

Politechnika Wrocławskiego
Wydział Informatyki i Telekomunikacji

Kierunek: **Zaufane Systemy Sztucznej Inteligencji (TAI)**

**PRACA DYPLOMOWA
MAGISTERSKA**

**Klasyfikacja wielomodalna przy użyciu metod
uczenia głębokiego na autorskim zbiorze
danych**

Jakub Grzana

Opiekun pracy
dr inż. Jakub Klikowski

Słowa kluczowe: klasyfikacja wielomodalna, uczenie głębokie, niebalansowany zbiór danych,
klasyfikacja wieloetykietowa

WROCŁAW 2024

STRESZCZENIE

W dzisiejszym cyfrowym świecie zasoby oraz media internetowe są tworzone przez użytkowników każdego dnia, w ilościach których manualne przetworzenie przez człowieka jest niemożliwe, pojawiła się potrzeba automatyzacji. Niniejsza praca prezentuje możliwości przetwarzania wielomodalnego obrazów za pomocą metod uczenia głębokiego w kontekście zadania klasyfikacji wieloetykietowej. Praca ta wnosi do dziedziny sztucznej inteligencji nowy, autorski zbiór danych zawierający 1542 obrazy wraz z etykietami i uzyskanymi dla niego wynikami dla modeli referencyjnych, a także innowatorską metodę ekstrakcji cech przy użyciu sieci konwolucyjnych oraz transformaty Fouriera.

ABSTRACT

In today's digital era, user-generated content and online media are produced daily in volumes that are unmanageable for humans to process manually, necessitating automation. This paper showcases the potential of multimodal image processing with deep learning techniques for multi-label classification tasks. It introduces a novel dataset of 1542 images, complete with labels, and presents the results from benchmark models. Additionally, it proposes an innovative feature extraction method that combines convolutional networks with the Fourier transform.

SPIS TREŚCI

1. Wprowadzenie	3
Cel i zakres pracy	3
2. Przegląd literatury	4
2.1. Klasyfikacja wielomodalna	4
2.2. Odczyt tekstu z obrazu - OCR	7
2.3. Augmentacja obrazów	8
2.4. Redukcja wymiarów	9
3. Zbiór danych	10
3.1. Etykiety	11
3.2. Statystyki	14
3.3. Zgrupowania i redukcja etykiet	20
4. Modele i metody	21
4.1. Przetwarzanie obrazu	22
4.1.1. Augmentacja danych	23
4.1.2. Modele ekstrakcji cech z obrazu	23
4.2. Przetwarzanie tekstu	24
4.2.1. Odczyt tekstu z obrazu	25
4.2.2. Modele ekstrakcji cech z tekstu	25
4.3. Generowanie nowej modalności: Transformata Fouriera	26
4.3.1. Reprezentacja transformaty	27
4.3.2. Zbiór danych	28
4.3.3. Model	29
4.3.4. Uczenie modelu FFT	29
4.4. Wykorzystanie PCA	30
4.5. Metody łączenia cech, normalizacja	30
4.6. Klasyfikator	32
4.7. Funkcja straty	33
4.8. Metryki	34
4.8.1. Progowanie, miary TP, TN, FP, FN	34
4.8.2. AUC	35
4.8.3. F1-score	36
4.8.4. AUPR	38
5. Wyniki	40

5.1. Przebieg badań	40
5.2. Wnioski	41
6. Podsumowanie	54
Bibliografia	55
Spis rysunków	59
Spis listingów	61
Spis tabel	62

1. WPROWADZENIE

W dzisiejszym cyfrowym świecie zasoby oraz media internetowe są tworzone przez użytkowników każdego dnia, w ilościach których manualne przetworzenie przez człowieka jest niemożliwe. Pojawiła się potrzeba automatyzacji, celem m.in. moderacji i odfiltrowywania treści nieodpowiednich (*NSFW*), czy też automatycznych rekomendacji. Wdrożenie systemów informatycznych realizujących te zadania wymaga zaprojektowania oraz implementacji modelu sztucznej inteligencji, a także dużych ilości obrazów z etykietami dla procesu uczenia.

Obecnie dostępne zbiory danych pochodzące z internetu są stosunkowo niewielkie, silnie ukierunkowane na konkretne tematy i przeznaczone głównie do wykrywania mowy nienawiści lub treści o charakterze seksualnym. Brakuje bardziej ogólnego zbioru, lepiej reprezentującego obrazy najczęściej udostępniane przez użytkowników internetu. Z tych powodów, celem tej pracy jest zebranie, zbadanie i opisanie nowego zbioru danych, o możliwie szerokim spektrum zawartości, który umożliwi pracę z tymi chaotycznymi, nieustrukturyzowanymi obrazami.

CEL I ZAKRES PRACY

Celem pracy jest przygotowanie autorskiego zbioru danych składającego się z obrazów i tekstów wraz z etykietami. Przygotowane dane zostaną poddane ewaluacji za pomocą wybranych referencyjnych modeli i metod uczenia głębokiego do klasyfikacji wieloetykietowej danych wielomodalnych. Ponadto opracowana zostanie autorska metoda (*wykorzystująca transformate Fouriera*), która zostanie poddana analizie porównawczej z metodami referencyjnymi.

Praca ta ma na celu wniesienie wkładu w dziedzinę sztucznej inteligencji poprzez dostarczenie kompleksowego, dobrze opisanego zbioru danych oraz zbadanie innowacyjnych technik poprawiających efektywność modeli klasyfikacji wielomodalnej.

2. PRZEGŁĄD LITERATURY

Zadanie stworzenia zbioru danych wielomodalnych oraz dostarczenia dla niego wyników referencyjnych wymaga zrozumienia kilku kluczowych aspektów. Przede wszystkim, konieczne jest zdefiniowanie pojęcia modalności oraz klasyfikacji wielomodalnej, jak również zapoznanie się z architekturami modeli przeznaczonych do jej realizacji. Niezbędne jest także poznanie metod maszynowego odczytu tekstu z obrazu oraz sposobów efektywnego wykorzystania ograniczonego zbioru danych poprzez techniki augmentacji. Ważnym elementem jest również zrozumienie metod redukcji wymiarowości. W tym rozdziale przedstawiam przegląd literatury dotyczącej każdego z tych zagadnień.

2.1. KLASYFIKACJA WIELOMODALNA

Zaprojektowanie modelu sztucznej inteligencji do pracy z zasobami pochodząymi od użytkowników internetu nie jest zadaniem trywialnym. Każdy użytkownik internetu może tworzyć obrazy bądź modyfikować istniejące według własnej wizji i umiejętności, przez co obrazy te są niejednorodne pod względem jakości, formatu oraz treści, często wymagają też znajomości szerszego kontekstu, posiadania zewnętrznej wiedzy [42]. Z tego powodu często wykorzystuje się w tym celu przetwarzanie wielomodalne.



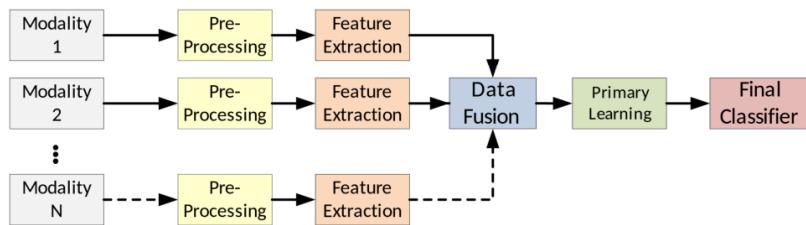
Rys. 2.1: Wpływ kontekstu i zewnętrznej wiedzy na interpretacje grafiki [21]

Modalność oznacza *stosunek osoby mówiącej do treści jej wypowiedzi, wyrażanego za pomocą środków językowych* [5], co można interpretować jako wykorzystanie formy oraz treści (*obrazu*) celem przekazania wiadomości. W kontekście sztucznej inteligencji, jako modalność najczęściej traktuje się różne sposoby reprezentacji danych wejściowych, np. obraz, odczytany z niego tekst, histogram kolorów [42]. Rys. 2.1 dobrze obrazuje istotność analizowania jednej modalności (*tekstu*) w kontekście innej (*obrazu*) - zarówno na obrazie goryla, jak i zdjęciu Anny Frank umieszczone napisy które samodzielnie są całkowicie niegroźne, natomiast umieszczone w kontekście nabierają charakteru mowy nienawiści.

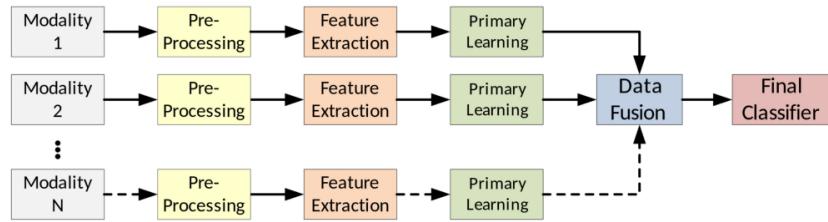
Klasyfikacja wielomodalna w dostępnej obecnie literaturze jest tematem stosunkowo niszowym, porównując z typową klasyfikacją obrazów czy tekstu. Ponadto w kontekście mediów internetowych najczęściej wykorzystywana jest głównie do wykrywania mowy nienawiści, co jest tematem pokrewnym acz odrębnym od bardziej ogólnej klasyfikacji, którą zajmuję się w ramach tej pracy. Dostępne zbiory danych najczęściej są stosunkowo niewielkie: przykładowo MultiOFF [46] wykorzystuje zbiór jedynie 743 obrazów do uczenia oraz ewaluacji modelu wykrywania mowy nienawiści (*klasyfikacja binarna*). Negatywny wpływ małego zbioru danych można ograniczyć poprzez zastosowanie transfer learningu, co MultiOFF [46] uczynił dla kanału (*modalności*) graficznego, wykorzystując model VGG16 wytrenowany wcześniej na zbiorze danych ImageNet [23], natomiast to samo zrobić można z kanałem tekstowym poprzez wykorzystanie np. modeli pochodnych BERTa [26].

Istnieje wiele metod projektowania modeli wielomodalnych. Typowa architektura składa się z modułów do wstępnego przetwarzania danych modalności oraz ekstrakcji cech, które następnie są łączone i przekazywane do klasyfikatora. (*jednej warstwy gęstej, dokonującej klasyfikacji*). Najczęściej modele wielomodalne wykorzystują też dodatkowe warstwy sieci do osiągnięcia lepszych wyników, umieszczane typowo przed algorytmem łączenia cech (*Late Fusion*) albo przed warstwą klasyfikatora końcowego (*Early Fusion*) [44]. Obydwie architektury przedstawiono na rysunkach 2.2 i 2.3. W ramach tej pracy zdecydowałem się na realizację architektury typu *Early Fusion* ze względu na jej popularność w literaturze.

Poruszając temat samych modalności: oprócz standardowych (*graficznej, tekstopowej*) wygenerować można wiele innych. Przykładowo, publikacja "*Detecting Hate Speech in multi-modal memes*" [21] wprowadza jako modalność tekst opisujący obraz, wygenerowany za pomocą modelu realizującego *image captioning* [21]. Praca "*Multi-channel CNN for*



Rys. 2.2: Przykładowa architektura sieci wykorzystująca *Early Fusion* [44]



Rys. 2.3: Przykładowa architektura sieci wykorzystująca *Late Fusion* [44]

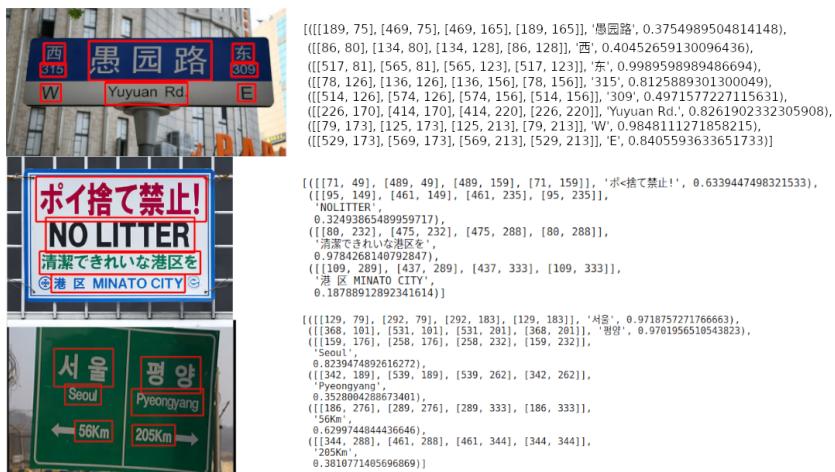
precise meme classification" [42] proponuje zastosowanie histogramu, co może być pomocne nie tylko dla zadania klasyfikacji internetowych memów, ale również do rozróżniania obrazów generowanych komputerowo od zdjęć.

Osobnym zagadnieniem w ramach klasyfikacji wielomodalnej jest algorytm łączenia cech wyekstraktowanych z poszczególnych modalności. Najbardziej naturalną metodą na połączenie kilku wektorów cech jest konkatenacja, natomiast artykuł "*Fakeddit: A New Multimodal Benchmark Dataset for Fine-grained Fake News Detection*" [35] proponuje alternatywne algorytmy: uśrednianie wartości cech, sumowanie ich bądź wybieranie maksimum.

2.2. ODCZYT TEKSTU Z OBRAZU - OCR

Odczytywanie tekstu z obrazu samo w sobie jest kolejnym zagadnieniem: musi to być proces zautomatyzowany, więc konieczny jest do tego osobny model, system optycznego rozpoznawania znaków (*OCR*). Istnieje wiele projektów i implementacji systemów *OCR*, najbardziej znane to rozwijany przez Google *Tesseract* [7] oraz *Texttract* [1] Amazonu.

Systemy *OCR* mogą wykorzystywać wiele różnych technologii, niekoniecznie bazowanych na głębokich sieciach neuronowych. Przykładowo *Tesseract* do trzeciej generacji wykorzystywał jedynie oparte na uczeniu maszynowym dopasowywanie wzorców - algorytm wyszukiwał wzorce na obrazie, wycinał z niego glif, czyli fragment obrazu potencjalnie zawierający literę i porównywał ze wszystkimi glifami w bazie danych [12] - co osiągało bardzo dobre wyniki, natomiast tylko dla dokumentów o znanej czcionce. Późniejsze modele wykorzystywały algorytmy przetwarzania obrazów (*np. detekcje krawędzi*) w połączeniu z uczonym maszynowo klasyfikatorem. Współcześnie dziedzina ta jest zdominowana przez modele wykorzystujące uczenie głębokie, szczególnie sieci konwolucyjne oraz LSTM.



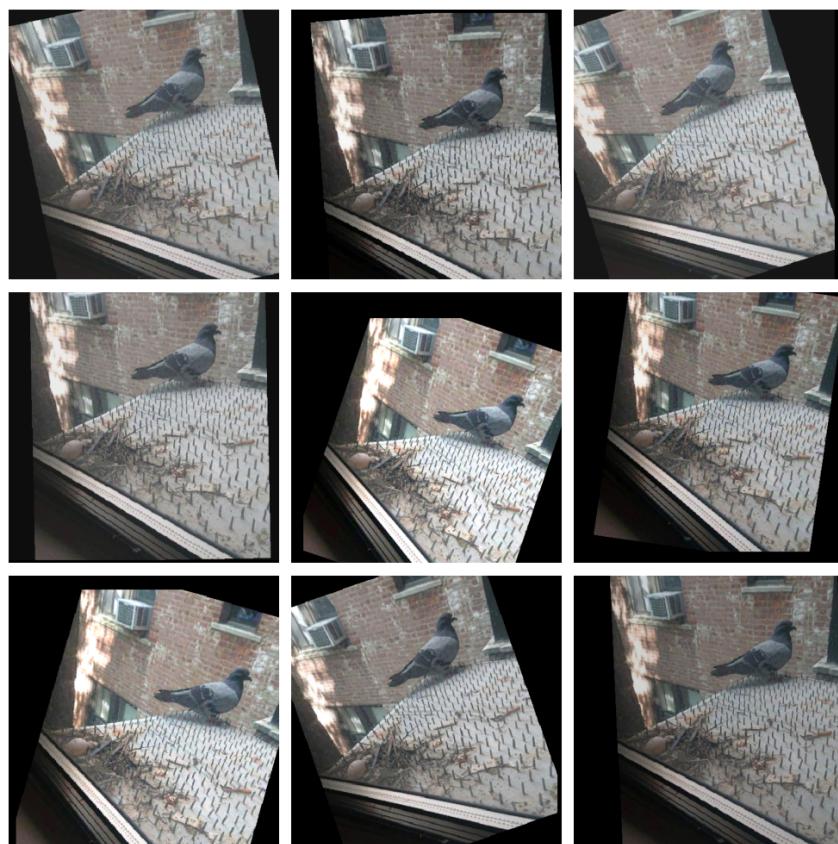
Rys. 2.4: Przykład działania OCR [30]

Na potrzeby tej pracy zdecydowałem się wybrać model EasyOCR [30]. Jet to model wykorzystujący sieci konwolucyjne, wytrenowany na stosunkowo różnorodnym zbiorze danych zawierającym m.in. skany dokumentów tekstowych, ale również zdjęcia znaków drogowych wykonanych przez kamery samochodowe a także zdjęcia plakatów i reklam - co powinno umożliwić mu pracę ze słabo ustrukturyzowanymi obrazami. Przykład działania przedstawiono na rysunku 2.4: model wyszukuje tekst, skanując obraz z góry na dół, od lewej do prawej. Znaleziony tekst dany jest *bounding boxem* określającym jego położenie na obrazie, a także prawdopodobieństwem (*stopniem pewności modelu*).

Według "Multi-channel CNN for precise meme classification" [42] spośród popularnych modeli Keras-OCR [33], Py-Tesseract [20] oraz EasyOCR ten ostatni osiąga najlepsze rezultaty dla zbiorów danych Memotion [40] oraz Memegenerator [15].

2.3. AUGMENTACJA OBRAZÓW

Augmentacja jest procesem, który poprzez losowe manipulacje obrazem tworzy z niego technicznie nowy, nie widziany wcześniej przez sieć obraz, sztucznie zawyżając rozmiar zbioru danych. Augmentowany obraz nie jest tak wartościowy jak obraz zupełnie nowy, natomiast umiejętne zastosowanie augmentacji może poprawić zdolność generalizacji modelu przy niewielkim nakładzie pracy. Typowe techniki augmentacji danych to: przerzut w pionie i poziomie, obrót o mały kąt, przesunięcie na osi X i Y, zbliżenie na losowy punkt, drgania kolorów (*ang. color jitter*) [8], **Nie wszystkie techniki nadają się do pracy z danymi wielomodalnymi:** w szczególności tekst widoczny na obrazie staje się nieczytelny po przeprowadzeniu przerzutu w pionie czy poziomie.



