



POLSKO-JAPOŃSKA
AKADEMIA TECHNIK
KOMPUTEROWYCH

Praca doktorska

METODY ROZPOZNAWANIA OBRAZÓW
W BADANIU DNA OKA DLA CELÓW
DIAGNOSTYKI MEDYCZNEJ

Autor: mgr inż. Tomasz Krzywicki

Promotor: Prof. dr hab. Lech Polkowski

07 września 2023

Podziękowania

Wiele osób zarówno w Uniwersytecie Warmińsko-Mazurskim (UWM) i poza nim służyło autorowi pomocą podczas przygotowywania tej rozprawy.

Autor składa podziękowanie Panu Profesorowi dr hab. Lechowi Polkowskiemu, kierownikowi Katedry Metod Matematycznych UWM, za jego pomoc w kształtowaniu ostatecznej postaci rozprawy, wsparcie i radę gdy były one potrzebne.

Podziękowanie autora płynie także do środowiska Katedry Metod Matematycznych Informatyki UWM i Wydziału Matematyki i Informatyki UWM za zapewnienie dobrej atmosfery do pracy i stworzenie warunków sprzętowych dla jej wykonywania.

Pani Profesor dr hab. Marii Orłowskiej, prorektor ds. nauki w Polsko-Japońskiej Akademii Technik Komputerowych autor chciałby podziękować za życzliwe przyjęcie i cenne wskazówki proceduralne.

Niezwykle istotna jest atmosfera stwarzana autorowi w jego środowisku prywatnym. Serdeczne wyrazy wdzięczności płyną do Rodziców oraz do Partnerki za nieustanne wsparcie, motywację i miłość, które były nieodzownym wsparciem podczas pisania rozprawy doktorskiej. Bez Ich nieocenionej obecności i wsparcia, ten sukces nie byłby możliwy.

Streszczenie

Celem rozprawy doktorskiej było opracowanie i zaproponowanie nowatorskich metod lokalizowania środków cech anatomicznych na zdjęciach dna oka: dołączka oraz tarczy nerwu wzrokowego. Badanie wykorzystало dwie różne strategie: podejście oparte na technikach przetwarzania obrazów i analizie kształtów oraz podejście wykorzystujące zaawansowane głębokie sieci neuronowe, uczenie transferowe i uczenie zespołowe.

W ocenie metod lokalizacji środków obiektów, takich jak sieci neuronowe czy ogólniej techniki AI, stosuje się miary efektywności, takie jak dokładność (*ang. accuracy*) i pokrycie (*ang. coverage*). Dokładność jest wyrażana jako średnia odległość euklidesowa (Średni Błąd Euklidesowy, SBE) między lokalizacjami środków wyznaczonymi przez eksperta i przez metodę: $SBE = \sqrt{\frac{\sum_{i=1}^n (\hat{y} - y)^2}{n}}$, gdzie \hat{y} – współrzędne środka obiektu wyznaczone przez eksperta, y – współrzędne środka wyznaczone przez metodę, n – liczba wyznaczonych środków. Pokrycie jest odsetkiem liczby środków obiektów wyznaczonych przez metodę w stosunku do liczby środków wyznaczonych przez eksperta: $coverage = \frac{\text{liczba środków wyznaczonych przez metodę}}{\text{liczba środków wyznaczonych przez eksperta}}$.

W ramach zastosowania pierwszego podejścia, skoncentrowano się na wykorzystaniu generycznych algorytmów przetwarzania obrazów. Zastosowano także metody analizy kształtów w celu zwiększenia efektywności identyfikacji cech anatomicznych siatkówki o charakterystycznej strukturze. Opracowana w ten sposób metoda była efektywna pod względem precyzji, a także wykorzystanych zasobów, jednak wykazywała pewne ograniczenia w kontekście zróżnicowanych warunków ekspozycyjnych zdjęć.

W drugim podejściu, zastosowano głębokie sieci neuronowe, które wykazują wysokie zdolności ekstrakcji złożonych wzorców występujących na obrazach. Wdrożono także uczenie transferowe, celem wykorzystania wiedzy zebranej

z innych zbiorów danych. Wykorzystanie uczenia zespołowego pozwoliło na efektywne zwiększenie ogólnej wydajności proponowanej metody. Głębokie sieci neuronowe potrafiły radzić sobie ze zróżnicowanymi warunkami i kształtami, lecz wymagały większych zasobów obliczeniowych i czasowych w porównaniu z pierwszym podejściem.

W celu porównania efektywności i konsumpcji zasobów opracowanych metod, przeprowadzono szereg testów. Oceny obejmowały precyzję i pokrycie lokalizacji, wydajność obliczeniową, zużycie pamięci oraz czas wykonywania metod w dwóch trybach: sekwencyjnym i równoległym. Wyniki wykazały, że metoda oparta na głębokich sieciach neuronowych osiągnęła wyższą efektywność i lepszą zdolność do radzenia sobie z różnorodnymi warunkami, kosztem wyższego zużycia zasobów.

Abstract

The goal of this Ph.D. thesis was to develop and propose novel methods for locating the centers of anatomical features on fundus images: fovea and optic nerve disc. The study used two strategies: an approach based on image processing techniques and shape analysis and an approach using advanced deep neural networks, transfer learning, and ensemble learning.

In the evaluation of methods for locating the centers of objects, such as neural networks or AI techniques more generally, performance measures such as accuracy and coverage are used. Accuracy is expressed as the average Euclidean distance (*pol. Średni Błąd Euklidesowy, SBE*) between the locations of the measures determined by the expert and by the method: $SBE = \sqrt{\frac{\sum_{i=1}^n (\hat{y}-y)^2}{n}}$, where \hat{y} – coordinates of the center of the object determined by the expert, y – coordinates of the center determined by the method, n – number of determined centers. Coverage is the percentage of the number of object centers determined by the method relative to the number of centers determined by the expert: $coverage = \frac{\text{number of centers determined by the method}}{\text{number of centers determined by the expert}}$.

In applying the first approach, the focus was on using generic image processing algorithms. Shape analysis methods were also applied to increase the efficiency of identifying anatomical features of the retina with characteristic structure. This approach proved effective in terms of efficiency and the resources used but showed some limitations in the context of the varying exposure conditions of the image.

In the second approach, deep neural networks were used, due to their high ability to extract complex patterns found in images. Transfer learning was also implemented to use the knowledge gathered from other datasets. The use of ensemble learning effectively increased the overall performance of

the proposed method. In comparison with the first approach, deep neural networks better cope with varying conditions and shapes but require larger computational and time resources.

A series of tests were performed to compare the efficiency and resource consumption of the developed methods. The evaluations included precision and location coverage, computational efficiency, memory consumption, and execution time of the methods in sequential and parallelized modes. The results showed that the method based on deep neural networks achieved higher efficiency and better ability to deal with various conditions at the cost of higher resource consumption.

Spis treści

1	Wstęp	28
1.1	Przedstawienie problemu	28
1.2	Cel i teza pracy	30
1.3	Główne wyniki rozprawy	31
1.4	Zastosowane narzędzia i dane	32
1.5	Zawartość pracy	33
2	Wprowadzenie	36
2.1	Przedstawienie problematyki poruszanej w rozdziale	36
2.2	Rola Sztucznej inteligencji	37
2.3	Diagnostyka dna oka	40
2.4	Rola lokalizacji cech anatomicznych dna oka	42
2.5	Podsumowanie	43
3	Metody rozpoznawania obrazów	46
3.1	Przedstawienie problematyki poruszanej w rozdziale	46

<i>SPIS TREŚCI</i>	8
3.2	Złożoność obliczeniowa algorytmu 47
3.2.1	Idea złożoności czasowej algorytmu i notacja big-O . . . 47
3.2.2	Złożoność pamięciowa 49
3.3	Obrazy cyfrowe i modele barw 50
3.3.1	Obraz cyfrowy 50
3.3.2	RGB 51
3.3.3	CIELab 53
3.3.4	YUV 54
3.4	Metody przetwarzania obrazów 56
3.4.1	Operacja splotu i zastosowania filtrów 56
3.4.2	Metody binaryzacji 60
3.4.3	Wyrównywanie histogramów 63
3.4.4	Wyznaczanie wypukłych powłok konturów 68
3.4.5	Analiza komponentów połączonych 70
3.5	Algorytmy topologiczne i algorytmy inspirowane naturą 71
3.5.1	Operacje morfologiczne 72
3.5.2	Sztuczne sieci neuronowe 79
3.6	Nowoczesne metody uczenia maszynowego i rozpoznawania obrazu 89
3.6.1	Uczenie transferowe 89
3.6.2	Uczenie zespołowe 90

<i>SPIS TREŚCI</i>	9
3.6.3 Detekcja i lokalizacja obiektów	94
3.7 Metody przyspieszania algorytmów	95
3.7.1 Rozwiązania wykorzystujące dedykowane algorytmy i struktury danych	95
3.7.2 Rozwiązania sprzętowe	96
3.8 Podsumowanie	99
4 Bazy danych wykorzystane w badaniach	100
4.1 Przedstawienie problematyki poruszanej w rozdziale	100
4.2 Baza Fundacji Okulistyka 21	102
4.3 Indian Diabetic Retinopathy Image Dataset	103
4.4 Podsumowanie	106
5 Zastosowane metody	108
5.1 Przedstawienie problematyki poruszanej w rozdziale	108
5.2 Istniejące rozwiązania	109
5.3 Metoda Szybkiej i Równoległej Lokalizacji Współrzędnych Środków Dołeczka i Tarczy Nerwu Wzrokowego na Zdjęciach Dna Oka przy użyciu Segmentacji oraz Analizy Komponentów Połączonych	111
5.3.1 Motywacja	111
5.3.2 Zasada działania	112
5.4 Metoda Symultanicznej Lokalizacji Współrzędnych Środków Dołeczka i Tarczy Nerwu Wzrokowego na Zdjęciach Dna Oka przy użyciu Zespołu Sieci Neuronowych	138

<i>SPIS TREŚCI</i>	10
5.4.1 Motywacja	138
5.4.2 Zasada działania	139
5.5 Podsumowanie	152
6 Wyniki własne i komparatyka	158
6.1 Przedstawienie problematyki poruszanej w rozdziale	158
6.2 Metodologia uzyskania wyników oraz ich porównania	159
6.3 Wyniki własne	161
6.4 Porównanie z innymi rozwiązaniami	164
6.5 Dyskusja	166
6.6 Uwagi o zrównoleglaniu metod na procesorach graficznych . . .	168
6.7 Podsumowanie	170
7 Podsumowanie	172
A Wyniki testów metod	174
A.1 Metoda MSiRL	174
A.2 Metoda MSLW	180
B Implementacje metod i narzędzi	188
B.1 Metoda MSiRL	188
B.2 Metoda MSLW	189
B.3 Narzędzie ORSG	189

<i>SPIS TREŚCI</i>	11
B.4 Narzędzie FPL	191
C Formuły z komentarzem	192
C.1 Formuły konwersji obrazów między modelami barw	192
C.1.1 CIELab	192
C.1.2 YUV	194
C.2 Formuły zastosowane w wynikach	195
C.2.1 Odległość Minkowskiego	195
D Otwarte problemy	198
D.1 Modelowanie decyzyjne i lokalizacja obiektów	198
D.2 Analiza i wydajność metod	199

Spis rysunków

1.1	Uproszczony schemat anatomiczny budowy dna oka. Wersja elektroniczna obrazu: https://krzywicki.pro/fig-1-1	29
2.1	Schemat anatomiczny budowy dna oka. Wersja elektroniczna obrazu: https://krzywicki.pro/fig-2-1	41
2.2	Przykład dwóch standardów centracji zdjęcia dna oka. Wersja elektroniczna obrazu: https://krzywicki.pro/fig-2-2	43
2.3	Przykład zdjęć dna oka lewego i prawego wycelowanych na dołeczek oraz na tarczę nerwu wzrokowego. Wersja elektroniczna obrazu: https://krzywicki.pro/fig-2-3	44
2.4	Zdjęcia dna oka o niskiej jakości. Wersja elektroniczna obrazu: https://krzywicki.pro/fig-2-4	45
3.1	Sześcian modelu barw RGB. Źródło: https://commons.wikimedia.org/wiki/File:RGB_color_solid_cube.png	52
3.2	Przykładowe zdjęcie dna oka. Wersja elektroniczna obrazu: https://krzywicki.pro/fig-3-2	52
3.3	Warstwy zdjęcia dna oka w modelu barw RGB. Wersja elektroniczna obrazu: https://krzywicki.pro/fig-3-3	53
3.4	Schemat modelu barw CIE Lab. Źródło: [LDF ⁺ 20]	53

- 3.5 Warstwy zdjęcia dna oka w modelu barw CIELab. Wersja elektroniczna obrazu: <https://krzywicki.pro/fig-3-5>. 54
- 3.6 Schemat modelu barw CIELab. Źródło: [IHKM12] 55
- 3.7 Warstwy zdjęcia dna oka w modelu barw YUV. Wersja elektroniczna obrazu: <https://krzywicki.pro/fig-3-7>. 55
- 3.8 Przykład zastosowania operacji filtrowania zdjęcia dna oka z wykorzystaniem filtru gaussowskiego o rozmiarach 3 i 5. Charakter zmian na zdjęciu ma kluczowe znaczenie dla dalszej analizy algorytmicznej, chociaż nie zawsze jest łatwo dostrzegalny go gołym okiem ludzkim. Wersja elektroniczna obrazu: <https://krzywicki.pro/fig-3-8>. 59
- 3.9 Przykład działania operatora Sobela na zdjęciu latarni morskiej w miejscowości Elaties. Wersja elektroniczna obrazu: <https://krzywicki.pro/fig-3-9>. 61
- 3.10 Przykład działania metody progowania prostego na zdjęciu „Lena” [Mun96]. Wersja elektroniczna obrazu: <https://krzywicki.pro/fig-3-10>. 62
- 3.11 Przykład działania binaryzacji metodą Otsu na zdjęciu „Lena” [Mun96]. Wersja elektroniczna obrazu: <https://krzywicki.pro/fig-3-11>. 64
- 3.12 Przykład działania metody wyrównywania histogramów. Wersja elektroniczna obrazu: <https://krzywicki.pro/fig-3-12>. 64
- 3.13 Przykład jednowarstwowego zdjęcia przetworzonego za pomocą globalnej metody wyrównywania histogramów. Wersja elektroniczna obrazu: <https://krzywicki.pro/fig-3-13>. 66
- 3.14 Ilustracja operacji przycinania wartości histogramu w metodzie CLAHE oraz ich ponownej redystrybucji. Źródło: <https://commons.wikimedia.org/wiki/File:Claheredist.svg>. 67

- 3.15 Przykład jednowarstwowego zdjęcia przetworzonego za pomocą metody CLAHE. Wersja elektroniczna obrazu: <https://krzywicki.pro/fig-3-15>. 68
- 3.16 Przykład wyznaczonej powłoki wypukłej konturu fragmentu dłoni, której kształt jest wielokątem. Kolorem zielonym oznaczono oryginalny kontur dłoni, czerwonym – uproszony wielokąt powstały w wyniku wyznaczenia wypukłej powłoki oryginalnego konturu. Wersja elektroniczna obrazu: <https://krzywicki.pro/fig-3-16>. 69
- 3.17 Przykład działania metody analizy komponentów połączonych. Wersja elektroniczna obrazu: <https://krzywicki.pro/fig-3-17>. 71
- 3.18 Przykład zastosowania operacji erozji morfologicznej na obrazie monochromatycznym z wykorzystaniem kwadratowego elementu strukturalnego. Grubość wraz z rozmiarem obiektu pierwszoplanowego uległy zmniejszeniu. Wersja elektroniczna obrazu: <https://krzywicki.pro/fig-3-18>. 73
- 3.19 Przykład zastosowania operacji dylacji morfologicznej na obrazie monochromatycznym z wykorzystaniem kwadratowego elementu strukturalnego. Grubość wraz z rozmiarem obiektu pierwszoplanowego uległy zwiększeniu. Wersja elektroniczna obrazu: <https://krzywicki.pro/fig-3-19>. 73
- 3.20 Przykład zastosowania operacji morfologicznej White top-hat na obrazie monochromatycznym z wykorzystaniem kwadratowego elementu strukturalnego. Na obrazie wynikowym pozostały jedynie te fragmenty obiektu pierwszoplanowego, które były mniejsze powierzchniowo od elementu strukturalnego. Wersja elektroniczna obrazu: <https://krzywicki.pro/fig-3-20>. 75

- 3.21 Przykład zastosowania operacji morfologicznej Black top-hat na obrazie monochromatycznym z wykorzystaniem kwadratowego elementu strukturalnego. Na obrazie wynikowym pozostały jedynie te fragmenty obiektu pierwszoplanowego, które były mniejsze powierzchniowo od elementu strukturalnego oraz ciemniejsze od otoczenia. Wersja elektroniczna obrazu: <https://krzywicki.pro/fig-3-21>. . . 76
- 3.22 Schemat sztucznego neuronu McCullocha-Pittsa. Wersja elektroniczna obrazu: <https://krzywicki.pro/fig-3-22>. . . 80
- 3.23 Schemat perceptronu. Wersja elektroniczna obrazu: <https://krzywicki.pro/fig-3-23>. 81
- 3.24 Schemat perceptronu wielowarstwowego. Wersja elektroniczna obrazu: <https://krzywicki.pro/fig-3-24>. 82
- 3.25 Schemat metody Boosting. Wersja elektroniczna obrazu: <https://krzywicki.pro/fig-3-25>. 91
- 3.26 Schemat metody Bagging. Wersja elektroniczna obrazu: <https://krzywicki.pro/fig-3-26>. 93
- 3.27 Schemat architektury GPU. ALU (*ang. Arithmetic Logic Unit*) – jednostka arytmetyczno-logiczna. Wersja elektroniczna obrazu: <https://krzywicki.pro/fig-3-27>. . . 98
- 4.1 Przykładowa zawartość bazy zdjęć dna oka Fundacji Okulistyka 21. Wersja elektroniczna obrazu: <https://krzywicki.pro/fig-4-1>. 104
- 4.2 Maska segmentacji tarczy nerwu wzrokowego bazy *IDRID*. Wersja elektroniczna obrazu: <https://krzywicki.pro/fig-4-2>. 105
- 4.3 Przykładowa zawartość bazy *IDRID*. Wersja elektroniczna obrazu: <https://krzywicki.pro/fig-4-3>. 106

- 5.1 Przykład konwersji do skali szarości prawidłowo oświetlonego zdjęcia dna oka o wysokiej jakości. Bez przeprowadzenia operacji korekty kontrastu obszar plamki żółtej jest mało wyraźny. Wersja elektroniczna obrazu: <https://krzywicki.pro/fig-5-1>. 113
- 5.2 Schemat działania proponowanej metody MSiRL. Zdjęcie jest przetwarzane niezależnie i równoległe pod kątem lokalizacji dołączka i tarczy nerwu wzrokowego. Wersja elektroniczna obrazu: <https://krzywicki.pro/fig-5-2>. 114
- 5.3 Plamka żółta oraz dołączek na zdjęciu dna oka. Oba obiekty przyjmują kształt kolisty. Wersja elektroniczna obrazu: <https://krzywicki.pro/fig-5-3>. 115
- 5.4 Plamka żółta o pogorszonej wyraźności i zmienionym kształcie. Strzałką wskazano orientacyjny obszar, który nie jest wyraźnie widoczny. Obszar dołączka jest niewidoczny. Wersja elektroniczna obrazu: <https://krzywicki.pro/fig-5-4>. . . . 116
- 5.5 Zdjęcie dna oka o niskiej jakości, gdzie obszar plamki żółtej jest niewidoczny poprzez niedoświetlenie. Wersja elektroniczna obrazu: <https://krzywicki.pro/fig-5-5>. 117
- 5.6 Zdjęcie dna oka z objawami DME zaburzającymi kompozycję kształtu plamki żółtej. Wersja elektroniczna obrazu: <https://krzywicki.pro/fig-5-6>. 118
- 5.7 Szczegółowy schemat działania metody segmentującej region plamki żółtej na zdjęciu dna oka. Wersja elektroniczna obrazu: <https://krzywicki.pro/fig-5-7>. 119
- 5.8 Przykład operacji korekty kontrastu na zdjęciu dna oka. Obszar plamki żółtej jest znacznie wyraźniejszy, niż na zdjęciu oryginalnym. Wersja elektroniczna obrazu: <https://krzywicki.pro/fig-5-8>. 120
- 5.9 Przykład wpływu połączonych transformacji morfologicznych na wyraźność plamki żółtej na przetwarzanym obrazie. Wersja elektroniczna obrazu: <https://krzywicki.pro/fig-5-9>. . . . 122

- 5.10 Porównanie histogramów obrazów przed i po wyrównaniu metodą CLAHE. Liczebność pikseli ciemnych jest nadal duża, co wynika z tła obszaru dna oka. Wersja elektroniczna obrazu: <https://krzywicki.pro/fig-5-10>. 123
- 5.11 Efekt korekty oświetlenia i kontrastu po dokonaniu łączonych transformacji morfologicznych. Wersja elektroniczna obrazu: <https://krzywicki.pro/fig-5-11>. 124
- 5.12 Obraz po przetworzeniu za pomocą kroku przetwarzania końcowego. Region plamki żółtej jest wyraźnie zauważalny. Wersja elektroniczna obrazu: <https://krzywicki.pro/fig-5-12>. 124
- 5.13 Oznaczone regiony będące kandydatami na plamkę żółtą. Każdy region ma przypisany inny kolor zgodny z wyznaczonym identyfikatorem. Wersja elektroniczna obrazu: <https://krzywicki.pro/fig-5-13>. 125
- 5.14 Wyznaczony środek regionu będącego plamką żółtą. Obraz w oryginalnym rozmiarze, będący końcowym efektem metody segmentującej plamkę żółtą. Wewnątrz okręgu znajduje się wyznaczona lokalizacja dołeczka. Wersja elektroniczna obrazu: <https://krzywicki.pro/fig-5-14>. 125
- 5.15 Wyznaczony przez algorytm region plamki żółtej na zdjęciu dna oka opisany okręgiem o promieniu najdłuższej przekątnej. Wewnątrz znajduje się wskazana przez algorytm lokalizacja dołeczka. Wersja elektroniczna obrazu: <https://krzywicki.pro/fig-5-15>. 126
- 5.16 Tarcza nerwu wzrokowego na zdjęciu dna oka. Wersja elektroniczna obrazu: <https://krzywicki.pro/fig-5-16>. . . 127
- 5.17 Wpływ nieprawidłowego oświetlenia dna oka na widoczność obszaru tarczy nerwu wzrokowego. Wersja elektroniczna obrazu: <https://krzywicki.pro/fig-5-17>. 127
- 5.18 Szczegółowy schemat działania metody segmentującej region tarczy nerwu wzrokowego na zdjęciu dna oka. Wersja elektroniczna obrazu: <https://krzywicki.pro/fig-5-18>. . . 128

- 5.19 Tarcza nerwu wzrokowego reprezentowana za pomocą wybranych warstw modeli barw. Wersja elektroniczna obrazu: <https://krzywicki.pro/fig-5-19>. 129
- 5.20 Pośrednia forma zdjęcia dna oka uwypuklająca tarczę nerwu wzrokowego. Wersja elektroniczna obrazu: <https://krzywicki.pro/fig-5-20>. 130
- 5.21 Pośrednia forma zdjęcia dna oka uwypuklająca tarczę nerwu wzrokowego po dokonaniu spłaszczenia regionów przypominających teksturę tarczę nerwu wzrokowego. Wersja elektroniczna obrazu: <https://krzywicki.pro/fig-5-21>. . . 131
- 5.22 Odszumiona pośrednia forma zdjęcia dna oka uwypuklająca tarczę nerwu wzrokowego po dokonaniu spłaszczenia regionów przypominających teksturę tarczę nerwu wzrokowego. Wersja elektroniczna obrazu: <https://krzywicki.pro/fig-5-22>. . . 132
- 5.23 Region kandydacki na tarczę nerwu wzrokowego po operacji binaryzacji metodą Otsu. Wersja elektroniczna obrazu: <https://krzywicki.pro/fig-5-23>. 133
- 5.24 Obraz będący efektem etapu przetwarzania końcowego algorytmu segmentującego region tarczy nerwu wzrokowego. Wersja elektroniczna obrazu: <https://krzywicki.pro/fig-5-24>. 133
- 5.25 Oznaczony region będący kandydatem na tarczę nerwu wzrokowego. Wersja elektroniczna obrazu: <https://krzywicki.pro/fig-5-25>. 134
- 5.26 Wyznaczony środek regionu będącego tarczą nerwu wzrokowego. Obraz w oryginalnym rozmiarze, będący końcowym efektem metody segmentującej tarczę nerwu wzrokowego. Wewnątrz okręgu znajduje się wyznaczona lokalizacja środka tarczy nerwu wzrokowego. Wersja elektroniczna obrazu: <https://krzywicki.pro/fig-5-26>. . . 135

