydajność modelu w zależności od zastosowanego ekstraktora cech. Jak wynika z analizy wykresu wydajność alternatywnego ekstraktora cech w postaci modelu EfficientNet B4 nie daje zauważalnych zmian w spółczynnikach SROCC oraz PLCC. Przy zamianie ekstraktora cech na model ConvNeXt Tiny zauważono poprawę wydajności modelu w SROCC jak i PLCC o odpowiednio 1,7% oraz 1,74%.

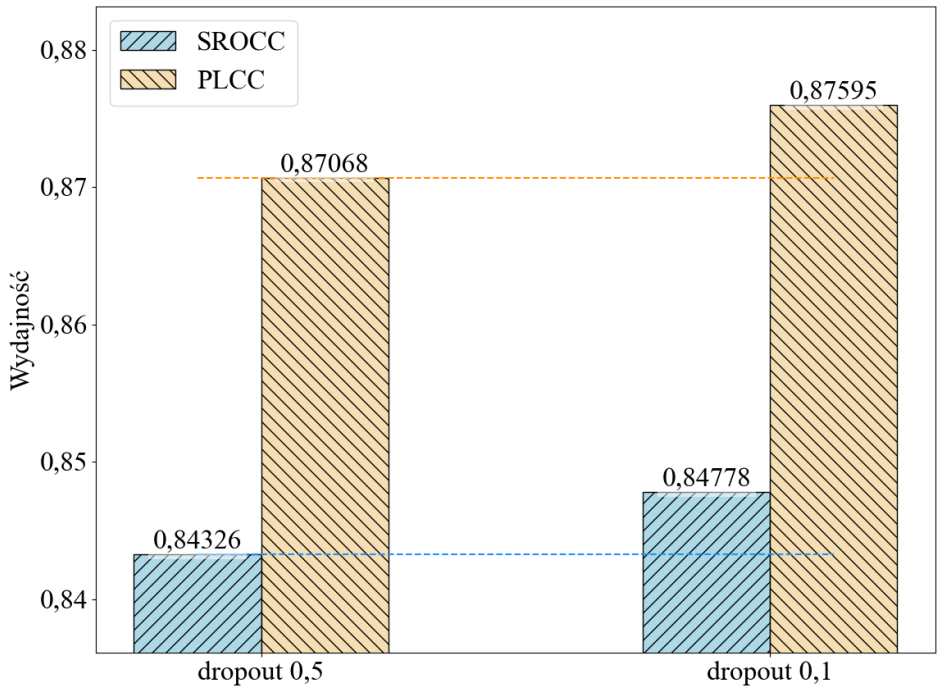


Rys. 16 Wpływ ekstraktora cech na wydajność modelu

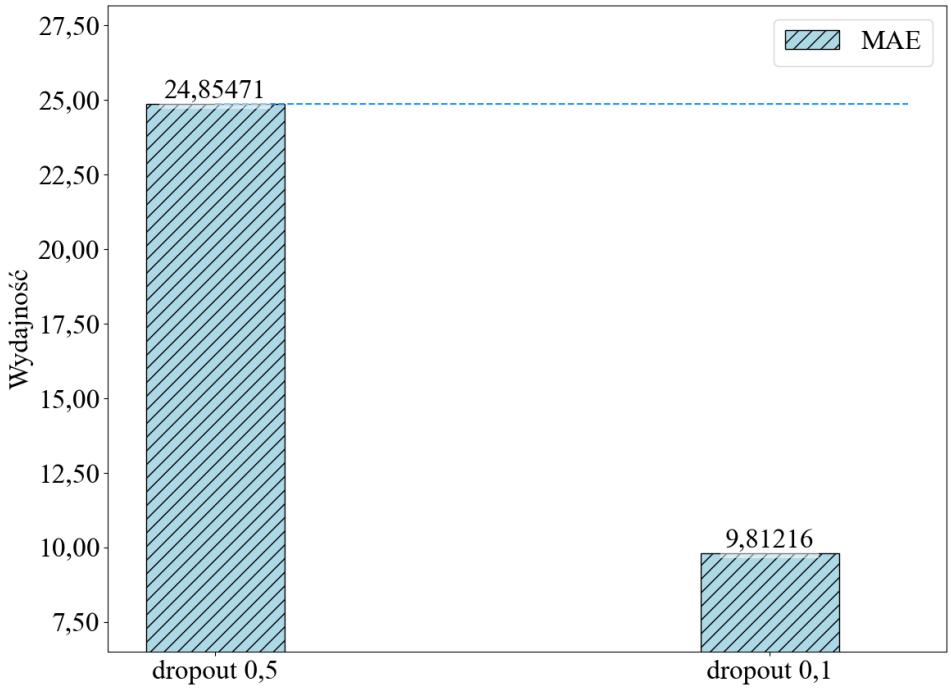
4.6. Eksperyment 3

W trzecim eksperymencie sprawdzono wpływ wartości dropout w blokach transformera na wydajność modelu. Podczas analizy kodu źródłowego oryginalnej implementacji zauważono stosunkowo duży współczynnik dropout wynoszący 0,5, jest on dosyć duży biorąc pod uwagę rozmiar wewnętrznej warstwy bloku FFN w transformerze wynoszący 64. Padło podejrzenie, że może to być zbyt duża wartość regularyzacji w stosunku do liczby neuronów. Warunki testu są takie same jak w *„4.4 Eksperyment 1”*.

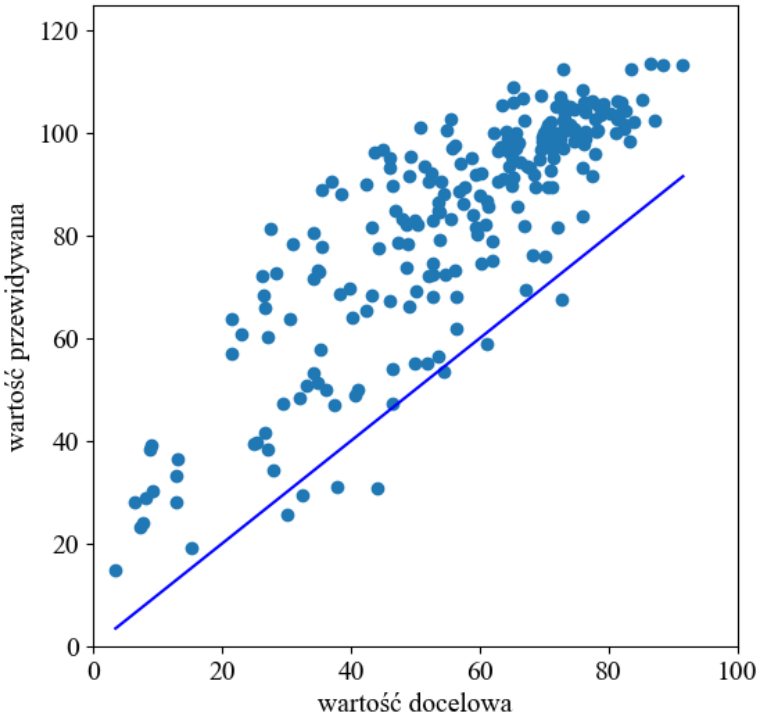
Na Rys. 17 pokazano porównanie wydajności dla wartości dropoutu wynoszącej odpowiednio 0,5 oraz 0,1. Zmniejszenie wartości dropoutu do 0,1 zwiększyło wydajność modelu we współczynnikach SROCC i PLCC odpowiednio o 0,54% oraz 0,6%. Natomiast zauważono dużo większą zmianę w średnim błędzie bezwzględnym, który został przedstawiony na Rys. 18. Zanotowano spadek tego współczynnika o 60%. Dodatkowo na Rys. 19 i Rys. 20 przedstawiono wykresy rozrzutu dla dropoutu odpowiednio 0,5 i 0,1. Pośród przeprowadzonych eksperymentów tylko zmniejszenie dropoutu pozwoliło zauważalnie zmniejszyć współczynnik MAE. Na osi poziomej jest wartość docelowa jakości obrazu a na osi pionowej wartość przewidywana jakości obrazu. W przypadku wykresu dla dropoutu 0,5 widoczne jest przesunięcie całego wykresu do góry, dodatkowo dla wartości dropoutu widać bardziej zacieśnione punkty. Co zostało odzwierciedlone przez poprawę współczynników SROCC i PLCC.