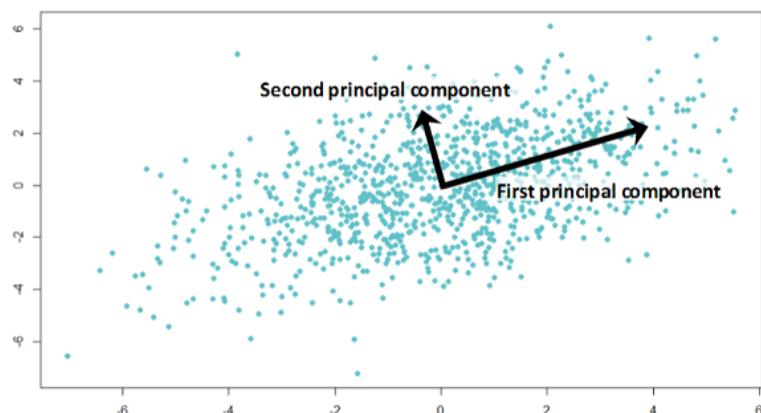
Rys. 2.5: Przykład przetworzonego i augmentowanego obrazu

Przykład augmentacji przedstawiłem na rysunku 2.5: z jednego obrazu wejściowego wygenerowałem dziewięć technicznie unikatowych obrazów. Zastosowanie tych operacji, oprócz zapewniania pozornie nieskończonej liczby obrazów, pomaga również np. nauczyć się abstrahowania obiektu od jego pozycji na obrazie, w załączonym przykładzie ten sam obiekt (*gołąb*) znajduje się w wielu różnych częściach obrazu, co teoretycznie powinno ułatwić sieci nauczenie się nie "czy *gołąb* znajduje się w tej i tej części obrazu", zamiast tego skupiając się na jego relacji względem innych obiektów.

2.4. REDUKCJA WYMIARÓW

Redukcja wymiarów to proces zmniejszania liczby zmiennych losowych podlegających rozważaniu poprzez uzyskanie zestawu głównych (*tzn. zawierających największą część informacji*) zmiennych. Techniki redukcji wymiarowości są popularnie wykorzystywane w modelach sztucznej inteligencji, ponieważ niewielkim kosztem skuteczności mogą w dużym stopniu ograniczyć zapotrzebowanie modelu na zasoby. Dla modeli wielomodalnych redukcja wymiarowości ma jeszcze inne zalety, które opiszę szerzej w rozdziale *Modele i metody: Metody łączenia cech, normalizacja*. Jedną z popularnych technik jest PCA (*Principle Component Analysis*).

U podstaw, analiza PCA polega na przekształceniu za pomocą liniowych operacji danych wejściowych do nowego układu współrzędnych, gdzie każda os (składowa główna) jest od wszystkich pozostałych niezależna (*są do siebie ortogonalne*). Składowe główne wybierane są jedna po drugiej w procesie dopasowywania modelu PCA do danych, w taki sposób by każda składowa maksymalizowała (*wyjaśniała*) wariancję względem danej osi, zachowując ortogonalność względem pozostałych - jest to dobrze widoczne na rysunku 2.6 [37].

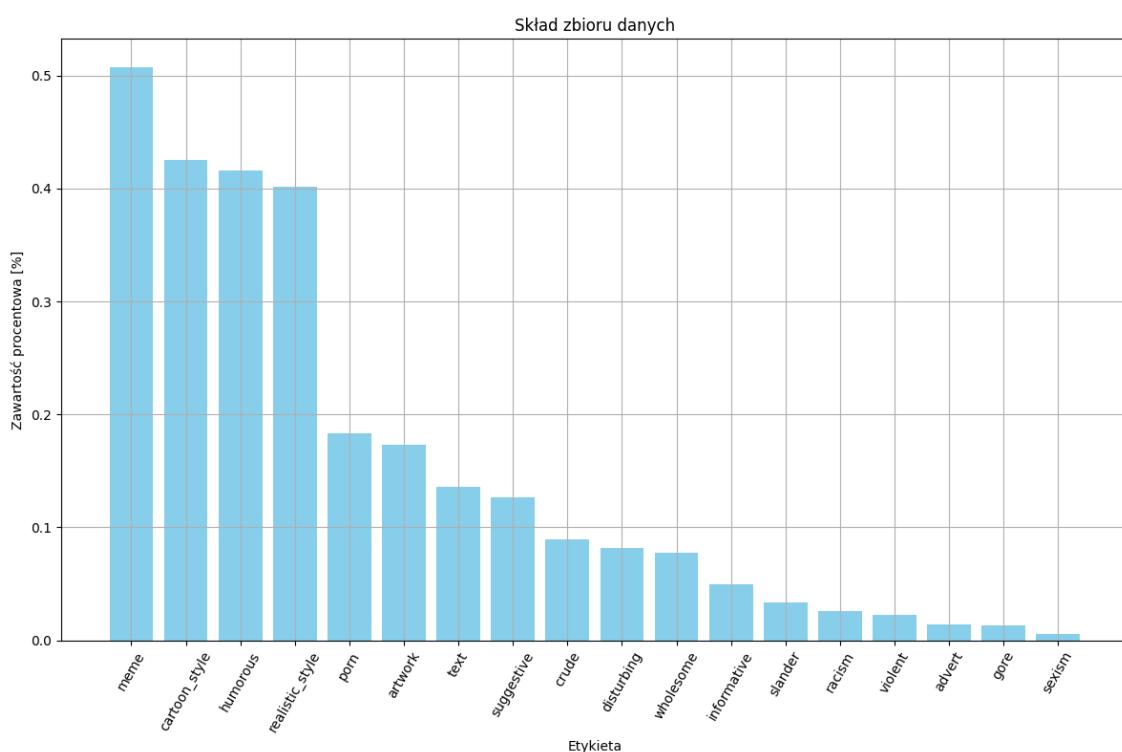


Rys. 2.6: Przykład działania PCA [17]

Ponieważ za każdym razem wybierana jest składowa główna o największej wariancji, każda kolejna składowa opisuje mniejszą część wariancji oryginalnych danych niż składowa poprzednia. Dla N cech możemy znaleźć N składowych głównych, natomiast ze względu na korelacje między cechami najczęściej nie jest to konieczne i znacznie mniejsza ich liczba wystarczy, by zachować zdecydowaną większość informacji w nich się zawierających.

3. ZBIÓR DANYCH

Zbiór danych zebrany w ramach pracy liczy sobie 76 163 obrazów, z czego 1542 posiada ręcznie przydzielone etykiety - **w ramach pracy wykorzystana została jedynie część z etykietami.** Obrazy zbierane były w okresie od lipca do października 2023 roku z bardzo różnych źródeł - większą część zbioru danych zebrałem osobiście bez używania tzw. scraperów, zadbałem również o wkład osób z innych środowisk niż moje, aby zapewnić różnorodność i ograniczyć *bias* zbioru danych. Na zbiór oprócz obrazów składają się przydzielone do nich etykiety oraz tekst odczytany za pomocą EasyOCR [30]. Zawartość procentowa zbioru danych została przedstawiona na rysunku 3.1, dokładne wartości umieszczone w tabeli 3.1 - wewnętrz tabeli wpisano również średnią liczbę próbek oraz średni udział etykiet należących do zbioru danych, wartości te są istotne przy analizie niektórych metryk.



Rys. 3.1: Zawartość procentowa etykiet w zbiorze danych

Etykieta	Liczność	Udział
meme	782	0.507
cartoon_style	656	0.425
humorous	642	0.416
realistic_style	619	0.401
porn	283	0.184
artwork	267	0.173
text	209	0.136
suggestive	195	0.126
crude	138	0.089
disturbing	126	0.082
wholesome	120	0.078
informative	77	0.050
slander	52	0.034
racism	40	0.026
violent	35	0.023
advert	22	0.014
gore	21	0.014
sexism	9	0.006
Średnia	239	0.154
Średnia (RD)	457	0.296

Tabela 3.1: Liczność i udział etykiet w zbiorze danych

3.1. ETYKIETY

Etykiety zostały przydzielone zgodnie z poniższymi definicjami oraz przykładami. Definicje te nie są ścisłe, więc subiektywna opinia osoby dokonującej etykietyzacji oraz jej *bias* mają tutaj duże znaczenie. Subiektywną naturę przydzielonych etykiet można określić ilościowo obliczając współczynnik Kappa Fleissa [3], natomiast to wymagałoby etykietyzacji tych samych obrazów przez wiele osób - w ramach tej pracy zdecydowałem się zamiast tego skupić się na etykietyzacji jak największej części zbioru. Większość etykiet w zbiorze została nadana przeze mnie osobieście.

Realistic style Odnosi się do stylu obrazu: zawiera w sobie rzeczywiste zdjęcia, a także realistyczne obrazy generowane komputerowo.

Cartoon style Odnosi się do stylu obrazu: zawiera w sobie komiksy, kreskówki, mangę, anime oraz rysunki.

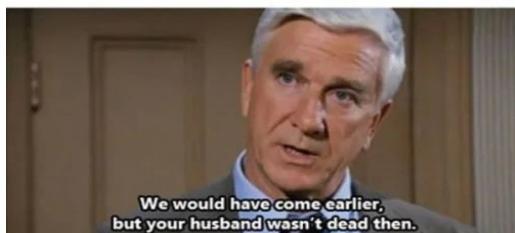
Meme Czy obraz "dąży do rozprzestrzeniania", tzn. czy wywołuje chęć do przesłania go dalej.

Advert Czy obraz jest reklamą, tzn. czy próbuje sprzedać produkt bądź usługę. Dobrym przykładem są: plakaty, billboardy.

Informative Czy obraz przekazuje jakąś konkretną informację lub idee, niekoniecznie

prawdziwą. Typowo: infografiki, figury z artykułów naukowych, notatki studenckie.

Text	Czy obraz składa się z czystego tekstu, tzn. czy można pominąć wszelkie elementy graficzne i sens obrazu zostanie zachowany. Przykłady to: zdjęcia notatek, prezentacji, postów w serwisach społecznościowych.
Artwork	Czy obraz przedstawia "artwork": rysunek artystyczny.
Suggestive	Czy na obrazie wyeksponowana jest seksualność, zmysłowość. Zawiera, m.in. artystyczną nagość.
Porn	Pornografia.
Sexism	Mowa nienawiści skierowana wobec płci: męski i żeński szowinizm, transfobia, homofobia.
Racism	Mowa nienawiści skierowana wobec rasy, narodowości czy religii.
Slander	Mowa nienawiści skierowana wobec konkretnej jednostki.
Crude	Wulgarne/niekulturalne obrazy.
Gore	Krwawe, brutalne obrazy, <i>body horror</i> .
Violent	Obrazy przedstawiające przemoc, np. uliczną, domową.
Humorous	Czy obraz został stworzony celem rozbawienia, niezależnie od efektu.
Disturbing	Obrazy które wywołują niepokojące uczucie, niekoniecznie przez efekt szoku, tylko ponieważ coś jest z nimi nie-tak. Dobrym przykładem jest efekt <i>uncanny valley</i> [34]
Wholesome	Urocze, miłe obrazki.



(a) realistic style, meme, slander, humorous



(b) cartoon style, meme, racism, humorous



THEONION.COM
Apartment Broker Recommends Brooklyn Residents Spend No More Than 150% Of Incom...

(c) realistic style, meme, text, humorous



(d) realistic style, meme, humorous, wholesome



(e) cartoon style, artwork



(f) realistic style, humorous, wholesome

Rys. 3.2: Przykładowe obrazy ze zbioru danych, wraz z etykietami

3.2. STATYSTYKI

Spośród 1542 próbek **1174 (76.1% zbioru)** zawiera tekst. Wyłącznie dla próbek zawierających tekst zmierzylem: długość tekstu (*w tokenach*) przypadającego na próbę oraz długość tokenu (*w znakach*), wyniki przedstawiono na rysunku 3.4 oraz umieszczone w tabeli 3.2. Do tokenizacji wykorzystałem tokenizer modelu BERT [26].

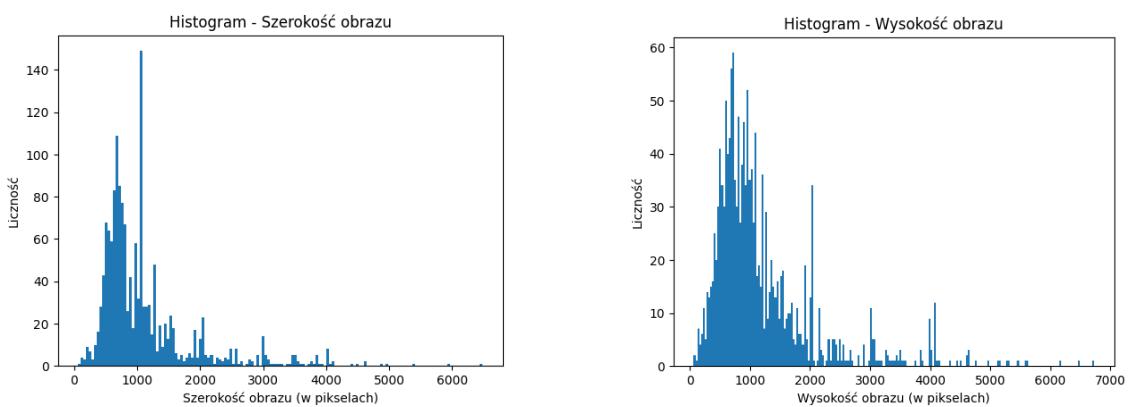
Średnia długość tekstu wraz z odchyleniem standardowym jest szczególnie użyteczna przy wyborze *maksymalnej liczby tokenów*, używanej przez tokenizer wybranego modelu tekstowego. Z przyczyn implementacyjnych, tekst musi być reprezentowany wektorem tokenów o stałej długości - dłuższe teksy są przycinane (*truncation, powoduje utratę informacji*), krótsze są wypełniane specjalnymi tokenami (*padding, może negatywnie wpływać na proces uczenia*). Korzystając z obliczonych wartości oraz histogramu, możemy wybrać taką długość bufora, która pomieści większość próbek ze zbioru danych bez przycinania lub ograniczy zbędne wypełnianie.

Łączna ilość tokenów w zbiorze wynosi 54055 - mniej niż w typowej powieści [18] - z czego 6856 (12.7%) jest unikatowe. **Wszystkie teksty w zbiorze danych są w języku angielskim.** Każdy token składa się z 3 ± 1.8 znaków, co oznacza że większość tokenów w zebranym zbiorze danych jest raczej krótka. Dane te sugerują, że zbiór zawiera zbyt mało danych tekstowych do samodzielnego zadań przetwarzania języka naturalnego. Jednakże, mogą one być użyteczne w zadaniu klasyfikacji wielomodalnej, jako uzupełnienie innych modalności.

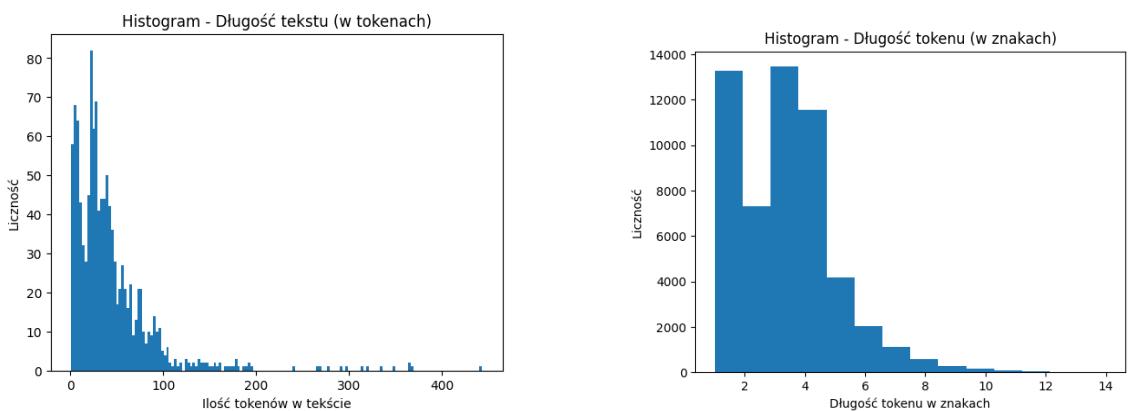
Wysokość i szerokość obrazu (*dane w pikselach*) znajdują się w tabeli 3.2 oraz na rysunku 3.3. Wartości te mogą być przydatne przy okazji uczenia własnej sieci np. konwolucyjnej na zbiorze danych.

	Minimum	Średnia	Odchylenie Standardowe	Maksimum
Szerokość obrazu (w pikselach)	61	1110	764	6480
Wysokość obrazu (w pikselach)	61	1192	883	6734
Długość tekstu na próbce (w tokenach)	1	46.0	82.8	2224
Długość tokenu (w znakach)	1	3.06	1.75	14
Etykiety na próbce	0	2.78	1.02	6

Tabela 3.2: Statystyki obrazów w zbiorze danych



Rys. 3.3: Statystyki obrazów w zbiorze danych



Rys. 3.4: Statystyki tekstu w zbiorze danych

Celem ułatwienia pracy ze zbiorem, zmierzyłem średnią i odchylenie standardowe każdego kanału obrazów dla całego zbioru danych. Wyniki umieściłem w tabeli 3.3 w postaci wartości pikseli modelu RGB (*dla czytelności*) oraz po normalizacji do przedziału 0..1 - w szczególności te drugie są przydatne w pracy z sieciami głębkimi, umożliwiając lepszą normalizację do zakresu (-1, 1) z jaką najczęściej pracują współczesne modele.

	Wartości pikseli (0..255)		Wartości pikseli (0..1)	
	Średnia	Odchylenie Standardowe	Średnia	Odchylenie Standardowe
R (<i>Czerwony</i>)	103.0	54.8	0.410	0.215
G (<i>Zielony</i>)	92.6	53.5	0.365	0.210
B (<i>Niebieski</i>)	87.4	53.5	0.343	0.210

Tabela 3.3: Średnia i odchylenie standardowe wartości pikseli w zbiorze danych

Przeprowadziłem również pomiar stopnia skorelowania między poszczególnymi parami etykiet. Stopień korelacji zdefiniowałem zgodnie ze wzorem 3.1,

$$Korelacja(L, P) = \frac{\bar{D}_{(L,P)}}{\min(\bar{D}_L, \bar{D}_P)} \quad (3.1)$$

gdzie L, P to etykiety, \bar{D}_X określa licznosć etykiety X w zbiorze D , $\bar{D}(X, L)$ określa licznosć próbek zawierających zarówno etykietę L i P . Wyniki umieściłem w tabeli 3.4, pomijając te o stopniu korelacji ≤ 0.05 .

Zdefiniowany w ten sposób współczynnik korelacji określa jaka czescią próbek etykiety mniej licznej należy również do etykiety bardziej licznej. Pewna korelacja między etykietami jest oczekiwana. Dla dwóch całkowicie niezależnych od siebie etykiet L i P o udziale X oraz Y wartość współczynnika korelacji wynosi $\approx \max(X, Y)$. Wynika to z teorii prawdopodobieństwa, gdzie w próbkach etykiety L spodziewamy się takiego samego udziału etykiety P jak w całym zbiorze danych, przy założeniu ich niezależności. Wartości powyżej tego progu sugerują *korelację dodatnią*, co oznacza nielosowy związek między próbками danych etykiet, skutkujący ich częstszym współwystępowaniem. Natomiast wartości poniżej progu sugerują *korelację ujemną*, wskazując na nielosowy związek, który sprawia, że etykiety te wzajemnie się wykluczają.

Wysokie wartości współczynnika korelacji mogą powodować problemy - dla przykładu korelacja etykiety *humorous* z większą etykietą *meme* wynosi ≈ 0.99 , co oznacza, że *humorous* jest podzbiorem etykiety *meme* i najlepiej jest wykluczyć ją z badań. W przypadku etykiet *artwork* oraz *cartoon style* korelacja wynosi ≈ 0.84 - jest nieco zbyt wysoka, przez co sieć może mieć problem ze znalezieniem cech które te dwie etykiety rozróżniają.