- 5.27 Wyznaczony przez algorytm region tarczy nerwu wzrokowego na zdjęciu dna oka opisany okręgiem o promieniu najdłuższej przekątnej. Wewnątrz znajduje się wskazana przez algorytm lokalizacja środka tarczy nerwu wzrokowego. Wersja elektroniczna obrazu: <https://krzywicki.pro/fig-5-27>. . . 136
- 5.28 Schemat działania proponowanej metody MSLW. Zdjęcie jest przetwarzane i analizowane pod kątem lokalizacji dołączka i tarczy nerwu wzrokowego za pomocą wspólnych operacji. Wersja elektroniczna obrazu: <https://krzywicki.pro/fig-5-28>. 141
- 5.29 Metoda podziału danych treningowych na potrzeby treningu metody MSLW. Podzbiór treningowy jest wielokrotnie dzielony na mniejsze podzbiory celem wytrenowania niezależnych sieci neuronowych. P – podzbiór, W – wariant przetwarzania, P_nW_m – *n*-ty podzbiór przetworzony *m*-tym wariantem. Wersja elektroniczna obrazu: <https://krzywicki.pro/fig-5-29>. 142
- 5.30 Proces przetwarzania wstępnego zdjęć dna oka w dwóch wariantach. Wersja elektroniczna obrazu: <https://krzywicki.pro/fig-5-30>. 145
- 5.31 Obszary plamki żółtej i tarczy nerwu wzrokowego oznaczone za pomocą ramek wyznaczonych przez 16 sieci neuronowych o architekturze YOLOv8. Kolorem niebieskim oznaczono plamkę żółtą, zielonym – tarczę nerwu wzrokowego. Wszystkie obszary wyznaczone przez sieci są do siebie zbliżone. Wersja elektroniczna obrazu: <https://krzywicki.pro/fig-5-31>. . . 149
- 5.32 Części wspólne (zaznaczone białą ramką) obszarów cech anatomicznych dna oka wyznaczonych przez modele bazowe. Wersja elektroniczna obrazu: <https://krzywicki.pro/fig-5-32>. 150
- 5.33 Zlokalizowane metodą MSLW i zaznaczone punktami środki dołączka (niebieski) oraz tarczy nerwu wzrokowego (zielony). Wersja elektroniczna obrazu: <https://krzywicki.pro/fig-5-33>. 152

- B.1 Schemat ORSG - narzędzia generującego próby zawartości baz zdjęć dna oka oraz atrybuty techniczne. ORSG jako wejście przyjmuje adres URL pliku z bazą lub jego identyfikator na platformie Kaggle [KAG23] oraz rozmiar próby. Wersja elektroniczna obrazu: <https://krzywicki.pro/fig-b-1>. . . . 190

Spis tabel

3.1	Czas wykonania rekonstrukcji geodezyjnej obrazu Lena [Mun96] na podstawie pracy [GTD13].	77
3.2	Czas operacji erozji i dylacji na obrazie binarnym dla trzech wariantów elementu strukturalnego.	78
3.3	Czas operacji erozji i dylacji na obrazie Lena dla trzech wariantów elementu strukturalnego.	78
3.4	Wyniki złożoności czasowej czterech wariantów operacji morfologicznych na obrazach o rozmiarze 512x512 px w trzech wariantach przyspieszenia.	79
3.5	Wyniki złożoności czasowej czterech wariantów operacji morfologicznych na obrazach o rozmiarze 1024x1024 px w trzech wariantach przyspieszenia.	79
3.6	Wyniki złożoności czasowej czterech wariantów operacji morfologicznych na obrazach o rozmiarze 2048x2048 px w trzech wariantach przyspieszenia.	79
4.1	Atrybuty techniczne baz zdjęć dna oka wykorzystanych w badaniach.	101
4.2	Atrybuty epidemiologiczne baz zdjęć dna oka wykorzystanych w badaniach.	101

4.3	Atrybuty anatomiczne baz zdjęć dna oka wykorzystanych w badaniach.	101
4.4	Atrybuty jakościowe baz zdjęć dna oka wykorzystanych w badaniach.	102
5.1	Parametry metody MSiRL. <i>FS</i> – proces segmentacji plamki żółtej, <i>ODS</i> – proces segmentacji tarczy nerwu wzrokowego. . .	137
5.2	Szczegóły podzbiorów bazy <i>BFO21</i> wykorzystywanych na etapie przygotowywania metody <i>MSLW</i>	143
5.3	Szczegóły wytrenowanych modeli bazowych w postaci sieci neuronowych o architekturze YOLOv8. Sieci z przyrostkiem “T” w symbolu były trenowane transferowo.	155
5.4	Parametry treningu modeli bazowych w postaci sieci neuronowych o architekturze YOLOv8.	156
5.5	Parametry metody <i>MSLW</i>	157
6.1	Zagregowane wyniki uzyskane przez metody zaproponowane w Rozdziale 5. <i>F</i> – proces lokalizacji środka dołączka, <i>OD</i> – proces lokalizacji środka tarczy nerwu wzrokowego.	162
6.2	Czas wyznaczania cech anatomicznych na zdjęciu dna oka przez metody zaproponowane w Rozdziale 5.	163
6.3	Średnie zużycie pamięci podczas wyznaczania cech anatomicznych na zdjęciu dna oka za pomocą metod zaproponowanych w Rozdziale 5.	163
6.4	Odległość Minkowskiego dla $p = 2$ między lokalizacjami cech anatomicznych, wyznaczonymi przez modele bazowe metody <i>MSLW</i> oraz ekspertów, dla wszystkich zdjęć z bazy IDRID. <i>D</i> – lokalizacja dołączka, <i>TNW</i> – lokalizacja tarczy nerwu wzrokowego.	164

6.5	Odległość Minkowskiego dla $p = 2$ między lokalizacjami cech anatomicznych, wyznaczonymi przez modele bazowe metody MSLW oraz ekspertów, dla zdjęć na których metoda MSLW zlokalizowała obydwie cechy anatomiczne. D – lokalizacja dołączka, TNW – lokalizacja tarczy nerwu wzrokowego.	165
6.6	Porównanie wyników uzyskanych przez rozwiązania lokalizujące środki dołączka oraz tarczy nerwu wzrokowego na zdjęciu dna oka.	166
6.7	Zrównoleżone na GPU kroki algorytmu MSiLR. FS – segmentacja regionu plamki żółtej, ODS – segmentacja regionu tarczy nerwu wzrokowego.	169
A.1	Pełne wyniki uzyskane przez metodę MSiRL na bazie IDRID. W kolumnach zawierających różnice między środkami znajdują się odległości wyrażone odległością Minkowskiego dla podanych wartości parametru p w kolejności: różnica odległości środka dołączka, różnica odległości środka tarczy nerwu wzrokowego. Wersja elektroniczna tabeli: https://krzywicki.pro/tab-a-1	174
A.1	Pełne wyniki uzyskane przez metodę MSiRL na bazie IDRID (kontynuacja).	175
A.1	Pełne wyniki uzyskane przez metodę MSiRL na bazie IDRID (kontynuacja).	176
A.2	Pełne wyniki wydajnościowe uzyskane przez metodę MSiRL na bazie IDRID. Wersja elektroniczna tabeli: https://krzywicki.pro/tab-a-2	176
A.2	Pełne wyniki wydajnościowe uzyskane przez metodę MSiRL na bazie IDRID (kontynuacja).	177
A.2	Pełne wyniki wydajnościowe uzyskane przez metodę MSiRL na bazie IDRID (kontynuacja).	178
A.2	Pełne wyniki wydajnościowe uzyskane przez metodę MSiRL na bazie IDRID (kontynuacja).	179

- A.3 Pełne wyniki uzyskane przez metodę MSLW na bazie IDRID. W kolumnach zawierających różnice między środkami znajdują się odległości wyrażone odległością Minkowskiego dla podanych wartości parametru p w kolejności: różnica odległości środka dołączka, różnica odległości środka tarczy nerwu wzrokowego. Wersja elektroniczna tabeli: <https://krzywicki.pro/tab-a-3>. 180
- A.3 Pełne wyniki uzyskane przez metodę MSLW na bazie IDRID (kontynuacja). 181
- A.3 Pełne wyniki uzyskane przez metodę MSLW na bazie IDRID (kontynuacja). 182
- A.4 Pełne wyniki wydajnościowe uzyskane przez metodę MSLW na bazie IDRID. Wersja elektroniczna tabeli: <https://krzywicki.pro/tab-a-4>. 182
- A.4 Pełne wyniki wydajnościowe uzyskane przez metodę MSLW na bazie IDRID (kontynuacja). 183
- A.4 Pełne wyniki wydajnościowe uzyskane przez metodę MSLW na bazie IDRID (kontynuacja). 184
- A.4 Pełne wyniki wydajnościowe uzyskane przez metodę MSLW na bazie IDRID (kontynuacja). 185
- A.5 Błąd między współrzędnymi środków dołączka oraz tarczy nerwu wzrokowego wyznaczonymi przez modele bazowe metody MSLW dla wszystkie zdjęć bazy IDRID za pomocą trzech wariantów odległości Minkowskiego. D – lokalizacja dołączka, TNW – lokalizacja tarczy nerwu wzrokowego, Max – wartość największa, Min – wartość najmniejsza. Wersja elektroniczna tabeli: <https://krzywicki.pro/tab-a-5>. . . . 186

- A.6 Błąd między współrzędnymi środków dołączka oraz tarczy nerwu wzrokowego wyznaczonymi przez modele bazowe metody MSLW, dla zdjęć IDRID na których metoda MSLW wykryła obie cechy, za pomocą trzech wariantów odległości Minkowskiego. D – lokalizacja dołączka, TNW – lokalizacja tarczy nerwu wzrokowego, Max – wartość największa, Min – wartość najmniejsza. Wersja elektroniczna tabeli: <https://krzywicki.pro/tab-a-6>. 187

Przedmowa

Szanowni Państwo,

Dla Państwa wygody i łatwości dostępu, pełna wersja rozprawy w formie elektronicznej dostępna jest pod adresem <https://krzywicki.pro/rozprawa>. Wprowadzenie elektronicznego formatu rozprawy ma na celu umożliwienie Państwu wygodnego odbioru materiału. Rysunki oraz obszerne tabele z wynikami zawierają w podpisach odnośniki do wersji elektronicznych, co pozwala na szczegółowe zapoznanie się z prezentowanymi pojęciami, danymi i wynikami.

Z uszanowaniem,
Tomasz Krzywicki.

Dla lepszej orientacji Czytelnika w stosowanej terminologii dodajemy nazwy anglojęzyczne omawianych pojęć i obiektów z uwagi na przewagę literatury specjalistycznej w języku angielskim.

Rozdział 1

Wstęp

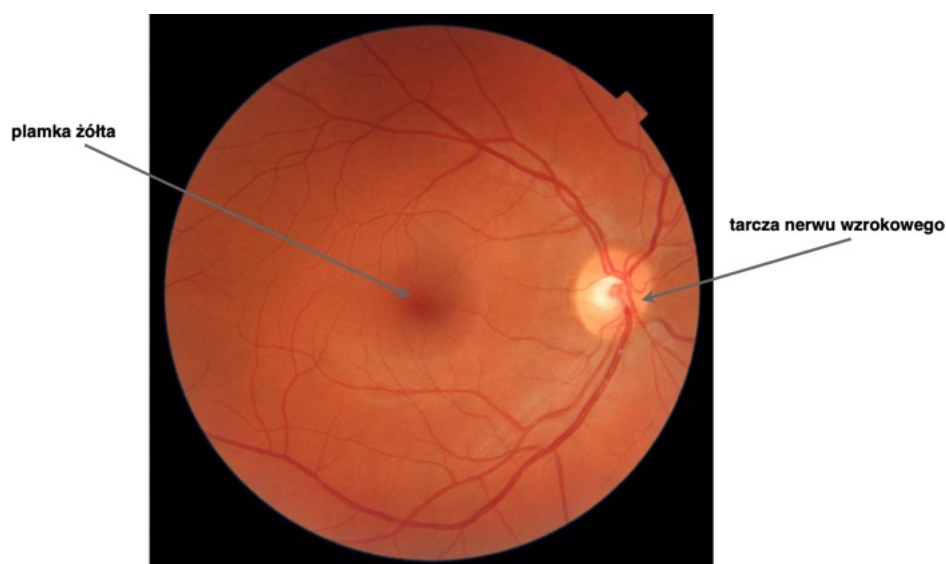
1.1 Przedstawienie problemu

W dzisiejszych czasach zastosowanie sztucznej inteligencji (*ang. Artificial Intelligence, AI*) w medycynie, a szczególnie w okulistyce, otwiera nowe możliwości i perspektywy.

Jednym z kluczowych obszarów okulistyki, gdzie AI zdobyła szczególne uznanie, jest diagnostyka retinopatii cukrzycowej (*ang. Diabetic Retinopathy, DR*) [FAG⁺04, GSNR23]. DR stanowi poważne zagrożenie dla pacjentów z cukrzycą i może prowadzić do trwałej utraty wzroku. Pacjenci z DR mogą doświadczać plam oraz zniekształceń w polu widzenia. Dzięki wykorzystaniu AI do analizy zdjęć dna oka, możliwe jest wykrywanie charakterystycznych zmian, takich jak mikroaneuryzmaty [NPR⁺09, BZW⁺23] czy zmiany naczyniowe [TSB⁺19, AME⁺22], co pozwala na szybkie wdrożenie odpowiednich procedur klinicznych. AI znalazła także zastosowanie we wsparciu diagnostyki innych okulistycznych jednostek chorobowych i stanów patologicznych, takich jak cukrzycowy obrzęk plamki żółtej (*ang. Diabetic Macular Edema, DME*) [GYA⁺69, SSA⁺23], zwyrodnienie plamki żółtej powiązane z wiekiem (*ang. Age-Related Macular Degeneration, AMD*) [Apt21, WLCL22], jaskra (*ang. glaucoma*) [DCP⁺16, CWA⁺23] czy obrzęk tarczy nerwu wzrokowego (*ang. Optic Disc Edema, ODE*) [Sta07, NAB⁺23].

AI znalazła także zastosowanie w diagnostyce chorób systemowych na podstawie zdjęć dna oka i stanowi obecnie prężnie rozwijający się kierunek w nauce. Wykorzystanie sieci neuronowych w połączeniu z metodami przetwarzania obrazów pozwala na wykrywanie charakterystycznych zmian w siatkówce, które mogą być związane z różnymi chorobami systemowymi, takimi jak nadciśnienie tętnicze czy choroby serca [IZW⁺23]. Interesującym faktem jest również zastosowanie AI w rozpoznawaniu płci, wieku oraz uzależnienia od tytoniu na podstawie zdjęć dna oka [KNB⁺20, IZW⁺23].

Jednymi ze wskaźników umożliwiającymi diagnostykę okulistycznych jednostek chorobowych są cechy anatomiczne dna oka takie jak plamka żółta i tarcza nerwu wzrokowego (Rysunek 1.1).



Rysunek 1.1: Uproszczony schemat anatomiczny budowy dna oka. Wersja elektroniczna obrazu: <https://krzywicki.pro/fig-1-1>.

Lokalizacja cech anatomicznych na zdjęciu dna oka, takich jak plamka żółta i tarcza nerwu wzrokowego, umożliwia dokonanie analizy ich najbliższego otoczenia, której celem jest wykrywanie zmian w postaci obrzęków, krwotoków lub uszkodzeń. Zmiany patologiczne występujące w otoczeniu cech anatomicznych mogą być związane z jednostkami chorobowymi siatkówki, w tym AMD, DME, jaskrą a także ze stanem patologicznym, takim jak ODE, co omówiono w Podrozdziale 2.3.

Lokalizacja cech anatomicznych na zdjęciu dna oka, takich jak plamka żółta i tarcza nerwu wzrokowego, może mieć także istotne znaczenie w ocenie

atrybutów anatomicznych i jakościowych zdjęć dna oka, takich jak standard wyśrodkowania, strona oka czy jakość wykonanego zdjęcia, co omówiono w Podrozdziale 2.4.

1.2 Cel i teza pracy

Głównym celem badań zaprezentowanych w rozprawie było opracowanie i zaproponowanie nowych metod lokalizujących dołeczek (punkt centralny plamki żółtej) oraz tarczę nerwu wzrokowego na zdjęciu dna oka. Do opracowania proponowanych metod wykorzystano dwa odmienne podejścia:

- wykorzystujące techniki przetwarzania obrazów oraz
- wykorzystujące sieci neuronowe.

Przy opracowywaniu proponowanych metod uwzględniono aspekty takie jak dokładność i pokrycie lokalizacji cech anatomicznych będących przedmiotem zainteresowania, efektywność obliczeniową w postaci zużytego czasu i pamięci i możliwości przyspieszania wykonania za pomocą rozwiązań sprzętowych. Realizacja badań miała na celu również dostarczenie wkładu w rozwój diagnostyki okulistycznej wspieranej za pomocą AI oraz poszerzenie bieżącego stanu wiedzy naukowej.

Na podstawie głównego celu pracy postawiono następujące cztery tezy.

1. Dokładność i pokrycie wyznaczonych lokalizacji cech anatomicznych na zdjęciach dna oka przy użyciu sieci neuronowych będzie wyższa, niż w przypadku zastosowania technik przetwarzania obrazów.
2. Czas wykonania obliczeń i zużycie pamięci operacyjnej będzie niższe w przypadku lokalizacji cech anatomicznych na zdjęciach dna oka przy użyciu technik przetwarzania obrazów, niż w przypadku zastosowania sieci neuronowych.
3. Zastosowanie przyspieszania sprzętowego zaproponowanych metod za pomocą zrównoleglania obliczeń na procesorach graficznych skróci czas obliczeń potrzebny na wyznaczenie lokalizacji cech anatomicznych na zdjęciach dna oka.

4. Dokładność i pokrycie lokalizacji cech anatomicznych na zdjęciach dna oka przy użyciu zespołu sieci neuronowych będą wyższe, niż w przypadku pojedynczej sieci neuronowej.

1.3 Główne wyniki rozprawy

Zgodnie z celem badawczym i tezami przedstawionymi w Podrozdziale 1.2 pomyślnie opracowano i zaimplementowano dwie metody lokalizujące dołeczek oraz tarczę nerwu wzrokowego na zdjęciach dna oka. Każdą z zaproponowanych metod poddano testom pod kątem:

- dokładności (*ang. accuracy*) i pokrycia (*ang. coverage*) wyznaczonych lokalizacji za pomocą pojedynczych sieci neuronowych oraz za pomocą zespołu sieci neuronowych,
- czasu wyznaczania lokalizacji oraz zużycia pamięci operacyjnej w wariantach implementacji wykonywanych sekwencyjnie na procesorze centralnym oraz częściowo równoległe na procesorze graficznym.

Analizując uzyskane wyniki testów można zauważyć, że dokładność i pokrycie wyznaczonych lokalizacji cech anatomicznych na zdjęciach dna oka są znacznie wyższe w przypadku wykorzystania zespołu sieci neuronowych, niż w przypadku wykorzystania technik przetwarzania obrazów. Warto podkreślić, że dokładność uzyskana za pomocą metody wykorzystującej zespół sieci neuronowych, przewyższa wyniki znane z literatury naukowej.

Zauważono, że metoda lokalizująca cechy anatomiczne na zdjęciu dna oka z wykorzystaniem technik przetwarzania obrazów zużywa mniej pamięci operacyjnej na potrzeby obliczeń oraz wykonuje się w krótszym czasie, niż metoda wykorzystująca zespół sieci neuronowych. Zastosowanie podejścia przyspieszania sprzętowego z wykorzystaniem zrównoleglania obliczeń na procesorze graficznym przyniosło efekt w postaci skrócenia czasu obliczeń w obydwu metodach. Zauważono jednak, że wykorzystanie procesora graficznego wpłynęło na nieznaczne zwiększenie zużycia pamięci operacyjnej.

Podczas analizy wyników testów dostrzeżono także, że zastosowanie zespołu sieci neuronowych przyniosło poprawę zarówno dokładności, jak

i pokrycia wyznaczonych lokalizacji cech anatomicznych na zdjęciach dna oka w porównaniu do zastosowania pojedynczej sieci neuronowej.

1.4 Zastosowane narzędzia i dane

Zaproponowane metody lokalizujące cechy anatomiczne na zdjęciach dna oka przygotowano przy użyciu podejść omówionych w Rozdziale 3. Podejścia te znacznie różnią się od siebie pod względem charakteru działania, możliwości oraz złożoności obliczeniowej. Zastosowanie różniących się podejść pozwoliło określić ich słabe i mocne strony w specyficznym zastosowaniu, jakim jest analiza zdjęć dna oka.

Do implementacji proponowanych metod wykorzystano język programowania *Python* [Pyt23] w wersji 3.11 wraz z bibliotekami zewnętrznymi, takimi jak *OpenCV* [Bra00] w wersji 4.7.0, *Ultralytics* [JCQ23] w wersji 8.0.94 oraz *PyTorch* [PGM⁺19] w wersji 2.0.1.

Język *Python* jako główny język implementacji został wybrany między innymi ze względu na jego popularność w środowisku programistów zajmujących się AI. Warty podkreślenia jest fakt, że język *Python* według rankingu *TIOBE* [TIO23] w latach 2022 oraz 2023 był najpopularniejszym językiem programowania, co czyni go także najpopularniejszym wyborem przez specjalistów od przetwarzania danych. Popularność języka *Python* przyczyniła się do powstania bogatego ekosystemu bibliotek zewnętrznych, które ułatwiają stosowanie zaimplementowanych i dokładnie przetestowanych metod przetwarzania danych, wpływając tym samym na wyższą niezawodność przygotowanych implementacji. Decydującym czynnikiem wpływającym na wykorzystanie języka *Python* jest 8-letnie doświadczenie komercyjne autora rozprawy w pracy z tym językiem (stan na 2023 rok).

Biblioteka *OpenCV* jest popularnym narzędziem umożliwiającym stosowanie technik przetwarzania obrazów. Wybrane metody posiadają implementacje równoległe wykonywane na procesorach graficznych, dzięki czemu niektóre z operacji mogą być wykonywane w znacznie krótszym czasie. Biblioteka *OpenCV* jest dostępna w wielu językach programowania, takich jak *Python*, *C/C++*, *Java*, *Go* oraz *MATLAB*.

Biblioteka *Ultralytics* to kolejne popularne narzędzie, którego celem jest dostarczanie sprawdzonych i przetestowanych implementacji sieci neuronowych przygotowanych przy użyciu biblioteki *PyTorch*. Biblioteka *Ultralytics* jako pierwsza dostarczyła implementacji architektury sieci neuronowej *YOLOv8* [RKHD23], którą wykorzystano w opracowaniu jednej z zaproponowanych metod.

W testach oraz opracowaniu wyników uzyskanych przez proponowane metody wykorzystano biblioteki *pandas* [pdt20, WM10] w wersji 2.0.2 oraz *SciPy* [VGO⁺20] w wersji 1.10.1, które umożliwiają sprawną analizę danych tabelarycznych wraz z zastosowaniem szeregu funkcji geometrycznych i statystycznych.

O poprawność implementacji napisanych w języku *Python* wraz z najwyższymi standardami jakości *PEP-8* [vRWC01] wyznaczonymi przez społeczność programistów dbano stosując narzędzia z rodziny *Flake8* [FLA23] w wersji 6.0.0 oraz *mypy* [MYP23] w wersji 1.2.0.

Do opracowania, wytrenowania i testów proponowanych metod wykorzystano dwa zbiory danych zawierające zdjęcia dna oka, które omówiono szczegółowo w Rozdziale 4.

1.5 Zawartość pracy

Rozprawę podzielono na siedem rozdziałów oraz cztery dodatki (A, B, C, D).

Rozdział drugi, „Wprowadzenie”, poświęcono zarysowaniu historii i teraźniejszości AI, a następnie jej zastosowań we wsparciu diagnostyki okulistycznej. Omówiono metody obrazowania dna oka wraz z przedstawieniem schematu budowy anatomicznej i omówieniem procesu diagnostycznego. Rozdział zwięźczo zakończono zaakcentowaniem istotnej roli plamki żółtej i tarczy nerwu wzrokowego w diagnostyce klinicznej oraz w ocenie jakości i standardu wykonania zdjęcia, a także w rozpoznaniu strony oka.