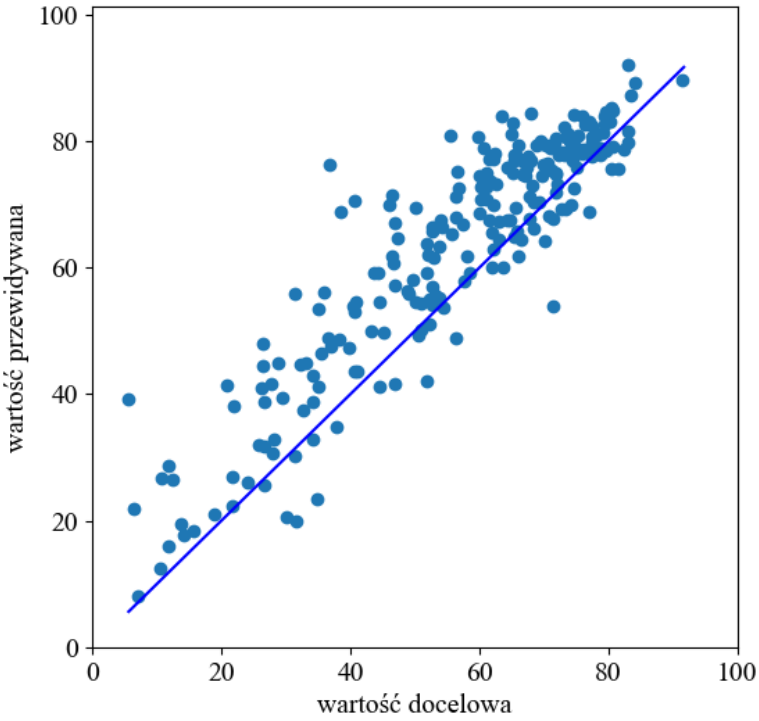
Rys. 17 Wpływ wartości dropoutu na SROCC i PLCC



Rys. 18 Wpływ wartości dropoutu na średni błąd bezwzględny



Rys. 19 Wykres rozrzutu dla dropoutu 0,5



Rys. 20 Wykres rozrzutu dla dropoutu 0,1

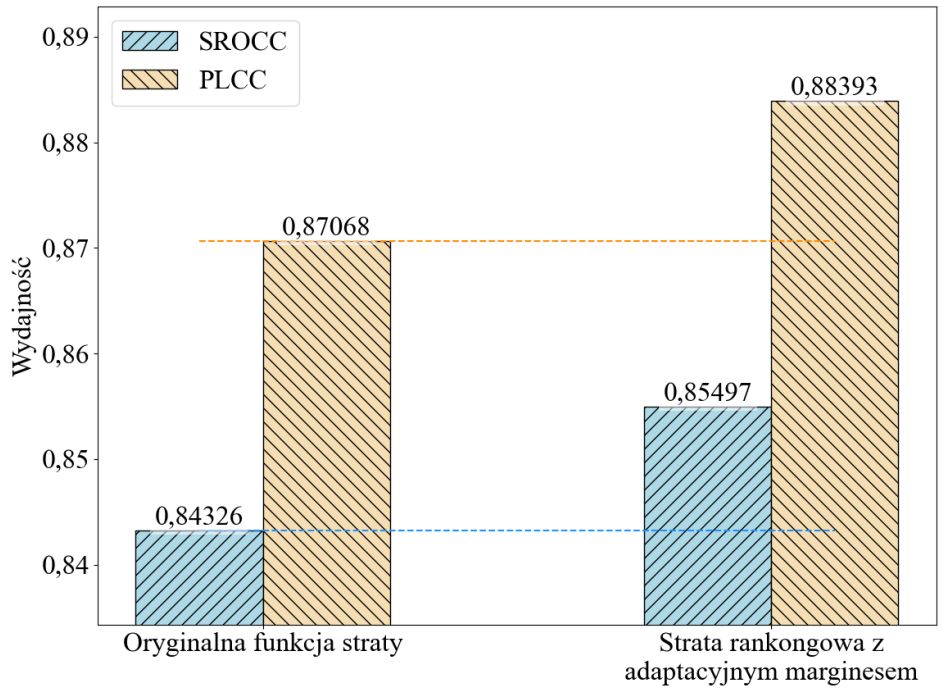
4.7. Eksperyment 4

W czwartym, ostatnim eksperymencie przeprowadzono testy wydajności z zastosowaniem proponowanej nowej funkcji strat opisanej w *„3.4.2 Strata rankingowa z adaptacyjnym* marginesem*”*. Dla współczynnika α zastosowano wartość 2. Dodatkowo proponowana funkcja strat była wspomagana funkcją strat z wykorzystaniem średniego błędu bezwzględnego. Wzór na ostateczną funkcję strat można przedstawić następująco (29):

$$L = L_{amr} + L_{mae} \tag{29}$$

Proponowaną funkcję strat zestawiono z oryginalną funkcją strat opisaną w rozdziale 3.3.1.

Na Rys. 21 przedstawiono porównanie oryginalnej funkcji strat wraz z proponowaną wcześniej funkcją strat we wzorze (29). Proponowana funkcja strat pozwoliła na zwiększenie wydajności wyrażoną współczynnikami SROCC i PLCC o odpowiednio 1,4% oraz 1,5%.



Rys. 21 Porównanie oryginalnej funkcji straty do proponowanej funkcji straty

4.8. Badanie zoptymalizowanej metody

Na podstawie wcześniej przeprowadzonych eksperymentów wybrano cechy, które zwiększały wydajność modelu. W pierwszym eksperymencie pokazano, że dodawanie kodowania pozycji co warstwę transformera zwiększa jego wydajność. Z drugiego eksperymentu wynika, że użycie sieci ConvNeXt Tiny jako ekstraktora cech zwiększa wydajność modelu. Z trzeciego eksperymentu wynika, że zmniejszenie dropoutu polepszyło znaczne średni błąd bezwzględny. Z ostatniego eksperymentu wynika, że proponowana funkcja strat daje lepsze wyniki.

Dlatego ostateczna ewaluacja odbędzie się na zoptymalizowanym modelu, z wykorzystaniem proponowanej funkcji strat, dodawaniem enkodowania pozycji co warstwę transformera, ekstraktorem cech w postaci wstępnie trenowanej sieci ConvNeXt Tiny oraz z wartością dropoutu na poziomie 0,1.

Ewaluacje zostaną przeprowadzone dziesięciokrotnie na każdym ze zbiorów KonIQ-10k, KADID-10k, CLIVE, LIVE, BID oraz TID2013, tak jak zostało to opisane w rozdziale „4.2 Metodologia przeprowadzonych badań”. Przy czym zbiory danych KADID-10k oraz KonIQ-10k były trenowane odpowiednio przez 30 i 50 epok, gdzie co 20 epok zmniejszano współczynnik uczenia 10-krotnie. Model na reszcie zbiorów był trenowany maksymalnie przez 150 epok oraz zmniejszano współczynnik uczenia co 50 epok.