Tabela 3.4: Korelacja etykiet

Para etykiet	Liczność wspólna	Liczność etykiety 1	Liczność etykiety 2	Korelacja
meme, humorous	636	782	642	0.99
meme, sexism	8	782	9	0.89
meme, slander	45	782	52	0.87
cartoon_style, artwork	224	656	267	0.84
meme, racism	32	782	40	0.80
meme, text	155	782	209	0.74
cartoon_style, gore	15	656	21	0.71
slander, humorous	37	52	642	0.71
sexism, humorous	6	9	642	0.67
racism, humorous	25	40	642	0.62
cartoon_style, porn	173	656	283	0.61
text, humorous	126	209	642	0.60

Para etykiet	Liczność wspólna	Liczność etykiety 1	Liczność etykiety 2	Korelacja
informative, text	45	77	209	0.58
cartoon_style, sexism	5	656	9	0.56
cartoon_style, suggestive	107	656	195	0.55
realistic_style, advert	12	619	22	0.55
meme, violent	19	782	35	0.54
gore, disturbing	11	21	126	0.52
meme, crude	71	782	138	0.51
cartoon_style, crude	70	656	138	0.51
violent, humorous	17	35	642	0.49
realistic_style, wholesome	58	619	120	0.48
meme, wholesome	58	782	120	0.48
realistic_style, meme	299	619	782	0.48
realistic_style, racism	19	619	40	0.47
cartoon_style, disturbing	59	656	126	0.47
meme, suggestive	91	782	195	0.47
cartoon_style, violent	16	656	35	0.46
crude, humorous	63	138	642	0.46
meme, disturbing	56	782	126	0.44
realistic_style, slander	23	619	52	0.44
cartoon_style, meme	290	656	782	0.44
realistic_style, disturbing	52	619	126	0.41
realistic_style, humorous	252	619	642	0.41
porn, crude	56	283	138	0.41
cartoon_style, wholesome	47	656	120	0.39
cartoon_style, racism	15	656	40	0.38
realistic_style, violent	13	619	35	0.37
realistic_style, porn	105	619	283	0.37
cartoon_style, humorous	237	656	642	0.37
realistic_style, crude	50	619	138	0.36
suggestive, humorous	70	195	642	0.36
humorous, wholesome	42	642	120	0.35
porn, gore	7	283	21	0.33
cartoon_style, slander	17	656	52	0.33
realistic_style, informative	25	619	77	0.32
meme, informative	25	782	77	0.32
artwork, suggestive	61	267	195	0.31
realistic_style, suggestive	58	619	195	0.30

Para etykiet	Liczność wspólna	Liczność etykiety 1	Liczność etykiety 2	Korelacja
humorous, disturbing	37	642	126	0.29
realistic_style, gore	6	619	21	0.29
porn, disturbing	36	283	126	0.29
gore, violent	6	21	35	0.29
violent, disturbing	10	35	126	0.29
artwork, wholesome	33	267	120	0.28
cartoon_style, advert	6	656	22	0.27
artwork, porn	69	267	283	0.26
suggestive, crude	33	195	138	0.24
artwork, gore	5	267	21	0.24
artwork, disturbing	29	267	126	0.23
meme, advert	5	782	22	0.23
advert, informative	5	22	77	0.23
advert, porn	5	22	283	0.23
advert, humorous	5	22	642	0.23
realistic_style, sexism	2	619	9	0.22
informative, sexism	2	77	9	0.22
text, sexism	2	209	9	0.22
text, slander	11	209	52	0.21
meme, gore	4	782	21	0.19
gore, humorous	4	21	642	0.19
artwork, violent	6	267	35	0.17
informative, humorous	13	77	642	0.17
text, suggestive	29	209	195	0.15
crude, disturbing	15	138	126	0.12
realistic_style, artwork	31	619	267	0.12
text, violent	4	209	35	0.11
crude, violent	4	138	35	0.11
artwork, sexism	1	267	9	0.11
porn, sexism	1	283	9	0.11
sexism, racism	1	9	40	0.11
text, racism	4	209	40	0.10
suggestive, disturbing	12	195	126	0.10
crude, gore	2	138	21	0.10
text, wholesome	11	209	120	0.09
realistic_style, text	19	619	209	0.09
cartoon_style, informative	7	656	77	0.09

Para etykiet	Liczność wspólna	Liczność etykiety 1	Liczność etykiety 2	Korelacja
advert, artwork	2	22	267	0.09
advert, suggestive	2	22	195	0.09
text, disturbing	11	209	126	0.09
slander, crude	4	52	138	0.08
meme, artwork	19	782	267	0.07
text, crude	9	209	138	0.07
artwork, crude	9	267	138	0.07
informative, slander	3	77	52	0.06
cartoon_style, text	12	656	209	0.06
suggestive, violent	2	195	35	0.06
porn, violent	2	283	35	0.06

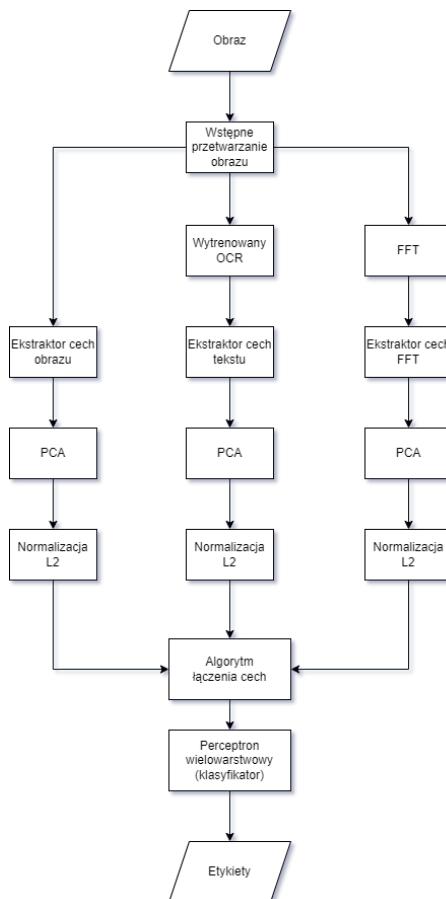
3.3. ZGRUPOWANIA I REDUKCJA ETYKIET

Część etykiet ze zbioru danych można połączyć w grupy, co jest szczególnie przydatne w przypadku tych o niskiej liczności. Np. etykiety *sexism*, *racism*, *slander* można zgrupować w jedną etykię, ***offensive*** - natomiast nawet wówczas nie osiągnie ona 100 obrazów. Etykietę *humorous* można usunąć ze zbioru danych, ponieważ jest *de facto* podzbiorem etykiety *meme* (korelacja 99%). Warto też rozważyć usunięcie etykiety *artwork*, która jest silnie skorelowana z etykietą *cartoon style*.

W późniejszej części pracy będę odwoływał się do "***zredukowanego zbioru danych***", **w skrócie RD**, celem dokładniejszego zbadania skuteczności zaprojektowanego modelu. Oznacza to zbiór danych w którym zachowałem tylko etykiety, których udział procentowy wynosi $\geq 10\%$. **Pełny zbiór danych** oznacza zbiór wykorzystujący wszystkie przydzielone etykiety.

4. MODELE I METODY

Projekt sieci neuronowej, oraz całego środowiska, przedstawiłem na rysunku 4.1. Wejściem jest obraz, który należy poddać przetwarzaniu, wyjściem natomiast są etykiety. Obraz wejściowy poddaje wstępemu przetwarzaniu i augmentacji, a następnie dokonuję na nim ekstrakcji cech za pomocą trzech metod: modelu graficznego (*VGG19* [43], *RESNET50* [29]), modelu tekstowego (*DistilRoBERTa* [39], *XLNet* [49]) oraz modelu pracującego na transformacie Fouriera tego obrazu. Uzyskane w ten sposób wektory cech są redukowane do wspólnej długości za pomocą *principal component analysis* (PCA), normalizowane, a następnie łączone za pomocą wybranego algorytmu (*konkatenacja, średnia, wybór maksimum*). Połączone cechy przekazywane są dalej do perceptronu wielowarstwowego [36], którego ostatnia warstwa dokonuje samej klasyfikacji, zwracając etykiety.



Rys. 4.1: Ogólny schemat modelu

Projekt zrealizowałem przy wykorzystaniu platformy Google Colab [16]. Kod źródłowy jest dostępny publicznie pod adresem <https://bit.ly/jakubgrzana>.

4.1. PRZETWARZANIE OBRAZU

Zaprojektowane i zaimplementowane przeze mnie przetwarzanie modalności graficznej (*wizualnej*) odbywa się w trzech głównych krokach: wstępnym przetwarzaniu obrazu, augmentacji oraz ekstrakcji cech za pomocą wytrenowanego modelu.

Wstępne przetwarzanie obrazu jest procesem, który ma na celu przygotowanie tego obrazu przed przekazaniem go sieci neuronowej w sposób ułatwiający przetworzenie go przez sieć. Typowe kroki wstępnego przetwarzania to: zmiana rozmiaru, filtracja, normalizacja wartości pikseli.

Wykorzystywane przeze mnie modele definiują własne kroki wstępnego przetwarzania, przez co implementacja kolejnych nie jest konieczna. Natomiast uznałem za stosowne przeprowadzić dwie operacje wstępnego przetwarzania:

Zmiana rozmiaru z paddingiem Rozmiar obrazu zostaje zmieniony na 512x512 pikseli, przy zachowaniu współczynnika proporcji. Brakujące piksele zastępowane są kolorem czarnym. Ma to na celu ochronę przed naruszeniem współczynnika proporcji przez wytrenowane modele.

Filtr medianowy Obraz zostaje poddany filtracji filtrem medianowym (*rozmiar kernelu = 3*) celem redukcji szumów, w szczególności typu *salt&pepper*.

Żaden z tych kroków nie ma jednoznacznie pozytywnego wpływu na przetwarzanie obrazu przez model. Zmiana rozmiaru z paddingiem zachowuje współczynnik proporcji, dzięki czemu zapobiega zniekształceniom obrazu, wprowadza natomiast padding który może zajmować znaczną część obrazu wynikowego. Filtr medianowy redukuje częściowo szумy co może mieć pozytywny wpływ na wynik klasyfikacji, natomiast nie eliminuje ich całkowicie, jednocześnie zacierając detale i krawędzie w obrazie. Z tego powodu ostateczne testy zostaną przeprowadzone zarówno dla obrazów wstępnie przetworzonych jak i dla nieprzetworzonych, a ich wyniki zostaną porównane.

4.1.1. Augmentacja danych

Na potrzeby tej pracy wykorzystałem następujące techniki augmentacji obrazów:

Random rotation	Obrót względem centrum obrazu o losowy kąt należący do przedziału ($-20^\circ, 20^\circ$).
Color Jitter	Manipuluje poziomem jasności, nasycenia i kontrastu w obrazie, zmieniając każdy z nich o losową wartość procentową należącą do przedziału ($-20\%, 20\%$).
Przesunięcie i przeskalowanie	Obraz zostaje przesunięty na osi X oraz Y, a także przeskalowany o losowe wartości procentowe należące do przedziału ($-10\%, 10\%$).

4.1.2. Modele ekstrakcji cech z obrazu

Jako modele wykorzystywane do ekstrakcji cech z obrazu wybrałem VGG19 [43] oraz RESNET50 [29], obydwa wytrenowane na zbiorze danych ImageNet [23]. Modele te zostały wybrane ze względu na swoją popularność w literaturze - uznałem, że dobrze nadają się jako modele bazowe.

Obydwa wytrenowane modele nie podlegają uczeniu, ich wagę są zamrożone, a cechy są pobieranie z przedostatniej warstwy (*ostatniej przed warstwą klasyfikatora*).

4.1.2.1. VGG19

Wykorzystałem model VGG19 z załadowanymi wagami pierwszej generacji, wytrenowanymi na zbiorze danych ImageNet (*IMAGENET1K_V1* [11]), ponieważ są to jedyne wagi dostępne dla tego modelu, wykorzystujące wybrany zbiór danych.

Wstępne przetwarzanie, zdefiniowane w dokumentacji modelu [11], przeprowadza następujące operacje:

Zmiana rozmiaru	Obraz przeskalowywany jest do wymiarów 256x256 pikseli.
Wycięcie środka	z angielskiego <i>Central Crop</i> , wycina z obrazu środkową część o wymiarach 224x224 pikseli.
Przeskalowanie wartości	Wartość każdego kanału w obrazie przeskalowana zostaje z 0..255 do 0..1.
Normalizacja	Przeprowadzona zostaje normalizacja, przy użyciu średnich (0.485, 0.456, 0.406) oraz odchyлеń standardowych (0.229, 0.224, 0.225)

Przetworzony obraz trafia na wytrenowany model z zamrożonymi wagami i obciętą ostatnią warstwą, zwracając **wektor cech o liczbie elementów równej 4096**.

4.1.2.2. RESNET50

Wykorzystałem model RESNET50 z załadowanymi wagami pierwszej generacji, wy-trenowanymi na zbiorze danych ImageNet (*IMAGENET1K_V1 [10]*). Pomimo dostępności generacji drugiej, która osiąga lepsze wyniki, postanowiłem pozostać przy generacji pierw-szej by móc uczciwie porównać skuteczność architektury RESNET z VGG.

Wstępne przetwarzanie, zdefiniowane w dokumentacji modelu [10], przeprowadza następujące operacje:

Zmiana rozmiaru	Obraz jest przeskalowywany do wymiarów 256x256 pikseli.
Wycięcie środka	z angielskiego <i>Central Crop</i> , wycina z obrazu środkową część o wymiarach 224x224 pikseli.
Przeskalowanie wartości	Wartość każdego kanału w obrazie zostaje przeskalowana z 0..255 do 0..1.
Normalizacja	Przeprowadzona zostaje normalizacja, przy użyciu średnich (0.485, 0.456, 0.406) oraz odchyleń standardowych (0.229, 0.224, 0.225)

Przetworzony obraz trafia na wytrenowany model z zamrożonymi wagami i obciętą ostatnią warstwą, zwracając **wektor cech o liczbie elementów równej 2048**.

4.2. PRZETWARZANIE TEKSTU

Zaprojektowane i zaimplementowane przeze mnie przetwarzanie modalności tekstowej odbywa się w dwóch głównych krokach: odczytanie tekstu z obrazu wejściowego za pomocą OCRa, oraz ekstrakcji cech z otrzymanego tekstu za pomocą wytrenowanego modelu (*DistilRoBERTa [39]*, *XLNet [39]*).

Tekst, tak jak i obraz, można poddawać wstępemu przetwarzaniu celem usunięcia z niego szumu, a wyciągnięcia cech prawdziwie istotnych. Przykładowe operacje wstępnego przetwarzania to: ujednolicenie wielkości liter, usunięcie znaków interpunkcyjnych i *stopwords*, stemmeryzacja tekstu (*obcinanie końcówek słów*) bądź lemmatyzacja (*sprowadzanie słów do formy kanonicznej*) [48]. Każda z tych operacji usuwa część informacji z tekstu, natomiast w przypadku pracy z danymi pozbawionymi struktury najczęściej informacji mamy zbyt dużo i ich redukcja oraz odszumienie pozwala uzyskać lepsze wyniki.

Modele które wykorzystałem w ramach pracy posiadają własne tokenizery, które przyjmują tekst na wejście po czym przeprowadzają wstępne przetwarzanie, tokenizację i kodowanie, po czym zwracają zwracając zakodowany tekst wraz z maską atencji.

Zdecydowałem się nie implementować własnego algorytmu wstępnego przetwarzania tekstu, ponieważ wytrenowane modele robią to same. Modyfikowanie oryginalnego tekstu mogłoby skutkować usunięciem informacji istotnej z perspektywy modelu.

Nie wykorzystuję również augmentacji tekstu. Operacje typu: zamiana losowych słów na synonimy, wielokrotne tłumaczenia tekstu (*np. angielski - polski - angielski*) lub też wykorzystanie modeli *text-to-text* mogą pomóc w uczeniu sieci [41], natomiast ich zapotrzebowanie na zasoby obliczeniowe okazało się zbyt duże bym mógł z nich skorzystać.

4.2.1. Odczyt tekstu z obrazu

Wybranym przeze mnie modelem OCR jest EasyOCR [30] - z powodów wymienionych w rozdziale *Przegląd literatury* - skonfigurowany do pracy z tekstem w języku angielskim.

Testy na zbiorze "*Multioff Covid*" [46] wykazały, że tekst odczytany za pomocą Easy-OCRa jest w większości poprawny - porównując go do tekstu dostarczonego przez twórców "*Multioff*" **osiągnął wynik 0.77 wedle metryki fuzz ratio** [27], obliczanej według wzoru 4.1

$$Ratio(a, b) = 1 - \frac{d}{\max(\bar{a}, \bar{b})} \quad (4.1)$$

gdzie d oznacza odległość Levenshtine'a między napisami a i b , \bar{a} oznacza długość napisu a .

4.2.2. Modele ekstrakcji cech z tekstu

Jako modele wykorzystywane do ekstrakcji cech z tekstu wybrałem DistilRoBERTa [39] oraz XLNet [49]. Obydwa wytrenowane modele nie podlegają uczeniu, ich wagi są zamrożone, a cechy są pobieranie z ostatniej warstwy. **Liczba elementów wektora cech w obydwu przypadkach wynosi 768.**

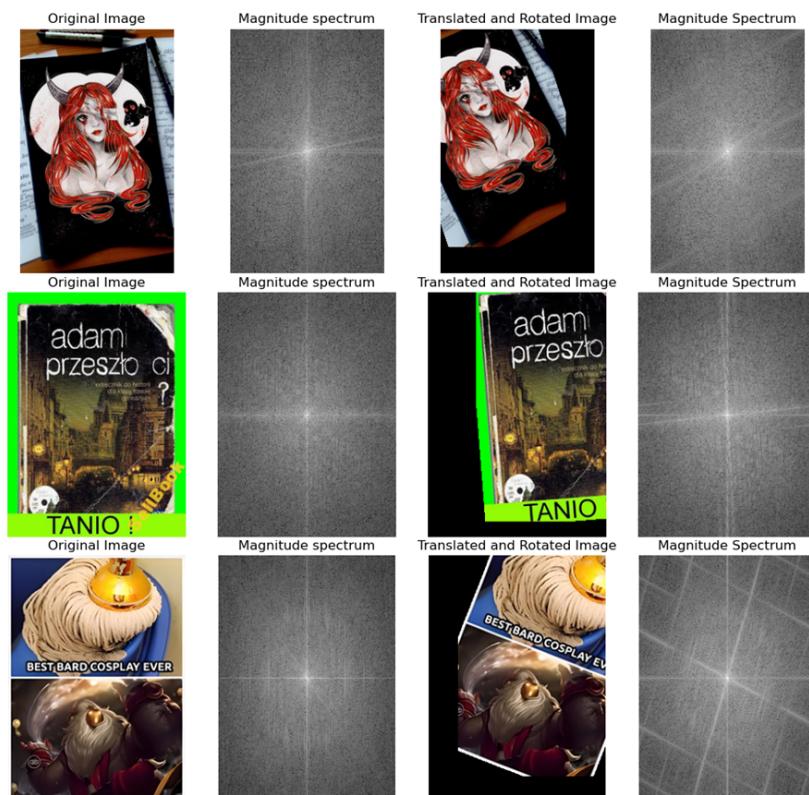
Uściślając, w ramach pracy wykorzystałem modele *distilroberta base* oraz *xlnet base cased* z serwisu HuggingFace [47, 14]. Modele te uwzględniają wielkość liter. W ramach tokenizacji przeprowadzam również przycięcie (*truncation*) / wypełnienie (*padding*) do zadanej liczby tokenów: 128. Jest to liczba, która pozwala na pomieszczenie zdecydowanej większości tekstów w zbiorze danych (*patrz rysunek 3.4*).