Rozdział trzeci, „Metody rozpoznawania obrazów”, przynosi omówienie wybranych metod i algorytmów generycznych przetwarzania i rozpoznawania obrazów stosowanych w niniejszej rozprawie dla tworzenia proponowanych rozwiązań lokalizujących środki dołączka oraz tarczy nerwu wzrokowego,

które omówiono w Rozdziale 5. Rozdział rozpoczął od wprowadzenia pojęcia złożoności obliczeniowej algorytmów. Następnie przedstawiono ideę obrazu cyfrowego wraz z modelami barw, takimi jak *RGB*, *CIELab* oraz *YUV*. W kolejnym kroku omówiono wybrane metody przetwarzania obrazów, takie jak filtrowanie obrazów, binaryzacja, wyrównywanie histogramów, wyznaczanie wypukłych powłok konturów wielokątów oraz analiza komponentów połączonych. Następnie przedstawiono algorytmy topologiczne i algorytmy inspirowane naturą, takie jak morfologia matematyczna oraz sieci neuronowe. W następnym etapie omówiono nowoczesne metody uczenia maszynowego, uczenia głębokiego i rozpoznawania obrazu, takie jak uczenie transferowe, uczenie zespołowe i detekcja obiektów. Rozdział zakończono omówieniem metod przyspieszania algorytmów.

Rozdział czwarty, „Bazy danych wykorzystane w badaniach”, poświęcono zbiorom danych zawierającym zdjęcia dna oka, które wykorzystano w procesie tworzenia proponowanych metod lokalizujących dołeczki i tarczę nerwu wzrokowego na zdjęciach dna oka. W Rozdziale szczegółowo omówiono dwa zbiory danych wraz z przedstawieniem autorskiego narzędzia wspomagającego automatyczną inspekcję podobnych zbiorów danych dostępnych w sieci.

Rozdział piąty, „Zastosowane metody”, poświęcono zaprezentowaniu i szczegółowemu omówieniu proponowanych metod lokalizujących środki dołeczki oraz tarczy nerwu wzrokowego. Rozdział rozpoczął od omówienia bieżącego stanu wiedzy naukowej oraz wyników uzyskanych przez najbardziej aktualne rozwiązania. Dla każdej z proponowanych metod przedstawiono motywację utworzenia oraz szczegółowy opis działania wraz z diagramami przepływu danych i przykładami działania etapów metod na przykładowych zdjęciach i obrazach.

Rozdział szósty, „Wyniki własne i komparatystyka”, poświęcono przedstawieniu oraz omówieniu wyników uzyskanych przez proponowane metody lokalizujące środki dołeczki oraz tarczy nerwu wzrokowego na zdjęciach dna oka, co w połączeniu z Rozdziałem 5 stanowi główną oś niniejszej rozprawy. Wyniki proponowanych metod poddano gruntownej analizie oraz porównano je z istniejącymi rozwiązaniami.

Rozdział siódmy, „Konkluzje”, poświęcono przedstawieniu wniosków wynikających z celu badań i postawionych tez, a także podsumowaniu niniejszej rozprawy.

Dodatki zorganizowano względem treści i charakteru rozdziałów celem uzupełnienia znajdującej się w nich informacji.

Dodatek A, „Wyniki testów metod”, poświęcono przedstawieniu całościowych wyników, które z uwagi na swoją obszerność, mogłyby odciągnąć uwagę Czytelnika od głównej treści Rozdziału szóstego.

W Dodatku B „Implementacje narzędzi oraz metod własnych” omówiono szczegóły implementacji zaproponowanych metod wraz z narzędziami dodatkowymi. W dodatku zawarto także niezbędne kroki do uzyskania kodów źródłowych wraz ze szczegółami dotyczącymi instalacji narzędzi uruchomienia środowiska wykonawczego.

Dodatek C, „Formuły z komentarzem”, poświęcono przedstawieniu i omówieniu matematycznych rozwinięć technik stosowanych w opracowaniu proponowanych metod oraz w wynikach.

W dodatku D, „Otwarte problemy”, przedstawiono problemy, których nie rozwiązano w niniejszej rozprawie. W dodatku zawarto wskazówki, które mogą ułatwić podjęcie się zadania rozwiązania przedstawionych problemów.

Rozdział 2

Wprowadzenie

2.1 Przedstawienie problematyki poruszanej w rozdziale

AI to dziedzina informatyki, która od dawna stanowi przedmiot zainteresowania naukowców i inżynierów reprezentujących rozmaite dziedziny, a w ostatnich latach przyniosła przełom w medycynie, a w szczególności, w diagnostyce okulistycznej. W niniejszym rozdziale skupiono się na zastosowaniach AI we wsparciu diagnostyki okulistycznej, szczególnie w kontekście analizy i interpretacji zdjęć dna oka. Omówiono historię powstania AI i jej rozwój oraz zastosowania w okulistyce.

Rozdział rozpoczęto rozpoczynając się od zarysowania historii AI, której korzenie sięgają lat 40. XX w. W kolejnym etapie omówiono zastosowania AI we wsparciu diagnostyki okulistycznej ze szczególnym uwzględnieniem analizy zdjęć dna oka. Następnie omówiono schemat anatomiczny wraz z procesem diagnostycznym dna oka. Zaakcentowano także istotną rolę plamki żółtej i tarczy nerwu wzrokowego w procesie diagnostycznym, których precyzyjna lokalizacja i analiza stanowią fundament diagnostyki okulistycznej oraz standardu i jakości wykonania zdjęcia dna oka.

2.2 Rola Sztucznej inteligencji

AI wyrosła z intuicji wielu badaczy zaangażowanych w studia nowych obszarów informatyki, których powstanie zawdzięczano konstruktorom pierwszych maszyn liczących w XX wieku: Johnowi Atanassoffowi i Cliffordowi Berry'emu w Iowa State College [RH00], Konradowi Zuse w Berlinie [Roj97], Johnowi Mauchly'emu i Presperowi Eckertowi w Philadelfii [Eni23]. Mimo niewielkich możliwości pierwszych maszyn liczących i kolejnych konstrukcji, wielu badaczy sondowało ich potencjalne możliwości. Jednym z pierwszych badaczy był twórca teorii informacji [Sha48] Claude E. Shannon z firmy Bell Labs. Shannon pisał pierwsze programy dla gry w szachy [Sha88], w których sformułował ideę oceny pozycji oraz *a – b pruningu*. Podobne programy pisał Alan M. Turing. Turing podjął ten temat w szerszej perspektywie [Tur50a]. Pytanie: „Can machines think?” rozwinął twierdząco nie tylko dla szachów ale dla wielu innych obszarów twórczości. Test Turinga (*the imitation game*) miał być kryterium orzekającym, że maszyna wykonała zadanie wymagające inteligencji człowieka.

Istotnym krokiem ku powstaniu paradygmatu AI były prace Olivera Selfridge'a na temat dopasowywania wzorców (*ang. pattern matching*), które zainspirowały Alana Newella do pracy nad uczeniem maszyn. Wspólnie z Herbertem Simonem przy współpracy Cliffa Shawa stworzyli w latach 1955-56 program *Logic Theorist* [NS56], który wykorzystali do dowodu maszynowego 38 propozycji w Principia Mathematica Russella i Whiteheada. *Logic Theorist* był w stanie wskazać prostszy dowód jednej z propozycji.

Sytuacja dojrzała do stworzenia ram formalnych dla nowej dziedziny. John McCarthy, asystent matematyki w Darmouth College w Hanover zwrócił się do Claude'a Shannona, Marvinina Minsky'ego i Nathana Rochester, z propozycją zwołania w Darmouth warsztatów na temat nowego paradygmatu, który nazwał *Artificial Intelligence*, czyli inteligencją artefaktów. We wniosku [MMRS06] złożonym do the *Rockefeller Foundation* sformułowali cele AI, między innymi, cytując:

„The study is to proceed on the basis of the conjecture that every aspect of learning or any other feature of intelligence can in principle be so precisely described that a machine can be made to simulate it. An attempt will be made to find how to make machines use language, form abstractions and concepts,

solve kinds of problems now reserved for humans, and improve themselves. We think that a significant advance can be made in one or more of these problems if a carefully selected group of scientists work on it together for a summer.”

Zaproponowano badania dziedzin jak: teoria obliczeń, przetwarzanie języka naturalnego, sieci neuronowe (zaproponowane w roku 1943 [MP43], aktualnie stosowane w uczeniu głębokim [Sch22]), abstrakcje, złożoność i uczenie autonomiczne, które do dzisiaj są aktualnymi tematami badań i wykorzystywane w chatbotach, których podstawą są metody przetwarzania języka naturalnego (*ang. Natural Language Processing*, NLP) [KKKS22], a także metody rozumienia języka naturalnego (*ang. Natural Language Understanding*, NLU) [Len23] w oparciu o głębokie sieci neuronowe (*ang. Deep Neural Networks*, DNN) [AZH⁺21] i głębokie uczenie ze wzmocnieniem (*ang. Deep Reinforcement Learning*, DRL) [YJD22].

Dzięki połączeniu tych paradygmatów powstały narzędzia przewyższające człowieka dokładnością w wielu dziedzinach. Dla przykładu w konkursie *ImageNet: Large Scale Visual Recognition Challenge 2017 (ILSVRC2017)* [Ima23], w kategorii 2a: *localization + detection* najlepszy wynik osiągnął błąd rzędu 6.22% dla zadania detekcji obiektów i 2.48% dla zadania lokalizacji obiektów na obrazach.

Według [ShCYE17], metody oparte na AI w połowie roku 2014 osiągnęły poziom dokładności 95%, który jest właściwy dla człowieka, a w roku 2017 osiągnęły poziom dokładności podany w wynikach konkursu *ImageNet* [Ima23]. Poziom dokładności rozpoznawania mowy (*ang. Audio and Speech Recognition*, ARS) 94% właściwy dla człowieka został uzyskany przez AI po roku 2015 [ARS23].

Przełomowym był artykuł Krizhevsky’ego i innych [KSH12] z roku 2012, w którym wprowadzono głębokie splotowe (konwolucyjne) sieci neuronowe (*ang. Deep Convolutional Neural Network*, DCNN). Autorzy zbudowali sieć celem przydzielenia obrazom o wysokiej rozdzielczości jednej z 1000 klas decyzyjnych. Zaproponowana sieć (*AlexNet*) składała się z pięciu warstw splotowych wraz z warstwami łączącymi. Sieć zawierała 500000 neuronów oraz $60 \cdot 10^6$ parametrów. Autorzy uzyskali dokładność 60.3% dla błędu *top-1* oraz 81.1% dla błędu *top-5*, co było istotnym postępowaniem w porównaniu z wynikami wcześniejszymi. Od ukazania się pracy [KSH12], liczba nowych rozwiązań opartych o AI wzrosła z 200 do 500 w roku 2017

[Top19]. Równolegle, wzrosła liczba publikacji z tematyki głębokich sieci neuronowych, od kilkuset w roku 2012 do 10000 w roku 2018 [Top19].

Obecnie, AI wnika w niemal wszystkie dziedziny nauki oraz twórczości i jest stosowana powszechnie w dziedzinie rozpoznawania obrazów i wzorców oraz ich dopasowywania. Oczekuje się, że AI istotnie wspomogę medycynę tradycyjną w rozpoznawaniu symptomów i w diagnostyce. Poprzez wzrost liczby pacjentów, przeciętny czas wizyty u lekarza w USA około roku 2020 wynosił 12 minut dla pacjenta nowego i 7 minut dla pacjenta stałego [Top19]. To szansa dla metod AI we wsparciu diagnostów dzięki systemom automatycznej diagnozy na podstawie dużych zbiorów przypadków klinicznych, cytując za [Top19]:

„In the past few years, several studies relying on deep learning have been published in leading peer-reviewed medical journals. Many in the medical community were frankly surprised by what deep learning could accomplish: studies that claim AI’s ability to diagnose some types of skin cancer as well as or perhaps even better than board-certified dermatologists; to identify specific heart-rhythm abnormalities like cardiologists; to interpret medical scans or pathology slides as well as senior, highly qualified radiologists and pathologists, respectively; to diagnose various eye diseases as well as ophthalmologists; and to predict suicide better than mental health professionals. These skills predominantly involve pattern recognition, with machines learning those patterns after training on hundreds of thousands, and soon enough millions, of examples. Such systems have just gotten better and better , with the error rates for learning from text-, speech-, and image-based data dropping well below 5 percent, whizzing past the human threshold.”

Oczekiwania od AI są bardzo duże, ponownie cytując za [Top19]:

- „Outperform doctors at all tasks;
- Diagnose the undiagnosable;
- Treat the untreatable;
- See the unseeable on scans, slides;

- Predict the unpredictable;
- Classify the unclassifiable;
- Eliminate workflow inefficiencies;
- Eliminate hospital admissions and readmissions;
- Eliminate the surfeit of unnecessary jobs;
- 100% medication adherence;
- Zero patient harm;
- Cure cancer”.

Zacytowane pozycje mogą wydawać się czysto życzeniowe, jednak oddają one oczekiwania środowiska medycznego wobec przyszłych narzędzi AI. Do specjalności, które z AI już korzystają, włączyła się okulistyka.

Niniejszą rozprawę poświęcono zastosowaniom AI w okulistyce [Grz20, Grz21, WKR11]. Okulistyka wkracza intensywnie w korzystanie z narzędzi bazujących na AI: liczba publikacji na temat AI-eye wzrosła od około 80 w roku 2012 do ponad 500 w roku 2020 [Grz21].

Celem specyficznym niniejszej rozprawy jest wsparcie diagnostyki tylnego odcinka dna oka za pomocą metod opartych o AI.

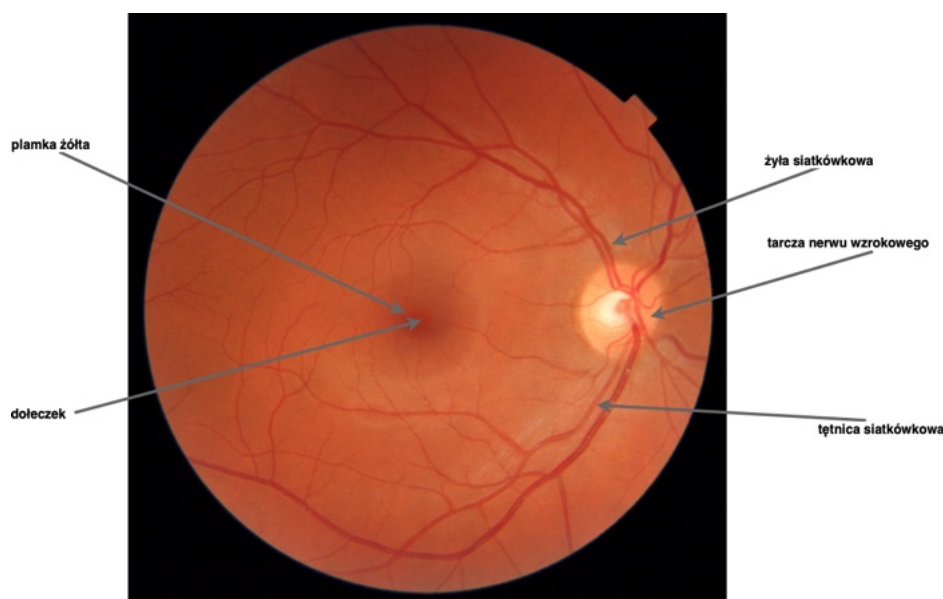
2.3 Diagnostyka dna oka

Fundoskopia to badanie okulistyczne, którego celem jest obrazowanie dna oka. Posługując się dedykowanymi narzędziami, specjalista może ocenić stan krytycznych składowych tylnego odcinka oka: plamki żółtej, tarczy nerwu wzrokowego, siatkówki oraz naczyń krwionośnych.

Obserwacja dna oka stała się możliwa dzięki konstrukcji oftalmoskopu [She89] przez Helmholtza, von Graefe i ich studentów w roku 1851 [WKR11]. Kolejne udoskonolenia doprowadziły do uzyskania fotografii barwnej tylnego odcinka oka (*ang. color retinal fundus photograph*) za pomocą *fundus kamery* (*ang. fundus camera*) [PHL⁺16]. W modelu barw *RGB* poszczególne kolory pozwalają dostrzec specyficzne cechy tylnego odcinka oka i w sumie razem syntezują pełny obraz: kolor zielony pozwala na lepszą obserwację naczyń

krwionośnych, kolor błękitny wyróżnia włókna nerwowe, a kolor czerwony wyodrębnia struktury pigmentowe [WKR11].

Dno oka to struktura anatomiczna, której obraz ukazuje cechy przedstawione na Rysunku 2.1.



Rysunek 2.1: Schemat anatomiczny budowy dna oka. Wersja elektroniczna obrazu: <https://krzywicki.pro/fig-2-1>.

Plamka żółta (*ang. macula*), której zadaniem jest zbieranie nadmiaru bodźców świetlnych leży w centrum siatkówki (*ang. retina*) składającej się z wielu warstw komórek światłoczułych i neuronów przekazujących sygnały wizualne do mózgu. W centrum plamki żółtej znajduje się dołeczek (*ang. fovea*), a w jej okolicach – tarcza nerwu wzrokowego (*ang. optic disc*), czyli miejsce, w którym nerw wzrokowy wychodzi z siatkówki. Na zdjęciu dna oka można również dostrzec struktury naczyniowe, takie jak żyły (*ang. retinal vein*) i tętnice (*ang. retinal artery*), które odpowiedzialne są za unaczynienie siatkówki.

W dzisiejszych czasach, zastosowanie AI w okulistyce otwiera nowe horyzonty w diagnostyce chorób oczu i chorób systemowych. Jednymi z najpopularniejszych rozwiązań, wykorzystywanych we wsparciu diagnostyki okulistycznej, są sieci neuronowe [Krz21]. Dzięki wysokim zdolnościom interpretacji informacji wizualnej, sieci neuronowe są w stanie dokładnie rozpoznawać cechy, struktury i charakterystyczne wzorce występujące na

zdjęciach dna oka, co pozwala na szybkie identyfikowanie okulistycznych i systemowych jednostek chorobowych. Warto podkreślić, że sieci neuronowe z powodzeniem uzyskują wyniki porównywalne z dokładnością specjalistów w autonomicznej diagnostyce DR [GBK⁺22].

2.4 Rola lokalizacji cech anatomicznych dna oka

Jednym z kluczowych elementów analizy zdjęć dna oka jest odpowiednia lokalizacja dołeczka oraz tarczy nerwu wzrokowego, które są istotne w diagnostyce okulistycznych jednostek chorobowych.

Lokalizacja dołeczka na zdjęciu dna oka umożliwia dokonanie analizy obszaru plamki żółtej i najbliższego otoczenia, której celem jest wykrywanie zmian, takich jak obrzęk czy uszkodzenia, które mogą być związane z chorobami siatkówki, w tym AMD [TWS⁺23, DPK⁺23] czy DME [SK23]. Ocena tarczy nerwu wzrokowego pozwala na identyfikację zmian związanych z chorobami nerwu wzrokowego, takimi jak jaskra [KYC23] czy ODE [BDMS⁺22].

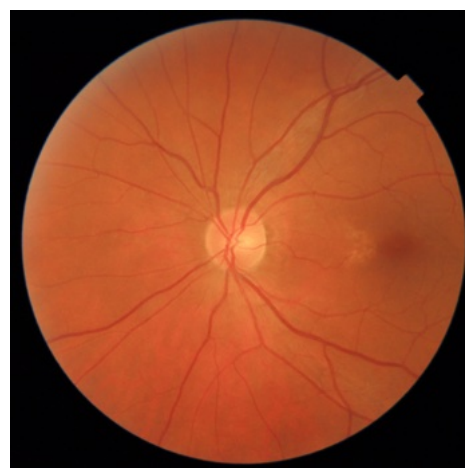
Lokalizacja struktur anatomicznych dna oka, takich jak tarcza nerwu wzrokowego oraz dołeczek, umożliwiają ocenę standardów wykonania zdjęć dna oka, które odgrywają istotną rolę w zapewnieniu dokładnej diagnostyki klinicznej. Centracja zdjęcia dna oka (na dołeczek lub tarczę nerwu wzrokowego, Rysunek 2.2) jest istotnym standardem z uwagi na możliwość dokładnej analizy ich obszarów, które są kluczowe dla funkcjonowania wzroku.

Lokalizacja tarczy nerwu wzrokowego i dołeczka odgrywa także kluczową rolę w określaniu strony oka (lewe lub prawe, Rysunek 2.3), co ma znaczenie podczas diagnostyki jednostek chorobowych na poziomie pacjenta oraz może wymagać analizy zdjęć zarówno oka lewego, jak i prawego.

Jakość zdjęcia dna oka to kolejny aspekt, w którym lokalizacja tarczy nerwu wzrokowego i dołeczka może odgrywać istotną rolę [RTM19]. Czytelne i poprawnie wykonane zdjęcie dna oka (Rysunek 2.1) charakteryzuje się wysoką ostrością, wyraźnym kontrastem, równomiernym oświetleniem i brakiem artefaktów optycznych. Prawidłowa ekspozycja zdjęcia dna



(a) Zdjęcie wycentrowane na dołeczek.



(b) Zdjęcie wycentrowane na tarczę nerwu wzrokowego.

Rysunek 2.2: Przykład dwóch standardów centracji zdjęcia dna oka. Wersja elektroniczna obrazu: <https://krzywicki.pro/fig-2-2>.

oka jest konieczna w celu wyraźnego ukazania struktur anatomicznych, których analiza jest istotna z punktu widzenia klinicznego. Nieprawidłowo oświetlone zdjęcia dna oka często charakteryzują się niewidocznymi cechami anatomicznymi, co przedstawiono na Rysunku 2.4. Można więc traktować widoczność plamki żółtej i tarczy nerwu wzrokowego jako jeden z wyznaczników jakości zdjęć dna oka.

2.5 Podsumowanie

Niniejszy rozdział stanowił wprowadzenie do tematyki zastosowań AI we wsparciu procesu diagnostyki okulistycznej na podstawie zdjęć dna oka. Analiza zdjęć dna oka jest istotną częścią diagnostyki okulistycznej, pozwalając specjalistom na wczesne wykrycie jednostek chorobowych i stanów patologicznych.

Dzięki AI, możliwe jest zastosowanie metod, które lokalizują kluczowe cechy na zdjęciach dna oka – dołeczek i tarczę nerwu wzrokowego. Metody te odgrywają istotną rolę w precyzyjnej diagnostyce stanu pacjenta i umożliwiają szybsze wdrożenie stosownych procedur klinicznych. Lokalizacja cech na zdjęciu dna oka może znaleźć także zastosowanie w ocenie



(a) Oko prawe wycelowane na dołeczek.



(b) Oko lewe wycelowane na dołeczek.

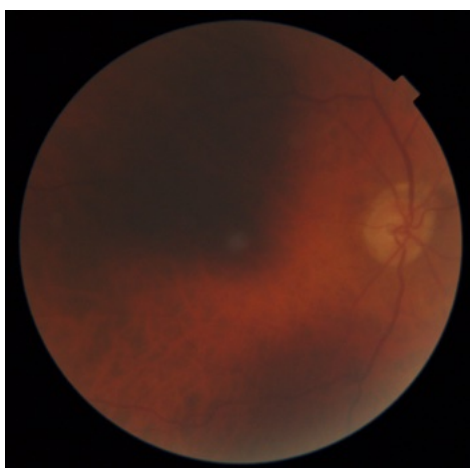


(c) Prawe oko wycelowane na tarczę nerwu wzrokowego.

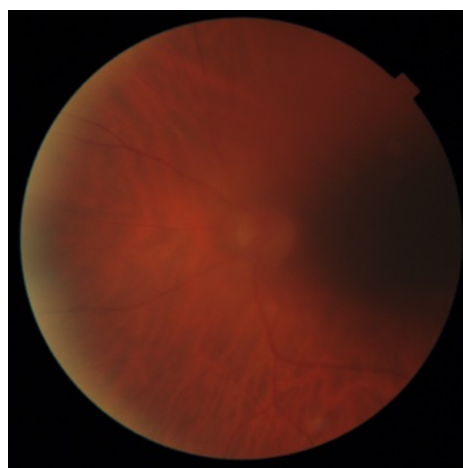


(d) Lewe oko wycelowane na tarczę nerwu wzrokowego..

Rysunek 2.3: Przykład zdjęć dna oka lewego i prawego wycelowanych na dołeczek oraz na tarczę nerwu wzrokowego. Wersja elektroniczna obrazu: <https://krzywicki.pro/fig-2-3>.



(a) Zdjęcie prawego oka wycentrowane na dołeczek z niedoświetlonym obszarem obejmującym około połowę siatkówki.



(b) Nieostre i niedoświetlone zdjęcie lewego oka wycentrowane na tarczycę nerwu wzrokowego. Zarówno plamka żółta, jak i tarcza nerwu wzrokowego są trudne lub niemożliwe do precyzyjnego wskazania.

Rysunek 2.4: Zdjęcia dna oka o niskiej jakości. Wersja elektroniczna obrazu: <https://krzywicki.pro/fig-2-4>.

standardu centracji, widocznej strony oka oraz w ocenie jakości zdjęcia.