Do trenowania wykorzystano optymalizator Adam, o początkowym współczynniku uczenia $1 \cdot 10^{-5}$ z regularyzacją L2 na poziomie $1 \cdot 10^{-4}$. Rozmiar partii wynosił 16 obrazów, z optymalizacją modelu co drugą partię, co tworzyło efektywną partię o rozmiarze 32.

Do porównania zoptymalizowanej metody zaczerpnięto wydajności innych metod z prac [9, 8].

W przypadku porównania metod na wykresach słupkowych, jeśli brakuje słupków dla danej metody, oznacza to, że dana metoda nie była testowana na danym zbiorze lub jej wydajność według danego współczynnika była mniejsza niż 10 percentyl.

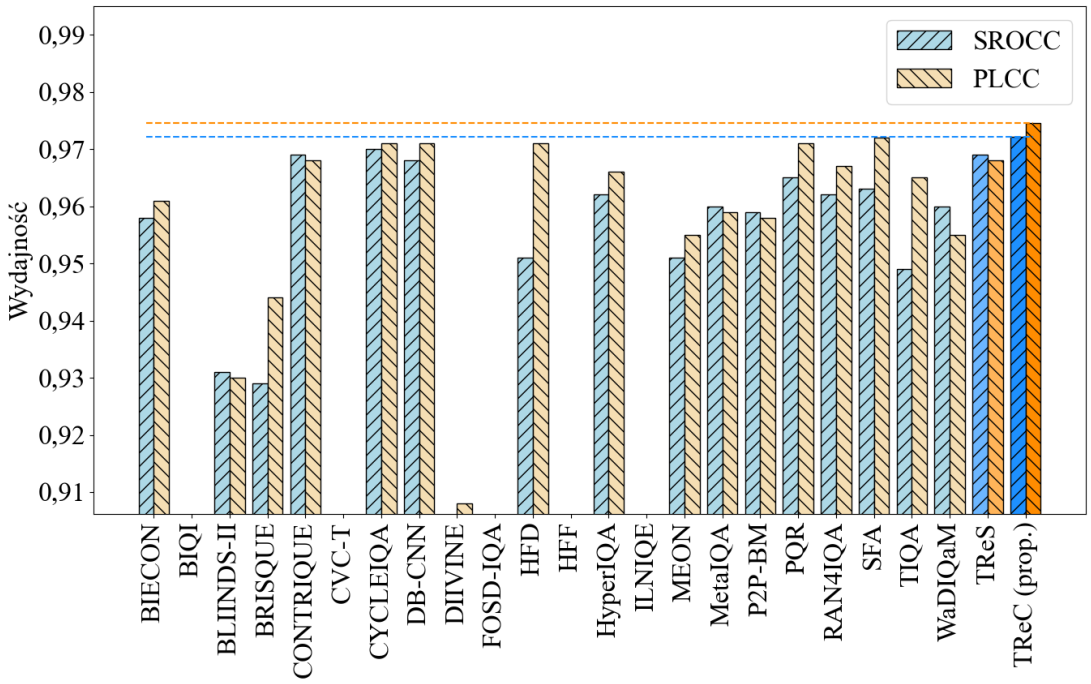
Na wykresach słupkowych w osi *y* oznaczono wydajność modelu, gdzie słupkiem niebieskim oznaczono miarę SROCC a pomarańczowym PLCC. Na osi *x* znajdują się nazwy porównywanych metod, gdzie ostatnią metodą jest proponowana zoptymalizowana metoda. Przedostatnią metodą jest oryginalna metoda TReS. Dodatkowo słupki proponowanej metody mają najciemniejszy odcień oraz słupki oryginalnej metody TReS posiadają ciemniejszy odcień niż reszta metod, ale jaśniejszy niż proponowana metoda.

Dokładne wartości wydajności metod zestawiono odpowiednio w Tabeli 6 oraz Tabeli 7, ze względu na to, że próba przedstawienia dokładnych wartości wydajności na wykresach słupkowych skutkowałaby zaczernieniem wykresu.

Dla zoptymalizowanej metody zaproponowano nazwę TReC (ang. ***T****ransformers,* ***Re****lative ranking with* ***C****onvNeXt*); nazwa została zainspirowana sposobem nazwania oryginalnej metody TReS.

4.8.1. LIVE

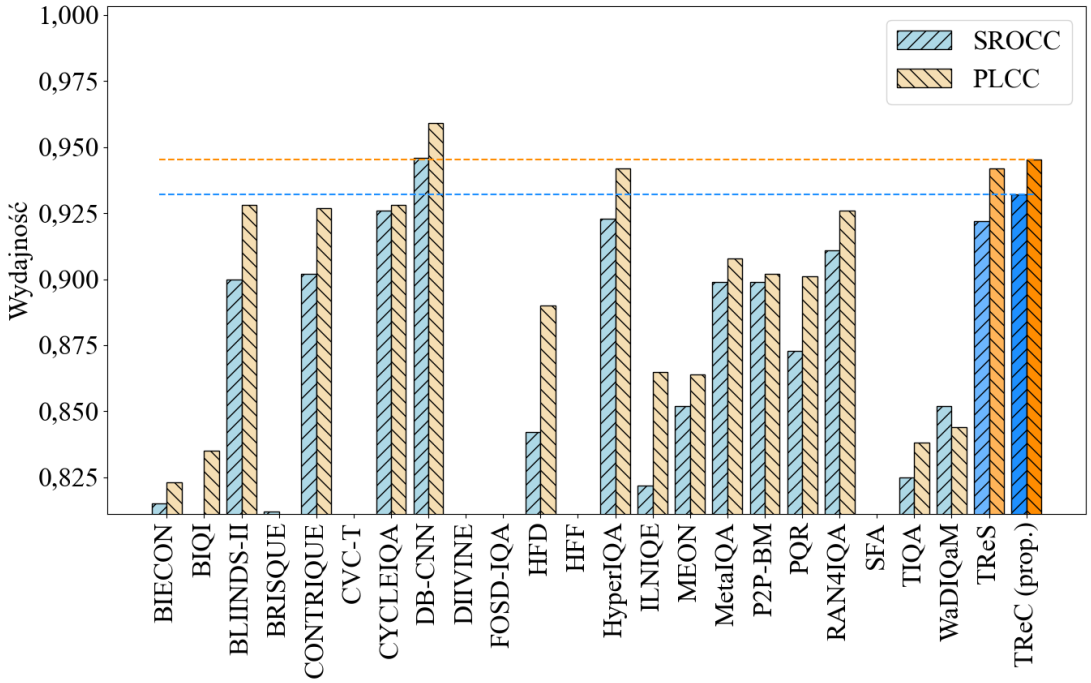
Na zbiorze danych syntetycznych LIVE, przedstawionym na wykresie słupkowym na Rys. 22 porównano wydajność z innymi metodami. Z wykresu wynika, że zoptymalizowana metoda osiąga najlepszą wydajność dla obu miar SROCC oraz PLCC. Przewyższa wydajność oryginalnej implementacji o 0,3% oraz 0,6% dla SROCC i PLCC.