4.3. GENEROWANIE NOWEJ MODALNOŚCI: TRANSFORMATA FOURIERA

Transformacja Fouriera przenosi sygnał z dziedziny czasu do dziedziny częstotliwości, umożliwiając łatwiejszą analizę składowych cyklicznych. Typowo stosowana jest w ramach wstępnego przetwarzania obrazu, do przeprowadzania filtracji bądź kompresji, może być jednak zastosowana również do np. analizy tekstur [4].

Wynikiem transformacji Fouriera jest transformata dana liczbami zespolonymi - z tego powodu każdy kanał obrazu zostaje zamieniony w dwa kanały, typowo reprezentującymi część rzeczywistą i urojoną transformaty. **Transformacja jest operacją odwracalną**, co oznacza że transformata zawiera wszystkie informacje zawarte w oryginalnym obrazie (*pomijając zaokrąglenia wynikające z implementacji*).

Postanowiłem wykorzystać transformacje Fouriera jako dodatkową, trzecią modalność. Nie znalazłem w literaturze żadnych prac poruszających ten temat, natomiast intuicja sugeruje że jest to dobry sposób na ekstrakcję cech z obrazu - natura transformaty zapewnia łatwiejszy dostęp do informacji o krawędziach w obrazie czy teksturach. Podanie takiej reprezentacji danych na sieć konwolucyjną, znaną ze swoich zdolności znajdowania wzorców, może dać dobre efekty. Na rysunku 4.2 przedstawiłem kilka obrazów przed i po augmentacji wraz z wizualizacją transformaty tych obrazów - wyraźnie widać przesunięcia i nachylenia, niemniej transformaty wydają się być bardziej skoncentrowane niż obrazy oryginalne, co mam nadzieję korzystnie wpłynie na wyniki.



Rys. 4.2: Przykładowe widma transformaty Fouriera różnych obrazów

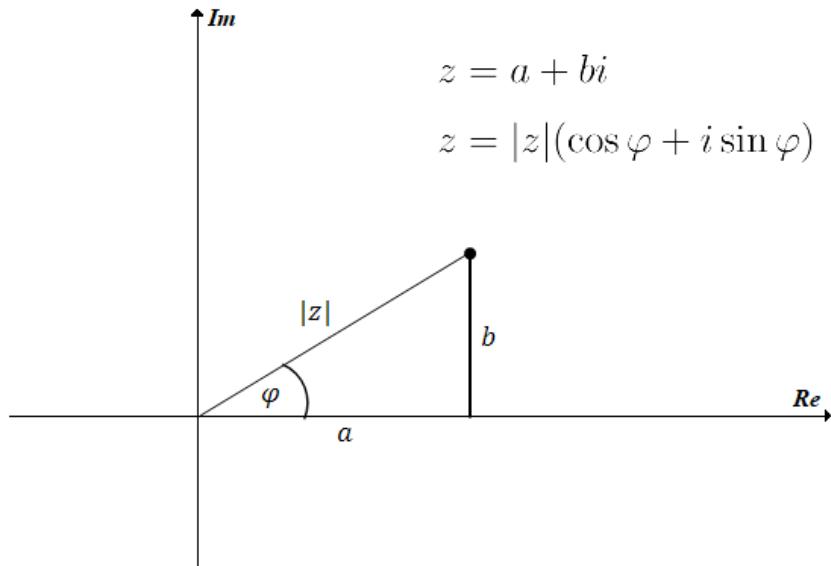
W ramach pracy wykorzystałem transformacje Fouriera w połączeniu z modelem RESNET18 [29] wytrenowanym przeze mnie na przetransformowanym zbiorze danych Tiny ImageNet [32], pracując z **obrazami w skali szarości**.

Zmiana obrazów na *grayscale* ma na celu ułatwienie dotarcia do cech przez sieć: tekstury oraz krawędzie powinny być dla sieci konwolucyjnej równie łatwe lub nawet łatwiejsze na jednym kanale niż na trzech, zakładam że stwierdzenie to aktualne jest również po zastosowaniu transformacji.

4.3.1. Reprezentacja transformaty

Transformata obrazu dana jest liczbami zespolonymi, zatem jeden kanał obrazu wejściowego transformowany jest w dwa kanały, zawierające część rzeczywistą i urojoną transformaty. Wprowadza to problem związany z normalizacją.

Dla obrazów w formacie RGB wartość każdego kanału jest liczbą całkowitą należącą do (0..255), w przypadku transformaty nie da się określić górnego ani dolnego ograniczenia części rzeczywistej ani urojonej - dla liczb zespolonych reprezentowanych w postaci kanonicznej. Z tego powodu zdecydowałem się na **reprezentacje transformaty za pomocą postaci trygonometrycznej liczby zespolonej**.

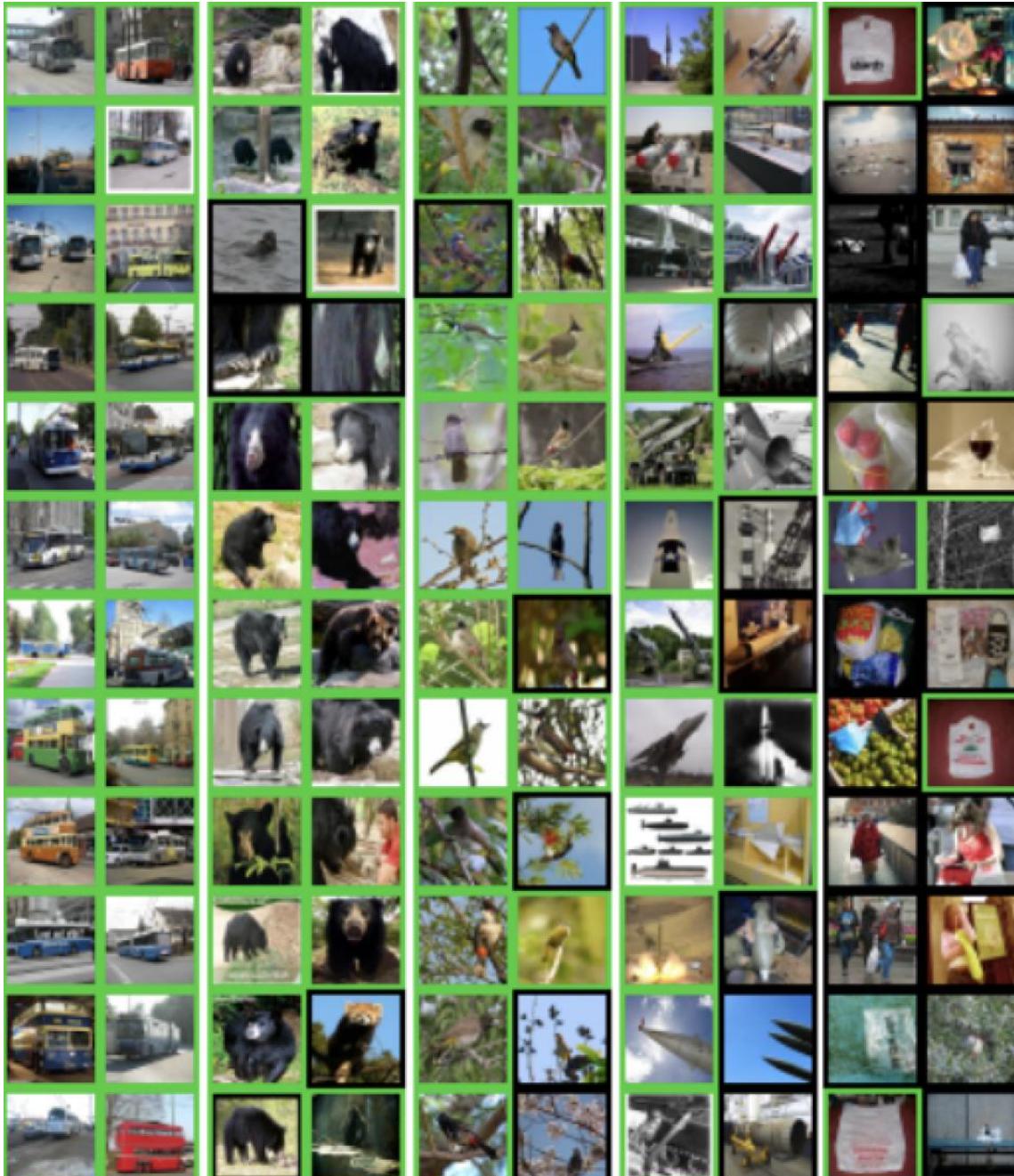


Rys. 4.3: Reprezentacje liczby zespolonej

Reprezentowana tak transformata składa się z modułu $|z|$ o wartościach $0 \leq |z| < \infty$ oraz argumentu φ o wartościach $-\pi \leq \varphi \leq \pi$. Zamknięcie tych wartości dolnymi i (*dla argumentu*) górnymi ograniczeniami teoretycznie powinno pomóc w normalizacji danych.

4.3.2. Zbiór danych

Wykorzystałem zbiór danych Tiny ImageNet [32], ponieważ jest on dość duży (*100 000 obrazów RGB 64x64, 200 klas*) a jednocześnie wystarczająco mały bym mógł przeprowadzić na nim uczenie przy użyciu dostępnych dla mnie zasobów obliczeniowych. Przykładowe obrazy z tego zbioru przedstawia rysunek 4.4

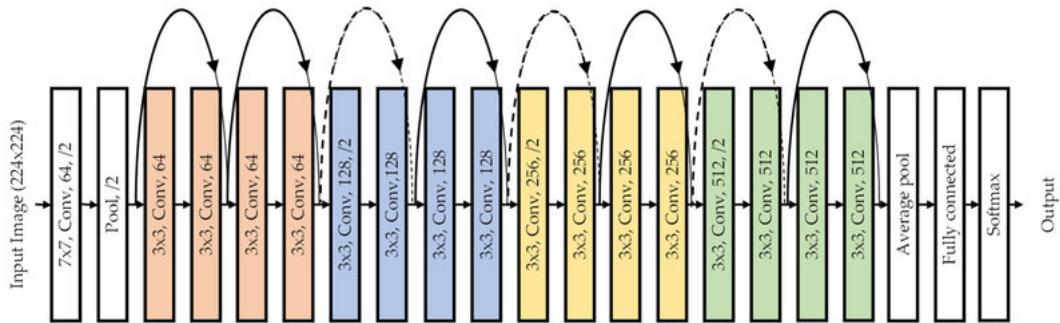


Rys. 4.4: Przykładowe obrazy ze zbioru Tiny ImageNet [32]

Tiny ImageNet jest również stosunkowo popularny, co umożliwiło mi łatwe porównanie uzyskanych wyników [13].

4.3.3. Model

Do ekstrakcji cech z transformaty obrazu wykorzystałem model RESNET18 [29] przedstawiony na rysunku 4.5, wytrenowany na przetransformowanym zbiorze danych Tiny Imagenet. Aby umożliwić modelowi pracę z dwoma kanałami (*moduł, argument*) zastąpiłem pierwszą warstwę konwolucyjną modelu warstwą przyjmującą dwa kanały wejściowe.

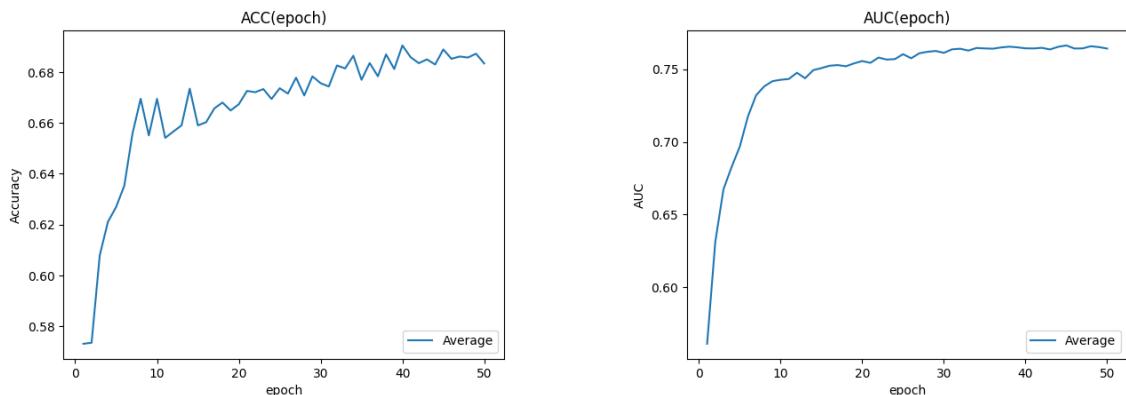


Rys. 4.5: Oryginalna architektura RESNET18 [19]

Wybrałem RESNET18 ponieważ jest to popularny model który dobrze radzi sobie z problemem zanikającego gradientu oraz osiąga przyzwoite wyniki przy niskim zapotrzebowaniu na zasoby.

4.3.4. Uczenie modelu FFT

Uczenie trwało 50 epok, przy zastosowaniu optymalizatora SGD [38] z parametrami *momentum=0.9*, *weight_decay=0.0001*. *Learning rate* ustawiłem na 0.001 i mnożyłem go co epokę przez 0.95. Jako funkcję straty wykorzystałem *Cross Entropy Loss*. Wyniki uczenia przedstawiłem na rysunku 4.6



Rys. 4.6: Wyniki wytrenowanego modelu RESNET18 na transformatach obrazów ze zbioru Tiny Imagenet

Uzyskane Accuracy wynosi około 68%, co jest oczekiwany wynikiem dla tego modelu i zbioru danych [13]. Nie ulega więc wątpliwości że model pracujący z transformatami obrazów zamiast z samymi obrazami działa skutecznie.

Po wytrenowaniu modelu zamroziłem jego wagę oraz usunąłem ostatnią warstwę (*klasyfikującą*), uzyskując ekstraktor cech zwracający **wektor cech o liczbie elementów równej 512**.

4.4. WYKORZYSTANIE PCA

Na potrzeby pracy wykorzystałem implementacje PCA z biblioteki *scikit-learn* [24] **dopasowaną dla wszystkich próbek ze zbioru uczącego**, osobno dla każdego z modeli - ekstraktorów cech. Dla każdego z modeli przeprowadziłem **redukcję do 512 cech**, ponieważ tyle cech dostarcza model FFT a z powodów opisanych w sekcji *Metody łączenia cech, normalizacja* chciałem zachować jednolity rozmiar wektorów cech.

Procent wariancji oryginalnych danych wyjaśniony przez 512 pierwszych składowych głównych dla poszczególnych modeli, wykorzystanych w ramach pracy:

VGG19	93.18%
RESNET50	98.32%
DistilRoBERTa	99.97%
XLNet	99.79%
Model FFT	100.0%

4.5. METODY ŁĄCZENIA CECH, NORMALIZACJA

Wyekstraktowane i zredukowane wektory cech poddaje, niezależnie od siebie, normalizacji L2 danej wzorem 4.2,

$$v = \frac{v}{\sqrt{\sum_{i=1}^n (v_i)^2}} \quad (4.2)$$

gdzie v jest wektorem, a v_i to i -ty element wektora v .

Następnie znormalizowane wektory cech $w_1 \dots w_n$ łączę za pomocą jednego z trzech algorytmów:

Konkatenacja Każdy kolejny wektor cech umieszczany jest na końcu poprzedniego, jak pokazałem we wzorze 4.3.

$$\begin{aligned} w_1 &= [1, 2, 3, 4] \\ w_2 &= [5, 6, 7, 8] \\ \text{concat}(w_1, w_2) &= [1, 2, 3, 4, 5, 6, 7, 8] \end{aligned} \quad (4.3)$$

Wybór maksimum Dla każdej wartości wektora wyjściowego v_i wybierane jest maksimum analogicznych wartości wektorów wejściowych (*wzór 4.4*).

$$v_i = \max(w_{1i}, w_{2i} \dots w_{ni}) \quad (4.4)$$

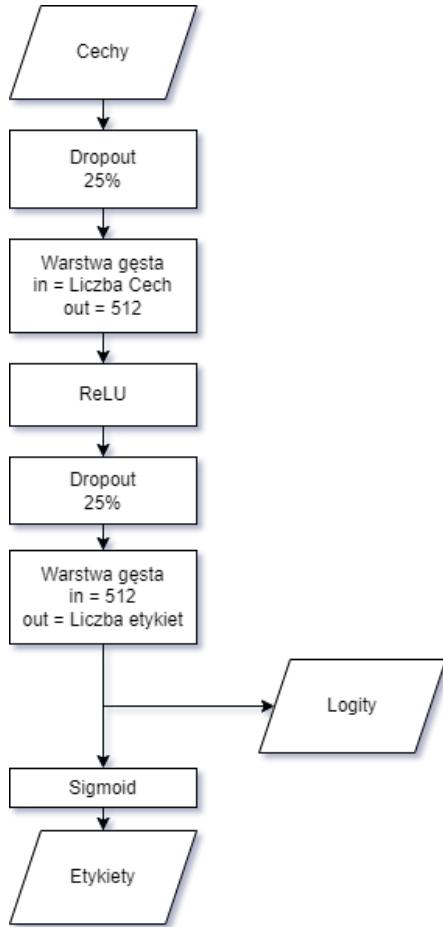
Uśrednienie Dla każdej wartości wektora wyjściowego v_i obliczana jest średnia (*arytmetyczna, nieważona*) analogicznych wartości wektorów wejściowych. (*wzór 4.5*).

$$v_i = \frac{\sum_{k=1}^n w_{ki}}{n} \quad (4.5)$$

Wybór maksimum oraz uśrednianie wymagają, aby wektory cech miały tę samą liczbę cech. Zachowanie jednolitej liczby cech między modalnościami może również pozytywnie wpłynąć na dalsze przetwarzanie w przypadku konkatenacji. Każda cecha wejściowa wpływa na ostateczny wynik klasyfikacji. Jeśli ekstraktor dostarczy dwukrotnie więcej cech niż inny, teoretycznie będzie miał dwukrotnie większy wpływ na wyniki. Normalizacja L2 przeciwdziała temu efektowi, skalując wartości cech tak, aby długość (*norma*) wektora wynosiła 1. Może to jednak prowadzić do zdominowania klasyfikacji przez cechy wektora o mniejszej liczbie cech, ponieważ po normalizacji wartości te będą większe niż w wektorze o większej liczbie cech.

4.6. KLASYFIKATOR

Projekt sieci klasyfikatora stanowi przykład perceptronu wielowarstwowego [36], składającego się z jednej warstwy ukrytej wraz z funkcją aktywacji ReLU, jednej warstwy wyjściowej (*klasyfikującej*) wraz z funkcją aktywacji Sigmoid, oraz dwóch warstw dropout. Schemat przedstawiłem na rysunku 4.7. Model klasyfikatora jest prosty, natomiast uznałem że jest adekwatny do zbioru danych zawierającego tylko 1542 próbki.



Rys. 4.7: Schemat modelu klasyfikatora

Dropout jest techniką regularyzacji, szczególnie skuteczną w zapobieganiu przesadnego dopasowania (*overfittingu*) modelu [45]. Tradycyjnie warstwy dropout umieszcza się pomiędzy warstwami gęstymi (*w pełni połączonymi*), umieszczenie takiej warstwy bezpośrednio na wejściu do sieci jest bardziej kontrowersyjne - warstwa ta wprowadza zaszumienie danych wejściowych poprzez usunięcie z nich niektórych wartości, co może poprawić zdolności generalizacyjne modelu. Dodatkowo, niektóre algorytmy łączenia cech (*np. wybór maksimum*) naruszają normalizację i mogą prowadzić do dominacji jednej cechy nad pozostałymi - dropout skutecznie temu zjawisku przeciwdziała.