Istotną do poruszenia kwestią jest zastosowanie zadania lokalizacji obiektów (cech anatomicznych dna oka) zamiast segmentacji, która polega na wydzieleniu regionów tych obiektów. Przesłanki, które kierowały za wyborem zadania (i metod) lokalizacji obiektów dotyczyły przede wszystkim podejścia za pomocą segmentacji. Operacja segmentacji może być mniej precyzyjna niż lokalizacja, zwłaszcza w przypadku obrazów o niskiej jakości. Oznaczenie regionów cech anatomicznych dna oka jest zadaniem znacznie bardziej czasochłonnym, niż wskazanie ramek otaczających te obiekty, co może przekładać się na znacznie dłuższy czas przygotowania bazy danych wykorzystywanych podczas treningu oraz testów opracowanych metod.

Rozdział 3

Metody rozpoznawania obrazów

3.1 Przedstawienie problematyki poruszanej w rozdziale

Rozpoznawanie obrazów jest obszarem informatyki, który umożliwia automatyczną analizę i interpretację treści obrazów cyfrowych. W Rozdziale 3 przedstawiono wybrane algorytmy i metody wykorzystywane w procesie rozpoznawania obrazów, które zastosowano podczas tworzenia proponowanych rozwiązań lokalizujących środki dołączka oraz tarczy nerwu wzrokowego omówionych w Rozdziale 5.

Rozdział rozpoczęto od przedstawienia pojęcia złożoności obliczeniowej algorytmów. Zrozumienie wymagań stawianych przez poszczególne algorytmy wobec zasobów obliczeniowych pozwoli na dokładną ocenę ich efektywności i właściwego zastosowania. Następnie przedstawiono ideę obrazu cyfrowego wraz z trzema kluczowymi modelami barw: RGB, CIELab i YUV celem przedstawienia sposobów reprezentacji kolorów oraz zastosowania tych modeli w analizie i przetwarzaniu zdjęć dna oka. W kolejnym kroku omówiono wybrane metody przetwarzania obrazów, takie jak filtrowanie obrazów, binaryzacja, wyrównywanie histogramów, wyznaczanie wypukłych powłok konturów wielokątów oraz analiza komponentów połączonych. Dla każdej z tych metod podano także ich obliczeniową złożoność, jeśli jest znana w bieżącym stanie wiedzy. W dalszej części skupiono się na algorytmach topologicznych, ze szczególnym naciskiem

na morfologię matematyczną, która umożliwia na analizę kształtu obiektów znajdujących na obrazie i wyodrębnianie ich cech. Następnie omówiono algorytmy inspirowane naturą, z głównym akcentem na sztucznych sieciach neuronowych wraz z omówieniem podstawowych pojęć uczenia maszynowego i uczenia głębokiego. W kolejnym etapie zaprezentowano nowoczesne metody uczenia maszynowego, uczenia głębokiego i rozpoznawania obrazu, takie jak uczenie transferowe, uczenie zespołowe i detekcja obiektów. W końcowej części rozdziału omówiono podejścia związane z zastosowaniami dedykowanych algorytmów i struktur danych oraz struktur sprzętowych w przyspieszaniu algorytmów.

3.2 Złożoność obliczeniowa algorytmu

Złożoność obliczeniowa jest kluczowym pojęciem w analizie algorytmów, które pozwala na ocenę ilości zasobów, takich jak czas i pamięć, potrzebnych do wykonania danej operacji. W tym rozdziale poruszono pojęcie złożoności obliczeniowej algorytmów oraz omówiono, jak ona wpływa na efektywność i wydajność rozwiązań.

3.2.1 Idea złożoności czasowej algorytmu i notacja big-O

Złożoność czasowa (*ang. temporal complexity*) algorytmu jest określana dla danego modelu obliczeń tj. mechanizmu który dla danej struktury wejściowej produkuje oczekiwaną strukturę wyjściową. Ideę złożoności obliczeniowej najlepiej jest wyjaśnić dla *maszyny Turinga* (*ang. Turing Machine*, TM) [Tur50b], czyli mechanizmu dla obliczeń reprezentowanego jako system $TM = \{T, P, H, Q, q_0, A, I, l, r, s\}$. Symbole w sygnaturze TM oznaczają kolejno: T – uporządkowany liniowo nieskończony zbiór komórek (analogia taśmy nieskończonej w obu kierunkach, podzielonej na równokształtne komórki w postaci kwadratów); P – zbiór tych komórek; H (head) – głowica czytająco-pisząca, przesuwaną się nad komórkami; Q – skończenie liczny zbiór stanów maszyny (reprezentant jej pamięci); q_0 – stan początkowy, w którym każdorazowo TM rozpoczyna obliczenie; A – alfabet maszyny (skończenie liczny zbiór symboli); I – zbiór instrukcji (program dla maszyny).

Symbole l, r, s to polecenia dla głowicy H , odpowiednio: przesunąć w lewo, w prawo oraz zostać w miejscu.

Niech A oznacza alfabet binarny: $A = \{0, 1\}$. Nad A , TM oblicza funkcję f_{TM} . Argumentami i wartościami tej funkcji są skończone ciągi nad A – elementy zbioru A^* . Podzbiory zbioru A^* są nazywane językami i są oznaczane symbolem L . Elementy L są nazywane słowami.

Znany jest sposób prowadzenia obliczeń przez TM [AB09, Pap93]. Złożoność obliczeniową funkcji f_{TM} określa funkcja $T(n)$ w postaci $T : N \rightarrow N$ na zbiorze liczb naturalnych N . Funkcja f_{TM} jest obliczana w czasie $T(n)$, jeśli dla każdego słowa $x \in A^*$, obliczenie wartości $f_{TM}(x)$ wymaga czasu (równego liczbie kroków w obliczeniu) $T(|x|)$ gdzie $|x|$ jest długością x . Językiem potocznym można określić, że TM pracuje w czasie $T(n)$, co oznaczono jako $time_{TM} = T(n)$ na potrzeby dalszych rozważań.

Funkcje o postaci T są ograniczone od dołu przez $T(n) = n$: dla ciągu o długości n , samo przejście TM przez ciąg wymaga n kroków.

Z uwagi na złożoność porównuje się funkcje, co wymaga miar dla porównania funkcji o wartościach i argumentach będących liczbami naturalnymi. Kryteria porównania pochodzą od Bachmanna [Bac06] i są wyrażone w notacji *big-O*.

W związku z tym dla pary funkcji $f, g : N \rightarrow N$, f ma złożoność $O(g)$ gdy istnieje liczba naturalna n_0 o własności 3.1 oraz stała $C > 0$ taka, że:

$$f(n) \leq Cg(n) \text{ dla } n \geq n_0 \quad (3.1)$$

W sytuacji określonej przez 3.1 piszemy też: $g = \Omega(f)$.

Notacja Bachmanna pozwala na określenie pewnych klas problemów w dwóch przypadkach: deterministycznym (D) i niedeterministycznym (ND). W przypadku deterministycznym, TM w każdym kroku ma możliwość co najwyżej jednej kontynuacji obliczenia, a w przypadku ND istnieją kroki obliczenia dla których istnieje kilka możliwości kontynuacji przez wybór losowy jednej z kilku instrukcji.

Elementy w A^* dla których f_{TM} oblicza wartość tworzą język L_{TM} . Język L_{TM} jest rozstrzygany (*ang. decided*) przez TM, co można uogólnić następująco: jeśli f_{TM} jest funkcją charakterystyczną dla L , t.j., dla każdego $x \in A^*$, $f_{TM}(x) = 1 \equiv x \in L$.

Klasa $DTIME(T(n))$ zawiera języki rozstrzygane przez maszyny TM pracujące w czasie $c \cdot T(n)$ dla pewnej stałej $c > 0$ (Równanie 3.2).

$$L \in DTIME(T(n)) \equiv (\exists TM [time_{TM} = T(n), (f_{TM}(x) = 1 \equiv x \in L)]) \quad (3.2)$$

Suma klas $DTIME(T(n^c))$ dla $c > 0$ to klasa P języków o złożoności wielomianowej:

$$P = \bigcup_{c>0} DTIME(T(n^c)) \quad (3.3)$$

Odpowiednik niedeterministyczny klasy P to klasa NP. Rozgraniczenie to odpowiada rozgraniczeniu obliczalny - weryfikowalny: każdy język w klasie P jest *obliczalny*, każdy język w klasie NP jest *weryfikowalny*, czyli każde w nim słowo ma certyfikat obliczalny: $L \in NP \equiv [\exists TM \exists p(n). time_{TM} = T(p(n)). \forall x \in A^*. (x \in L \equiv \exists y \in A^{p(|x|)}. f_{TM}(xy) = 1)]$. Słowo y jest certyfikatem dla x . W związku z tym każda instancja problemu z klasy NP jest weryfikowalna w czasie wielomianowym ale sam problem nie jest rozstrzygalny w czasie wielomianowym. Przykładem problemu z klasy NP jest *SAT* [Coo71] – problem spełnialności formuł rachunku zdań.

3.2.2 Złożoność pamięciowa

Złożoność pamięciowa (*ang. space complexity*) służy do określania zasobów pamięci TM i klasyfikowania języków według zasobów pamięci koniecznych dla ich rozstrzygalności lub weryfikacji. W tym przypadku nie istnieje ograniczenie dolne dla funkcji określających złożoność. Formalna definicja klas złożoności pamięciowej jest w analogii z definicjami w Podrozdziale 3.2.1.

Dla funkcji $S : N \rightarrow N$ język L jest w klasie $SPACE(S(n))$ jeśli istnieje TM oraz $c > 0$ takie, że TM rozstrzyga L oraz w tym procesie TM używa co najwyżej $c \cdot S(n)$ komórek taśmy dla każdego $x \in L$, z wyłączeniem komórek zajętych przez input (dla TM o kilku taśmach, brane pod uwagę są zajęte komórki na taśmach roboczych).

Jeśli TM jest niedeterministyczna (NTM), to klasa jest oznaczana jako $NSPACE(S(n))$.

Podstawowe relacje pomiędzy tymi klasami podaje poniższy ciąg inkluzji zawarty w Formule 3.4.

$$DTIME(S(n)) \subseteq SPACE(S(n)) \subseteq NSPACE(S(n)) \subseteq DTIME(S(2^{O(S(n))})) \quad (3.4)$$

Krótkie wprowadzenie do problematyki złożoności obliczeniowej jest jedynie anonsem tej rozbudowanej dziedziny i służy jako prolog do Rozdziału 3.

3.3 Obrazy cyfrowe i modele barw

Obrazy cyfrowe stanowią ważny element w dziedzinie obrazowania medycznego, gdzie odgrywają kluczową rolę w diagnozie, analizie i monitorowaniu stanu zdrowia. W tym podrozdziale omówiono główne aspekty związane z przetwarzaniem obrazów – ideę obrazu cyfrowego oraz modele barw wykorzystywane w pracy ze zdjęciami dna oka.

3.3.1 Obraz cyfrowy

Obrazy cyfrowe reprezentują rzeczywiste obiekty, które powstają w wyniku rzutowania ich na powierzchnię światłoczułą, przykładowo matrycę CCD lub CMOS [MPM15] znajdującą się w aparacie cyfrowym.

W ujęciu formalnym, jednowarstwowy obraz cyfrowy jest funkcją $p(x, y)$ reprezentowaną za pomocą macierzy, której elementy są wartościami sygnału dla pary współrzędnych x, y (odpowiadające pikselom, Formuła 3.5).

$$p(x, y) = \begin{bmatrix} p(1, 1) & p(1, 2) & \cdots & p(1, n) \\ p(2, 1) & p(2, 2) & \cdots & p(2, n) \\ \vdots & \vdots & \vdots & \vdots \\ p(m, 1) & p(m, 2) & \cdots & p(m, n) \end{bmatrix} \quad (3.5)$$

gdzie:

- m – wysokość obrazu,
- n – szerokość obrazu.

Zwykle sygnał jest reprezentacją intensywności światła w danym punkcie, lecz może odzwierciedlać także przykładowo odbiciowość radarową [Mer87]. Obrazy cyfrowe, w przeciwieństwie do obrazów analogowych, przyjmują skończony zbiór wartości dla zadanej pary współrzędnych. Każdy obraz cyfrowy składa się ze skończonej liczby pikseli, z których każdy ma przypisaną pozycję oraz wartość.

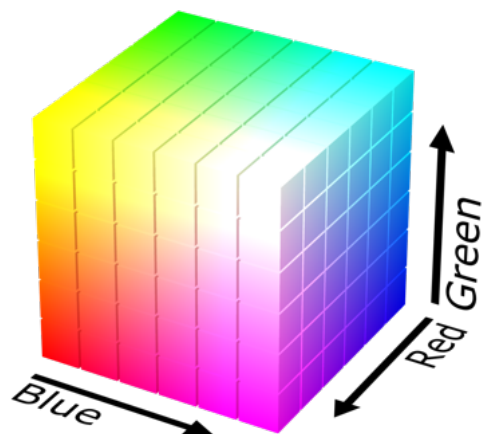
W praktyce często operuje się na obrazach kolorowych, które są reprezentowane za pomocą wielu macierzy (warstw) gdzie każda z nich odpowiada za inną składową wybranego modelu barw. Model barw jest matematyczną metodą opisu kolorów w określonym układzie współrzędnych. W kolejnych podrozdziałach omówiono wybrane modele barw, które wykorzystano w badaniach omówionych w Rozdziale 5.

3.3.2 RGB

Model barw *RGB* jest najpopularniejszą metodą reprezentacji obrazów kolorowych, często wykorzystywana jako domyślna w popularnych formatach plików. Każdy obraz kolorowy, opisany za pomocą modelu *RGB*, składa się z trzech warstw reprezentujących barwy: R – czerwony (*ang. red*), G – zielony (*ang. green*), B – niebieski (*ang. blue*). Barwa każdego piksela jest wyznaczana za pomocą kombinacji kolorów składowych warstw. Model barw *RGB* można przedstawić w formie sześcianu (Rysunek 3.1), gdzie każda z osi reprezentuje kolory składowe. Każdy z punktów sześcianu (woksel) reprezentuje poszczególny kolor.

W modelu barw *RGB* każdy piksel w każdej warstwie przyjmuje wartość od 0 do 255, gdzie 0 w każdej składowej oznacza kolor czarny, a 255 kolor biały.

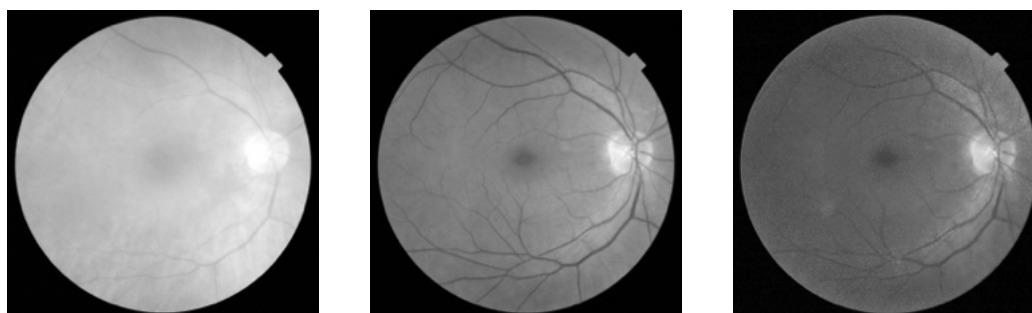
Na Rysunku 3.3 przedstawiono warstwy zdjęcia dna oka z Rysunku 3.2 w modelu barw *RGB*. Można zauważyć, że tarcza nerwu wzrokowego wraz z plamką żółtą najwyraźniej są zauważalne na warstwie G, która reprezentuje nasycenie barwy zielonej (Rysunek 3.3b).



Rysunek 3.1: Sześcian modelu barw RGB. Źródło: https://commons.wikimedia.org/wiki/File:RGB_color_solid_cube.png.



Rysunek 3.2: Przykładowe zdjęcie dna oka. Wersja elektroniczna obrazu: <https://krzywicki.pro/fig-3-2>.



(a) Warstwa R.

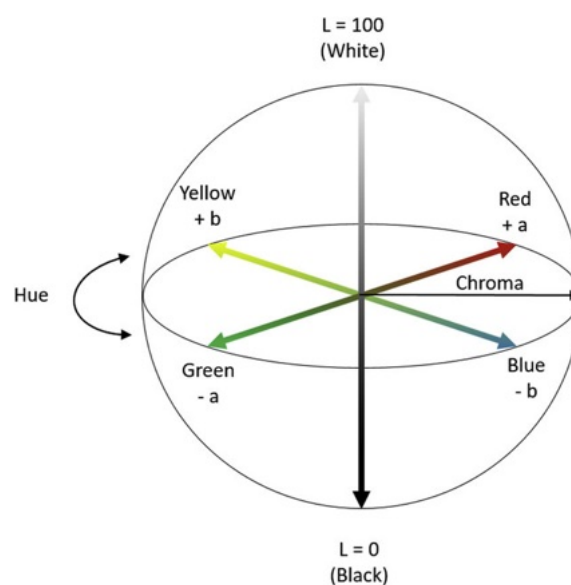
(b) Warstwa G.

(c) Warstwa B.

Rysunek 3.3: Warstwy zdjęcia dna oka w modelu barw RGB. Wersja elektroniczna obrazu: <https://krzywicki.pro/fig-3-3>.

3.3.3 CIELab

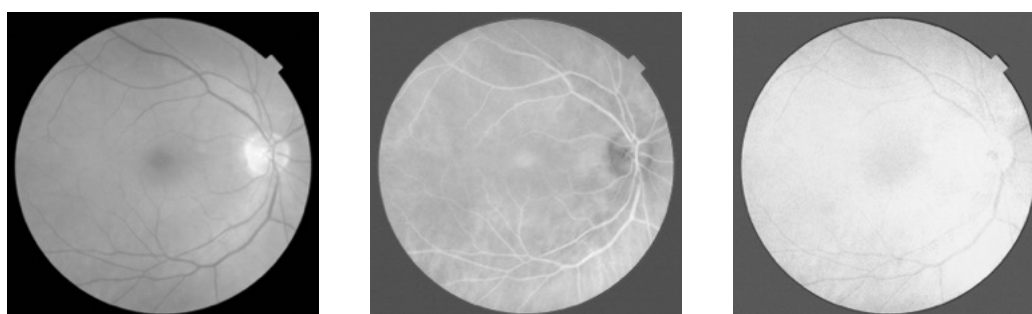
Model barw *CIELab* [Luo15] opisuje obraz za pomocą trzech składowych: L – jasność (*ang. lightness*), a – wartość barwy od czerwonej dla wartości dodatnich do zielonej dla wartości ujemnych (*ang. chroma*) oraz b – wartość barwy od niebieskiej dla wartości dodatnich do żółtej dla wartości ujemnych (*ang. hue*). Barwa każdego piksela jest wyznaczana na podstawie kombinacji osi wartości barw a i b oraz na podstawie jasności L (Rysunek 3.4).



Rysunek 3.4: Schemat modelu barw CIELab. Źródło: [LDF⁺ 20]

Model barw *CIELab* znajduje zastosowania w korekcji kontrastu obrazów [AAM14], a także w analizie zdjęć dna oka [JB15].

Na Rysunku 3.5 przedstawiono warstwy zdjęcia dna oka z Rysunku 3.2 w modelu barw *CIELab*. Można zauważyć, że tarcza nerwu wzrokowego wraz z plamką żółtą najwyraźniej są zauważalne na warstwie L, która reprezentuje jasność (Rysunek 3.5a).



(a) Warstwa L.

(b) Warstwa a.

(c) Warstwa b.

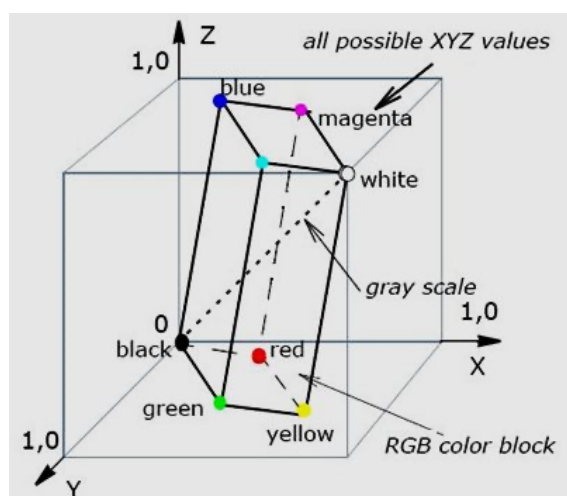
Rysunek 3.5: Warstwy zdjęcia dna oka w modelu barw *CIELab*. Wersja elektroniczna obrazu: <https://krzywicki.pro/fig-3-5>.

Metodę konwersji obrazów między modelami *RGB* i *CIELab* można znaleźć w Podrozdziale C.1.1 znajdującym się w Dodatku C.1

3.3.4 YUV

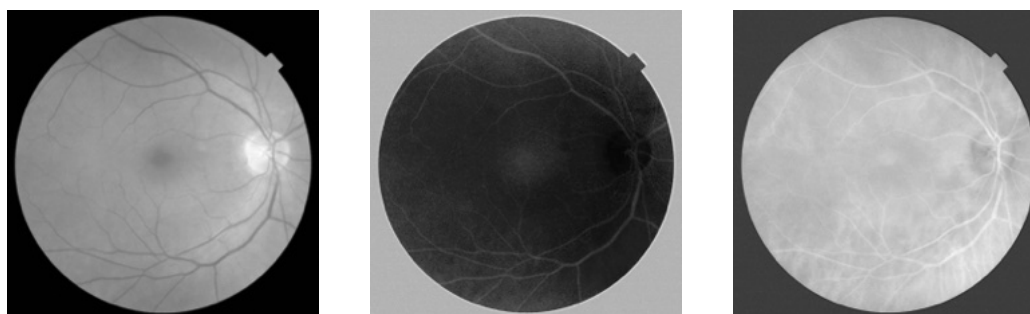
Model barw *YUV* [PKKJ14] opisuje obraz za pomocą trzech składowych: L – luminacja (*ang. luminance*), U – chrominacja (nasycenie i odcień) (*ang. chrominance*) barwy niebieskiej oraz b – chrominacja barwy czerwonej. Barwa każdego piksela jest wyznaczana na podstawie kombinacji osi chrominacji *U* i *V* oraz na podstawie luminacji *Y* (Rysunek 3.6).

Model barw *YUV* wywodzi się z sygnału telewizyjnego, który w okresie wprowadzania kolorowych odbiorników przekazywany był w sposób ujednolicony umożliwiający prawidłowy odbiór zarówno na telewizorach „czarno-białych”, jak i kolorowych. Obecnie znajduje szerokie spektrum zastosowań, między innymi w analizie obrazów medycznych [Din21].



Rysunek 3.6: Schemat modelu barw CIE Lab. Źródło: [IHKM12]

Na Rysunku 3.7 przedstawiono warstwy zdjęcia dna oka z Rysunku 3.2 w modelu barw YUV . Można zauważyć, że tarcza nerwu wzrokowego wyraźnie jest dostrzegalna (w postaci negatywu) na warstwie U , która reprezentuje chrominację barwy niebieskiej (Rysunek 3.7b). Warto mieć także na uwadze fakt, że warstwy Y z modelu barw YUV oraz warstwa L z modelu barw $CIE Lab$ reprezentują oświetlenie i są do siebie podobne, lecz mimo tego nie są takie same.



(a) Warstwa Y .

(b) Warstwa U .

(c) Warstwa V .

Rysunek 3.7: Warstwy zdjęcia dna oka w modelu barw YUV . Wersja elektroniczna obrazu: <https://krzywicki.pro/fig-3-7>.

Metodę konwersji obrazów między modelami RGB i YUV można znaleźć w Podrozdziale C.1.2 znajdującym się w Dodatku C.1.

3.4 Metody przetwarzania obrazów

Przetwarzanie obrazów odgrywa istotną rolę w automatycznym wsparciu diagnostyki medycznej z uwagi na zdolność wydobycia wartościowych informacji z cyfrowych reprezentacji struktur anatomicznych i patologicznych. Metody przetwarzania obrazów można podzielić na generyczne (stosowane w wielu rodzajach obrazów) i dziedzinowe (przystosowane do konkretnego typu obrazu, przykładowo optyczna koherentna tomografia *Optical Coherence Tomography* – *OCT* [ADFM19]).

W tym podrozdziale omówiono wybrane generyczne metody przetwarzania obrazów, które mają szczególne zastosowanie także w analizie medycznej i zostały wykorzystane w przygotowaniu metod przedstawionych w Rozdziale 5.