Rys. 22 Porównanie wydajności modelu na zbiorze danych LIVE

4.8.2. CISQ

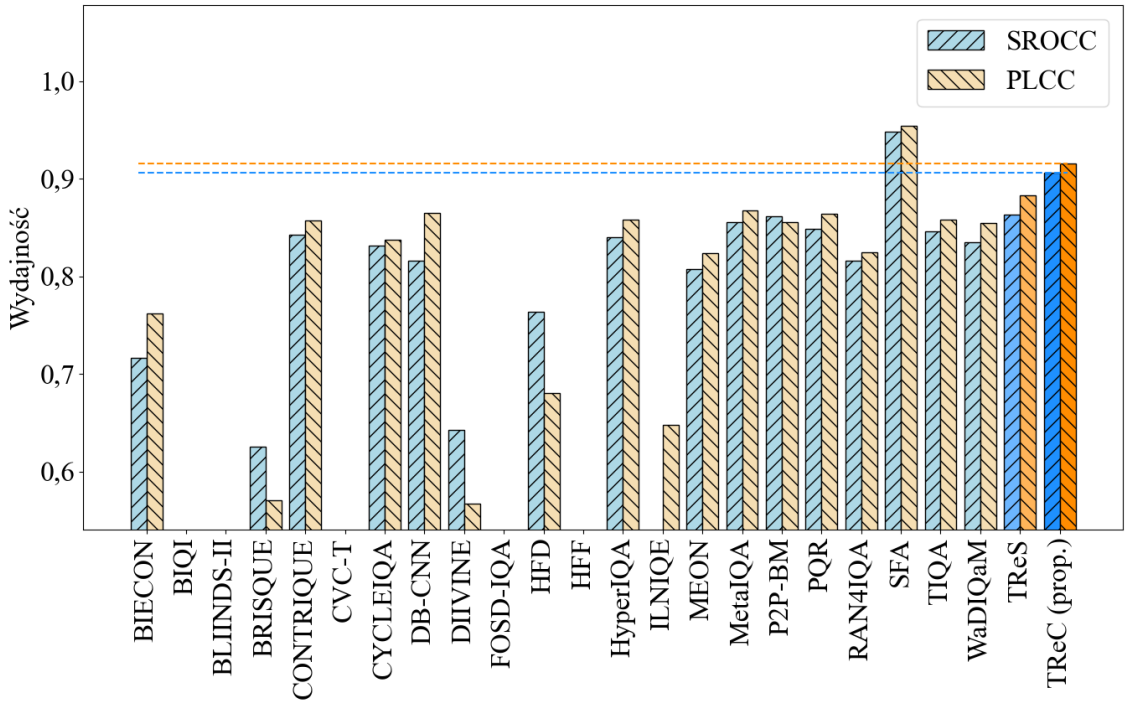
Drugie porównanie przeprowadzono na syntetycznym zbiorze danych CISQ. Wykres słupkowy porównujący wydajność metod przedstawiono na Rys. 23. Z wykresu wynika, że zoptymalizowana metoda osiąga drugą najlepszą wydajność dla obu miar SROCC oraz PLCC zaraz po metodzie DB-CNN. Zoptymalizowana metoda osiąga lepsze wyniki dla SROCC i PLCC o odpowiednio 1% i 0,3% względem oryginalnej implementacji.



Rys. 23 Porównanie wydajności modelu na zbiorze danych CISQ

4.8.3. TID2013

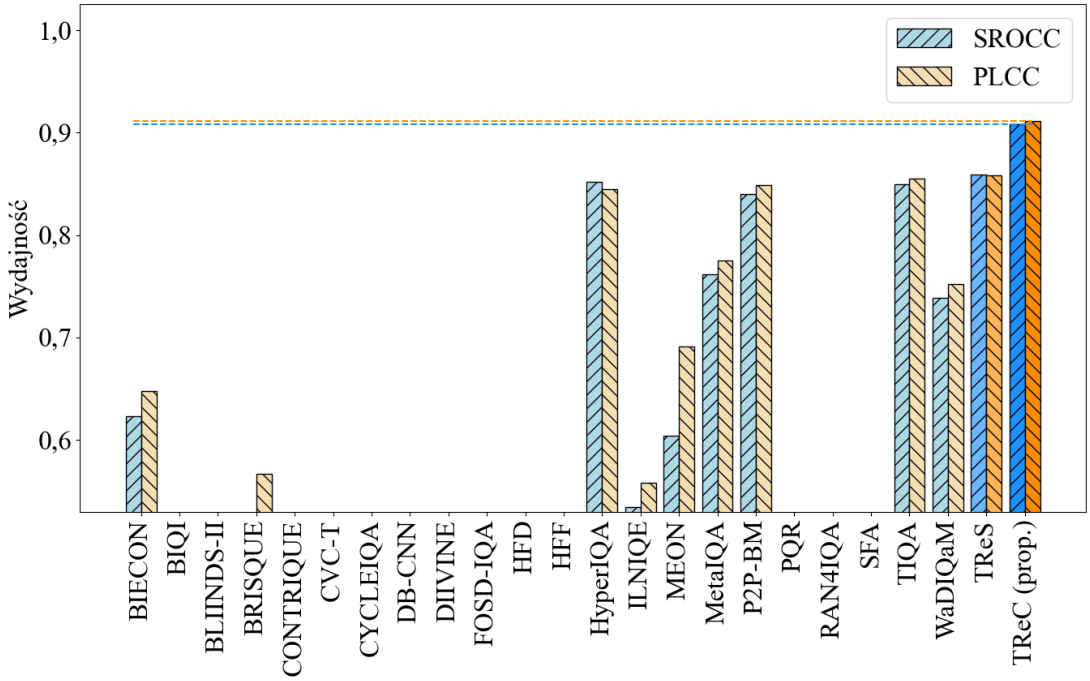
Trzecie porównanie przeprowadzono na syntetycznym zbiorze danych TID2013. Wykres słupkowy przedstawiający porównanie wydajności metod przedstawiono na Rys. 24. Z analizy wykresu wynika, że zoptymalizowana metoda osiągnęła drugi najwyższy wynik po metodzie SFA. Zoptymalizowana metoda osiąga znacząco lepsze wyniki dla SROCC i PLCC o odpowiednio 5% i 3,7% względem oryginalnej implementacji.



Rys. 24 Porównanie wydajności modelu na zbiorze danych TID2013

4.8.4. KADID-10k

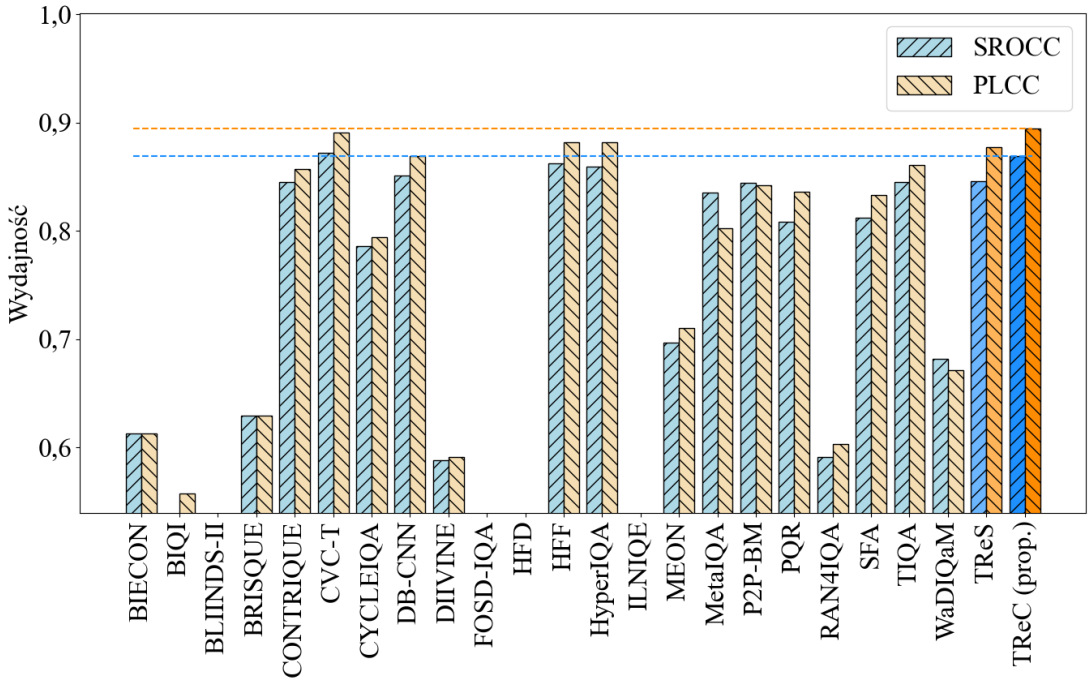
Ostatnie porównanie na syntetycznych zbiorach przeprowadzono na zbiorze danych KADID-10k. Wykres słupkowy z porównaniem metod przedstawiono na Rys. 25. Z analizy wykresu wynika, że zoptymalizowana metoda posiada najlepszą wydajność dla SROCC oraz PLCC. Dodatkowo względem oryginalnej implementacji uzyskano znacząco lepsze wyniki o 5,7% oraz 6,2% dla odpowiednio SROCC oraz PLCC.