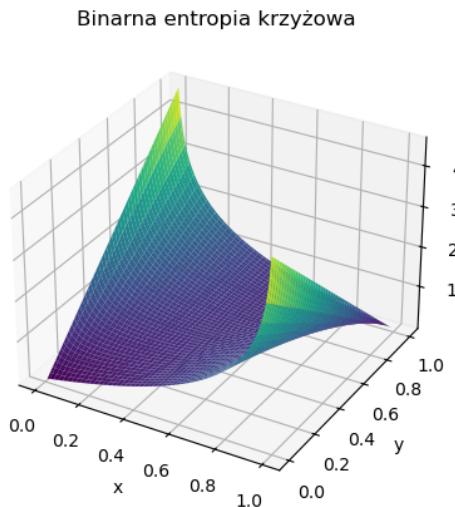
4.7. FUNKCJA STRATY

Klasyfikacja wieloetykietowa polega na przypisaniu do danej próbki wszystkich pasujących do niej etykiet, w przeciwieństwie do klasyfikacji wieloklasowej gdzie przypisywana jest tylko jedna klasa. Oznacza to, że dla N dostępnych etykiet warstwa klasyfikatora składa się z N neuronów dokonujących klasyfikacji binarnej, której wynik jest niezależny od pozostałych neuronów w tej warstwie.

Jako funkcje straty zastosowałem *binarną entropię krzyżową* (ang. *Binary Cross Entropy* [9]) daną wzorem 4.6.

$$H(x, y) = -(y \log(x) + (1 - y) \log(1 - x)) \quad (4.6)$$

Dla każdego neuronu warstwy wyjściowej obliczana jest strata $H(x, y)$, gdzie x oznacza odpowiedź neuronu, y jest wartością docelową. Jeżeli $x \approx y$ to $H(x, y) \approx 0$ - im różnica między x, y jest większa, tym większą wartość przyjmuje $H(x, y)$, co przedstawiłem na rysunku 4.8. Wartość funkcji straty poszczególnych neuronów jest następnie uśredniania.



Rys. 4.8: Wykres binarnej entropii krzyżowej

Funkcja straty jest wykorzystywana do obliczania poprawek wag wewnętrz sieci klasyfikatora. Ponieważ wykorzystywany przez mnie zbiór danych jest niebalansowany, istnieje duże ryzyko, że sieć zignoruje etykiety o małym udziale w zbiorze. Próba optymalizacji parametrów sieci w celu prawidłowej klasyfikacji etykiety o udziale rzędu 10% może spowodować wzrost wartości funkcji straty przez wyniki fałszywie pozytywne. W efekcie optymalizator może skupić się na poprawianiu predykcji dla dominujących etykiet, zaniedbując te mniej liczne. Aby temu przeciwdziałać wykorzystałem **ważoną funkcję straty**, gdzie wartość funkcji straty poszczególnych neuronów (*przed uśrednieniem*) jest mnożona przez wagę obliczoną według wzoru 4.7.

$$waga_{Li} = \frac{\frac{\bar{D}}{D_{Li}}}{\sum_{k=1}^n \frac{\bar{D}}{D_{Lk}}} \quad (4.7)$$

gdzie \bar{D} oznacza liczbę próbek w zbiorze D , \bar{D}_{Li} oznacza liczbę próbek zawierających etykietę Li , n oznacza liczbę etykiet.

4.8. METRYKI

Wykorzystany zbiór danych jest niebalansowany. Udział poszczególnych etykiet waha się od 0.6% do 50.7% (*tabela 3.1*), co wymaga wykorzystania metryk odpornych na niebalansowanie. Do tego celu wybrałem metryki: AUC (*pole pod wykresem krzywej ROC*), F1-score oraz AUPR (*pole pod wykresem krzywej Precision-Recall*).

Wszystkie opisane metryki (*AUC*, *F1-score*, *AUPR*), są zdefiniowane dla klasyfikacji binarnej, natomiast można je zastosować do klasyfikacji wieloetykietowej. Wynika to z faktu, że przewidywanie każdej etykiety jest niezależnym zadaniem klasyfikacji binarnej. Otrzymane metryki dla poszczególnych etykiet można analizować osobno lub agregować. W tej pracy do agregacji zastosowałem średnią arytmetyczną nieważoną (*macro average*).

4.8.1. Progowanie, miary TP, TN, FP, FN

Klasyfikator sieci zwraca dla poszczególnych etykiet wartość liczbową należącą do zakresu $[0, 1]$, którą można interpretować jako stopień pewności klasyfikatora w odniesieniu do przypisania danej etykiety próbce. Jednakże, aby ocenić jakość klasyfikacji, potrzebne są jednoznaczne wyniki określające, czy danej próbce przypisać daną etykietę. Konieczne jest więc przeprowadzenie progowania (*ang. thresholding*).

Progowanie (*binarne*) jest operacją sprowadzającą wartość ciągłą do wartości binarnej, wedle wzoru 4.8

$$p(x, \text{threshold}, \text{low}, \text{up}) = \begin{cases} \text{low}, & \text{if } x < \text{threshold} \\ \text{up}, & \text{if } x \geq \text{threshold} \end{cases} \quad (4.8)$$

gdzie x oznacza wyjście klasyfikatora. Dla zadania klasyfikacji, dolnym ograniczeniem *low* jest 0 (*brak etykiety*), górnym *up* jest 1, Problemem jest wartość progu (*threshold*), która może być dowolną liczbą rzeczywistą, natomiast której dobór ma ogromny wpływ na wyniki. **Na potrzeby badań wybieram wartość progu maksymalizującą wartość metryki F1-score dla zbioru uczącego.** Każda etykieta ma swoją niezależną wartość progu, aktualizowaną co epokę uczenia.

Po przeprowadzeniu progowania otrzymane wyniki są porównywane z rzeczywistymi etykietami, aby znaleźć wartości TP, TN, FP, FN dla każdej etykiety:

TP (True Positive) Liczba próbek, którym dana etykieta została prawidłowo przypisana.

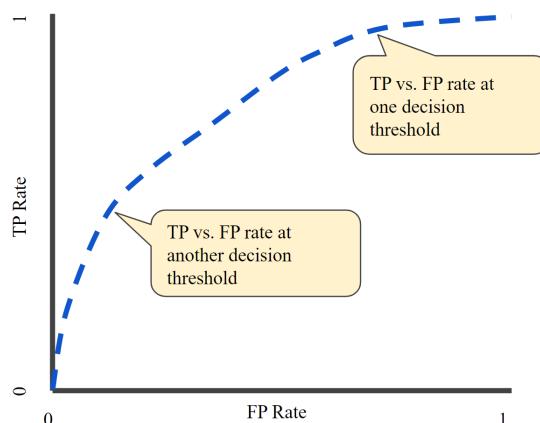
TN (True Negative) Liczba próbek, którym dana etykieta została prawidłowo nieprzypisana.

FP (False Positive) Liczba próbek, którym dana etykieta została błędnie przypisana.

FN (False Negative) Liczba próbek, którym dana etykieta została błędnie nieprzypisana.

4.8.2. AUC

Krzywa ROC przedstawia zależność między odsetkiem prawdziwie pozytywnych wyników (*True Positive Rate, TPR*) a odsetkiem fałszywie pozytywnych wyników (*False Positive Rate, FPR*) dla różnych progów klasyfikacji. Metryka AUC określa pole pod wykresem krzywej ROC. Przykład krzywej ROC przedstawiono na rysunku 4.9



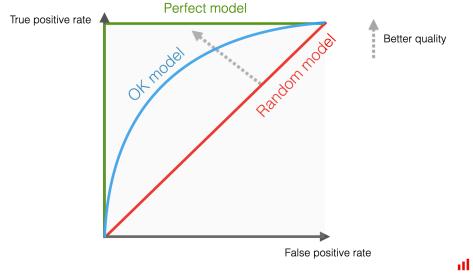
Rys. 4.9: Przykładowa krzywa ROC [28]

Każdy punkt na wykresie krzywej ROC opisuje wartość TPR oraz FPR dla danej wartości progu - sama wartość progu nie jest tutaj szczególnie istotna, ważny jest jedynie kształt krzywej tworzonej przez pary (*FPR, TPR*) dla wielu różnych progów. Wartości FPR i TPR obliczane są ze wzorów: 4.9, 4.10 [28].

$$FPR = \frac{FP}{FP + TN} \quad (4.9)$$

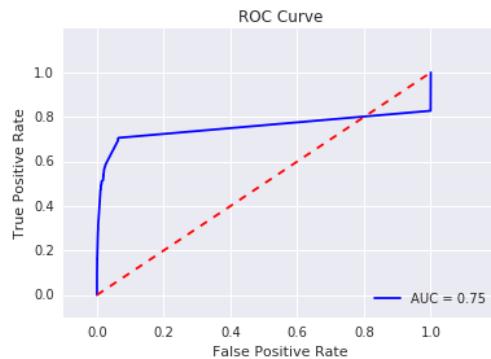
$$TPR = \frac{TP}{TP + FN} \quad (4.10)$$

Na rysunku 4.10 przedstawiono wykresy ROC dla trzech klasyfikatorów: idealnego, dobrego oraz losowego. **Dla klasyfikatora losowego pole pod wykresem ROC wynosi 0.5**, ponieważ niezależnie od przyjętej wartości progu, wśród przydzielonych etykiet pozytywnych będzie mniej więcej równy udział prawdziwie pozytywnych i fałszywie pozytywnych, co powoduje, że $FPR \approx TPR$, a krzywa ROC jest prostą linią od punktu $(0,0)$ do $(1,1)$. Im klasyfikator jest skuteczniejszy, tym bardziej średnie wartości (*FPR, TPR*) zbliżają się do punktu $(0,1)$, który oznacza brak wyników fałszywie pozytywnych oraz 100% wyników prawdziwie pozytywnych - gdy punkt ten zostanie osiągnięty, pole pod wykresem wynosi 1.



Rys. 4.10: Krzywa ROC dla idealnego, dobrego oraz losowego klasyfikatora [2]

AUC jest miarą w większości niezależną od proporcji klas, co czyni tę metrykę szczególnie przydatną w kontekście niebalansowanych zbiorów danych. Natomiast dla zbiorów silnie niebalansowanych, gdzie etykieta przydzielana jest rzadko występuje zjawisko przesunięcie wykresu w lewo: jest to spowodowane wystąpieniem TN w mianowniku wzoru 4.9 definiującego FPR, co może sztucznie zawyżyć pole pod wykresem [22], przykład czego przedstawiono na rysunku 4.11. Nie jest to metryka idealna.



Rys. 4.11: Przykład przesunięcia krzywej ROC dla ekstremalnie niebalansowanego zbioru danych [6]

4.8.3. F1-score

Metryka F1-score jest średnią harmoniczną dwóch innych metryk: *Precision (precyzja)* i *Recall (Trafność)*.

Precision jest obliczane według wzoru 4.11 i opisuje jaka część wszystkich przypisań wybranej etykiety jest prawidłowa. $Precision = 0.8$ oznacza, że ze wszystkich próbek, którym przypisano daną etykietę, 80% zostało sklasyfikowane prawidłowo.

$$Precision = \frac{TP}{TP + FP} \quad (4.11)$$

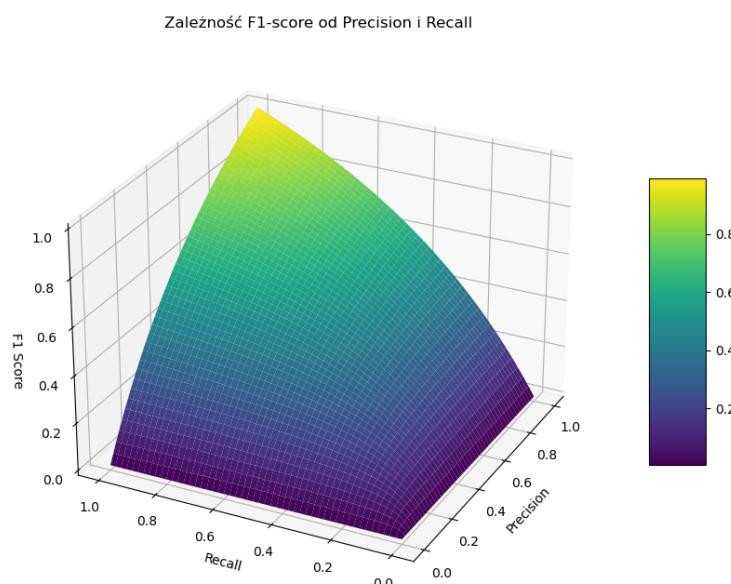
Recall jest obliczane według wzoru 4.12 i opisuje jaka część wszystkich wystąpień danej etykiety w zbiorze została prawidłowo wykryta i przypisana. $Recall = 0.6$ oznacza, że klasyfikatorowi udało się prawidłowo przypisać etykietę dla 60% próbek, które prawdziwie ją zawierają.

$$Recall = \frac{TP}{TP + FN} \quad (4.12)$$

Zarówno *precision* jak i *recall* mogą zostać oszukane: klasyfikator przypisujący etykietę do wszystkich możliwych próbek osiąga wynik $Recall = 1$, klasyfikator który prawidłowo przypisze etykietę jednej próbce po czym zignoruje wszystkie pozostałe osiągnie $Precision = 1$. Natomiast **niemożliwe jest jednocześnie *Precision* oraz *Recall***.

F1-score jest średnią harmoniczną *Precision* oraz *Recall*, daną wzorem 4.13. Zależność między F1-score a jej składowymi przedstawiłem na rysunku 4.12: osiąga maksimum, gdy zarówno *Precision* oraz *Recall* wynoszą 1, natomiast gdy jedna z tych wartości zbliża się do 0, wartość F1-score również dąży do zera.

$$F1 = 2 \frac{Precision * Recall}{Precision + Recall} \quad (4.13)$$



Rys. 4.12: Zależność F1-score od *Precision* i *Recall*

Klasyfikator osiągający wysokie wyniki F1-score musi jednocześnie wykrywać większość przypadków pozytywnych (*wpływ Recall*) oraz być pewnym swojej decyzji, gdy tą etykietę przydziela (*wpływ Precision*) - dla klasyfikatora losowego, **F1-score jest równy udziałowi danej etykiety w zbiorze. Metryka ta jest odporna na niezbalansowanie w zbiorze danych, ponieważ skupia się wyłącznie na przypadkach pozytywnych**. Jej wadą jest konieczność ustalenia progu do binaryzacji wyjścia klasyfikatora - ostateczna jakość oceny jest więc zależna od jakości dobranego progu.

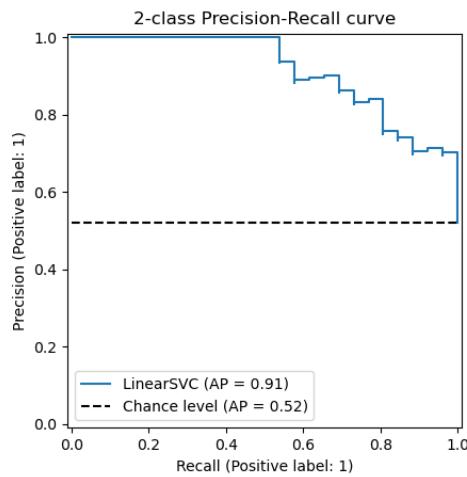
Dokonując analizy F1-score trzeba brać pod uwagę udział danej etykiety w zbiorze. Z tego też powodu agregacja tej metryki do formy średniej jest mniej informatywna, niż

w przypadku AUC - dla dwóch etykiet o udziale kolejno 0.5 oraz 0.1 klasyfikator losowy osiągnie $AUC \approx 0.5$, natomiast $F1 \approx 0.3$, co może zasugerować nieuwrażnemu badaczowi lepszą skuteczność klasyfikacji niż rzeczywista.

4.8.4. AUPR

Krzywa PR przedstawia zależność pomiędzy *Precision* oraz *Recall* dla różnych progów klasyfikacji. Metryka AUPR określa pole pod wykresem krzywej PR, przykład przedstawiono na rysunku 4.13. Każdy punkt opisuje parę wartości (*Recall*, *Precision*) dla danego progu - ponownie, wartość tego progu nie jest istotna, ważny jest jedynie kształt krzywej. Warto zauważyć, że TPR (wzór 4.10) oraz *Recall* (wzór 4.12) to te same wielkości, natomiast w przypadku krzywej PR *Recall* znajduje się dolnej poziomiej, podczas gdy w przypadku krzywej ROC dany jest na osi pionowej.

Krzywa PR zaczyna się w punkcie $(0, 1)$ - w tym miejscu klasyfikator przydziela etykietę bardzo małej liczbie próbek ($Recall \approx 0$), natomiast jeżeli już to robi to poprawnie. Dalej, wraz ze zmianą wartości progu stopniowo rośnie liczba przydzielanych etykiet i wartość *Recall*, aż do momentu w którym drobny zysk na wartości *Recall* powoduje zauważalny spadek wartości *Precision*. Idealny klasyfikator dąży do $Precision \approx 1$ oraz $Recall \approx 1$, czyli do punktu $(1, 1)$ położonego w prawym górnym rogu wykresu, natomiast w rzeczywistości rzadko jest on osiągany. Typowa krzywa PR kończy się w punkcie $(1, n)$ gdzie n oznacza udział danej etykiety w zbiorze danych [22, 25]..



Rys. 4.13: Przykład krzywej PR [25]

Pole pod wykresem krzywej PR przyjmuje wartości od 0 do 1. **Dla klasyfikatora losowego *Precision*, a co za tym idzie AUPR, jest równe udziałowi analizowanej etykiety w zbiorze danych.** Dokładny opis działania tej metryki wykracza poza zakres tej pracy.

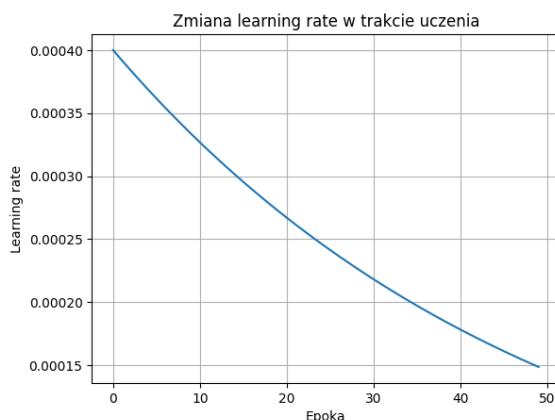
Metryka AUPR sprawdza się doskonale w przypadku silnie niezbalansowanych zbiorów danych [22], natomiast dokonując jej analizy trzeba brać pod uwagę udział danej etykiety w zbiorze, dokładnie tak samo jak w przypadku F1-score.

5. WYNIKI

5.1. PRZEBIEG BADAŃ

Zbiór danych podzieliłem na zbiór uczący (75%) i testowy (25%). Ze względu na dosyć mały zbiór oraz dużą liczbę etykiet zrezygnowałem z zastosowania trzeciego zbioru walidacyjnego.

Dla każdej możliwej kombinacji modelu graficznego z modelem tekstowym oraz modelem wykorzystującym FFT przeprowadziłem uczenie na pełnym zbiorze danych, a także na zredukowanym (*tzn. pomijając etykiety o udziale poniżej 0.1, liczba próbek pozostała taka sama*). Do uczenia wykorzystałem optymalizator Adam [31] o *learning rate* zależnym wykładniczo od epoki uczenia (patrz rys. 5.1). Uczenie każdego z modeli trwało 50 epok.