3.4.1 Operacja splotu i zastosowania filtrów

Filtrowanie obrazów (filtracja) jest operacją matematyczną dokonywaną na obrazie źródłowym, w wyniku której uzyskiwany jest nowy, przekształcony obraz. Filtrację określa się jako transformację kontekstową z uwagi na konieczność wykorzystania informacji z wielu sąsiadujących pikseli w celu wyznaczenia nowej wartości piksela w obrazie docelowym. Filtrowanie obrazów jest operacją dokonywaną za pomocą *operacji splotu* [BAV13] (w języku angielskim określana jako *convolution* – konwolucja) lub transformaty obrazu [SMKA19] i wybranego filtru w postaci macierzy kwadratowej o nieparzystym rozmiarze.

Celem filtracji jest wydobycie szeregu użytecznej informacji z obrazu w oryginalnej formie przed dokonaniem dalszych N kroków przetwarzania, przykładowo: położenia krawędzi lub rogów obiektów. Filtracja może także znaleźć zastosowanie w redukcji szumów czy także w wyostrzaniu szczegółów obrazów. Filtrację można przeprowadzać w dziedzinie przestrzennej oraz częstotliwościowej. W celu dokonania operacji filtrowania w dziedzinie częstotliwościowej zastosowanie znajduje operacja mnożenia transformat obrazu oraz filtru. Przy znacznie bardziej popularnej operacji filtrowania w dziedzinie przestrzennej (liniowej) zastosowanie znajduje *operacja splotu*.

Splot

Splot dyskretny [BAV13] (Równanie 3.6) jest matematycznym działaniem polegającym na przekształceniu obrazu za pomocą filtru (maski) celem przygotowania do dalszych kroków analizy.

$$q(x, y) = \sum_k \sum_l p(x - k, y - l) \cdot h(k, l) \quad (3.6)$$

gdzie:

- $q(x, y)$ – wartość piksela o współrzędnych (x, y) obrazu wyjściowego,
- $p(x, y)$ – wartość piksela o współrzędnych (x, y) obrazu wejściowego,
- $h(k, l)$ to wartość filtru w punkcie (k, l) .

Najważniejsza idea operacji splotu dyskretnego polega na przesuwaniu okna na obrazie wejściowym p wzdłuż sygnału h oraz na dokonaniu operacji iloczynu skalarnego obydwu macierzy. W ten sposób piksele wynikowego obrazu q powstają na podstawie odpowiadających sąsiednich pikseli obrazu oryginalnego p .

W 1989 roku LeCun i inni [LBD⁺89] zaproponowali zastosowanie operacji splotu w sieci neuronowej, co zapoczątkowało oddzielny nurt zastosowań polegający na automatycznym uczeniu wyszukiwania wzorców na obrazach za pomocą *splotowych sieci neuronowych*.

W przeciwieństwie do splotowych sieci neuronowych, w operacji filtrowania obrazów za pomocą operacji splotu wykorzystuje się odgórnie ustalone filtry będące macierzami kwadratowymi. Operacja filtracji jest także często dokonywana na obrazach jednowarstwowych.

Złożoność operacji splotu wynosi $O(k^2 \cdot HW)$ [WTW⁺22], gdzie:

- k – rozmiar filtru,
- H – wysokość obrazu,
- W – szerokość obrazu.

Wybrane zastosowania filtrów

Filtr gaussowski w połączeniu z operacją splotu dokonuje odszumiania obrazu poprzez rozmycie i usunięcie niewielkich obiektów. *Filtr gaussowski* występuje w różnych rozmiarach, według zależności: im większy rozmiar filtru, tym wyraźniejsza redukcja szumów i rozmycie. Ogólny wzór wyznaczania filtru gaussowskiego przedstawiono w Równaniu 3.7.

$$h(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (3.7)$$

gdzie:

- $h(x, y)$ – wartość filtru gaussowskiego w punkcie o współrzędnych x, y ,
- σ – odchylenie standardowe reprezentujące rozmiar filtru.

Przykładowe filtry gaussowskie o rozmiarach 3 i 5 przedstawiono w Formułach 3.8 i 3.9.

$$h = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \quad (3.8)$$

$$h = \frac{1}{256} \begin{bmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & 36 & 24 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix} \quad (3.9)$$

Przykładowe działanie filtrów gaussowskich na zaszumionym obrazie przedstawiono na Rysunku 3.8. Można zauważyć, że struktury anatomiczne siatkówki uległy niewielkiemu rozmyciu, co wiąże się także z redukcją szumów.

Filtr gaussowski można stosować zarówno na obrazach jednowarstwowych, jak i wielowarstwowych, przy czym operację filtrowania należy przeprowadzić na każdej warstwie z osobna.



(a) Przykładowe zdjęcie dna oka.

(b) Zdjęcie dna oka po operacji filtrowania z wykorzystaniem filtru gaussowskiego o rozmiarze 3.

(c) Zdjęcie dna oka po operacji filtrowania z wykorzystaniem filtru gaussowskiego o rozmiarze 5.

Rysunek 3.8: Przykład zastosowania operacji filtrowania zdjęcia dna oka z wykorzystaniem filtru gaussowskiego o rozmiarach 3 i 5. Charakter zmian na zdjęciu ma kluczowe znaczenie dla dalszej analizy algorytmicznej, chociaż nie zawsze jest łatwo dostrzegalny go gołym okiem ludzkim. Wersja elektroniczna obrazu: <https://krzywicki.pro/fig-3-8>.

Innym przykładem filtru może być *operator (lub filtr) Sobela* [SF68, KVB88], którego przeznaczeniem jest wykrywanie krawędzi obiektów. *Filtr Sobela* jest operatorem dyskretnego różniczkowania, umożliwiającą aproksymację pochodnych kierunkowych nasycenia pikseli obrazu w ośmiu kierunkach co 45° . *Operator Sobela* jest najczęściej stosowany na obrazach jednowarstwowych.

Każdy z ośmiu kierunków pochodnych kierunkowych wyznaczany jest za pomocą macierzy asymetrycznych w stosunku do kierunku wykrywanej krawędzi. Każda z ośmiu przeznaczonych do tego macierzy umożliwia wyznaczenie kierunku od 0° do 315° z krokiem wynoszącym 45° , gdzie kierunek 0° oznacza krawędź pionową, a 90° – poziomą. W Formułach 3.10 – 3.13 przedstawiono filtry dla kierunków 0° – 135° z krokiem wynoszącym 45° . Kolejne filtry otrzymuje się obracając kolejno filtry podane w Formułach 3.10 – 3.13 o 180° .

$$h_0 = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \quad (3.10)$$

$$H_{45} = \begin{bmatrix} -2 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 2 \end{bmatrix} \quad (3.11)$$

$$H_{90} = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad (3.12)$$

$$H_{135} = \begin{bmatrix} 0 & 1 & 2 \\ -1 & 0 & 1 \\ -2 & -1 & 0 \end{bmatrix} \quad (3.13)$$

Przykład działania *operatora Sobela* na obrazie przedstawiono na Rysunku 3.9.

3.4.2 Metody binaryzacji

Binaryzacja to operacja przetwarzająca obrazy (najczęściej jednowarstwowe) do postaci uproszczonej, gdzie każdy piksel na całym obrazie lub w ustalonym lokalnym obszarze, może przyjąć jedną z dwóch wartości, co w znaczny sposób redukuje ilość zawartej w nim informacji. Binaryzacja często realizowana jest za pomocą progowania, które polega na ustaleniu wartości progowej, poniżej której piksele obrazu kwalifikowane są jako piksele głównego obiektu, natomiast pozostałe piksele są kategoryzowane jako piksele tła niewchodzące w skład tekstury obiektu.

Wśród najpopularniejszych podejść do binaryzacji obrazu można wyróżnić globalne i lokalne oraz polegające na odgórnie przydzielonej i automatycznie wyznaczonej wartości progowej.

Progowanie proste

Metoda progowania prostego polega na zamianie wartości piksela obrazu na podstawie odgórnie ustalonej wartości progowej, co przedstawiono w Równaniu 3.14.



(a) Przykładowe zdjęcie.



(b) Zdjęcie z wykrytymi krawędziami w kierunku 0°.



(c) Zdjęcie z wykrytymi krawędziami w kierunku 45°.



(d) Zdjęcie z wykrytymi krawędziami w kierunku 90°.



(e) Zdjęcie z wykrytymi krawędziami w kierunku 135°.



(f) Połączone krawędzie wykryte w kierunkach 0-135°.

Rysunek 3.9: Przykład działania operatora Sobela na zdjęciu latarni morskiej w miejscowości Elaties. Wersja elektroniczna obrazu: <https://krzywicki.pro/fig-3-9>.

$$q(x, y) = \begin{cases} 0, & \text{gdy } p(x, y) < th \\ 255, & \text{gdy } p(x, y) \geq th \end{cases} \quad (3.14)$$

gdzie:

- p – obraz oryginalny,

- q – obraz będący efektem operacji progowania prostego,
- x, y – współrzędne piksela na obrazie,
- th – wartość progu.

Jak można zauważyć w Równaniu 3.14, w zależności od postaci zwartości obrazu pikselom obiektów będących przedmiotami zainteresowania nadaje się wartość minimalną (0), pikselom tła wartość maksymalną (255). Przykład działania metody progowania prostego dla różnych wartości progowych przedstawiono na Rysunku 3.10. Można zauważyć, że przyjęcie wartości progowych oddalonych od środka zakresu progów (127, Rysunek 3.10c) powoduje zmniejszenie widoczności szczegółów na obrazie poprzez „prześwietlenie” (Rysunek 3.10b) lub „niedoświetlenie” (Rysunek 3.10d).



(a) Lena.

(b) Wynik operacji progowania prostego z progiem o wartości 64.

(c) Wynik operacji progowania prostego z progiem o wartości 127.

(d) Wynik operacji progowania prostego z progiem o wartości 191.

Rysunek 3.10: Przykład działania metody progowania prostego na zdjęciu „Lena” [Mun96]. Wersja elektroniczna obrazu: <https://krzywicki.pro/fig-3-10>.

Złożoność metody progowania prostego wynosi $O(H \cdot W)$, gdzie H to wysokość obrazu wyrażona w pikselach, a W to szerokość.

Metoda Otsu

Wśród globalnych metod binaryzacji obrazów na szczególną uwagę zasługuje metoda Otsu [Ots79], której działanie polega na dokonaniu operacji progowania prostego dla wszystkich dostępnych wartości progowych oraz na wyznaczeniu wariancji dla powstałych w ten sposób dwóch klas

wartości pikseli na każdym z obrazów zbinaryzowanym za pomocą każdego z wykorzystanych wartości progowych (Równanie 3.15).

$$\sigma_{w^2} = w_1(t)\sigma_1^2 + w_2(t)\sigma_2^2 \quad (3.15)$$

gdzie:

- $w_1(t)$, $w_2(t)$ – względne wagi dla obydwu klas pikseli, zbinaryzowanego obrazu za pomocą metody progowania prostego, wyrażone za pomocą liczby pikseli,
- σ_1^2 , σ_2^2 – wariancje w obydwu klasach pikseli.

Metoda Otsu automatycznie wyznacza próg binaryzacji poprzez minimalizację wyrażenia przedstawionego w Równaniu 3.15. W związku z tym automatycznie wyznaczoną wartością progową zostaje ta, dla której σ_{w^2} uzyska najmniejszą wartość.

Na Rysunku 3.11 przedstawiono przykład binaryzacji *metodą Otsu* na podstawie progów wyznaczonego automatycznie wynoszącego 117.

Można zauważyć, że wartość progowa została wyznaczona poprawnie, co widać w szczegółowości powstałego obrazu i jest bliska środkowi zakresu progów (127). Bliskość wartości progowej wyznaczonej *metodą Otsu* do środka zakresu progów wynika z charakteru obrazu „Lena”, który ma wyrównaną rozpiętość tonalną, dzięki czemu jest popularny w literaturze dotyczącej przetwarzania obrazów.

Złożoność *metody Otsu* wynosi $O(2 \cdot W \cdot H + 256)$ [PTC⁺20], gdzie H to wysokość obrazu wyrażona w pikselach, a W to szerokość.

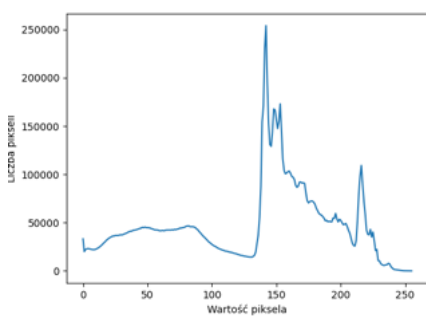
3.4.3 Wyrównywanie histogramów

Wyrównywanie histogramów jest jedną z metod przetwarzania obrazów, której celem jest uzyskanie histogramu obrazu z bardziej równomierną dystrybucją, niż w przypadku obrazu oryginalnego (Rysunek 3.12).

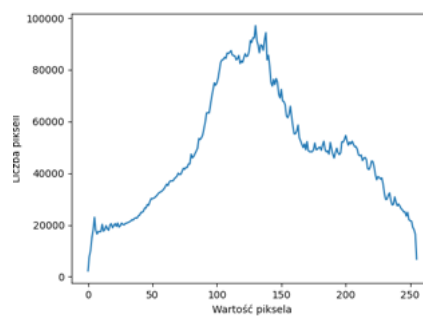


Rysunek 3.11: Przykład działania binaryzacji metodą Otsu na zdjęciu „Lena” [Mun96]. Wersja elektroniczna obrazu: <https://krzywicki.pro/fig-3-11>.

Dzięki zastosowaniu operacji wyrównywania histogramów obrazy zyskują na kontrastowości i szczegółowości.



(a) Histogram przykładowego obrazu jednowarstwowego.



(b) Histogram przykładowego obrazu po zastosowaniu operacji wyrównania.

Rysunek 3.12: Przykład działania metody wyrównywania histogramów. Wersja elektroniczna obrazu: <https://krzywicki.pro/fig-3-12>.

Wśród metod wykorzystywanych w wyrównywaniu histogramów można wyróżnić wykorzystujące podejścia globalne i lokalne. Podejścia globalne operują na pełnej dystrybucji histogramu obrazu, co może prowadzić do pewnych ograniczeń. W przypadku, gdy różne obszary obrazu mają znacząco odbiegające od siebie rozkłady wartości pikseli, metoda globalna może wpływać na prześwietlenie i znaczne zaszumienie, co jest niepożądanym efektem. Podejścia adaptacyjnej, w przeciwieństwie do metod globalnych, operują na fragmentach obrazów, co może ograniczyć niepożądane efekty występujące przy zastosowaniu metod globalnych.

Globalne wyrównywanie histogramów

Operację globalnego wyrównania histogramu obrazu jednowarstwowego można przedstawić za pomocą Równania 3.16.

$$h_{new}(v) = \text{round} \left(\frac{cdf(v) - cdf_{min}}{(H \cdot W) - cdf_{min}} \cdot L - 1 \right) \quad (3.16)$$

gdzie:

- $h_{new}(v)$ – funkcja zwracająca nową wartość przyporządkowaną pikselowi o wartości v ,
- H, W – wysokość i szerokość obrazu,
- cdf – histogram skumulowany obrazu źródłowego,
- cdf_{min}, cdf_{max} – wartość minimalna (niezerowa) oraz maksymalna histogramu skumulowanego obrazu źródłowego,
- L – liczba wartości przyjmowanych przez piksele (zazwyczaj 256),
- round – funkcja zaokrąglająca przekazaną wartość do postaci całkowitej.

Celem transformacji przedstawionej w Równaniu 3.16 jest normalizacja wartości każdego piksela poprzez jej podmianę na podstawie funkcji h_{new} .



(a) Jednowarstwowe zdjęcie (w skali szarości) przedstawiające miasto Zakynthos.



(b) Zdjęcie 3.13a przetworzone za pomocą globalnej metody wyrównywania histogramów.

Rysunek 3.13: Przykład jednowarstwowego zdjęcia przetworzonego za pomocą globalnej metody wyrównywania histogramów. Wersja elektroniczna obrazu: <https://krzywicki.pro/fig-3-13>.

Na Rysunku 3.13 przedstawiono efekt działania operacji globalnego wyrównania histogramu obrazu. Można zauważyć, że zdjęcie zyskało na kontrastowości, co widać szczególnie w obszarze chmur.

Adaptywne wyrównywanie histogramów

Adaptywne (lokalne) wyrównywanie histogramu obrazu [KLW⁺76, Hum77] (AHE, ang. *Adaptive Histogram Equalization*) to operacja polegająca na zastosowaniu metody globalnej niezależnie na małych fragmentach obrazu źródłowego – analogicznie do operacji splotu.

W odpowiedzi na metodę AHE, która ma tendencję do nadmiernego wzmocnienia kontrastu w wybranych obszarach obrazu, powstała metoda CLAHE [Rez04] (ang. *Contrast Limited Adaptive Histogram Equalization*) będąca modyfikacją AHE, która ogranicza kontrastowość zdjęcia zmniejszając jednocześnie widoczność szumów na obrazie.

Metoda CLAHE, w przeciwieństwie do AHE, ogranicza kontrastowość obrazu poprzez przycięcie histogramu obrazu źródłowego za pomocą parametru *CL* (ang. *Clip Limit*) przed operacją wyznaczenia histogramu skumulowanego *cdf* (Równanie 3.17), gdzie wartości przycięte służą do ponownej redystrybucji histogramu obrazu źródłowego, co przedstawiono na

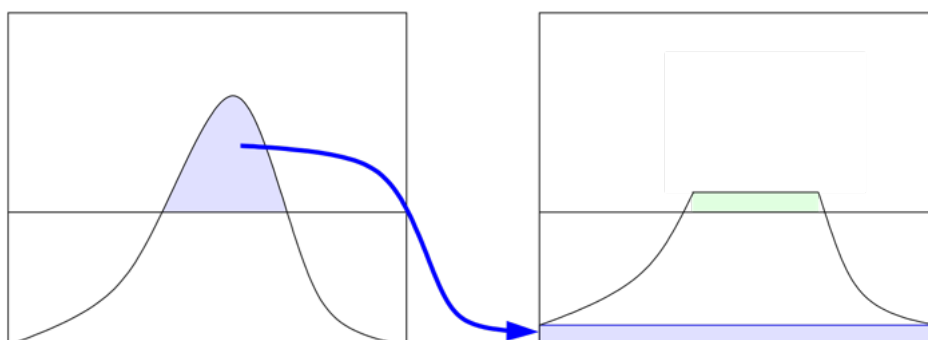
Rysunku 3.14.

$$new_v = v + h_{original}(v) - CL \quad (3.17)$$

gdzie:

- v – oryginalna wartość intensywności piksela,
- new_v – nowa wartość intensywności piksela,
- $h_{original}(v)$ – liczebność pikseli o wartości v na obrazie źródłowym.

Metody *AHE* oraz *CLAHE* można stosować także na obrazach wielowarstwowych, co omówiono w pracach odpowiednio [Hum77] oraz [Rez04].



Rysunek 3.14: Ilustracja operacji przycinania wartości histogramu w metodzie *CLAHE* oraz ich ponownej redystrybucji. Źródło: <https://commons.wikimedia.org/wiki/File:Clahc-redist.svg>.

Przykład działania metody *CLAHE* zaprezentowano na Rysunku 3.15.

Operacji *CLAHE* poddano zdjęcie 3.13a wykorzystane w przykładzie działania metody globalnego wyrównania histogramów. Można zauważyć, że wynikiem metody *CLAHE* jest obraz o bardziej zrównoważonym kontraście, niż w przypadku metody globalnego wyrównania histogramów (Rysunki 3.13b oraz 3.15b).



(a) Jednowarstwowe zdjęcie (w skali szarości) przedstawiające miasto Zakynthos.



(b) Zdjęcie 3.15a przetworzone za pomocą metody CLAHE.

Rysunek 3.15: Przykład jednowarstwowego zdjęcia przetworzonego za pomocą metody CLAHE. Wersja elektroniczna obrazu: <https://krzywicki.pro/fig-3-15>.

3.4.4 Wyznaczanie wypukłych powłok konturów

Jednym z etapów przetwarzania obrazów jest wyodrębnianie obszarów obiektów stanowiących przedmiot zainteresowania. Powstałe w ten sposób obiekty mogą przyjmować postać wielokątów, które mogą wymagać uproszczenia poprzez znalezienie wypukłej powłoki ich konturu celem dokonania optymalizacji analizy kształtu lub przynależności punktowej.

Powłoka wypukła konturu to wielokąt obejmujący obiekt w całości, który składa się jedynie z punktów należących oryginalnie do obiektu i w każdym miejscu jest wypukły, co oznacza że kąt wewnętrzny między każdą kolejną trójką punktów jest mniejszy niż 180° (Rysunek 3.16).

Wśród podejść służących do wyznaczania wypukłej powłoki konturów wielokątów istnieje algorytm *Convex hull* [Sk182], którego zasada działania polega na znalezieniu najmniejszego wielokąta wypukłego, który zawiera zbiór punktów wchodzący w skład analizowanego konturu.

Algorytm *Convex hull* rozpoczyna swoje działanie od wyznaczenia punktu początkowego, którym może być punkt konturu o najmniejszej lub największej wartości współrzędnej x , lub y . Następnie punkty konturów są sortowane według kątów, jakie tworzą względem punktu początkowego. Kolejne punkty konturu są dodawane do zewnętrznego wypukłego konturu



(a) Przykładowe zdjęcie dłoni z wyciętym tłem.



(b) Wypukła powłoka konturu obiektu ze zdjęcia 3.16a.

Rysunek 3.16: Przykład wyznaczonej powłoki wypukłej konturu fragmentu dłoni, której kształt jest wielokątem. Kolorem zielonym oznaczono oryginalny kontur dłoni, czerwonym – uproszony wielokąt powstały w wyniku wyznaczenia wypukłej powłoki oryginalnego konturu. Wersja elektroniczna obrazu: <https://krzywicki.pro/fig-3-16>.

wynikowego w odpowiedniej kolejności, aby tworzyć wierzchołki wielokąta, dopóki nie dotrze do punktu początkowego, uzyskując tym samym zamknięty kontur. W wyniku tego etapu usunięte zostają punkty wewnętrzne, a pozostawione jedynie punkty krawędzi, które tworzą najmniejszy wypukły wielokąt.

Przykład 3.16b wchodzący w skład Rysunku 3.16 przedstawiającego przykład wyznaczonej powłoki wypukłej konturu fragmentu dłoni wyznaczono za pomocą omawianej metody *Convex hull*.

Złożoność metody *Convex hull* wynosi $O(n \cdot \log n)$ gdzie n jest rozmiarem wejścia (liczbą punktów konturu).

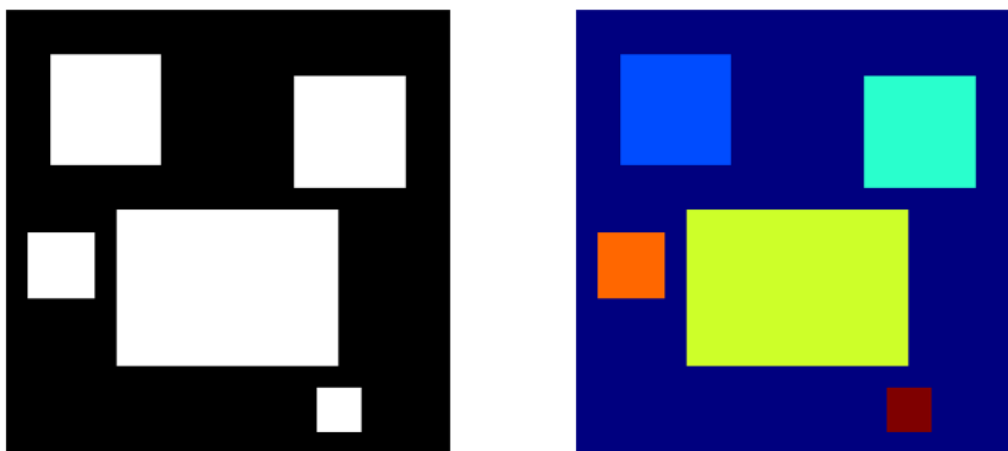
3.4.5 Analiza komponentów połączonych

Analiza komponentów połączonych jest procesem polegającym na przydzielaniu na obrazach unikalnych etykiet obiektom pierwszoplanowym, które charakteryzują się ciągłym obszarem i homogeniczną teksturą. W związku z tym *analiza komponentów połączonych* jest stosowana zwykle na obrazach będących wynikiem procesu segmentacji. *Analiza komponentów połączonych* jest często wykorzystywana w zadaniach segmentacji tła oraz obiektów jako filtr etapu przetwarzania końcowego celem usuwania plam szumów, regionu tła a także w metodach realizujących zadania optycznego rozpoznawania znaków (*ang. optical character recognition, OCR*) [ALT22].