Rys. 25 Porównanie wydajności modelu na zbiorze danych KADID-10k

4.8.5. CLIVE

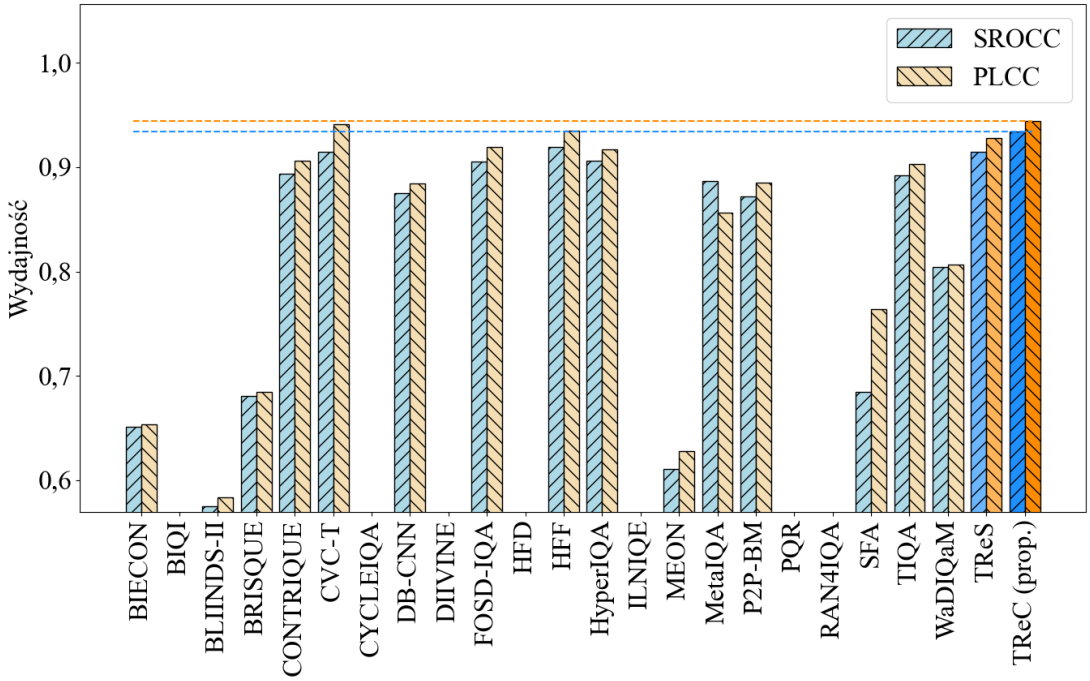
Czwarte porównanie przeprowadzono na autentycznym zbiorze danych CLIVE. Wykres słupkowy z porównaniem metod przedstawiono na Rys. 26. Z analizy wykresu wynika, że zoptymalizowana metoda osiąga najlepsze wyniki, na równi z metodą CVC-T. CVC-T osiągnęło lepszy współczynnik SROCC, a zoptymalizowana metoda osiągnęła lepszy współczynnik PLCC. Dodatkowo względem oryginalnej metody uzyskano lepsze wyniki o 2,7% oraz 2% dla odpowiednio SROCC oraz PLCC.



Rys. 26 Porównanie wydajności modelu na zbiorze danych CLIVE

4.8.6. KonIQ-10k

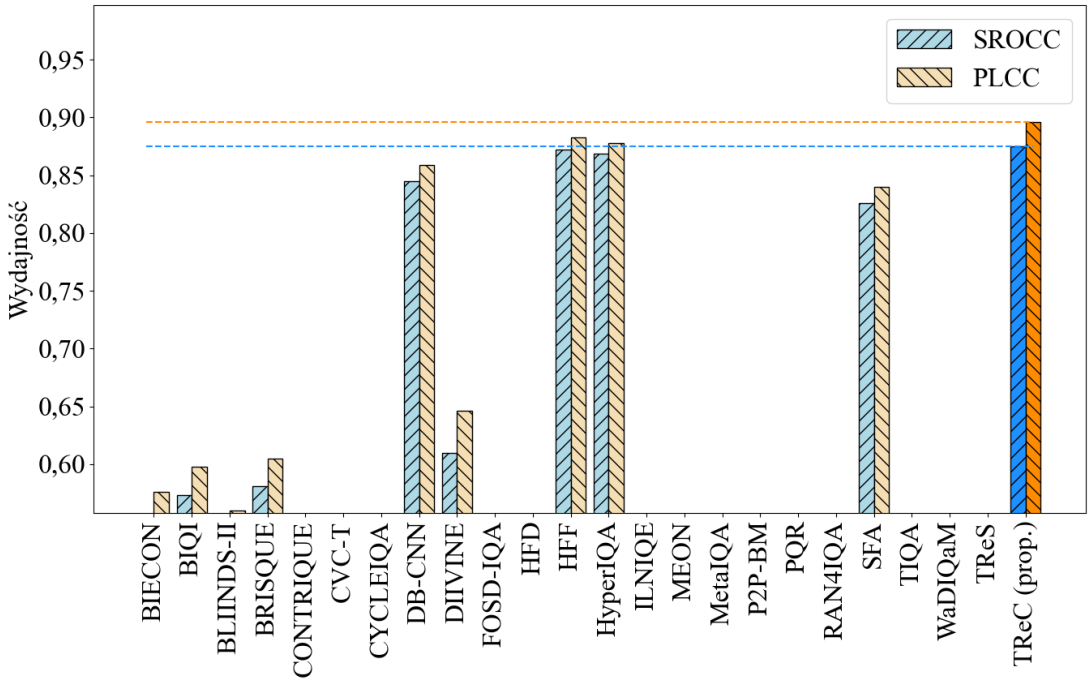
Piąte porównanie przeprowadzono na autentycznym zbiorze danych KonIQ-10k. Wykres słupkowy z porównaniem metod przedstawiono na Rys. 27. Z analizy wykresu wynika, że zoptymalizowana metoda osiąga najlepszy wynik dla SROCC oraz odrobinę lepszy wynik dla PLCC względem metody CVC-T. Względem oryginalnej metody uzyskano lepsze wyniki o 2% oraz 1,7% dla odpowiednio SROCC oraz PLCC.



Rys. 27 Porównanie wydajności modelu na zbiorze danych KonIQ-10k

4.8.7. BID

Ostatnie porównanie przeprowadzono na autentycznym zbiorze danych BID. Wykres słupkowy z porównaniem metod przedstawiono na Rys. 28. Z analizy wykresu wynika, że zoptymalizowana metoda osiąga najlepsze wyniki dla obu współczynników.



Rys. 28 Porównanie wydajności modelu na zbiorze danych BID

4.8.8. Zestawienie wydajności dla wszystkich zbiorów danych

W Tabeli 6 przedstawiono porównanie wydajności metod dla syntetycznych zbiorów danych. W Tabeli 7 przedstawiono porównanie wydajności metod dla autentycznych zbiorów danych. Kolorem zielonym zaznaczono najlepszą wydajność dla danego współczynnika w danym zbiorze danych. Natomiast kolorem niebieskim zaznaczono drugą najlepszą wydajność dla danego współczynnika w zbiorze danych. Jeśli metoda nie była testowana na zbiorze, to zostało to oznaczone symbolem „-”. Zoptymalizowana metoda znajduje się w ostatnim wierszu w każdej z tabel. Natomiast oryginalna implementacja metody znajduje się w przedostatnim wierszu w każdej z tabel.