Rys. 5.1: Zmiana learning rate w procesie uczenia

Nie wykorzystywałem *early stoppingu* ani nie zachowywałem modelu o najlepszych osiągach, zamiast tego do ostatecznej ewaluacji **wybrałem 3 epoki, w których model osiągnął najlepsze wyniki F1-score**, po czym uśredniłem (*średnia arytmetyczna nieważona*) wyniki AUC, Precision, Recall, F1-score oraz AUPR dla wybranych epok. Takie podejście jednocześnie zapewnia miarodajne wyniki, pozwalające ocenić skuteczność jednego modelu na tle innych, z drugiej strony unikając *cherry pickingu*.

Dla dwóch wybranych modeli przeprowadziłem badania skuteczności różnych metod łączenia wektora cech, a także wpływu wstępniego przetwarzania obrazu. Do badań tych wykorzystałem zredukowany zbiór danych.

Uczenie wszystkich modeli przeprowadzone zostało dla tych samych hiperparametrów, dla tego samego schematu klasyfikatora oraz dla tych samych zbiorów uczących i testowych.

5.2. WNIOSKI

Otrzymane wyniki dla pełnego zbioru danych, zaprezentowane w tabeli 5.1, są raczej słabe. Średnia wartość $F1 \approx 0.41$ oraz $AUC \approx 0.78$ sugerują, że klasyfikator posiada pewną zdolność do rozróżniania przypadków pozytywnych i negatywnych, natomiast do osiągnięcia użyteczności w rzeczywistych zastosowaniach dużo brakuje. Analiza wyników poszczególnych etykiet dla wybranych modeli przedstawionych w tabelach 5.2, 5.4, 5.3 w kontekście składu zbioru danych (*tablica 3.1*) ujawnia jedną z przyczyn: klasyfikator nie uczy się prawidłowej klasyfikacji dla etykiet o bardzo małym udziale. Szczególnie dobrze widać to również na rysunku 5.2 ukazującym historię uczenia modela. Dla etykiet *racism*, *slander* oraz *violent* (*każda o udziale około 3%*) wynik F1-score wyniósł ≈ 0.1 , dla etykiety *sexism* (*na którą składa się jedynie 9 z 1542 obrazów*) F1-score wyniósł ≈ 0 , nie widać również tendencji wzrostowej na wykresie F1-score w zależności od epoki.. Etykiety te zanizują średnie wyniki, ponadto ze względu na wykorzystaną przez mnie ważoną funkcję straty klasyfikator próbuje się do nich dostosować bardziej, niż do pozostałych etykiet, mimo tego że ze względu na zbyt małą liczbę danych uczących nie jest w stanie tego zrobić, co prowadzi do niższej skuteczności również dla etykiet o wysokim udziale. Porównując z wynikami dla zredukowanego zbioru danych (*zawierającego jedynie etykiety o udziale powyżej 10%*), zaprezentowanych w tabelach 5.7 oraz 5.6, wyniki F1-score dla etykiet *meme*, *cartoon style*, *realistic style* są gorsze o około 0.02.

Model	AUC	Precision	Recall	F1	AUPR
VGG19	0.762	0.375	0.499	0.400	0.386
RESNET50	0.767	0.364	0.554	0.396	0.377
DistilRoBERTa	0.658	0.269	0.550	0.309	0.268
XLNet	0.642	0.224	0.591	0.287	0.237
FFT	0.652	0.246	0.518	0.304	0.259
VGG19 + DistilRoBERTa	0.781	0.368	0.560	0.403	0.387
VGG19 + XLNet	0.780	0.382	0.566	0.407	0.390
RESNET50 + DistilRoBERTa	0.777	0.383	0.543	0.405	0.392
RESNET50 + XLNet	0.787	0.375	0.575	0.402	0.384
VGG19 + DistilRoBERTa + FFT	0.767	0.379	0.531	0.398	0.384
VGG19 + XLNet + FFT	0.778	0.378	0.557	0.398	0.381
RESNET50 + DistilRoBERTa + FFT	0.775	0.396	0.532	0.413	0.392
RESNET50 + XLNet + FFT	0.780	0.382	0.566	0.407	0.390

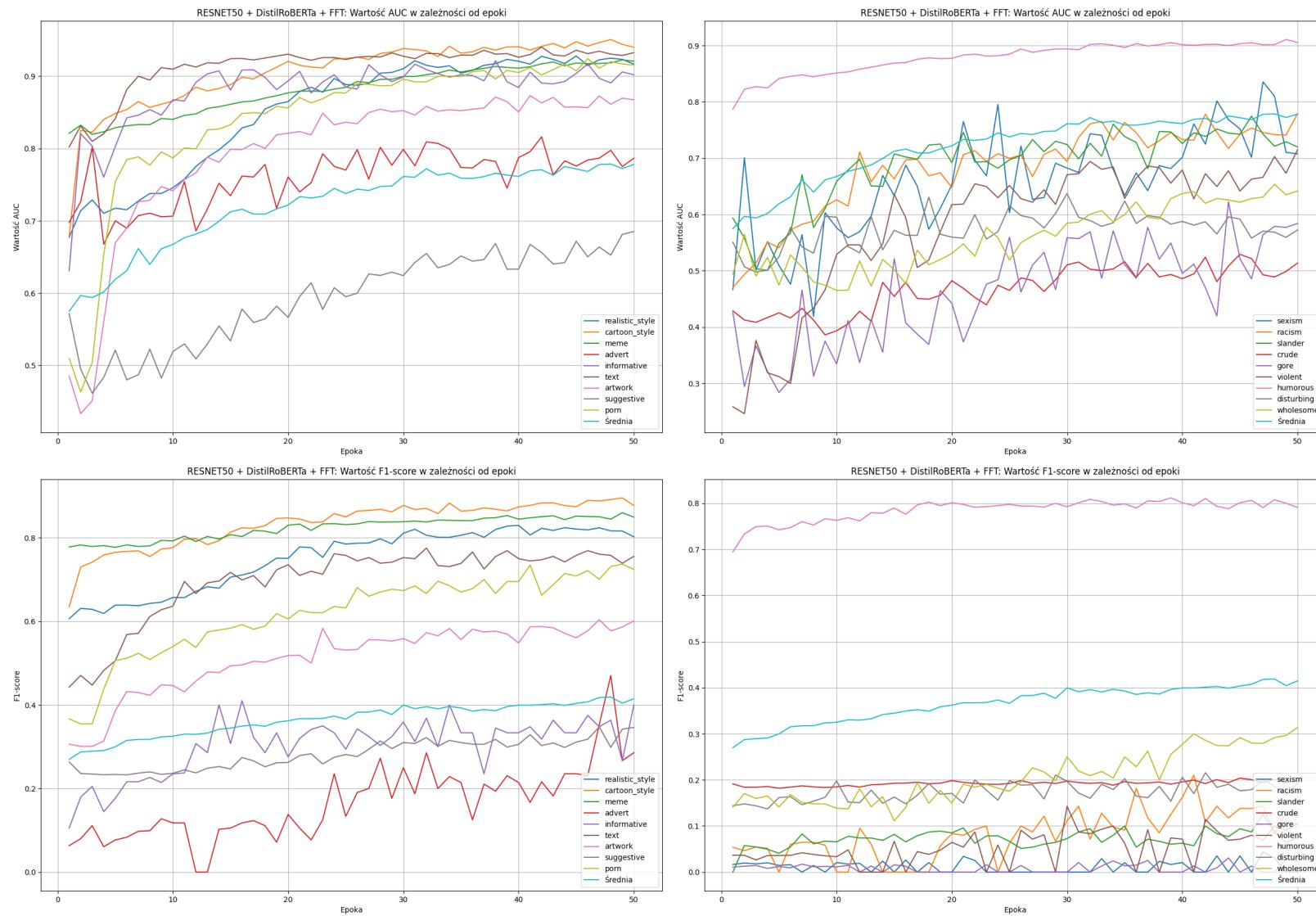
Tabela 5.1: Uśrednione wyniki dla pełnego zbioru danych

Etykieta	AUC	Precision	Recall	F1	AUPR
realistic_style	0.921	0.748	0.897	0.816	0.866
cartoon_style	0.945	0.901	0.877	0.888	0.925
meme	0.919	0.850	0.854	0.851	0.905
advert	0.786	0.264	0.429	0.321	0.195
informative	0.903	0.546	0.308	0.351	0.381
text	0.932	0.829	0.696	0.757	0.822
artwork	0.865	0.463	0.814	0.589	0.495
suggestive	0.667	0.235	0.580	0.330	0.210
porn	0.918	0.655	0.810	0.723	0.689
sexism	0.753	0.008	0.200	0.015	0.024
racism	0.752	0.096	0.222	0.123	0.078
slander	0.737	0.063	0.200	0.088	0.062
crude	0.503	0.110	0.935	0.198	0.112
gore	0.558	0.007	0.200	0.013	0.016
violent	0.684	0.060	0.225	0.089	0.058
humorous	0.905	0.768	0.833	0.799	0.842
disturbing	0.566	0.148	0.259	0.185	0.109
wholesome	0.638	0.383	0.240	0.292	0.268
Średnia	0.775	0.396	0.532	0.413	0.392

Tabela 5.2: Wyniki modelu RESNET50 + DistilRoBERTa + FFT dla pełnego zbioru danych

Etykieta	AUC	Precision	Recall	F1	AUPR
realistic_style	0.926	0.785	0.887	0.832	0.866
cartoon_style	0.951	0.871	0.905	0.888	0.934
meme	0.923	0.847	0.867	0.856	0.912
advert	0.798	0.122	0.314	0.165	0.120
informative	0.898	0.492	0.292	0.365	0.361
text	0.938	0.889	0.664	0.760	0.834
artwork	0.868	0.455	0.835	0.588	0.505
suggestive	0.667	0.202	0.706	0.313	0.210
porn	0.924	0.684	0.798	0.734	0.712
sexism	0.738	0.003	0.067	0.005	0.023
racism	0.748	0.113	0.289	0.159	0.080
slander	0.757	0.060	0.356	0.099	0.058
crude	0.497	0.111	0.960	0.198	0.120
gore	0.559	0.011	0.333	0.021	0.015
violent	0.641	0.008	0.050	0.014	0.057
humorous	0.908	0.775	0.844	0.807	0.845
disturbing	0.628	0.159	0.319	0.209	0.146
wholesome	0.620	0.308	0.293	0.272	0.259
Średnia	0.777	0.383	0.543	0.405	0.392

Tabela 5.3: Wyniki modelu RESNET50 + DistilRoBERTa dla pełnego zbioru danych



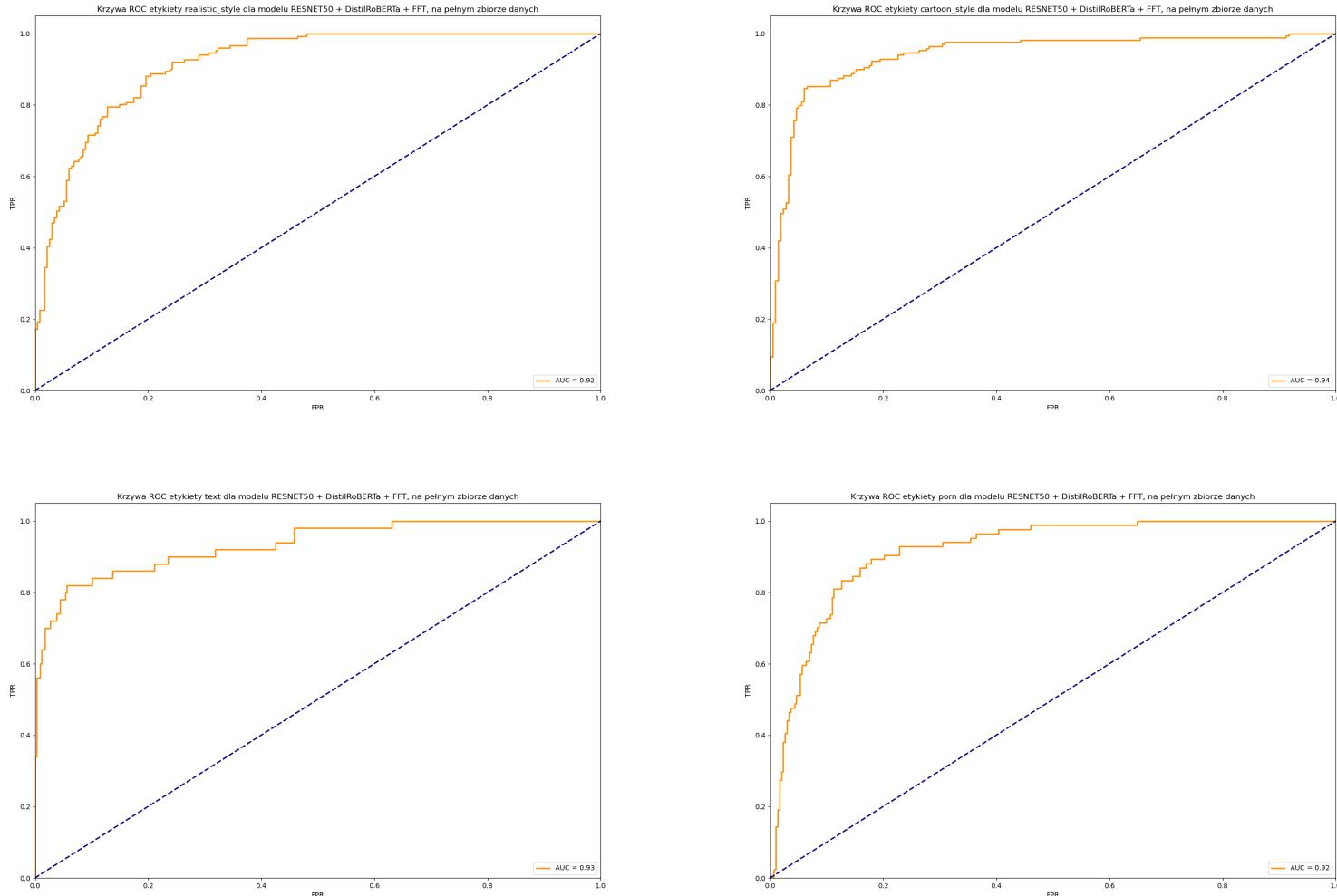
Rys. 5.2: Wyniki AUC, F1-score modelu RESNET50 + DistilRoBERTa + FFT dla pełnego zbioru danych

Etykieta	AUC	Precision	Recall	F1	AUPR
realistic_style	0.929	0.788	0.861	0.822	0.882
cartoon_style	0.952	0.850	0.927	0.886	0.935
meme	0.899	0.817	0.868	0.842	0.856
advert	0.776	0.092	0.486	0.151	0.074
informative	0.884	0.249	0.338	0.254	0.288
text	0.943	0.879	0.696	0.777	0.843
artwork	0.855	0.446	0.780	0.567	0.512
suggestive	0.696	0.225	0.745	0.344	0.215
porn	0.923	0.654	0.824	0.726	0.745
sexism	0.699	0.017	0.333	0.032	0.023
racism	0.714	0.052	0.200	0.081	0.057
slander	0.803	0.134	0.200	0.131	0.096
crude	0.600	0.180	0.335	0.231	0.147
gore	0.710	0.015	0.200	0.027	0.034
violent	0.549	0.027	0.725	0.052	0.025
humorous	0.891	0.718	0.882	0.791	0.805
disturbing	0.596	0.180	0.326	0.226	0.159
wholesome	0.642	0.306	0.347	0.322	0.263
Średnia	0.781	0.368	0.560	0.403	0.387

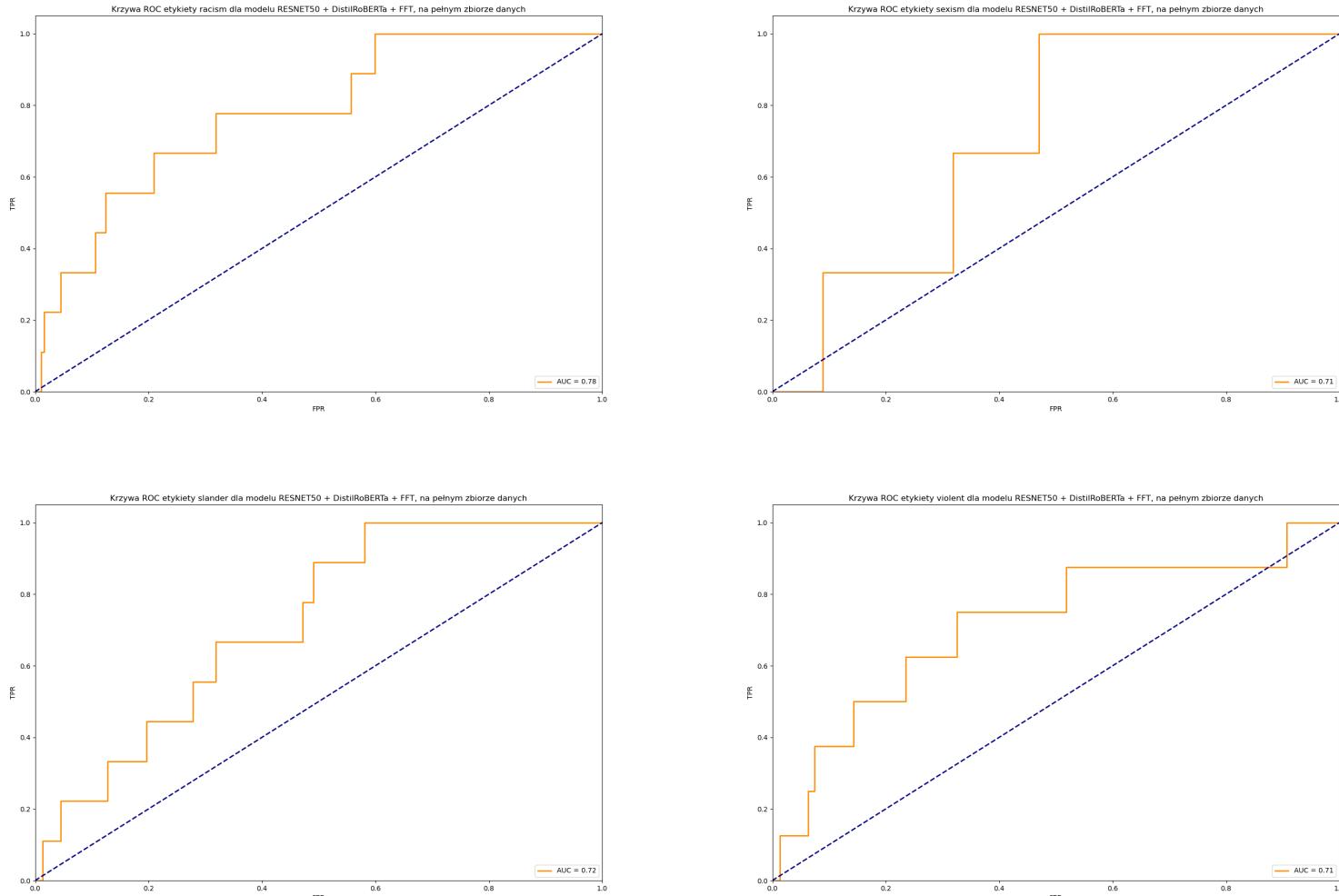
Tabela 5.4: Wyniki modelu VGG19 + DistilRoBERTa dla pełnego zbioru danych

Warto również zauważyć, że dla pełnego zbioru danych wartość AUC dla niektórych etykiet (*zaznaczona kolorem czerwonym w tabeli 5.2*) znacznie odbiega od oczekiwanych. Dla etykiet *sexism*, *racism*, *slander* model osiągnął $AUC \approx 0.75$ co jest przyzwoitym wynikiem, mimo że według metryki F1-score zdolność klasyfikacji tego modelu jest tragiczna ($F1 \approx 0.1$), uzyskana wartość AUPR również wskazuje, że klasyfikator jest tylko minimalnie lepszy niż klasyfikator losowy dla tych etykiet. Jest to prawdopodobnie spowodowane zjawiskiem, które opisałem w sekcji *Metryki - AUC* nie jest miarodajną metryką dla etykiet o ekstremalnie niskim udziale w zbiorze danych. Na rysunkach 5.3 oraz 5.4 umieściłem krzywe ROC dla wybranych etykiet, które uzyskały kolejno bardzo dobre oraz bardzo złe wyniki F1-score. W przypadku tych złych spodziewałem się zobaczyć widoczne przesunięcie krzywej w lewą stronę i istotnie, dla etykiet *racism* oraz *violent* jest ono subtelne ale widoczne, krzywa etykiety *sexism* ze względu na skrajnie małą liczbę próbek pozytywnych jest zbyt chaotyczna do sensownej analizy.