Zasady działania metod *analizy komponentów połączonych* są do siebie zbliżone i polegają na przejściu przez obraz poruszając się w jednym kierunku oraz przypisywaniu etykiet na bieżąco w zależności od wartości sąsiednich pikseli. Dla każdego piksela o wartości > 0 (wchodzącego w skład komponentu) są sprawdzane piksele znajdujące się w jego sąsiedztwie i na tej podstawie podejmowana jest decyzja, czy dany piksel powinien być przypisany do już istniejącej etykiety (składnika), czy też powinien otrzymać nową, unikalną etykietę.

Na Rysunku 3.17 przedstawiono przykład działania *analizy komponentów połączonych*. Można zauważyć, że każdy z obiektów (łącznie z tłem) został oznaczony unikalną etykietą zobrazowaną za pomocą koloru.

Metody *analizy komponentów połączonych* na przestrzeni lat ulegały zmianom głównie z uwagi na wydajność oraz dokładność działania [RP66, HC15, HRG⁺17, ABG20]. Warto wśród nich wyróżnić jedno z najnowszych podejść – algorytm *Spaghetti Labeling* [BABG20], który w celu optymalizacji wydajności wykorzystuje ideę zmodyfikowanego drzewa decyzyjnego [Qui86, BCBG18]. Bolelli i inni w pracy [BABG20] porównali czas działania zaproponowanej metody wraz z innymi rozwiązaniami, co pozwoliło przesądzić że algorytm *Spaghetti labeling* okazał się najwydajniejszy.



(a) Przykładowy obraz przedstawiający pięć rozłącznych obiektów.

(b) Obiekty z obrazu 3.17a oznaczone etykietami w postaci kolorów.

Rysunek 3.17: Przykład działania metody analizy komponentów połączonych. Wersja elektroniczna obrazu: <https://krzywicki.pro/fig-3-17>.

3.5 Algorytmy topologiczne i algorytmy inspirowane naturą

Maszyny Turinga to jeden z wielu modeli obliczeń. Teza Churcha-Turinga, zweryfikowana dla między innymi rachunku λ Churcha, systemów Thue oraz Markowa, gramatyk formalnych itd. postuluje, że każdy fizycznie realizowalny model obliczeń jest w zakresie wielomianowej złożoności emulowany przez maszynę Turinga.

Tematyka rozprawy – *Metody Rozpoznawania Obrazów w Badaniu Dna Oka dla Celów Diagnostyki Medycznej* – skłania do wkroczenia w obszar algorytmów nienumerycznych lub hybrydowych – analizy struktur wzorowanych na strukturach biologicznych ale przenoszących wartości numeryczne, znanych jako struktury teksturowe (*ang. texture structures*). Opracowanie Érdiego [Erd08] przynosi obszerną analizę systemów o dużej złożoności, pokrewnych w poruszanych badaniach.

3.5.1 Operacje morfologiczne

Morfologia matematyczna wyrosła z powodów praktycznych – badania stali z uwagi na porowatość [MS02], jednak jej idee pochodzą z topologii matematycznej. Operacje topologii dokonują się z pomocą operatorów wnętrza Int i domknięcia Cl . Przestrzeń topologiczna to zbiór X z rodziną jego podzbiorów τ zamkniętą z uwagi na operacje sumy dowolnej podrodziny i przecięcia dla podrodzin skończonych.

Dla danego podzbioru $A \subseteq X$, wnętrze $IntA$ jest określone jako $\bigcup\{O \in \tau : O \subseteq A\}$: jest to największy element τ zawarty w X .

Domknięcie Cl_X jest definiowane jako $Int(X \setminus A)$, co można wyrazić w postaci $\bigcap\{X \setminus O : O \cap A = \emptyset\}$ i można ująć jako najmniejszy zbiór postaci $X \setminus O$ dla O rozłącznego z X .

Zbiory z rodziny τ nazywa się zbiorami otwartymi, dopełnienia zbiorów otwartych to zbiory domknięte. Zatem Cl_X jest przecięciem wszystkich zbiorów domkniętych zawierających X , wnętrze Int_X jest sumą wszystkich zbiorów otwartych zawartych w X . Różnica $Cl_A \setminus Int_A$ jest ograniczeniem A oznaczanym symbolem Fr_A . Jest to zbiór wszystkich elementów x przestrzeni X mających tę własność, że każdy zbiór otwarty zawierający x przecina A oraz $X \setminus A$.

Morfologia matematyczna [Soi04] przeniosła te operatory do sfery praktycznej – miejsce rodziny zbiorów otwartych zajął jeden zbiór S zwany elementem strukturalnym. Z jego pomocą wykonuje się operacje wnętrza i domknięcia. Warto przyjrzeć im się dokładniej.

Morfologiczne operacje otwierania i domykania

W formie pomocniczej warto rozważyć najpierw operacje erodowania (*ang. erosion*, Rysunek 3.18) i dylacji (*ang. dilation*, Rysunek 3.19).



(a) Przykładowy obraz.

(b) Przykładowy obraz po zastosowaniu operacji erozji morfologicznej.

Rysunek 3.18: Przykład zastosowania operacji erozji morfologicznej na obrazie monochromatycznym z wykorzystaniem kwadratowego elementu strukturalnego. Grubość wraz z rozmiarem obiektu pierwszoplanowego uległy zmniejszeniu. Wersja elektroniczna obrazu: <https://krzywicki.pro/fig-3-18>.



(a) Przykładowy obraz.

(b) Przykładowy obraz po zastosowaniu operacji dylacji morfologicznej.

Rysunek 3.19: Przykład zastosowania operacji dylacji morfologicznej na obrazie monochromatycznym z wykorzystaniem kwadratowego elementu strukturalnego. Grubość wraz z rozmiarem obiektu pierwszoplanowego uległy zwiększeniu. Wersja elektroniczna obrazu: <https://krzywicki.pro/fig-3-19>.

Są one wykonywane za pomocą elementu strukturalnego S – ustalonego obiektu o wybranym kształcie. Element strukturalny jest obrany w kontekście

obrazu lub sygnału. Typowe elementy w przypadku dwuwymiarowej przestrzeni metrycznej to dysk, sześciokąt, kwadrat, krzyż oraz segment liniowy. Jest istotne, by element strukturalny miał wskazany środek s .

Operacja erodowania oznaczona symbolem $E(S, \cdot)$ daje dla obiektu X nowy obiekt $E(S, X) = \{z \in X : S(z) \subseteq X\}$, gdzie $S(z)$ oznacza element strukturalny S usytuowany tak, że $s = z$. Jest to szczególny przypadek operatora wnętrza z jednym zbiorem otwartym S i warunkiem geometrycznym $s = z$.

Operacja dylacji to odpowiednik operacji domknięcia: $D(S, X) = \{z : S(z) \cap X \neq \emptyset\}$.

Jak można oczekiwać, erozja i dylacja są związane dualnością właściwą dla operatorów wnętrza i domknięcia: $E(S, X) = -D(S, -X)$, gdzie $-$ oznacza operację dopełnienia (komplementacji).

Operacja otwarcia (*ang. opening*) przypomina operator wnętrza: $O(S, X) = \bigcup_z \{S(z) : S(z) \subseteq X\}$. Dla operacji domknięcia należy użyć komplementacji: $C(S, X) = \bigcap_z \{-S(z) : X \subseteq -S(z)\}$.

Operacje cylindryczne

White top-hat (cylinder via otwieranie, lub skrótowo *Top-hat*: Rysunek 3.20) $W(S, X) = X \setminus O(S, X)$ oraz Black top-hat (cylinder via domykanie, lub skrótowo *Black-hat*: Rysunek 3.21) $B(S, X) = D(S, X) - X$ dają w sumie obszar wolny od elementu strukturalnego: $W(S, X) \cup B(S, X) = D(S, X) \setminus O(S, X)$.

Operacje cylindryczne stosowane są w wielu procedurach analizy obrazu lub sygnału. Przedstawione tutaj warianty są najprostszymi. Przykładowo, dla obrazów w skali szarości, powyższe formuły (ideowo analogiczne) mogą mieć bardziej złożoną postać.



(a) Przykładowy obraz.

(b) Przykładowy obraz po zastosowaniu operacji morfologicznej White top-hat.

Rysunek 3.20: Przykład zastosowania operacji morfologicznej White top-hat na obrazie monochromatycznym z wykorzystaniem kwadratowego elementu strukturalnego. Na obrazie wynikowym pozostały jedynie te fragmenty obiektu pierwszoplanowego, które były mniejsze powierzchniowo od elementu strukturalnego. Wersja elektroniczna obrazu: <https://krzywicki.pro/fig-3-20>.

Hit-or-miss

By zrozumieć ideę *hit-or miss*, warto cofnąć się do wersji topologicznej [Pol23]. Metafora *hit-or-miss* odnosi się do sytuacji, gdy dana jest rodzina podzbiorów \mathcal{A} przestrzeni topologicznej (X, τ) oraz rodziny \mathcal{P} i \mathcal{Q} podzbiorów X przy czym rodzina \mathcal{P} jest zamknięta na sumy skończone.

W tym kontekście, indukowana jest topologia $\tau_{\mathcal{P}, \mathcal{Q}}$ którą oznaczymy skrótem τ^* . Jej bazę otwartą stanowią zbiory postaci $\{A \in \mathcal{A} : A \cap P = \emptyset, A \cap Q_i \neq \emptyset\}$ dla pewnego $P \in \mathcal{P}$ oraz dla $Q_1, Q_2, \dots, Q_k \in \mathcal{Q}$.

Morfologia czerpie z tej konstrukcji, w przypadku obrazu binarnego, dwa elementy strukturalne, przedni S^+ i tylny S^- , które są rozłączne.

Z pomocą elementów S^+, S^- , powstaje operacja hit-or-miss: $HM(S^+, S^-, X) = \{z : S^+(z) \subseteq X, S^-(z) \subseteq -X\}$. Można zauważyć, że $HM(S^+, S^-, X) = E(S^+, X) \cap E(S^-, -X)$. Hit-or-miss pozwala także na skeletonizację obrazu [Soi04].



(a) Przykładowy obraz.

(b) Przykładowy obraz po zastosowaniu operacji morfologicznej Black top-hat.

Rysunek 3.21: Przykład zastosowania operacji morfologicznej Black top-hat na obrazie monochromatycznym z wykorzystaniem kwadratowego elementu strukturalnego. Na obrazie wynikowym pozostały jedynie te fragmenty obiektu pierwszoplanowego, które były mniejsze powierzchniowo od elementu strukturalnego oraz ciemniejsze od otoczenia. Wersja elektroniczna obrazu: <https://krzywicki.pro/fig-3-21>.

W przypadku operacji na obrazach w skali szarości, definicje operacji morfologicznych należy zmodyfikować by uwzględnić rozkłady stopni szarości w obrazie i w elemencie strukturalnym.

Przyjmijmy, że dla pikseli (x, y) w obrazie jednowarstwowym i pikseli (u, v) w elemencie strukturalnym, intensywności szarości są wyrażone, odpowiednio, jak $p(x, y)$ i $q(u, v)$. Wynikające wartości erozji, dylacji, otwierania i domykania, są przedstawione, odpowiednio, przez funkcje intensywności $E(x, y)$, $D(x, y)$, $O(x, y)$, $C(x, y)$: określone następująco:

$$(E) \quad E(x, y) = \min\{p(x + u, y + v) - q(u, v)\},$$

$$(D) \quad D(x, y) = \max\{p(x - u, y - v) + q(u, v)\},$$

$$(O) \quad O(x, y) = D(q(u, v), E(x, y)),$$

$$(C) \quad C(x, y) = E(q(u, v), D(p(x, y), q(u, v))).$$

Złożoność operacji morfologicznych

Literatura dostarcza pewnych danych o złożoności czasowej. Dane te są zależne od rodzaju obiektu poddawanego operacjom, od elementu strukturalnego i od rodzaju zadania. Wystarczy tu odnieść się do operacji erozji i dylacji, gdyż one określają pozostałe operacje.

Podstawowa złożoność operacji morfologicznych wynosi $O(n \cdot s)$ gdzie n jest rozmiarem obrazu w pikselach, a s jest rozmiarem elementu strukturalnego. W wyniku optymalizacji można uzyskać złożoności $O(n \cdot s^{\frac{1}{2}})$, $O(n \cdot \log s)$, czy nawet $O(n)$ [APHB07].

Z powodu różnic w wartościach złożoności uzyskanych dla różnych algorytmicznych organizacji operacji morfologicznych, złożoność czasową podaje się zazwyczaj w sekundach czasu wykonania operacji.

Złożoność operacji morfologicznych zależy od algorytmu użytego do wykonania operacji. Géraud, Talbot i Van Droogenbroeck [GTD13] podają kilka typów algorytmów i dla nich wykazują czas wykonania rekonstrukcji geodezyjnej z użyciem dylacji obrazu Lena [Mun96] o rozmiarze 512x512 pikseli w odcieniach szarości z topologią 4 sąsiadów. Ich wyniki przedstawiono w Tabeli 3.1.

LP	Typ algorytmu	Czas wykonania [s]
1	parallel	25.28
2	sequential	3.18
3	queue-based	0.65
4	hybrid	0.34
5	union-find	0.34

Tabela 3.1: Czas wykonania rekonstrukcji geodezyjnej obrazu Lena [Mun96] na podstawie pracy [GTD13].

Autorzy pracy [GTD13] podają przykład wykonania dylacji obrazu binarnego X z elementem strukturalnym S za pomocą splotu $X * S$ i szybkiego przekształcenia Fouriera. Operacja wykonuje się w czasie stałym względem S .

O złożoności decyduje też wybór elementu strukturalnego. W pracy Fanga i innych [FWWL20], poświęconej konstrukcji adaptacyjnego

elementu strukturalnego, autorzy określili złożoność erozji i dylacji dla złożonego obrazu binarnego (butterfly) oraz dla obrazu Lena w skali szarości dla wybranych elementów strukturalnych. Wyniki czasowe operacji przedstawiono w Tabelach 3.2 i 3.3.

LP	Element strukturalny	Czas – erozja [s]	Czas – dylacja [s]
1	Krzyż	0.081	0.097
2	Dysk	0.094	0.099
3	Adaptacyjny	3.519	3.698

Tabela 3.2: Czas operacji erozji i dylacji na obrazie binarnym dla trzech wariantów elementu strukturalnego.

LP	Element strukturalny	Czas – erozja [s]	Czas – dylacja [s]
1	Krzyż	0.388	0.221
2	Dysk	0.256	0.239
3	Adaptacyjny	2.271	2.308

Tabela 3.3: Czas operacji erozji i dylacji na obrazie Lena dla trzech wariantów elementu strukturalnego.

Można stwierdzić, że dysk jest optymalnym wyborem elementu strukturalnego.

Izmantoko i Choi [IC13] podają wyniki badań bliskich tematycznie badaniom omówionych w Rozdziale 5 Rozprawy – autorzy badali skany rezonansu magnetycznego (*ang. magnetic resonance*, MR) mózgu. Przedstawiony w pracy [IC13] obraz mózgu był aksjalnym MR obrazem w trzech klasach rozdzielczości: 512x512, 1024x1024 oraz 2048x2048. Obraz w rozdzielczości 512x512 był oryginalnie uzyskanym, pozostałe rozdzielczości zostały uzyskane w drodze interpolacji dwuliniowej.

W testach autorzy badania wykorzystali procesor CPU *Intel Xeon 2.6 GHz* oraz procesor graficzny *NVIDIA Quadro 56*. Wyniki złożoności przedstawiono w Tabelach 3.4, 3.5 oraz 3.6, gdzie można zauważyć przyspieszenie czasu trwania obliczeń w stosunku do CPU.

Rodzaj operacji / jądro operacji	CPU [ms]	naive [ms]	global [ms]	shared [ms]
Erozja	89.58	11.35	32.99	3.10
Dylacja	73.84	20.95	20.70	21.31
Otwarcie	159.74	32.33	53.76	24.03
Domknięcie	159.66	35.42	54.61	24.03

Tabela 3.4: Wyniki złożoności czasowej czterech wariantów operacji morfologicznych na obrazach o rozmiarze 512x512 px w trzech wariantach przyspieszenia.

Rodzaj operacji / jądro operacji	CPU [ms]	naive [ms]	global [ms]	shared [ms]
Erozja	326.55	44.71	130.58	11.22
Dylacja	289.74	76.81	81.95	80.84
Otwarcie	621.27	122.08	212.50	92.14
Domknięcie	618.40	122.09	212.58	92.20

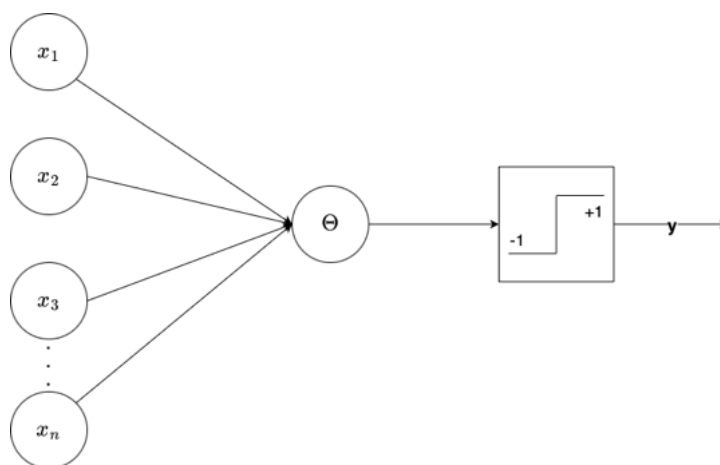
Tabela 3.5: Wyniki złożoności czasowej czterech wariantów operacji morfologicznych na obrazach o rozmiarze 1024x1024 px w trzech wariantach przyspieszenia.

3.5.2 Sztuczne sieci neuronowe

Sztuczne sieci neuronowe emulują układy neuronów fizjologicznych i ich struktura jest modelowana jako struktura neuronu neurofizjologicznego [yC91]. Wprowadzenie przez Alana M. Turinga modelu *maszyny Turinga* inspirowało do tworzenia konkurencyjnych modeli obliczeń i jednym z nich był, stworzony przez Warrena McCullocha i Waltera Pittsa model sztucznego neuronu, nazwany *neuronem McCullocha-Pittsa* [MP43] (Rysunek 3.22).

Rodzaj operacji / jądro operacji	CPU [ms]	naive [ms]	global [ms]	shared [ms]
Erozja	1226.74	189.30	527.90	45.16
Dylacja	1133.52	313.80	330.20	329.68
Otwarcie	2362.88	503.15	857.78	374.92
Domknięcie	2361.64	503.19	858.04	374.30

Tabela 3.6: Wyniki złożoności czasowej czterech wariantów operacji morfologicznych na obrazach o rozmiarze 2048x2048 px w trzech wariantach przyspieszenia.



Rysunek 3.22: Schemat sztucznego neuronu McCullocha-Pittsa. Wersja elektroniczna obrazu: <https://krzywicki.pro/fig-3-22>.

Miejsce somy neuronu zajmuje sumator który z wejść dla bitów x_1, x_2, \dots, x_n odpowiadającym dendrytom neuronu tworzy sumę $\sum_i x_i$. Potencjał aktywacyjny neuronu po przekroczeniu którego neuron wysyła sygnał poprzez swój akson, wyraża próg aktywacji Θ . Wynik obliczenia y na wyjściu neuronu jest dany jako $y = 1$, gdy $\sum_i x_i \geq \Theta$, $y = 0$, gdy $\sum_i x_i < \Theta$.

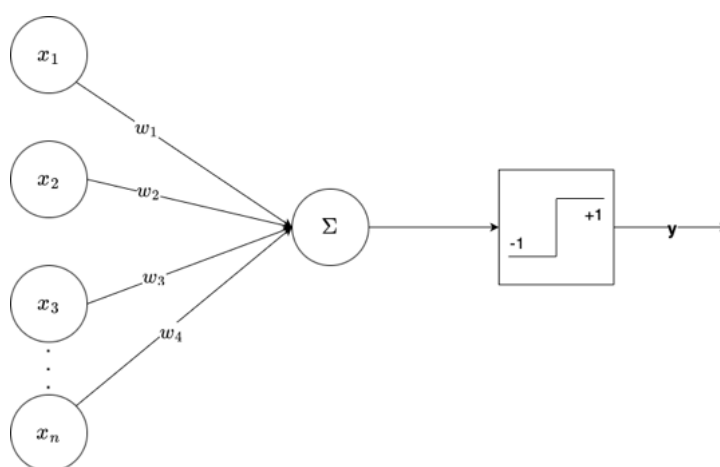
Od czasu Santiago Ramóna y Cajala (około 1880 roku) wiadomo, że neurony w tkankach organizmów są izolowane, więc sztuczne neurony tworzą sieci neuronów połączonych w sposób następujący: wyjścia neuronów danej warstwy są wejściami dla neuronów kolejnej warstwy.

Sieć neuronowa to zbiór warstw L_1, L_2, \dots, L_k , przy czym warstwa L_1 to warstwa wejściowa (*ang. input layer*), warstwa L_k to warstwa wyjściowa (*ang. output layer*), warstwy L_2, \dots, L_{k-1} to warstwy ukryte (*ang. hidden layers*). Każda warstwa L_i składa się ze skończenie wielu neuronów, które w tej warstwie nie są połączone. Każdy neuron warstwy L_i dla $i = 1, 2, \dots, k - 1$ jest połączony z każdym neuronem warstwy L_{i+1} . W tej architekturze dokonują się obliczenia, dla danego wektora $\bar{\Theta}$ progów aktywacji w neuronach sieci, dla każdego wektora \bar{x} bitów wejściowych, sieć oblicza wartość $f_{\bar{\Theta}}(\bar{x}) = \bar{y}$. Sieć neuronowa oblicza lub wyucza się funkcji logicznej (boolowskiej) f jeśli $f = f_{\bar{\Theta}}$.

Sieci neuronów McCullocha-Pittsa mają znaczące zdolności obliczeniowe – dla każdej funkcji Boolowskiej o skończenie wielu literałach, istnieje sieć $N(f)$ neuronów McCullocha-Pittsa, która oblicza funkcję f .

Sieci o postaci $N(f)$ nie mają zdolności uczenia. Pewną adaptacyjność można zapewnić jednak za pomocą kontrolowanej w ramach pewnej strategii zmiany progów aktywacji.

Istotne wzbogacenie zdolności uczenia się przez sieć przyniósł *perceptron* [Ros58] (Rysunek 3.23) i *sieci perceptronów*. Neuron perceptronowy (p-neuron) jest rozszerzonym neuronem McCullocha-Pittsa, gdzie autorzy wprowadzili dodatkowo wagi w postaci liczb rzeczywistych na wejściach bitów.



Rysunek 3.23: Schemat perceptronu. Wersja elektroniczna obrazu: <https://krzywicki.pro/fig-3-23>.

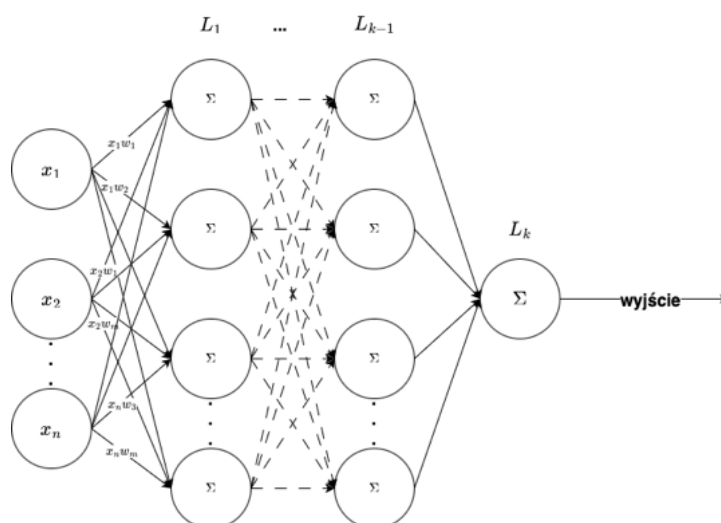
Reguła obliczeń dla p-neuronu jest następująca: $y = 1$, gdy $\sum_i w_i \cdot x_i \geq \Theta$, $y = 0$, if $\sum_i w_i \cdot x_i < \Theta$.

Perceptron wielowarstwowy (sieć złożona z perceptronów, Rysunek 3.24) jest obecnie jednym z najpopularniejszych podejść do tworzenia sztucznych sieci neuronowych.

Uczenie maszynowe

Poruszenie tematyki uczenia maszynowego (*ang. Machine Learning, ML*) dostarczy języka do dalszej dyskusji.

Podstawowym terminem jest koncept (pojęcie). Gdy dany jest zbiór U obiektów nazywany uniwersum, to koncepty są definiowane jako podzbiory



Rysunek 3.24: Schemat perceptronu wielowarstwowego. Wersja elektroniczna obrazu: <https://krzywicki.pro/fig-3-24>.

uniwersum.