Z analizy wyników z Tabeli 6 oraz Tabeli 7 wynika, że wydajność zoptymalizowanej metody daje lepsze wyniki w każdym zbiorze danych, gdzie zoptymalizowana metoda jest wydajniejsza we współczynniku SROCC średnio o 2,8% oraz o 2,4% wydajniejsza dla współczynnika PLCC.

Z analizy Tabeli 7 można zauważyć dodatkowo, że zoptymalizowana metoda osiągnęła wyniki 0,869 oraz 0,895 dla odpowiednio SROCC oraz PLCC, co jest wyższą wartością niż osobne zastosowanie modelu ConvNeXt Tiny lub rankingowej funkcji strat z adaptacyjnym marginesem, gdzie wartości dla tego zbioru wynosiły odpowiednio 0,858 oraz 0,888 dla modelu z modelu ConvNeXt Tiny oraz 0,855 i 0,884 dla modelu z rankingową funkcją strat z adaptacyjnym marginesem. To pokazuje, że zastosowanie tych zmian miało wpływ na optymalizację modelu, a w szczególności na wydajność według miary SROCC.

Tabela 6 Zestawienie wydajności metod dla syntetycznych zbiorów danych

	LIVE		CISQ		TID2013		KADID-10k	
Metoda	**SROCC**	**PLCC**	**SROCC**	**PLCC**	**SROCC**	**PLCC**	**SROCC**	**PLCC**
BIECON	0,958	0,961	0,815	0,823	0,717	0,762	0,623	0,648
BIQI	0,820	0,821	0,760	0,835	0,349	0,366	-	-
BLIINDS-II	0,931	0,930	0,900	0,928	0,536	0,538	-	-
BRISQUE	0,929	0,944	0,812	0,748	0,626	0,571	0,528	0,567
CONTRIQUE	0,969	0,968	0,902	0,927	0,843	0,857	-	-
CVC-T	-	-	-	-	-	-	-	-
CYCLEIQA	0,970	0,971	0,926	0,928	0,832	0,838	-	-
DB-CNN	0,968	0,971	0,946	0,959	0,816	0,865	-	-
DIIVINE	0,892	0,908	0,804	0,776	0,643	0,567	0,413	0,435
FOSD-IQA	-	-	-	-	-	-	-	-
HFD	0,951	0,971	0,842	0,890	0,764	0,681	-	-
HFF	-	-	-	-	-	-	-	-
HyperIQA	0,962	0,966	0,923	0,942	0,840	0,858	0,852	0,845
ILNIQE	0,902	0,906	0,822	0,865	0,521	0,648	0,534	0,558
MEON	0,951	0,955	0,852	0,864	0,808	0,824	0,604	0,691
MetaIQA	0,960	0,959	0,899	0,908	0,856	0,868	0,762	0,775
P2P-BM	0,959	0,958	0,899	0,902	0,862	0,856	0,840	0,849
PQR	0,965	0,971	0,873	0,901	0,849	0,864	-	-
RAN4IQA	0,962	0,967	0,911	0,926	0,816	0,825	-	-
SFA	0,963	0,972	-	-	0,948	0,954	-	-
TIQA	0,949	0,965	0,825	0,838	0,846	0,858	0,850	0,855
WaDIQaM	0,960	0,955	0,852	0,844	0,835	0,855	0,739	0,752
TReS	0,969	0,968	0,922	0,942	0,863	0,883	0,859	0,858
TReC (prop.)	0,972	0,974	0,932	0,945	0,906	0,916	0,908	0,911

Tabela 7 Zestawienie wydajności metod dla autentycznych zbiorów danych

	CLIVE		KonIQ-10k		BID	
Metoda	**SROCC**	**PLCC**	**SROCC**	**PLCC**	**SROCC**	**PLCC**
BIECON	0,613	0,613	0,651	0,654	0,539	0,576
BIQI	0,532	0,557	-	-	0,573	0,598
BLIINDS-II	0,463	0,507	0,575	0,584	0,532	0,560
BRISQUE	0,629	0,629	0,681	0,685	0,581	0,605
CONTRIQUE	0,845	0,857	0,894	0,906	-	-
CVC-T	0,872	0,891	0,915	0,941	-	-
CYCLEIQA	0,786	0,794	-	-	-	-
DB-CNN	0,851	0,869	0,875	0,884	0,845	0,859
DIIVINE	0,588	0,591	0,546	0,558	0,610	0,646
FOSD-IQA	-	-	0,905	0,919	-	-
HFD	-	-	-	-	-	-
HFF	0,862	0,882	0,919	0,935	0,872	0,883
HyperIQA	0,859	0,882	0,906	0,917	0,869	0,878
ILNIQE	0,508	0,508	0,523	0,537	-	-
MEON	0,697	0,710	0,611	0,628	-	-
MetaIQA	0,835	0,802	0,887	0,856	-	-
P2P-BM	0,844	0,842	0,872	0,885	-	-
PQR	0,808	0,836	-	-	-	-
RAN4IQA	0,591	0,603	-	-	-	-
SFA	0,812	0,833	0,685	0,764	0,826	0,840
TIQA	0,845	0,861	0,892	0,903	-	-
WaDIQaM	0,682	0,671	0,804	0,807	-	-
TReS	0,846	0,877	0,915	0,928	-	-
TReC (prop.)	0,869	0,895	0,934	0,944	0,875	0,896

5. Podsumowanie

W ramach tego rozdziału zostanie przybliżony opis wykonanych prac, następnie zostaną opisane wnioski płynące z tej pracy oraz przybliżone zostaną proponowane kierunki rozwoju oraz dalszych badań w dziedzinie bezreferencyjnej oceny jakości obrazów oraz dalszej optymalizacji metody TReS.

5.1. Opis wykonanych prac

W ramach niniejszej pracy dokonano analizy literatury dotyczącej bezreferencyjnej oceny jakości obrazu. Podczas analizy przyjrzano się problemowi percepcji obrazu, gdzie jakość obrazu często zależy nie tylko od jego ostrości czy rozkładu jasności, ale też od subiektywnej oceny estetyki czy kontekstu. Przeanalizowano również dostępne typy metod bezreferencyjnej oceny jakości obrazu, co miało kluczowy wpływ przy wyborze późniejszej metody do implementacji. W ramach tej analizy dokonano również wyboru zbiorów danych, takich jak: LIVE, CISQ, TID2013, KADID-10k, CLIVE, BID, KonIQ-10k. Wybór tych zbiorów był podykotowany popularnością ich użycia przy testowaniu innych metod bezreferencyjnej oceny jakości obrazu. Podczas analizy zbiorów danych zauważono wykorzystanie dwóch metryk oceny jakości obrazów, takich jak MOS i DMOS, z czego DMOS jest wykorzystywany wyłącznie w syntetycznych zbiorach, natomiast MOS jest wykorzystywany głównie w autentycznych zbiorach.