45



Rys. 5.3: Krzywe ROC dla wybranych etykiet, RESNET50 + DistilRoBERTa + FFT, pełny zbiór danych, mierzone dla ostatniej epoki uczenia



Rys. 5.4: Krzywe ROC dla wybranych etykiet, RESNET50 + DistilRoBERTa + FFT, pełny zbiór danych, mierzone dla ostatniej epoki uczenia

Wyniki dla zredukowanego zbioru danych, ukazane w tabeli 5.5, są zdecydowanie lepsze, osiągając średnie F1-score na poziomie ≈ 0.75 . Analizując wyniki poszczególnych etykiet dla wybranych modeli, przedstawionych w tabelach 5.7 oraz 5.6, widać że model jest w stanie prawidłowo sklasyfikować większość etykiet. Drobnym zaskoczeniem jest tutaj skuteczność dla etykiety *text*, która pomimo dość niskiego udziału w zbiorze danych ($\approx 14\%$) osiąga bardzo dobre wyniki: $AUC \approx 0.93$ oraz $F1 \approx 0.79$. Stosunkowo niska skuteczność dla etykiet *suggestive* oraz *artwork* sugeruje, że etykiety te są trudne do rozróżnienia przez klasyfikator, niskie wartości *Precision* oraz wysokie *Recall* sugerują że etykiety te przydzielane są zbyt często, zbyt liberalnie - może to wynikać ze skorelowania z etykietami o wysokim udziale. Przykładowo *artwork* jest silnie skorelowane z etykietą *cartoon style* - zdefiniowana przeze mnie korelacja wynosi 0.84, a więc 84% próbek *artwork* należy również do etykiety *cartoon style* (patrz tabela 3.4), natomiast nie jest to jedyna potencjalna przyczyna. Innym powodem niskiego *Precision* oraz wysokiego *Recall* może być niedouczenie - ponieważ dla etykiety *suggestive* nie ma aż tak dużych korelacji z innymi etykietami, niedouczenie wydaje się bardziej prawdopodobnym źródłem problemu.

Model	AUC	Precision	Recall	F1	AUPR
VGG19	0.898	0.713	0.801	0.747	0.758
RESNET50	0.908	0.704	0.825	0.753	0.777
DistilRoBERTa	0.767	0.513	0.741	0.588	0.543
XLNet	0.709	0.445	0.825	0.557	0.480
FFT	0.758	0.498	0.751	0.582	0.539
VGG19 + DistilRoBERTa	0.904	0.722	0.815	0.751	0.761
VGG19 + XLNet	0.896	0.698	0.799	0.735	0.753
RESNET50 + DistilRoBERTa	0.911	0.722	0.840	0.764	0.769
RESNET50 + XLNet	0.904	0.711	0.805	0.745	0.773
VGG19 + DistilRoBERTa + FFT	0.904	0.722	0.805	0.750	0.764
VGG19 + XLNet + FFT	0.895	0.703	0.813	0.740	0.755
RESNET50 + DistilRoBERTa + FFT	0.912	0.723	0.816	0.759	0.773
RESNET50 + XLNet + FFT	0.901	0.713	0.796	0.747	0.767

Tabela 5.5: Uśrednione wyniki dla zredukowanego zbioru danych

Wykorzystany model cierpi na niedouczenie, widoczne na rysunkach 5.5 oraz 5.6 - funkcja straty na zbiorze testowym zbiega do minimum, po czym w nim utyka. Oznacza to, że odpowiednio dostrajając oraz powiększając model prawdopodobnie da się uzyskać lepsze wyniki.

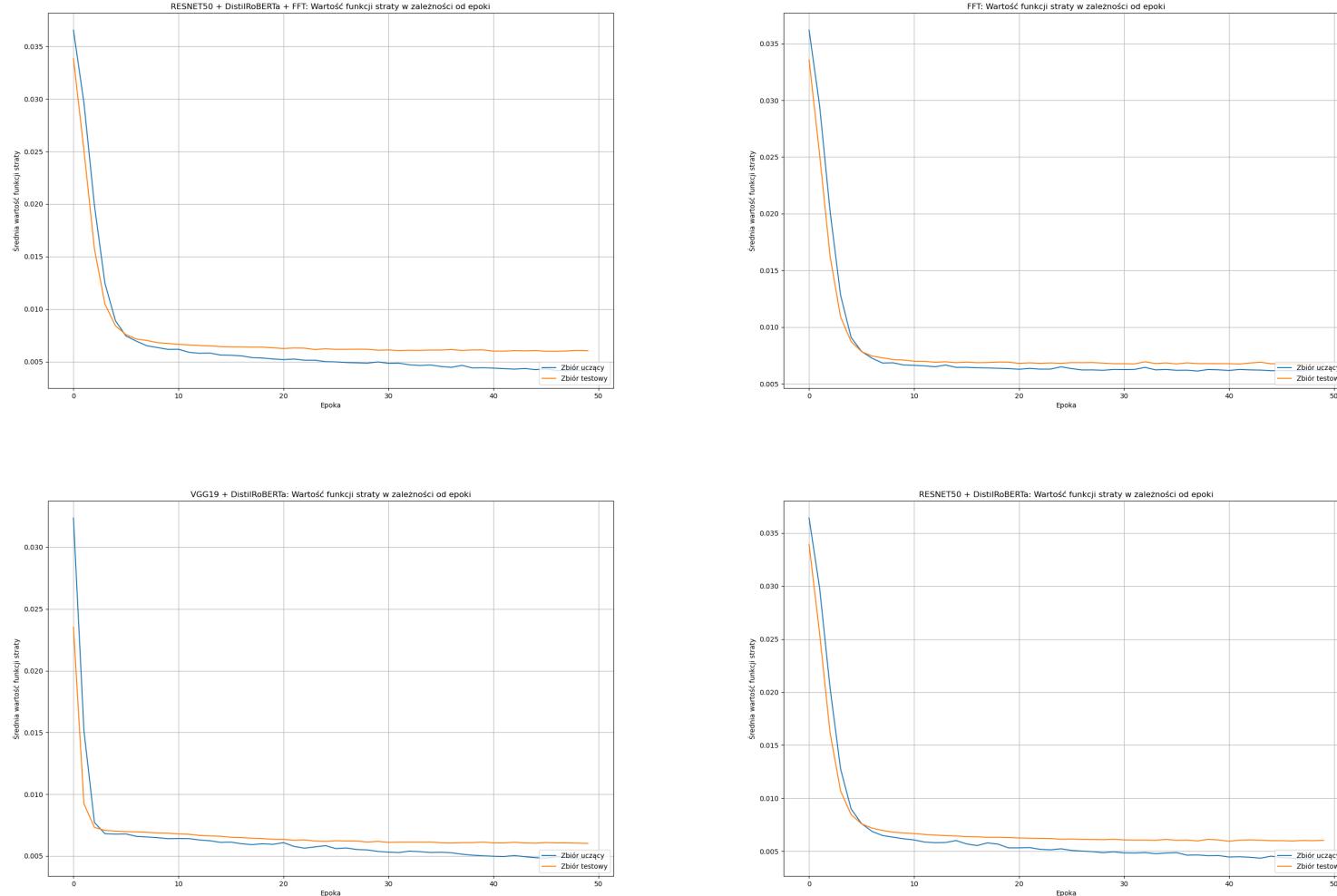
Choć wymagałoby to dalszych badań, to uważam że umieszczenie warstwy dropout na wejściu sieci było błędem, Teoretycznie zmusza to sieć do wykorzystywania nie tylko najbardziej oczywistych cech ale również tych bardziej subtelnych, to ten sam efekt powinna dać warstwa dropout umieszczona pomiędzy warstwą ukrytą a wyjściową - jednocześnie pozwalając sieci korzystać ze wszystkich cech obrazu zamiast losowo wybranych 75% z nich.

Etykieta	AUC	Precision	Recall	F1	AUPR
realistic_style	0.936	0.801	0.889	0.842	0.882
cartoon_style	0.957	0.881	0.911	0.895	0.935
meme	0.939	0.897	0.859	0.877	0.935
text	0.940	0.875	0.752	0.807	0.867
artwork	0.888	0.500	0.870	0.634	0.600
suggestive	0.784	0.337	0.710	0.455	0.312
porn	0.934	0.717	0.840	0.773	0.765
humorous	0.914	0.770	0.890	0.826	0.859
Średnia	0.911	0.722	0.840	0.764	0.769

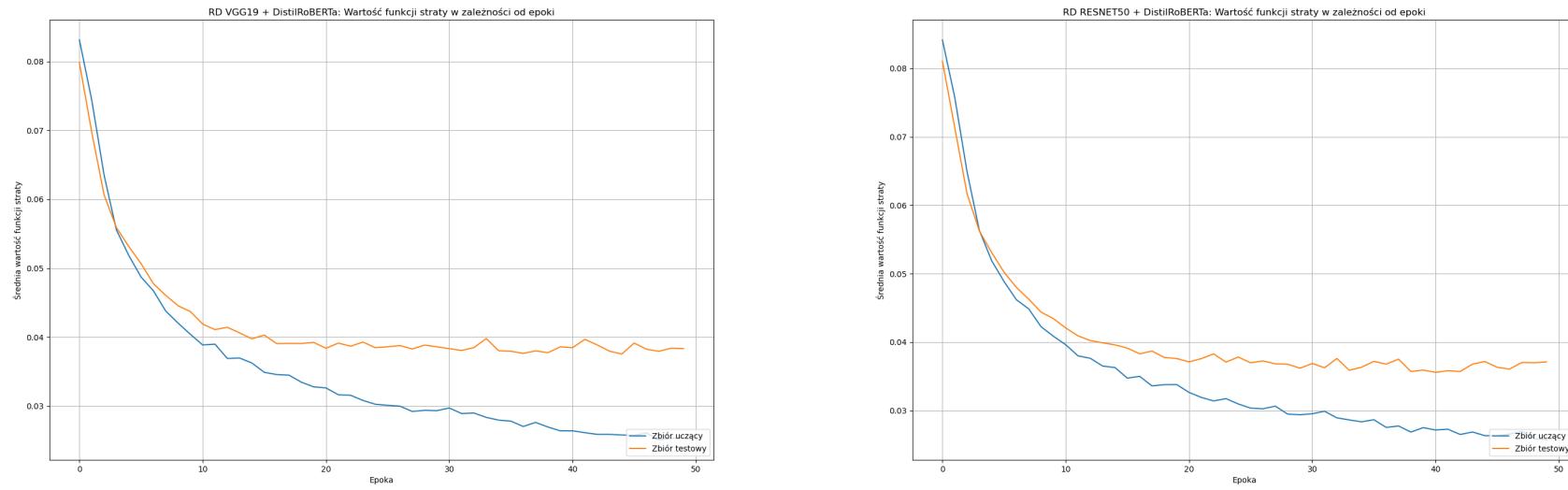
Tabela 5.6: Wyniki modelu RESNET50 + DistilRoBERTa dla zredukowanego zbioru danych

Etykieta	AUC	Precision	Recall	F1	AUPR
realistic_style	0.938	0.816	0.898	0.855	0.895
cartoon_style	0.958	0.875	0.928	0.901	0.942
meme	0.928	0.872	0.860	0.865	0.922
text	0.931	0.915	0.696	0.789	0.848
artwork	0.872	0.463	0.829	0.594	0.586
suggestive	0.771	0.317	0.643	0.423	0.310
porn	0.926	0.764	0.764	0.764	0.740
humorous	0.908	0.749	0.899	0.817	0.844
Średnia	0.904	0.722	0.815	0.751	0.761

Tabela 5.7: Wyniki modelu VGG19 + DistilRoBERTa dla zredukowanego zbioru danych



Rys. 5.5: Funkcje straty dla wybranych modeli



Rys. 5.6: Funkcje straty modeli VGG19/RESNET50 + DistilRoBERTa dla zredukowanego (RD) zbioru danych

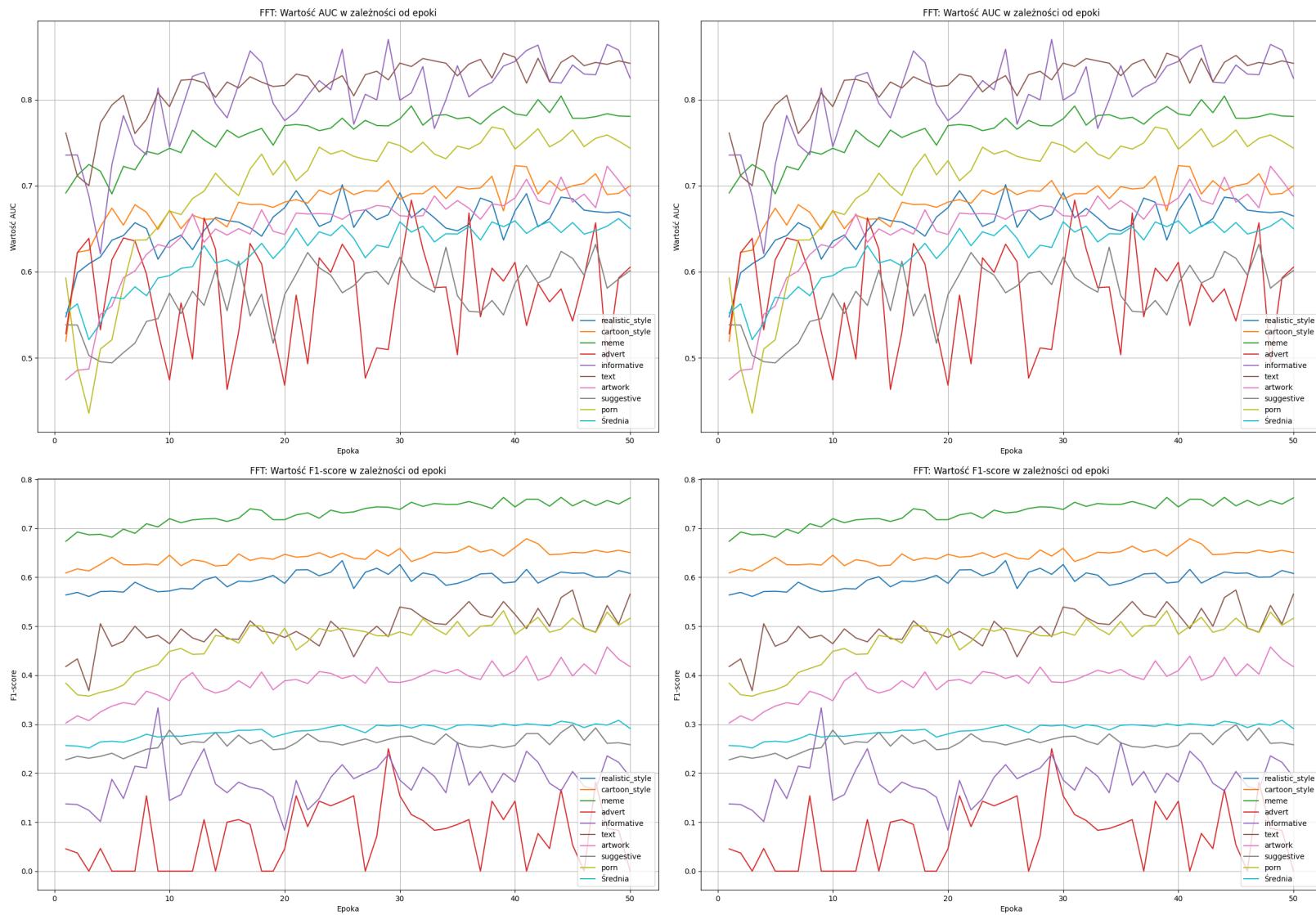
Uzyskane wyniki dla etykiet o wysokim udziale: *meme*, *cartoon style*, *realistic style* są na tyle wysokie by można było wykorzystać je w rzeczywistych zadaniach, również dla etykiet *text*, *porn* są one stosunkowo dobre. Etykietę *humorous* pomijam w tym miejscu, ponieważ jest ona podzioborem etykiety *meme* (patrz tabela 3.4).

Zarówno dla pełnego (tabela 5.1), jak i zredukowanego zbioru danych (tabela 5.5) najlepsze wyniki osiągnęły modele wykorzystujące modalność wizualną oraz tekstową. Wprowadzona przeze mnie modalność związana z transformatą Fouriera bardziej przeszka-dza niż pomaga w klasyfikacji, choć nie w przypadku każdej z wykorzystanych kombinacji modeli.

Zaproponowane przeze mnie wykorzystanie transformaty Fouriera, mimo że nie wniosło istotnej poprawy do realizacji zadania klasyfikacji, pokazuje że rozwiązywanie to ma potencjał. Należy pamiętać, że wykorzystany model RESNET18 jest znacznie mniej zaawansowany niż RESNET50 czy VGG19, gorsze wyniki nie są więc niczym dziwnym - zwłaszcza że wytrenowany na obrazach ze zbioru Tiny ImageNet ma do dyspozycji jedynie 64×64 pikseli. Dokładne wyniki modelu FFT przedstawiłem w tabeli 5.8 oraz na rysunku 5.7.

Etykieta	AUC	Precision	Recall	F1	AUPR
realistic_style	0.674	0.475	0.852	0.607	0.565
cartoon_style	0.696	0.537	0.847	0.655	0.630
meme	0.788	0.708	0.813	0.756	0.720
advert	0.568	0.054	0.200	0.082	0.050
informative	0.843	0.140	0.431	0.207	0.129
text	0.843	0.494	0.588	0.537	0.493
artwork	0.696	0.359	0.528	0.421	0.342
suggestive	0.598	0.175	0.690	0.275	0.199
porn	0.758	0.383	0.757	0.509	0.419
sexism	0.530	0.003	0.200	0.006	0.013
racism	0.560	0.034	0.511	0.062	0.033
slander	0.544	0.028	0.244	0.048	0.032
crude	0.528	0.177	0.380	0.195	0.148
gore	0.580	0.003	0.067	0.006	0.021
violent	0.697	0.044	0.400	0.077	0.043
humorous	0.787	0.616	0.826	0.705	0.636
disturbing	0.500	0.094	0.415	0.150	0.089
wholesome	0.554	0.106	0.580	0.167	0.104
Średnia	0.652	0.246	0.518	0.304	0.259

Tabela 5.8: Wyniki modelu FFT dla pełnego zbioru danych



Rys. 5.7: Wyniki AUC, F1-score modelu FFT dla pełnego zbioru danych

Odnosząc się do algorytmów łączenia cech; wyniki widoczne są w tabeli 5.9. Najlepsze wyniki osiągnęła konkatencja, której przewaga wedle metryki F1-score nad wyborem maksimum czy uśrednianiem wynosi ≈ 0.1 dla odpowiadających sobie modeli. Jest to duża, istotna różnica. Wynik ten nie zgadza się z pracą "A New Multimodal Benchmark Dataset for Fine-grained Fake News Detection" [35], wedle której najlepsze wyniki uzyskał wybór maksimum - może to być spowodowane tym, że w zebranym przez mnie zbiorze danych tylko $\approx 76\%$ zawiera tekst, przez co kanał tekstowy jest mniej informatywny niż kanał graficzny.