Perceptron o wektorze wag \bar{w} , $\bar{w} = [w_1, w_2, \dots, w_k]$ działa na przestrzeni wektorowej \mathbb{R}^k . Rozmaitość liniowa Q o równaniu $\sum_i w_i \cdot x_i = \Theta$, nazywana rozmaitością separującą wyznacza półprzestrzenie $Q^+(\bar{x} = \{x : \sum_i w_i \cdot x_i \geq \Theta\})$ oraz $Q^- = \{x : \sum_i w_i \cdot x_i < \Theta\}$.

Elementy uniwersum U są dla celów uczenia zakodowane jako wektory $[x_1, x_2, \dots, x_k]$ (wybór k jest podyktowany naturą obiektów w U). Symbol U^c to zbiór kodów obiektów w U . Uczenie p-neuronu oznacza wyuczenie danego konceptu $C \subseteq U$.

Uczenie maszynowe używa pojęcia próbki (*ang. sample*) S . P-neuron dokonuje klasyfikacji binarnej. Podstawą dla procedury klasyfikacji jest próba testowa S , która jest skończonym zbiorem wektorów ze zbioru U^c . Próba S jest sumą dwu rozłącznych podzbiorów S^+ i S^- . Symbolem S^+ oznaczymy te elementy S , które należą do pojęcia C , są to przykłady pozytywne dla C . Elementy zbioru S^- do zbioru C nie należą, są to przykłady dla C negatywne.

Niech $Q^+(S)$ oznacza te elementy próby S , które należą do Q^+ , a $Q^-(S)$ pozostałe elementy w S . Perceptron P o zbiorze progów $\bar{\Theta}$ i zbiorze wag \bar{w} indukuje funkcję $f_{\bar{\Theta}, \bar{w}}$. W przypadku standardowym, progi nie ulegają modyfikacjom, a jedynie wagi. Stąd przyjmując, że progi są ustalone, $f_{\bar{w}}$

będzie oznaczała funkcję perceptronu.

Perceptron wyuczył się konceptu C na podstawie próby S , jeżeli $S^+ = Q^+(S)$ i $S^- = Q^-(S)$. Istnieją dwa rodzaje błędnej klasyfikacji Błąd typu I oznacza, że element w S^+ zaliczony został do zbioru $Q^-(S)$. W przypadku błędu typu II element w S^- został zaliczony do zbioru $Q^+(S)$.

Twierdzenie o uczeniu perceptronu [Nov62, Ros58] orzeka, że dla konceptu C wyuczalnego, czyli takiego, że pewna rozmaitość liniowa oddziela elementy C od elementów zbioru $U \setminus C$, perceptron zaczynając klasyfikację próby S pd niezerowego losowego zbioru wag, w skończenie wielu krokach poprawnie sklasyfikuje wszystkie elementy próby czyli wyuczy się konceptu C na podstawie próby S [Roj96].

Algorytm uczący polega na zmianie wag w przypadku błędu w klasyfikacji. Dla błędu typu I na obiekcie \bar{x} , $\bar{w}_new \leftarrow \bar{w} + \bar{x}$, a w przypadku błędu typu II, $\bar{w}_new \leftarrow \bar{w} - \bar{x}$. Twierdzenie jest nieefektywne – liczba kroków do uzyskania poprawnej klasyfikacji jest zawsze skończona, ale zależy ona od rozkładu elementów S , dokładniej odległości η zbioru $Q^+(S)$ od zbioru $Q^-(S)$ określonej jako $\min_{x,y} \{|x - y| : x \in Q^+(S), y \in Q^-(S)\}$. Liczba kroków procedury jest proporcjonalna do $\frac{1}{\eta}$.

Kolejny krok w celu zwiększenia efektywności uczenia/klasyfikacji przez sieci neuronowe to zastosowanie procedury optymalizacji wielostopniowej podczas modyfikacji wag sieci jako procedury propagacji wstecznej błędu (*ang. backpropagation*) [RHW86]. Dla sieci neuronowej o warstwach L_1, L_2, \dots, L_k , realizowane jest uczenie nadzorowane (*ang. supervised learning*). Dla danego wektora wejściowego \bar{x} , przekazywany jest oczekiwany wektor wyjściowy \bar{t} . Dla danego zbioru wag na połączeniach neuronów w sieci \bar{w} , określany jest błąd obliczenia (Równanie 3.18).

$$E(\bar{w}) = \frac{1}{2} \cdot \sum_j (y_j - t_j)^2 \quad (3.18)$$

Celem jest zmniejszanie błędu poprzez zmianę wag za pomocą metody optymalizacji gradientowej [Soy20], gdzie *metoda spadku wzdłuż gradientu* [RM51, KW52] (*ang. gradient descent*) jest jednym z najpopularniejszych rozwiązań. W celu zastosowania *metody spadku wzdłuż gradientu* należy różniczkować funkcję E co oznacza, że funkcje aktywacji neuronów w sieci

muszą być różniczkowalne. Należy odejść więc od skokowych (Heaviside'a) funkcji aktywacji i od logiki progowej (*ang. threshold logic*) do logiki funkcji aktywacji, przykładowo sigmoidalnych, określonych w przypadku najprostszym w Równaniu 3.19. Funkcje te mogą być modyfikowane parametrycznie. Podobniekształtne funkcje uzyskuje się za pomocą funkcji *tangensa hiperbolicznego* (Równanie 3.20).

$$f(x) = \frac{1}{1 + e^{-x}} \quad (3.19)$$

$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (3.20)$$

Dla neuronu n_{ij} - j-tego neuronu w warstwie L_i , sygnał z_{ij} wchodzący do n_{ij} od neuronów warstwy poprzedniej ($i > 1$), jest przekształcany w sygnał wychodzący do każdego neuronu warstwy następnej L_{i+1} o wartości $f_{ij}(z_{ij})$. Dla ustalonych funkcji $\{f_{ij}\}$ dla wszystkich neuronów w $\cup_{i < k} L_i$, funkcja błędu E jest różniczkowalną funkcją wag.

Niech w_{jk}^i oznacza wagę połączenia neuronu n_{ij} z neuronem $n_{i+1,k}$. Gradient funkcji $E(\bar{w})$ jest wektorem $[\frac{\partial E}{\partial w_{jk}^i}]_{i,j,k}$. Reguła łańcuchowa (*ang. chain rule*) w rachunku różniczkowym pozwala na kolejne różniczkowania w przypadku funkcji złożonych, w efekcie błąd jest propagowany od warstwy wyjściowej do warstwy przedostatniej i kolejno od warstwy do warstwy poprzedzającej.

Reguły zmiany wag są następujące. $w_{new} \leftarrow w - \eta \cdot \frac{\partial E}{\partial w}$ gdzie $\frac{\partial E}{\partial w}$ jest pochodną cząstkową E względem w . Sedno propagacji wstecznej polega na tym, że pochodne cząstkowe są obliczane retrorekurencyjnie z warstwy na warstwę [Roj96].

Złożoność jednej iteracji algorytmu propagacji wstecznej zależy od złożoności mnożenia macierzy która w podstawowym oszacowaniu wynosi $O(n^3)$ dla macierzy o rozmiarze $n \times n$. *Algorytm Strassena* [Str69] zmniejsza złożoność do $O(n^{\log n})$. Najlepsze oszacowanie to $O(n^{2.37188})$ uzyskane w pracy Duana i innych [DWZ22]. Analogicznie można określić złożoność operacji mnożenia macierzy prostokątnych. Całkowita złożoność operacji propagacji wstecznej zależy od liczby iteracji algorytmu i można ją określić jako $O(N \cdot M^3)$ gdzie N jest liczbą iteracji i M jest maksymalnym wymiarem wierszowym lub kolumnowym dla macierzy mnożenia aktywnych w algorytmie.

Metoda propagacji wstecznej pozwala na zmniejszenie błędu dla sygnału wejściowego w drodze iteracji algorytmu. Warto jednak zwrócić uwagę, że podczas procesu klasyfikacji konceptu (jak w przypadku perceptronu), próbe dla konceptu przedstawia się jako sumę zbiorów przykładów pozytywnych i przykładów negatywnych, a zmiana wag dokonuje się w wyniku błędu w klasyfikacji. Błąd jest zatem funkcją rozmiaru próby i złożoności konceptu.

Paradygmat PAC uczenia

Formalizację *PAC* (Probably Approximately Correct) [Val84] warto zacząć od ustalenia środowiska obliczeniowego. Niech X będzie zbiorem instancji konceptu C oraz $Y = \{0, 1\}$ – zbiorem sygnatur instancji. W kontekście klasyfikacji binarnej koncept C reprezentuje funkcja $I_C : X \rightarrow Y$ i $C = I_C^{-1}(1)$. Jako przedmiot uczenia podana jest klasa \mathcal{C} konceptów nad X . Niech na zbiorze X określony będzie rozkład prawdopodobieństwa P . System uczący się otrzymuje wylosowaną zgodnie z P próbe $S = \{x_1, x_2, \dots, x_k\}$ oraz indeksację $\{I_C(x_1), \dots, I_C(x_k)\}$ dla konceptu C .

System uczący się rozważa zbiór H hipotez celem wyboru hipotezy $h_C \in H$ o możliwie małym błędzie. Błąd uczenia dla próby S i hipotezy h to błąd empiryczny $Err(h) = \frac{1}{k} \cdot |\{x_i : h(x_i) \neq I_C(x_i)\}|$.

Niech A będzie algorytmem uczenia, a $h(S)$ hipotezą otrzymaną po wykonaniu na S algorytmu A . Algorytm A jest PAC-algorytmem uczenia dla podanego powyżej kontekstu, jeśli istnieje wielomian p taki, że:

dla każdego wyboru parametrów $\varepsilon, \delta > 0$, dla każdego rozkładu prawdopodobieństwa P na zbiorze X i dla każdego konceptu $C' \in \mathcal{C}$, dla każdej liczebności próby $|S| \geq p(\frac{1}{\varepsilon}, \frac{1}{\delta}, \max_{x \in X} |x|, |\mathcal{C}|)$, jest spełniony warunek:

$[Prob(Err(h(S)) > \varepsilon) < \delta$ oraz złożoność czasowa algorytmu A wynosi $O(p(\frac{1}{\varepsilon}, \frac{1}{\delta}, \max_{x \in X} |x|, |\mathcal{C}|))$, gdzie $|x|$ jest rozmiarem reprezentacji instancji $|x|$ oraz $|\mathcal{C}|$ jest maksymalnym rozmiarem reprezentacji konceptu z \mathcal{C} .

Liczba $p(\frac{1}{\varepsilon}, \frac{1}{\delta}, \max_{x \in X} |x|, |\mathcal{C}|)$ jest złożonością próby (*ang. sample complexity*).

W szczególnym przypadku skończonego zbioru H hipotez, złożoność próby spełnia warunek [Che19]:

$$|S| \geq \frac{1}{\epsilon} \cdot (\log|H| + \log\frac{1}{\delta})$$

W przypadku nieskończonego zbioru $|H|$ hipotez, jest możliwe uczenie PAC, co pokazują znane przykłady uczenia pojęcia prostokąta lub kwadratu na płaszczyźnie [Che19]. Istotnym czynnikiem analizy w tym przypadku jest ocena złożoności zbioru hipotez. Spośród kilku metod, *wymiar Vapnika-Chervonenkisa* jest wybierany jako najprostsza miara tej złożoności.

Wymiar Vapnika-Chervonenkisa

Powiemy, że zbiór H rozbija (*ang. shatters*) skończony zbiór m elementów S , jeśli dla każdego podzbioru $Y \subseteq S$ istnieje hipoteza $h \in H$ o własności $h \cap S = Y$. Zatem, zbiór H rozróżnia wszystkie koncepty w S . Symbolem $\Pi_H(m)$ oznacza się liczbę podzbiorów zbioru m -elementowego rozróżnionych przez zbiór H .

Wymiar Vapnika-Chervonenkisa [VC15, Vap00] (dim_{VC} , wymiar VC) jest określony następująco (Równanie 3.21):

$$dim_{VC}(H) = \max\{m : \Pi_H(m) = 2^m\} \quad (3.21)$$

Zatem, $dim_{VC}(H) = m$ oznacza, że pewien zbiór m -elementowy jest rozbity przez H ale żaden zbiór o mocy większej od m nie jest rozbity przez H . Kombinatoryczne własności dim_{VC} wyraża lemat Sauera [Sau72]: jeśli $dim_{VC} = m$ to dla każdej liczby naturalnej n : $\Pi_H(n) \leq \sum_{i=0}^m \binom{n}{i}$.

$\Pi_H(m)$ jest nazywana funkcją wzrostu. Lemat Sauera [Sau72] implikuje, że złożoność funkcji wzrostu jest $O(m^{dim_{VC}(H)})$ [Che19]. Zatem skończony wymiar VC implikuje, że funkcja wzrostu ma złożoność wielomianową, jeśli $dim_{VC}(H) = \infty$, to $\Pi_H(m) = 2^m$ dla każdego m , więc złożoność jest wykładnicza.

Wynik w pracy Ehrenfeuchta, Hausslera, Kearnsa i Valianta [EHKV89], wiąże wymiar VC i złożoność próby dla uczenia klasy konceptów \mathcal{C} (Równanie 3.22).

$$m = \Omega\left(\frac{1}{\varepsilon}\left(\ln\frac{1}{\delta} + \dim_{VC}(\mathcal{C})\right)\right) \quad (3.22)$$

Zatem, uczenie PAC jest możliwe tylko w przypadku skończonego wymiaru VC .

Wymiar VC dla sieci neuronowych

Wymiar VC dla sieci neuronowej omówili Barlett i Maass w pracy [BM03].

W najprostszym przypadku neuronu McCullocha-Pittsa N o n bitach wejściowych, $\dim_{VC}(N) = n + 1$. Sieć neuronów N McCullocha-Pittsa o n wagach spełnia warunek $\dim_{VC}(N) = O(n \cdot \log n)$ [BH89].

Uogólnieniem sieci McCullocha-Pittsa są sieci, w których funkcje aktywacji neuronów są kawałkami wielomianowe, wymiar VC takich sieci spełnia równanie $\dim_{VC}(N) = O(w \cdot k \cdot \log w + w \cdot k^2)$ gdzie k jest liczbą warstw, w jest liczbą wag, funkcje aktywacji są ustalone i każda z tych funkcji ma ustaloną liczbę wielomianów [BMM98].

Na koniec pozostają sieci binarne o sigmoidalnych funkcjach aktywacji. Jest dla nich znany wynik $\dim_{VC}(N) = O(t^4)$, gdzie t oznacza liczbę parametrów sieci [KM97].

Powyższy przegląd odnosił się do sieci nad alfabetem $\{0,1\}$. Sieci nad alfabetem R liczb rzeczywistych R tworzą oddzielną klasę. Tutaj powstaje nowy rodzaj wymiaru VC . Niech X oznacza zbiór, a F rodzinę funkcji z X w R wraz ze skończonym podzbiorem $D \subseteq X$ o k elementach: $D = \{x_1, x_2, \dots, x_k\}$. Dla liczby rzeczywistej $\gamma > 0$, rodzina F γ -rozbija D jeśli istnieją liczby rzeczywiste r_1, r_2, \dots, r_k o tej własności, że dla każdego ciągu $q \in \{0, 1\}^k$ istnieje funkcja $f_q \in F$ taka, że:

$$f_q(x_i) \geq r_i + \gamma \text{ w przypadku gdy } q_i = 1; f_q(x_i) \leq r_i - \gamma \text{ w przypadku gdy } q_i = 0 \quad (3.23)$$

W takim przypadku (Równanie 3.23) określamy $\text{fat}(\gamma) - \text{dim}_{VC}$ jako największą liczebność zbioru γ -rozbitego przez F .

Dla funkcji rzeczywistych realizujących logikę progową, występuje twierdzenie [AB99]: jeśli rodzina F ma skończony wymiar fat-dim_{VC} i $(x_1, y_1), \dots, (x_k, y_k)$ jest próbą z $X \times [0, 1]$, to istnieją stałe c_1, c_2 takie, że dla $m \geq \frac{c_1(\text{fat-dim}_{VC}(F) \cdot (c_2 \cdot \varepsilon) \cdot (\log^2 \frac{1}{\varepsilon}) + \log \frac{1}{\delta})}{\varepsilon^2}$, prawdopodobieństwo wynosi $\geq 1 - \delta$ tego, że każda funkcja $f \in F$ dla której wartość średnia z próby odbiega o co najwyżej $\frac{1}{m^{\frac{1}{2}}}$ od minimum wartość średniej dla F , spełnia warunek (Równanie 3.24).

$$E(f(x) - y)^2 \leq \inf_{g \in F} [E(g(x) - y^2)] + \varepsilon \quad (3.24)$$

gdzie, E oznacza wartość oczekiwaną.

Wymiar VC i złożoność dla głębokich sieci neuronowych

Barlett i Maass [BM03] wskazali, że oszacowanie $O(t^4)$ dla wymiaru Vapnika-Chervonenkisa sieci z sigmoidalnymi funkcjami aktywacji jest prawdziwe również dla głębokich sieci neuronowych (*ang. Deep Neural Network, DNN*), gdzie t jest liczbą wolnych parametrów.

Basu i inni [BMK⁺18] uzyskali oszacowanie dla wymiaru VC splotowych sieci neuronowych (*ang. Convolutional Neural Network, CNN*) [AMAZ17] w postaci (Równanie 3.25)

$$O\left(\frac{m^4 k^4 s^{2l-2}}{l^2}\right) \quad (3.25)$$

gdzie:

- m – liczba map cech,

- k – rozmiar jądra,
- s – współczynnik operacji podpróbkiowania (*ang. subsampling*),
- l – liczba warstw.

W pracy Maji i Mullinsa [MM18] można znaleźć metody przyspieszenia dla CNN.

3.6 Nowoczesne metody uczenia maszynowego i rozpoznawania obrazu

W tym podrozdziale przedstawiono nowoczesne metody uczenia maszynowego i rozpoznawania obrazu. Omówiono trzy wybrane techniki: uczenie transferowe, które umożliwia przenoszenie wiedzy z jednego modelu na inny w celu efektywnego rozwiązywania różnych zadań, uczenie zespołowe, w którym łączone są wyniki wielu modeli bazowych celem uzyskania dokładniejszych wyników, oraz metody detekcji i lokalizacji obiektów na obrazach z naciskiem na *YOLO* – architekturę sieci neuronowej.

3.6.1 Uczenie transferowe

Uczenie transferowe (*ang. transfer learning*) [YXL22] to technika uczenia sieci neuronowych polegająca na wykorzystaniu wiedzy zdobytej podczas treningu celem rozwiązania jednego problemu do rozwiązywania także innych zadań. W przeciwieństwie do sieci neuronowych tradycyjnie wyuczonych od podstaw, co wymaga znacznych zasobów obliczeniowych i czasowych, *uczenie transferowe* pozwala znacząco zredukować te koszty poprzez wykorzystanie wcześniej wyuczonych modeli jako punktu wyjścia dla nowych zadań.

Główną ideą uczenia transferowego jest fakt, że wiedza nabyta przez sieć neuronową (model) podczas uczenia się jednego zadania może być także przydatna w rozwiązywaniu podobnych zadań. W związku z tym, sieć przenosi swoją zdolność do rozpoznawania wzorców, wykrywania cech i generalizowania na nowe dane, co może znacząco przyspieszyć proces treningu i poprawić jego ogólną jakość.

W praktyce, *uczenie transferowe* może być realizowane na różne sposoby. Jednym z popularnych podejść jest *fine-tuning* (dostrojenie), który polega na modyfikacji i dotrenowywaniu wyuczonych warstw modelu na nowym zadaniu lub nowym zbiorze danych. W ten sposób, model jest w stanie dostosować się do specyficznych cech i wzorców występujących w nowym zbiorze danych. Warto mieć na uwadze fakt, że technika *fine-tuning* nie polega na pozbawieniu modelu dotychczas zdobytej wiedzy na rzecz nowej, lecz na sprawnym douczeniu rozwiązywania innych problemów na podstawie zarówno poprzedniej, jak i nowej wiedzy.

Innym popularnym podejściem w *uczeniu transferowym* jest ekstrakcja cech, gdzie wytrenowane warstwy modelu są wykorzystywane jako uniwersalne ekstraktory cech na potrzeby innego zadania. Proces ekstrakcji cech polega na wykorzystaniu wytrenowanych warstw w połączeniu z klasyfikatorem przystosowanym do innego zadania. Przykładowo, wytrenowane warstwy splotowe pod kątem klasyfikacji gatunków zwierząt mogą zostać wykorzystane do ekstrakcji cech obrazów przedstawiających różne rasy królików, a następnie do ich klasyfikacji za pomocą oddzielnego klasyfikatora. Warto mieć na uwadze fakt, że klasyfikator wykorzystujący proces ekstrakcji cech wymaga również treningu pod kątem rozwiązywania konkretnego problemu decyzyjnego, w przeciwieństwie do warstw realizujących ekstrakcję cech, które powinny pozostać niezmodyfikowane.

Należy jednak pamiętać, że mimo wszystkich zalet, *uczenie transferowe* ma swoje ograniczenia. Ważne jest, aby dane używane w dotrenowywaniu pod kątem nowego zadania miały zgodny charakter z danymi wykorzystanymi do trenowania modelu oryginalnego. Co więcej, niektóre zadania mogą wymagać bardziej specjalistycznego podejścia, co w przypadku *uczenia transferowego* nie zawsze będzie osiągalne.

3.6.2 Uczenie zespołowe

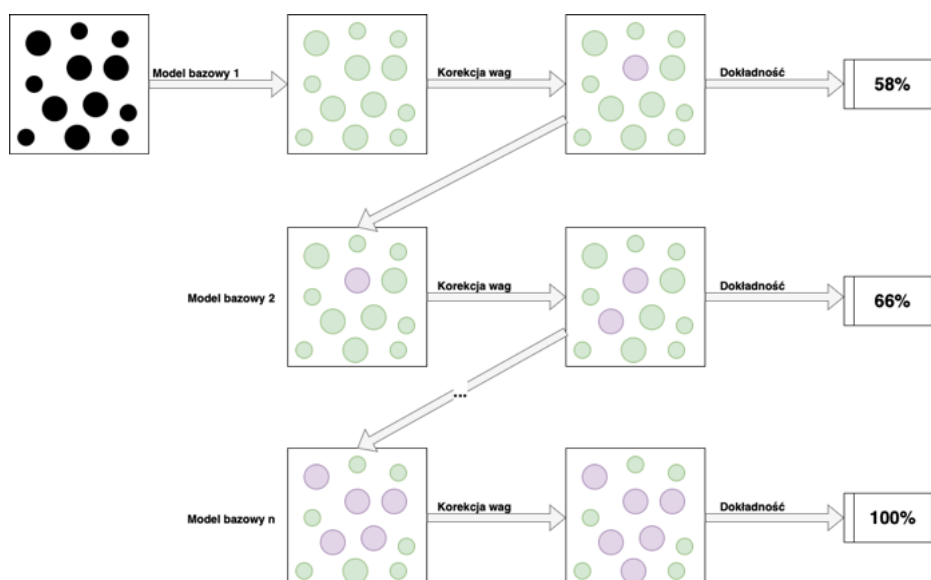
Uczenie zespołowe (*ang. ensemble learning*) [DYC⁺19] jest metodą uczenia maszynowego, która polega na łączeniu wyników kilku modeli bazowych w celu poprawy skuteczności predykcyjnej oraz ogólnej niezawodności. Główną ideą *uczenia zespołowego* jest założenie, że połączenie predykcji wielu słabszych modeli prowadzi do uzyskania bardziej efektywnego i dokładniejszego wyniku końcowego. Wśród metod *uczenia zespołowego*

warto wyróżnić trzy metody: *boosting* [Sch90], *bagging* [Bre96] oraz *pasting* [Bre99], gdzie każda z nich wykorzystuje unikalne podejście do wykorzystania wiedzy grupy modeli wytrenowanych niezależnie.

Boosting

Boosting [Sch90] to metoda uczenia zespołowego, która ma na celu poprawę skuteczności słabych modeli poprzez sekwencyjne skupienie się na próbkach, które zostały błędnie zaklasyfikowane przez modele wcześniejsze. Główną ideą metody *Boosting* jest przydzielanie większej wagi błędnie zaklasyfikowanym próbkom, co pozwala kolejnym modelom skoncentrować się na trudniejszych elementach danych. Proces *boostingu* polega na iteracyjnym trenowaniu serii modeli bazowych, gdzie każdy kolejny model stara się poprawić błędy swoich poprzedników. Ostateczna predykcja jest dokonywana przez połączenie głosów wszystkich modeli, biorąc także pod uwagę ich wagi. Wśród popularnych algorytmów bazujących na metodzie *boosting* mamy *AdaBoost* [FS95] oraz *Gradient Boosting Machines* [Fri01].