W części implementacyjnej zdecydowano się na zaimplementowanie metody o nazwie TReS, zdecydowano się na nią z powodu wykorzystywania przez nią w architekturze mechanizmu uwagi w formie transformera. Modele wykorzystujące transformer potrafią uchwycić zarówno lokalne jak i globalne zależności w obrazie, dzięki czemu uzyskują lepsze wyniki, a w szczególności na autentycznych zbiorach danych. W ramach implementacji modelu TReS dokonano jego dokładnej analizy, dzięki temu, że autorzy tego modelu udostępnili kod źródłowy w serwisie *github*, po przeprowadzeniu autorskiej implementacji możliwe było porównanie oryginalnej implementacji z tym co jej autorzy opisali w pracy. Podczas tej analizy zauważono niestandardową implementację transformera, gdzie kodowanie pozycyjne było

dodawane co warstwę transformera oraz tylko do wartości q oraz k oraz normalizacja została przeniesiona na początek bloku transformera. Zauważenie tych zmian podyktowało część eksperymentów w dalszej części pracy. Dodatkowo podczas analizy zauważono stosunkowo wysoką wartość dropoutu w transformatorze wynoszącą 0,5. Normalnie nie jest ta wartość za wysoka, ale biorąc pod uwagę, że liczba neuronów w warstwie bloku FFN wynosiła 64, sugerowało to, że model może się niedouczać. Kolejnymi etapami implementacji była próba znalezienia sposobu na zoptymalizowanie wydajności tej metody. Dlatego postanowiono zamienić oryginalnie wykorzystywany ekstraktor cech ResNet 50 na EfficientNet B4 oraz ConvNeXt Tiny. Dodatkowo podczas analizy oryginalnie wykorzystywanej funkcji strat do optymalizacji modelu TReS, zainspirowano się i zaproponowano wykorzystanie rankingowej funkcji strat z adaptacyjnym marginesem.

W części badawczej opisano dokładnie metodologię przeprowadzanych badań wraz z opisem wykorzystywanych miar, takich jak: SROCC, PLCC, MAE, do testowania wydajności modelu na zbiorach danych. W ramach części poświęconej metodologii przeprowadzonych badań opisano sposób podziału zbiorów danych na treningowy i testowy, przedstawiono zastosowane techniki sztucznego zwiększenia zbioru danych poprzez augmentację danych oraz zapewniono, że w syntetycznych zbiorach danych zdegradowane obrazy pochodzące od jednego obrazu źródłowego będą tylko w jednym podzbiorze, treningowym lub testowym. Później przeprowadzono cztery eksperymenty, z czego pierwszy polegał na zbadaniu wpływu różnic w implementacji transformera na wydajność modelu. Drugi eksperyment polegał na zbadaniu wpływu zmian ekstraktora cech na wydajność modelu. Trzeci eksperyment polegał na zbadaniu, czy wartość dropoutu 0,5 rzeczywiście była za duża i powodowała niedouczenie modelu. W ostatnim eksperymencie zbadano, czy proponowana rankingowa funkcja strat z adaptacyjnym marginesem wpłynie pozytywnie na wydajność modelu. Po przeprowadzeniu tych eksperymentów wyciągnięto wnioski i przeprowadzono ostateczną optymalizację metody. Na końcu przeprowadzono ostateczne badania zoptymalizowanej metody na wcześniej opisanych zbiorach danych oraz porównano zoptymalizowaną metodę do innych metod z aktualnego stanu wiedzy.

W ostatniej części pracy podsumowano całość pracy, wyciągnięto idące z tego wnioski oraz zaproponowano dalsze kierunki rozwoju.

5.2. Wnioski i dalsze kierunki badań

W ramach analizy literatury oraz przeprowadzonych badań nad zaimplementowaną metodą oraz zoptymalizowaną wersją tej metody nasuwają się następujące wnioski:

- przeprowadzona analiza literatury pozwoliła na zrozumienie aktualnego stanu wiedzy w dziedzinie bezreferencyjnej oceny jakości obrazu oraz wskazała model TReS jako jedną z najbardziej obiecujących metod do dalszych badań,
- implementacja modelu TReS umożliwiła dokładne poznanie jego struktury oraz identyfikację rozbieżności względem publikacji, pozwoliło to na zbadanie wpływu tych różnic na wydajność modelu,
- w ramach badań nad wpływem różnic w implementacji transformera zauważono, że dodawanie enkodowania pozycji co warstwę transformera, do wartości *q* oraz *k* zwiększa wydajność modelu o 3% dla SROCC oraz 15% dla PLCC. Standardowo enkodowanie pozycji dodaje się przed transformerami, co daje efektywnie dodawanie enkodowania pozycji do wszystkich trzech wartości *q, k* oraz *v*. Dlatego jako dalszy kierunek badań sugeruje się sprawdzenie, czy na poprawę wydajności miało wpływ dodawanie enkodowania pozycji co warstwę, czy dodanie enkodowania pozycji tylko do wartości *q* oraz *k*,
- w przypadku zaimplementowanego modelu zastosowanie normalizacji na początku bloku transformera, względem normalizacji na końcu bloku transformera, nie miało znaczącego wpływu na wydajność modelu,
- podczas zmiany ekstraktora cech z bazowego modelu ResNet 50 na EfficientNet B4 nie dało znaczącej poprawy wydajności, natomiast zamiana na ConvNeXt Tiny podniosła wydajność modelu o 1,7% oraz o 1,74% dla odpowiednio SROCC oraz PLCC. Sugeruje to, że model o większej wydajności dla klasyfikacji obrazów, wcale może nie radzić sobie lepiej z zadaniem w postaci bezreferencyjnej oceny jakości obrazów. Dlatego w celu wybrania optymalnego modelu do ekstrakcji cech z obrazu pod kątem bezreferencyjnej oceny jakości obrazu z wykorzystaniem architektury TReS, konieczne byłoby przeprowadzenie większej ilości badań pod kątem przetestowania większej liczby ekstraktorów cech,
- w ramach badań pokazano, że współczynnik dropoutu na poziomie 0,5 był zbyt wysoki i skutkował niedouczeniem modelu, co objawiało się słabym współczynnikiem MAE. Po zmniejszeniu współczynnika dropoutu do 0,1 zauważono znaczny spadek współczynnika MAE o 60%. Dodatkowo współczynniki SROCC i PLCC delikatnie wzrosły o odpowiednio 0,54% oraz

o 0,6%. Wszystkie te trzy współczynniki pokazują, że model zwiększył swoją wydajność. Wartość współczynnika dropoutu na poziomie 0,1 może nie być optymalną wartością dla tego modelu, dlatego w celu dalszej optymalizacji wydajności modelu możliwa jest dalsza optymalizacja tego współczynnika,

- podczas badań nad proponowaną rankingową funkcją strat z adaptacyjnym marginesem, zauważono wzrost wydajności modelu o 1,4% SROCC oraz o 1,5% PLCC względem oryginalnej funkcji strat. Dzięki temu pokazano, że samo ulepszenie funkcji strat może zwiększyć ostateczną wydajność trenowanego modelu. Proponowana funkcja strat uwzględniała względny ranking pomiędzy bezpośrednimi sąsiadami w partii. Jako ulepszenie tej metody sugeruje się branie pod uwagę rankingu pomiędzy wszystkimi sąsiadami w partii. Rozwiązanie to może być jednak zbyt skomplikowane obliczeniowo, dlatego konieczne byłyby dalsze badania w tym kierunku,
- w ostatnich badaniach pokazano, że zoptymalizowana metoda pozwoliła osiągnąć poziom aktualnego stanu wiedzy, gdzie we wszystkich testowanych zbiorach danych osiągnęła ona konkurencyjne wyniki a w większości autentycznych zbiorów danych zoptymalizowana metoda osiągnęła najlepsze wyniki. Przy czym dla każdego zbioru danych zoptymalizowana metoda przewyższyła oryginalną implementację TReS o średnio 2,8% dla SROCC oraz o 2,4% dla PLCC.

5.3. Informacje końcowe

W ramach niniejszej pracy dokonano analizy literatury, udało się zaimplementować oraz przebadać metodę TReS. Dodatkowo w ramach przeprowadzonych badań udało się zoptymalizować metodę TReS. Wszystkie poczynione usprawnienia zostały poparte odpowiednimi badaniami. Dzięki tym działaniom można stwierdzić, że postawiony cel pracy został zrealizowany.