Model	AUC	Precision	Recall	F1	AUPR
Konkat. VGG19 + DistilRoBERTa	0.904	0.722	0.815	0.751	0.761
Konkat. RESNET50 + DistilRoBERTa	0.911	0.722	0.840	0.764	0.769
Maks. VGG19 + DistilRoBERTa	0.829	0.623	0.714	0.653	0.652
Maks. RESNET50 + DistilRoBERTa	0.835	0.606	0.760	0.663	0.657
Średnia, VGG19 + DistilRoBERTa	0.846	0.620	0.753	0.666	0.677
Średnia, RESNET50 + DistilRoBERTa	0.852	0.613	0.797	0.679	0.685

Tabela 5.9: Uśrednione wyniki dla badań łączenia cech, na zredukowanym (RD) zbiorze

W przypadku badań wpływu zmiany rozmiaru z paddingiem, wyniki zapisałem w tabeli 5.10. Różnice między modelami stosującymi padding a tymi, które go nie stosują jest pomijalna, można więc wyciągnąć z tego wniosek, że jest to operacja zbędna.

Model	AUC	Precision	Recall	F1	AUPR
VGG19 + DistilRoBERTa	0.904	0.722	0.815	0.751	0.761
RESNET50 + DistilRoBERTa	0.911	0.722	0.840	0.764	0.769
VGG19 + DistilRoBERTa, bez pad.	0.904	0.727	0.803	0.752	0.761
RESNET50 + DistilRoBERTa, bez pad.	0.914	0.720	0.828	0.760	0.773

Tabela 5.10: Uśrednione wyniki dla badań wpływu paddingu, na zredukowanym (RD) zbiorze

6. PODSUMOWANIE

Zebrany zbiór danych liczy sobie 76 163, z czego 1542 posiada ręcznie przydzielone etykiety. Zebrany zbiór został przeze mnie zastosowany do realizacji zadania klasyfikacji wielomodalnej, przy wykorzystaniu wytrenowanych modeli VGG19, RESNET50, DistilRoBERTa, XLNet, a także autorskiego modelu RESNET18 pracującego na transformacie Fouriera. Uzyskane surowe wyniki, dostarczone w ramach tej pracy, stanowią wartości referencyjne dla zebranego zbioru, w połączeniu z dokładnym opisem zastosowanych metod, modeli i wykorzystanych parametrów powinny być użyteczne dla badaczy chcących ten zbiór wykorzystać,

Najlepsze wyniki osiągnął model RESNET50 w połączeniu z DistilRoBERTa. Model wykorzystujący transformatę Fouriera wykazał potencjał, którego jednak nie udało się w pełni wykorzystać ze względu na zbyt prosty model - mimo to, w niektórych przypadkach poprawiał skuteczność klasyfikacji. Badania nad algorytmami łączenia cech wykazały, że spośród uśredniania, wyboru maksimum oraz konkatenacji, zdecydowanie najlepsze wyniki osiąga konkatenacja wektorów cech. Badania wpływu zmiany rozmiaru obrazu z zachowaniem proporcji za pomocą *paddingu* wskazują, że operacja ta nie ma istotnego wpływu na wyniki klasyfikacji.

Wytrenowane na tym zbiorze modele są w stanie skutecznie ($F1 \approx 0.85$) klasyfikować obrazy o realistycznym oraz kreskówkowym stylu graficznym, a także internetowe memy. Nieco gorzej, choć nadal dobrze ($F1 \approx 0.8$) model radzi sobie z wykrywaniem pornografii oraz obrazów zdominowanych przez tekst.

Praca pozostawia wiele możliwości do poprawy. Zbiór danych jest niebalansowany, co częściowo rekompensowałem stosując ważoną funkcję straty oraz odpowiednie metryki, natomiast można zrobić wiele więcej, szczególny potencjał może mieć losowe powielanie próbek ze zbioru danych posiadających rzadkie etykiety (*oversampling*). Ponadto, oprócz analizowanych w ramach tej pracy 1542 obrazach które już posiadają etykietę nie należy zapomnieć o pozostałych 74 621 które zebrałem a którym nie zdążyłem przydzielić etykiet - zatem istnieje duży potencjał na znalezienie większej liczby próbek dla etykiet mniejszościowych. Model wykorzystujący transformatę Fouriera może z pewnością uzyskać lepsze wyniki, jednak wymaga to zmiany architektury na RESNET50 i wytrenowania go na zbiorze danych ImageNet.

BIBLIOGRAFIA

- [1] *Amazon extract*, Dostęp w serwisie Amazon Web Services (AWS): <https://aws.amazon.com/textract/>. [dostęp 10.06.2024].
- [2] *Evidently ai team, "how to explain the roc curve and roc auc score?"*, Dostęp w serwisie internetowym Evidentlyai: <https://www.evidentlyai.com/classification-metrics/explain-roc-curve>. [dostęp 09.06.2024].
- [3] "*fleiss' kappa*", Angielska wikipedia, https://en.wikipedia.org/wiki/Fleiss%27_kappa. [dostęp 26.05.2024].
- [4] *The mathworks, inc, fourier transform*, Dokumentacja programu MATLAB: <https://www.mathworks.com/help/images/fourier-transform.html>. [dostęp 09.06.2024].
- [5] "*modalność*", Słownik języka polskiego PWN, <https://sjp.pwn.pl/slowniki/modalno%C5%82%C4%85%C5%84.html>. [dostęp 27.05.2024].
- [6] *Roc curve shows strange results for imbalanced dataset*, Dostęp na forum internetowym DataScience Stackexchange <https://datascience.stackexchange.com/questions/24315/roc-curve-shows-strange-results-for-imbalanced-dataset>. [dostęp 09.06.2024].
- [7] *tesseract-ocr/tesseract*, Repozytorium w serwisie Github: [https://github.com/tesseract-ocr/tesseract/](https://github.com/tesseract-ocr/tesseract). [dostęp 10.06.2024].
- [8] *Torch contributors, dokumentacja augmentacji i transformacji danych, pytorch*: <https://pytorch.org/vision/main/transforms.html>. [dostęp 09.06.2024].
- [9] *Torch contributors, dokumentacja bcewithlogitsloss, pytorch*: <https://pytorch.org/docs/stable/generated/torch.nn.BCEWithLogitsLoss.html>. [dostęp 09.06.2024].
- [10] *Torch contributors, dokumentacja resnet50, pytorch*: <https://pytorch.org/vision/main/models/generated/torchvision.models.resnet50.html#torchvision.models.resnet50>. [dostęp 09.06.2024].
- [11] *Torch contributors, dokumentacja vgg19, pytorch*: <https://pytorch.org/vision/main/models/generated/torchvision.models.vgg19.html>. [dostęp 09.06.2024].
- [12] *What is ocr (optical character recognition)?*, Dostęp w serwisie Amazon Web Services (AWS): https://aws.amazon.com/what-is/ocr/?fbclid=IwZKh0bgNhZWOCMTAAAR1Ey43xbkgK26dqm3pg89kaej6heTp2kg_hovyIEGOQMsWAT1uRZTL-IA_aem_ZmFrZWR1bW15MTZieXRlcw. [dostęp 10.06.2024].
- [13] *Wyniki resnet18 dla zbioru danych tinyimagenet*, Dostęp w serwisie internetowym Kaggle: https://paperswithcode.com/sota/image-classification-on-tiny-imagenet-1?tag_filter=3. [dostęp 09.06.2024].

- [14] *Xlnet (base-sized model)*, Dokumentacja XLNet w serwisie Huggingface: <https://huggingface.co/xlnet/xlnet-base-cased>. [dostęp 09.06.2024].
- [15] *Meme generator dataset*, Repozytorium PyTesseract w serwisie Github: <https://github.com/madmaze/pytesseract>. 17.05.2018. [dostęp 09.06.2024].
- [16] *Google llc. google colaboratory*. 2024. [dostęp 18.06.2024].
- [17] avcontentteam, *What is principal component analysis (pca)?*, Dostęp w serwisie internetowym Analytics Vidhya: <https://www.analyticsvidhya.com/blog/2016/03/pca-practical-guide-principal-component-analysis-python/>. 22.05.2024. [dostęp 09.06.2024].
- [18] Bingham, H., *How many words are there in a novel?*, Dostęp w serwisie internetowym Jericho Writers <https://jerichowriters.com/average-novel-wordcount/>. [dostęp 18.06.2024].
- [19] Brown, J., Gharineiat, Z., Raj, N., *Cnn based image classification of malicious uavs*, Applied Sciences. 2022, tom 13, str. 240.
- [20] co., S.H., Repozytorium PyTesseract w serwisie Github: <https://github.com/madmaze/pytesseract>. [dostęp 09.06.2024].
- [21] Das, A., Wahi, J.S., Li, S., *Detecting hate speech in multi-modal memes*. 2020.
- [22] Davis, J., Goadrich, M., *The relationship between precision-recall and roc curves*, w: *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06 (Association for Computing Machinery, New York, NY, USA, 2006), str. 233–240.
- [23] Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L., *Imagenet: A large-scale hierarchical image database*, w: *2009 IEEE conference on computer vision and pattern recognition* (Ieee, 2009), str. 248–255.
- [24] scikit-learn developers, *Pca*, Dokumentacja Scikit-Learn, <https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html>. [dostęp 06.06.2024].
- [25] scikit-learn developers, *"precision-recall"*, Dokumentacja Scikit-Learn, https://scikit-learn.org/stable/auto_examples/model_selection/plot_precision_recall.html. [dostęp 06.06.2024].
- [26] Devlin, J., Chang, M.W., Lee, K., Toutanova, K., *Bert: Pre-training of deep bidirectional transformers for language understanding*. 2019.
- [27] Dutta, M., *Fuzzy string matching – a hands-on guide*, Dostęp w serwisie internetowym Analytics Vidhya: <https://www.analyticsvidhya.com/blog/2021/07/fuzzy-string-matching-a-hands-on-guide/>. 19.09.2023. [dostęp 09.06.2024].
- [28] Google, *Classification: Roc curve and auc*, Kurs na platformie Google Developers: <https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc>. 18.07.2022. [dostęp 09.06.2024].
- [29] He, K., Zhang, X., Ren, S., Sun, J., *Deep residual learning for image recognition*. 2015.
- [30] JaidedAI, Repozytorium EasyOCR w serwisie GitHub, <https://github.com/jaidedai/easyocr>. [dostęp 09.06.2024].
- [31] Kingma, D.P., Ba, J., *Adam: A method for stochastic optimization*. 2017.

- [32] mnmoustafa, M.A., *Tiny imagenet*, <https://kaggle.com/competitions/tiny-imagenet>. 2017.
- [33] Morales, F., Internetowa dokumentacja Keras-OCR: <https://keras-ocr.readthedocs.io/en/latest/>. 2019. [dostęp 09.06.2024].
- [34] Mori, M., MacDorman, K.F., Kageki, N., *The uncanny valley [from the field]*, IEEE Robotics Automation Magazine. 2012, tom 19, 2, str. 98–100.
- [35] Nakamura, K., Levy, S., Wang, W.Y., *Fakeddit: A new multimodal benchmark dataset for fine-grained fake news detection*, w: *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pod red. N. Calzolari, F. Béchet, P. Blache, K. Choukri, C. Cieri, T. Declerck, S. Goggi, H. Isahara, B. Maegaard, J. Mariani, H. Mazo, A. Moreno, J. Odijk, S. Piperidis (European Language Resources Association, Marseille, France, 2020), str. 6149–6157.
- [36] Popescu, M.C., Balas, V., Perescu-Popescu, L., Mastorakis, N., *Multilayer perceptron and neural networks*, WSEAS Transactions on Circuits and Systems. 2009, tom 8.
- [37] Pound, M., *Data analysis 6: Principal component analysis (pca) - computerphile*, YouTube, Computerphile <https://youtu.be/TJdH6rPA-TI?si=ahYXqPIFffq6LXhV>. [dostęp 09.06.2024].
- [38] Ruder, S., *An overview of gradient descent optimization algorithms*, arXiv preprint arXiv:1609.04747. 2016.
- [39] Sanh, V., Debut, L., Chaumond, J., Wolf, T., *Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter*, ArXiv. 2019, tom abs/1910.01108.
- [40] Sharma, C., Paka, Scott, W., Bhageria, D., Das, A., Poria, S., Chakraborty, T., Gambäck, B., *Task Report: Memotion Analysis 1.0 @SemEval 2020: The Visuo-Lingual Metaphor!*, w: *Proceedings of the 14th International Workshop on Semantic Evaluation (SemEval-2020)* (Association for Computational Linguistics, Barcelona, Spain, 2020).
- [41] shashwatswain, *Text augmentation techniques in nlp*, Dostęp w serwisie internetowym Geeks for Geeks: <https://www.geeksforgeeks.org/text-augmentation-techniques-in-nlp/>. 03.01.2024. [dostęp 09.06.2024].
- [42] Sherratt, V., Pimblet, K., Dethlefs, N., *Multi-channel convolutional neural network for precise meme classification*, w: *Proceedings of the 2023 ACM International Conference on Multimedia Retrieval, ICMR '23* (Association for Computing Machinery, New York, NY, USA, 2023), str. 190–198.
- [43] Simonyan, K., Zisserman, A., *Very deep convolutional networks for large-scale image recognition*. 2015.
- [44] Sleeman, W.C., Kapoor, R., Ghosh, P., *Multimodal classification: Current landscape, taxonomy and future directions*, ACM Comput. Surv. 2022, tom 55, 7.
- [45] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R., *Dropout: A simple way to prevent neural networks from overfitting*, Journal of Machine Learning Research. 2014, tom 15, 56, str. 1929–1958.
- [46] Suryawanshi, S., Chakravarthi, B.R., Arcan, M., Buitelaar, P., *Multimodal meme dataset (MultiOFF) for identifying offensive content in image and text*, w: *Proceedings of the Second*

Workshop on Trolling, Aggression and Cyberbullying, pod red. R. Kumar, A.K. Ojha, B. Lahiri, M. Zampieri, S. Malmasi, V. Murdock, D. Kadar (European Language Resources Association (ELRA), Marseille, France, 2020), str. 32–41.

- [47] Victor Sanh, Lysandre Debut, J.C.T.W., *Model card for distilroberta base*, Dokumentacja DistilRoBERTa w serwisie Huggingface: <https://huggingface.co/distilbert/distilroberta-base>. [dostęp 09.06.2024].
- [48] Weng, J., *Nlp text preprocessing: A practical guide and template*, Dostęp w serwisie internetowym Towards Data Science: <https://towardsdatascience.com/nlp-text-preprocessing-a-practical-guide-and-template-d80874676e79>. 30.08.2019. [dostęp 05.02.2021].
- [49] Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R., Le, Q.V., *Xlnet: Generalized autoregressive pretraining for language understanding*. 2020.

SPIS RYSUNKÓW

2.1	Wpływ kontekstu i zewnętrznej wiedzy na interpretacje grafiki [21]	4
2.2	Przykładowa architektura sieci wykorzystująca <i>Early Fusion</i> [44]	5
2.3	Przykładowa architektura sieci wykorzystująca <i>Late Fusion</i> [44]	6
2.4	Przykład działania OCR [30]	7
2.5	Przykład przetworzonego i augmentowanego obrazu	8
2.6	Przykład działania PCA [17]	9
3.1	Zawartość procentowa etykiet w zbiorze danych	10
3.2	Przykładowe obrazy ze zbioru danych, wraz z etykietami	13
3.3	Statystyki obrazów w zbiorze danych	15
3.4	Statystyki tekstu w zbiorze danych	15
4.1	Ogólny schemat modelu	21
4.2	Przykładowe widma transformaty Fouriera różnych obrazów	26
4.3	Reprezentacje liczby zespolonej	27
4.4	Przykładowe obrazy ze zbioru Tiny ImageNet [32]	28
4.5	Oryginalna architektura RESNET18 [19]	29
4.6	Wyniki wytrenowanego modelu RESNET18 na transformatach obrazów ze zbioru Tiny Imagenet	29
4.7	Schemat modelu klasyfikatora	32
4.8	Wykres binarnej entropii krzyżowej	33
4.9	Przykładowa krzywa ROC [28]	35
4.10	Krzywa ROC dla idealnego, dobrego oraz losowego klasyfikatora [2]	36
4.11	Przykład przesunięcia krzywej ROC dla ekstremalnie niebalansowanego zbioru danych [6]	36
4.12	Zależność F1-score od <i>Precision</i> i <i>Recall</i>	37
4.13	Przykład krzywej PR [25]	38
5.1	Zmiana learning rate w procesie uczenia	40
5.2	Wyniki AUC, F1-score modelu RESNET50 + DistilRoBERTa + FFT dla pełnego zbioru danych	43
5.3	Krzywe ROC dla wybranych etykiet, RESNET50 + DistilRoBERTa + FFT, pełny zbiór danych, mierzone dla ostatniej epoki uczenia	45
5.4	Krzywe ROC dla wybranych etykiet, RESNET50 + DistilRoBERTa + FFT, pełny zbiór danych, mierzone dla ostatniej epoki uczenia	46
5.5	Funkcje straty dla wybranych modeli	49

5.6	Funkcje straty modeli VGG19/RESNET50 + DistilRoBERTa dla zredukowanego (RD) zbioru danych	50
5.7	Wyniki AUC, F1-score modelu FFT dla pełnego zbioru danych	52

SPIS LISTINGÓW

SPIS TABEL

3.1	Liczność i udział etykiet w zbiorze danych	11
3.2	Statystyki obrazów w zbiorze danych	14
3.3	Średnia i odchylenie standardowe wartości pikseli w zbiorze danych	16
3.4	Korelacja etykiet	17
5.1	Uśrednione wyniki dla pełnego zbioru danych	41
5.2	Wyniki modelu RESNET50 + DistilRoBERTa + FFT dla pełnego zbioru danych . .	42
5.3	Wyniki modelu RESNET50 + DistilRoBERTa dla pełnego zbioru danych	42
5.4	Wyniki modelu VGG19 + DistilRoBERTa dla pełnego zbioru danych	44
5.5	Uśrednione wyniki dla zredukowanego zbioru danych	47
5.6	Wyniki modelu RESNET50 + DistilRoBERTa dla zredukowanego zbioru danych . .	48
5.7	Wyniki modelu VGG19 + DistilRoBERTa dla zredukowanego zbioru danych . . .	48
5.8	Wyniki modelu FFT dla pełnego zbioru danych	51
5.9	Uśrednione wyniki dla badań łączenia cech, na zredukowanym (RD) zbiorze . . .	53
5.10	Uśrednione wyniki dla badań wpływu paddingu, na zredukowanym (RD) zbiorze . .	53