Bibliografia

[1] M. Wang, Z. Xu, M. Xu i W. Lin, „Blind Multimodal Quality Assessment of Low-light Images," *arXiv,* nr 2303.10369, 2023.

[2] L. Wang, „A Survey on IQA," *arXiv,* nr 2109.00347, 2022.

[3] Z. Wang i E. Simoncelli, „Reduce-Reference Image Quality Assessment Using a Wavelet-Domain Natural Image Statistic Model," *Proceedings of SPIE - The International Society for Optical Engineering,* tom 5666, 2005.

[4] https://evidentscientific.com/en/microscope-resource/knowledge-hub/lightandcolor/humanvisionintro. K. R. Spring, T. J. Fellers i M. W. Davidson, „Human Vision and Color Perception," (dostęp: 20.8.2025)

[5] M. Wang, „Blind Image Quality Assessment: A Brief Survey," *arXiv,* nr 2312.16551, 2023.

[6] D. Ghadiyaram i A. Bovik, „Massive Online Crowdsourced Study of Subjective and Objective Picture Quality," *IEEE Transactions on Image Processing,* tom 25, nr 1, pp. 372-387, 01.2016.

[7] https://live.ece.utexas.edu/research/ChallengeDB/index.html. D. Ghadiyaram i A. Bovik, „LIVE In the Wild Image Quality Challenge Database," (dostęp: 2.06.2025)

[8] P. Yang, J. Sturtz i L. Qingge, „Progress in Blind Image Quality Assessment: A Brief Review," *Mathematics,* tom 11, nr 12, 2023.

[9] S. A. Golestaneh, S. Dadsetan i K. M. Kitani, „No-Reference Image Quality Assessment via Transformers, Relative Ranking, and Self-Consistency," *2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV),* pp. 3989-3999, 2022.

[10] http://live.ece.utexas.edu/research/quality. H. Sheikh, Z.Wang, L. Cormack i A. Bovik, „LIVE Image Quality Assessment Database Release 2," (dostęp: 2.czerwiec.2025)

[11] Z. Wang, A. Bovik, H. Sheikh i E. Simoncelli, „Image Quality Qssessment: From Error Visibility to Structural Similarity," *IEEE Transactions on Image Processing,* tom 13, nr 4, pp. 600- 612, 04.2004.

[12] H. Sheikh, M. Sabir i A. Bovik, „A Statistical Evaluation of Recent Full Reference Image Quality Assessment Algorithms," *IEEE Transactions on Image Processing,* tom 15, nr 11, pp. 3440-3451, 11.2006.

[13] N. Ponomarenko, L. Jin, O. Ieremeiev, V. Lukin, K. Egiazarian, J. Astola, B. Vozel, K. Chehdi, M. Carli, F. Battisti i C.-C. J. Kuo, „Image Database TID2013: Peculiarities, Results," *Signal Processing: Image Communication,* tom 30, pp. 57-77, 01.2015.

[14] N. Ponomarenko, O. Ieremeiev, V. Lukin, K. Egiazarian, L. Jin, J. Astola, B. Vozel, K. Chehdi, M. Carli, F. Battisti i C.-C. J. Kuo, „Color Image Database TID2013:

Peculiarities and Preliminary Results," *Proceedings of 4th Europian Workshop on Visual Information Processing EUVIP2013,* pp. 106-111, 10-12.06.2013.

[15] N. Ponomarenko, O. Ieremeiev, V. Lukin, K. Egiazarian, L. Jin, J. Astola, B. Vozel, K. Chehdi, M. Carli, F. Battisti i C.-C. J. Kuo, „A New Color Image Database TID2013," *Innovations and Results, Proceedings of ACIVS,* pp. 402-413, 10.2013.

[16] E. C. Larson i D. M. Chandler, „Most Apparent Distortion: Full-Reference Image Auality Assessment and The Role of Strategy," *Journal of Electronic Imaging,* tom 19, nr 1, 03.2010.

[17] L. Hanhe, H. Vlad i S. Dietmar, „KADID-10k: A Large-scale Artificially Distorted IQA Database," *2019 Tenth International Conference on Quality of Multimedia Experience (QoMEX), IEEE,* pp. 1-3, 2019.

[18] L. Hanhe, V. Hosu i D. Saupe, „DeepFL-IQA: Weak Supervision for Deep IQA Feature Learning," *arXiv preprint arXiv:2001.08113,* 2020.

[19] V. Hosu, H. Lin, T. Sziranyi i D. Saupe, „KonIQ-10k: An Ecologically Valid Database for Deep Learning of Blind Image Quality Assessment," *IEEE Transactions on Image Processing,* tom 29, pp. 4041-4056, 2020.

[20] https://github.com/zwx8981/UNIQUE. zwx8981, „UNIQUE," Github, (dostęp: 2.07.2025)

[21] W. Zhang, K. Ma, G. Zhai i X. Yang, „Uncertainty-Aware Blind Image Quality Assessment in The Laboratory and Wild," *IEEE Transactions on Image Processing,* tom 30, pp. 3474--3486, 2021.

[22] W. Zhang, K. Ma, G. Zhai i X. Yang, „Learning to Blindly Assess Image Quality in The Laboratory and Wild," *IEEE International Conference on Image Processing,* pp. 111-115, 2020.

[23] https://github.com/isalirezag/TReS. TRes, „Github," (dostęp: 14.07.2025)

[24] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg i L. Fei-Fei, „ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV),* tom 115, nr 3, pp. 211-252, 2015.

[25] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser i I. Polosukhin, „Attention Is All You Need," *arXiv,* p. 1706.03762, 2023.

[26] https://docs.pytorch.org/docs/stable/generated/torch.nn.TripletMarginLoss.html. „TripletMarginLoss," PyTorch, (dostęp: 24.7.2025)

[27] https://docs.pytorch.org/vision/stable/models. „Models and Pre-Trained Weights," PyTorch, (dostęp: 25.07.2025)

[28] https://docs.pytorch.org/docs/stable/generated/torch.nn.MarginRankingLoss.html. „MarginRankingLoss," PyTorch, (dostęp: 12.07.2025)

Spis skrótów i symboli

FR-IQA	Ocena jakości obrazu z pełnym odniesieniem (ang. *Full Reference Image Quality Assessment*)
RR-IQA	Ocena jakości obrazu z częściowym odniesieniem (ang. *Reduced Reference Image Quality Assessment*)
NR-IQA	Ocena jakości obrazu bez obrazu odniesienia (ang. *No Reference Image Quality Assessment*) lub
BIQA	Ślepa ocena jakości obrazu (ang. *Blind Image Quality Assessment*)
HVS	System wzrokowy człowieka (ang. *Human Vision System*)
NSS	Naturalne statystyki sceny (ang. *Natural Scene Statistics*).
DNN	Głęboka sieć neuronowa (ang. *Deep Neural Network*).
CNN	Konwolucyjna sieć neuronowa (ang. *Convolutional Neural Network*)
MOS	Średnia ocena subiektywna (ang. *Mean Opinion Score*)
DMOS	Różnicowa ocena subiektywna (ang. *Difference Mean Opinion Score*)
SROCC	Współczynnik korelacji rangowej Spearmana (ang. *Spearman Rank Order Correlation Coefficient*)
PLCC	Współczynnik liniowej korelacji Pearsona (ang. Pearson Linear Correlation Coefficient)
MAE	Średni błąd bezwzględny (ang. *Mean Absolute Error*)

Lista dodatkowych plików, uzupełniających tekst pracy

W systemie do pracy dołączono dodatkowe pliki zawierające:

- Kod źródłowy.

Spis rysunków

Spis tabel