Schemat metody *Boosting* przedstawiono na Rysunku 3.25.



Rysunek 3.25: Schemat metody *Boosting*. Wersja elektroniczna obrazu: <https://krzywicki.pro/fig-3-25>.

Dla algorytmu *AdaBoost* złożoność w procesie treningu wynosi $O(n \cdot p \cdot n_{models})$, a predykcji $O(n \cdot n_{models})$

gdzie:

- n – liczba próbek treningowych,
- p – liczba atrybutów opisujących każdą z próbek,
- n_{models} – liczba modeli bazowych.

Bagging

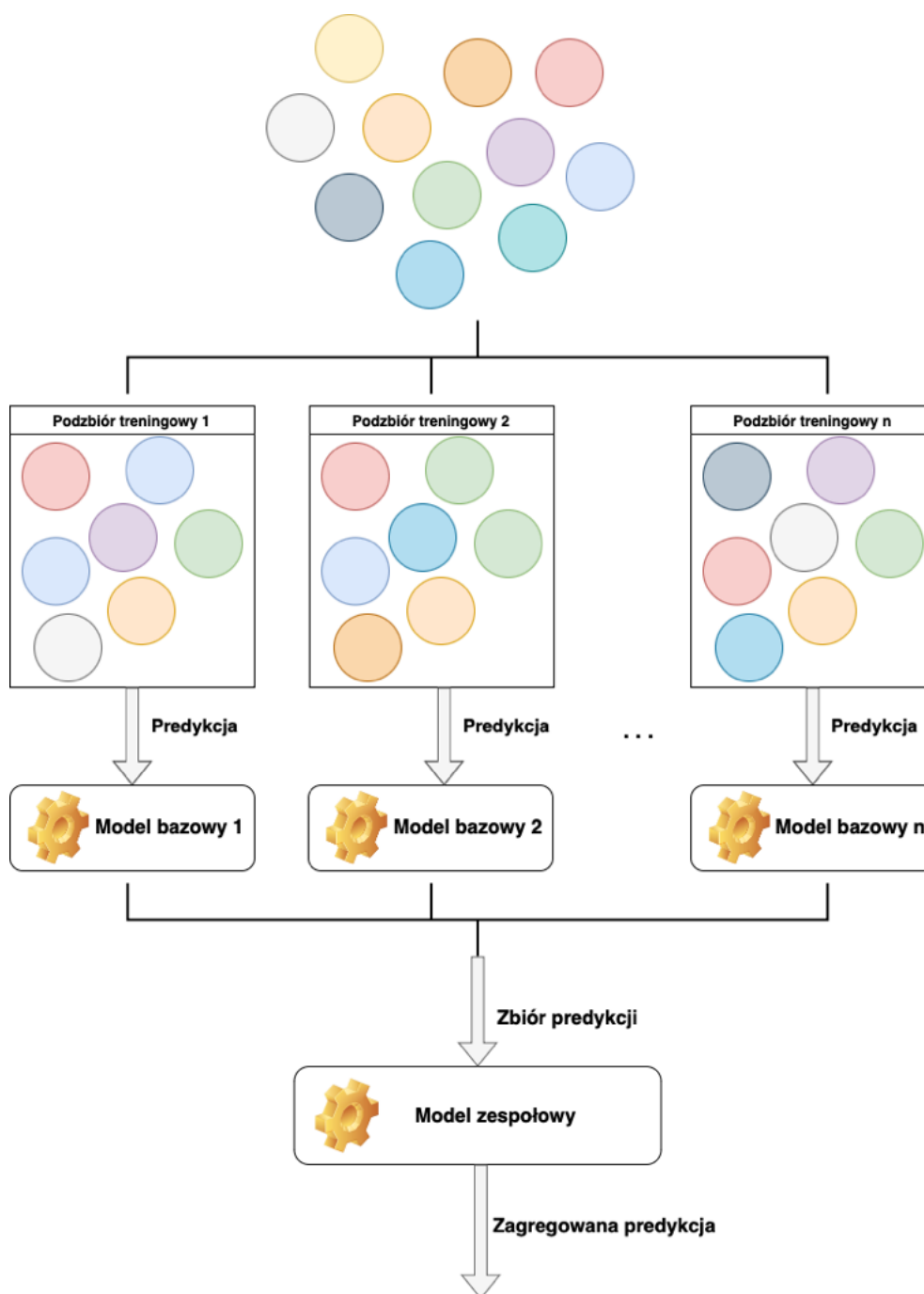
Bagging, (*ang. bootstrap aggregating*) [Bre96], jest metodą uczenia zespołowego, która ma na celu zmniejszenie nadmiernego dopasowania i wariancji predykcji (Podrozdział 3.4 w [Krz22]) zespołu modeli. *Bagging* tworzy kilka instancji tego samego modelu, trenując każdy z nich na pseudolosowo dobranym podzbiorze danych treningowych z powtórzeniami (bootstrapping). Po zakończonym procesie treningu modeli bazowych, każdy z nich samodzielnie dokonuje predykcji na nowych próbkach, a ostateczna predykcja jest dokonywana przez agregację wyników poszczególnych modeli – najczęściej przez głosowanie większościowe w przypadku problemów decyzyjnych lub uśrednianie dla zadań regresji. Jednym z najpopularniejszych algorytmów wykorzystujących metodę *bagging* jest las losowy (*ang. Random Forest*) [Bre01], który wykorzystuje drzewa decyzyjne jako modele bazowe.

Schemat metody *Bagging* przedstawiono na Rysunku 3.26.

Złożoność metody *lasu losowego* w procesie treningu wynosi $O(n_{trees} \cdot n \cdot \log(n) \cdot p)$, a predykcji $O(p \cdot n_{trees})$

gdzie:

- n – liczba próbek treningowych,
- p – liczba atrybutów opisujących każdą z próbek,
- n_{trees} – liczba drzew decyzyjnych będących modelami bazowymi.



Rysunek 3.26: Schemat metody Bagging. Wersja elektroniczna obrazu: <https://krzywicki.pro/fig-3-26>.

Pasting

Pasting [Bre99] to modyfikacja metody *bagging*, która również tworzy kilka instancji tego samego modelu trenowanych na podzbiorach niezawierających powtórzeń. Brak duplikatów sprawia, że każdy model otrzymuje unikalny fragment danych treningowych. Podobnie jak w metodzie *bagging*, każdy model dokonuje predykcji na nowych próbkach, a ostateczna predykcja jest dokonywana przez agregację wyników indywidualnych modeli. Metoda *pasting* może być przydatna w sytuacjach, gdy zbiór danych jest względnie duży (powyżej 2000 próbek) – powtarzanie instancji w podzbiorach nie jest wymagane. W przypadku mniejszych zbiorów danych metoda *bagging* z powtórzeniami może zapewnić lepszą różnorodność między modelami bazowymi.

3.6.3 Detekcja i lokalizacja obiektów

Detekcja obiektów na obrazach to jedno z zadań rozpoznawania obrazów, które pozwala na automatyczne rozpoznawanie różnych elementów w obrazach i wyznaczenie ich lokalizacji. Sieci neuronowe są obecnie najpopularniejszymi rozwiązaniami wykorzystywanymi w tym celu. Wśród najpopularniejszych architektur warto wyróżnić *You Only Look Once (YOLO)* [RDGF16] oraz *Single Shot MultiBox Detector (SSD)* [LAE⁺16].

YOLO [RDGF16] to architektura sieci neuronowej przeznaczona do detekcji obiektów w czasie rzeczywistym, która charakteryzuje się wysoką szybkością i wydajnością. *YOLO* traktuje detekcję obiektów jako problem lokalnej adaptywnej regresji, co oznacza że sieć neuronowa jednocześnie przetwarza całe zdjęcie i przewiduje prostokątne ramki otaczające wykrywane obiekty, a także przypisuje im odpowiednie klasy wraz z prawdopodobieństwem ich wystąpienia (współczynnik ufności).

Architektura *YOLO* w swojej pierwotnej wersji składała się z 24 bloków warstw splotowych oraz łączących, co umożliwiało ekstrakcję cech z obrazu dzięki zastosowanemu uczeniu transferowemu. Obecne wersje architektury *YOLO* [RKHD23] składają się z kilkuset bloków zawierających warstwy splotowe, łączące, próbkujące, rezydujące, itd.

Jedną z głównych zalet *YOLO* jest jej wydajność w czasie rzeczywistym, co czyni ją dobrym rozwiązaniem dla przypadków, w których wymagane jest szybkie i precyzyjne rozpoznawanie obiektów, takich jak systemy monitoringu, samochodowe systemy bezpieczeństwa, czy robotyka. Architektura *YOLO* ma także pewne ograniczenia. Jako sieć jednopoziomowa, może być mniej precyzyjna w przypadku detekcji bardzo małych obiektów. Dodatkowo, w niektórych przypadkach, sieć może błędnie interpretować obiekty, które nachodzą na siebie lub są bardzo zbliżone.

Architektura *YOLO* dzięki swojej popularności doczekała się serii nowych wersji [TCE23] zawierających szereg usprawnień, dzięki którym możliwe jest osiągnięcie dokładniejszych wyników przy stopniowym pozbywaniu się słabych stron znanych z pierwszych wersji.

3.7 Metody przyspieszania algorytmów

Przyspieszenie algorytmów jest procesem umożliwiającym efektywne wykorzystanie zasobów sprzętowych, pamięciowych, czasowych i energetycznych. W informatyce wykorzystywany jest szereg technik i metod przyspieszania algorytmów, które mogą być stosowane w zależności od problematyki realizowanej przez algorytm oraz dostępnego sprzętu.

Wśród metod przyspieszania algorytmów warto zwrócić uwagę na najpopularniejsze podejścia: wykorzystujące zasoby sprzętowe [PMMB22] oraz wykorzystujące podejścia związane z wykorzystaniem przeznaczonych do tego algorytmów i struktur danych [Vaz01, Wei95].

3.7.1 Rozwiązania wykorzystujące dedykowane algorytmy i struktury danych

Jedną z częstych przyczyn wolnego wykonywania się algorytmów są niewłaściwie dobrane struktury danych w implementacjach. Dobór odpowiednich struktur danych może mieć znaczący wpływ na szybkość wykonania algorytmów. Odpowiednio dobrana struktura danych może zmniejszyć czas dostępu do informacji, zwiększyć efektywność operacji wyszukiwania, dodawania czy usuwania elementów.

Często popełnianym błędem podczas implementacji algorytmów jest wybór listy jedno- lub dwukierunkowej jako kontenera pełniącego rolę wektora danych niezależnie od rodzaju wykonywanych operacji. Z uwagi na charakter listy, odczyt wartości znajdującej się w n -tym węźle wymaga odczytania także $n - 1$ węzłów znajdujących się bliżej jej początku lub końca, co może rzutować na wyższe zużycie zasobów obliczeniowych oraz dłuższy czas wykonywania się algorytmu. W przypadku kontenera, który jest często odczytywany, wydajniejsze może okazać się wykorzystanie struktury tablicy, która w celu dostępu do wartości na n -tej pozycji wymaga jedynie jednej operacji – obliczenia adresu pozycji w pamięci procesu.

W przypadkach, gdy problem jest znany jako trudny do rozwiązania w sposób dokładny (NP-trudny) lub dokładność wyników nie jest kluczowa, stosowanie algorytmów aproksymacyjnych [Vaz01] może pozwolić na znalezienie przybliżonych rozwiązań w akceptowalnym czasie. Aproksymacja polega na wyznaczeniu przybliżonej formy wyniku, co może skrócić czas obliczeń kosztem pewnej utraty precyzji. Aproksymacja często stosowana jest również w algorytmach o złożoności wykładniczej, które byłyby zbyt kosztowne obliczeniowo dla danych wejściowych o dużym rozmiarze.

3.7.2 Rozwiązania sprzętowe

Rozwiązania sprzętowe [PMMB22] są jednymi z najpopularniejszych rozwiązań stosowanych w przyspieszaniu algorytmów, które polegają na wykorzystaniu wielu rdzeni lub procesorów w celu równoczesnego przetwarzania danych. Przyspieszanie sprzętowe szczególnie dobrze się sprawdza w algorytmach które łatwo można podzielić na niezależne etapy. Każdy z tych etapów może wówczas być wykonywany niezależnie, w tym samym czasie, na innej jednostce obliczeniowej.

Obecnie jednymi z najpopularniejszych rozwiązań w sprzętowym przyspieszaniu algorytmów są procesory graficzne (*ang.* *Graphical Processing Unit*, GPU), które zostały oryginalnie zaprojektowane pod kątem wydajnego przetwarzania grafiki trójwymiarowej (gry komputerowe, tomografia, rezonans magnetyczny). W ostatnim czasie popularność zdobywa zastosowanie GPU także w przetwarzaniu sygnału audio [GPU23]. Wśród dostępnych GPU na rynku, najczęściej wykorzystywane są modele produkcji firmy *NVIDIA*, które wspierają technologię *NVIDIA CUDA* [NVI23] z uwagi na szeroki wachlarz dostępnych narzędzi dla programistów

[Cud23a, CuD23b].

Przetwarzanie równoległe

Przetwarzanie równoległe za pomocą GPU to technika, w której wykorzystuje się moc obliczeniową kart graficznych do wykonywania równoczesnych operacji na danych. W przeciwieństwie do *CPU*, które posiadają kilka lub kilkanaście rdzeni, *GPU* są wyposażone w tysiące rdzeni, co umożliwia wykonywanie równoległych operacji na dużych zbiorach danych jednocześnie. Dzięki temu przetwarzanie równoległe przy użyciu *GPU* z powodzeniem znalazło zastosowanie w trenowaniu głębokich sieci neuronowych czy w obliczeniach naukowych (prognozy pogody, równania Schrödingera).

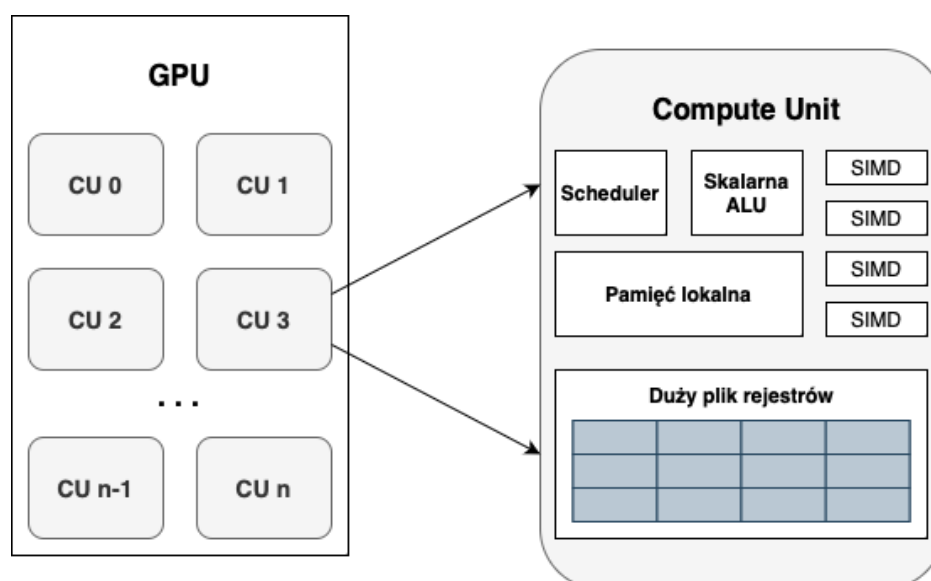
SIMD (Single Instruction Multiple Data) [Hug15] to popularna architektura komputerowa, która umożliwia równoległe wykonanie tej samej operacji dla różnych danych reprezentowanych za pomocą wektora, co oznacza równoległe wykonanie instrukcji na wektorze. *SIMD* jest dostępna, między innymi, w procesorach Intel 64 (x86-64) z technologiami MMX/SSE/AVX [HHM16] czy ARM z technologią NEON [JKK11].

Równoległe przetwarzanie obrazów wymaga zastosowania *SIMD* w większej skali, co jest zadaniem dla *GPU*. Zasada zrównoleglania obliczeń na *GPU* polega na powieleniu koncepcji *SIMD* w strukturze jednostek obliczeniowych (*ang. compute unit*, *CU*), które są w stanie wykonać tą samą instrukcję na wektorze (Rysunek 3.27).

Dodatkowo, dzięki znacznej liczbie rejestrów obliczeniowych, przełączanie wątków wykonywanych na *CU* jest mało kosztowne, co wpływa na efektywność wykonywania powtarzalnych operacji na *GPU* na dużych porcjach danych.

Programowanie GPU

Na rynku istnieją rozwiązania natywne umożliwiające wykonywanie obliczeń przy współpracy *GPU* i *CPU* (*NVIDIA CUDA* [NVI23], *AMD Vulkan* [AMD23c], *Apple Metal* [App23a]). Niewątpliwą zaletą stosowania rozwiązań natywnych jest ich wysoka wydajność oraz większa kontrola



Rysunek 3.27: Schemat architektury GPU. ALU (ang. Arithmetic Logic Unit) – jednostka arytmetyczno-logiczna. Wersja elektroniczna obrazu: <https://krzywicki.pro/fig-3-27>.

nad wykonywaniem instrukcji na sprzęcie. Rozwiązania natywne są jednak przystosowane do jednej platformy sprzętowej, co uniemożliwia wykonanie kodu źródłowego zaimplementowanego na platformę *NVIDIA CUDA* na GPU o architekturze *Apple Silicon* [App23b, KC22], do programowania której przeznaczona jest platforma (oraz język) *Metal*.

Wraz z rozwiązaniami natywnymi często wykorzystywane są języki programowania *C* [KR06] oraz *C++* [Str13], które umożliwiają zachowanie wysokiej kontroli nad wykonywaniem instrukcji na sprzęcie oraz nad pamięcią. Za sprawą rosnącej popularności języka *Python* [Pyt23], rozwiązanie *NVIDIA CUDA* zostało dostosowane również do tego języka, co jednak z uwagi na jego specyfikę, nie umożliwia kontroli ani nad wykonywaniem instrukcji na sprzęcie ani nad pamięcią.

Wśród rozwiązań wieloplatformowych warto zwrócić uwagę na bibliotekę *OpenCL* (Open Computing Language) [SGS10], która umożliwia wykonywanie obliczeń równoległych na platformach heterogenicznych składających się z różnych typów procesorów. *OpenCL* jest platformą niezależną od sprzętu, co oznacza, że aplikacje napisane w tym standardzie mogą być uruchamiane na różnych platformach, niezależnie od producenta sprzętu.

Język programowania *OpenCL* oparty jest na językach C/C++, dzięki czemu można z łatwością przystosować implementacje do wydajnej współpracy z procesorami różnych typów. Mimo, że platforma *OpenCL* została stworzona przez firmę *Apple*, obecnie nie jest przystosowana do współpracy z architekturą *Apple Silicon*.

3.8 Podsumowanie

W tym rozdziale omówiono algorytmy i metody stosowane w rozpoznawaniu obrazów. Rozdział rozpoczął od wprowadzenia pojęcia analizy złożoności obliczeniowej algorytmów celem omówienia ich wydajności. Następnie przedstawiono ideę obrazu cyfrowego i trzy modele barw: RGB, CIELab i YUV. Omówiono także wybrane techniki przetwarzania obrazów, takie jak filtrowanie, binaryzacja i wyrównywanie histogramów. Przedstawiono algorytmy topologiczne, w tym morfologię matematyczną, oraz algorytmy inspirowane naturą, takie jak sieci neuronowe. Następnie zaprezentowano nowoczesne metody uczenia maszynowego, takie jak uczenie transferowe, uczenie zespołowe i detekcja obiektów na obrazach za pomocą sieci neuronowych. Na zakończenie omówiono podejścia związane z dedykowanymi algorytmami, strukturami danych i podejściami sprzętowymi do przyspieszania algorytmów.

Rozdział 4

Bazy danych wykorzystane w badaniach

4.1 Przedstawienie problematyki poruszanej w rozdziale

Rozdział 4. jest poświęcony bazom zawierającym zdjęcia dna oka, które wykorzystano w procesie tworzenia metod lokalizujących cechy anatomiczne tylnego odcinka oka, takie jak dołeczek oraz tarcza nerwu wzrokowego. Zbieranie i analiza tych danych jest kluczowym krokiem w rozwijaniu skutecznych metod monitorowania i diagnostyki chorób siatkówki.

Bazy zawierające zdjęcia dna oka są zwykle generowane dwuetapowo. W pierwszej kolejności gromadzone są zdjęcia dna oka wykonywane za pomocą fundoskopii, jako jednej z technik obrazowania medycznego. Następnie, zdjęcia są etykietowane przez ekspertów pod kątem występowania jednostek chorobowych lub zaznaczane są na nich istotne obszary jako adnotacje obejmujące cechy anatomiczne lub widoczne zmiany chorobowe.

W badaniach wykorzystano dwie bazy kolorowych zdjęć dna oka, które omówiono w dwóch kolejnych podrozdziałach. W ich ramach przedstawiono szczegółową charakterystykę zebranych danych wraz z informacjami dotyczącymi źródła obrazowania oraz metody oceny i adnotacji. Jakość zdjęć w bazach oceniono za pomocą metody zaproponowanej przez

ROZDZIAŁ 4. BAZY DANYCH WYKORZYSTANE W BADANIACH 101

Fu i in. [FWS⁺19], która osiągnęła dokładność na poziomie 88% i jest dostępna publicznie w formie wyuczonego modelu. Do zbadania atrybutów technicznych baz (Tabela 4.1) wykorzystano autorskie narzędzie *Ophthalmic Repository Sample Generator (ORSG)* (Podrozdział B.3), którego celem jest generowanie pseudolosowych reprezentatywnych prób zawartości obszernych baz zdjęć dna oka oraz generowanie podsumowań w postaci atrybutów technicznych.

LP	Nazwa	Liczba zdjęć	Rozmiary zdjęć	Opisy zdjęć	Dostęp
1	<i>BFO21</i>	3626	1956x1934 px	Etykiety i adnotacje	Zamknięty
2	<i>IDRID</i>	103	4288x2848 px	Etykiety i adnotacje	Otwarty [IDR23a]

Tabela 4.1: Atrybuty techniczne baz zdjęć dna oka wykorzystanych w badaniach.

Atrybuty epidemiologiczne (Tabela 4.2) baz uzupełniono na podstawie informacji od autorów oraz danych znalezionych na stronach źródłowych.

LP	Nazwa	Liczba pacjentów	Jednostki chorobowe	Kraj pochodzenia
1	<i>BFO21</i>	830	DR, DME	Polska
2	<i>IDRID</i>	Brak danych	DR, DME	Indie

Tabela 4.2: Atrybuty epidemiologiczne baz zdjęć dna oka wykorzystanych w badaniach.

Atrybuty anatomiczne (Tabela 4.3) oraz jakościowe (Tabela 4.4) dotyczące artefaktów optycznych opisano samodzielnie za pomocą autorskiego narzędzia *Fundus Photo Labeller (FPL)* [Fun23] (implementacja w Podrozdziale B.4 w Dodatku B). Każda baza zdjęć dna oka została w pełni zanonimizowana, dzięki czemu może być wykorzystywana w badaniach.

LP	Nazwa	Liczba zdjęć zdeję prawego oka	Liczba zdjęć zdeję lewego oka	Liczba zdjęć wycentrowanych na dołeczek	Liczba zdjęć wycentrowanych na tarczę nerwu wzrokowego	Liczba zdjęć niewycentrowanych
1	<i>BFO21</i>	1816	1814	1788	1682	160
2	<i>IDRID</i>	47	56	101	0	2

Tabela 4.3: Atrybuty anatomiczne baz zdjęć dna oka wykorzystanych w badaniach.

Wnioski płynące z tego rozdziału będą stanowiły fundament dla dalszych badań i doskonalenia metod rozpoznających obrazy włączając w to zdjęcia dna oka. Mogą także znacząco wpłynąć na polepszenie diagnostyki i opieki zdrowotnej w obszarze okulistyki.

LP	Nazwa	Liczba zdjęć wysokiej jakości	Liczba zdjęć akceptowalnej jakości	Liczba zdjęć niskiej jakości	Liczba zdjęć z przysłaniającymi artefaktami optycznymi	Liczba zdjęć z nieprzysłaniającymi artefaktami optycznymi	Liczba zdjęć bez artefaktów optycznych
1	<i>BFO21</i>	2515	94	1017	298	1713	1619
2	<i>IDRID</i>	89	6	8	0	17	86

Tabela 4.4: Atrybuty jakościowe baz zdjęć dna oka wykorzystanych w badaniach.

4.2 Baza Fundacji Okulistyka 21

Baza zdjęć dna oka będąca przedmiotem analizy w tym podrozdziale została pozyskana we współpracy z Fundacją *Okulistyka 21*¹. Przeznaczeniem zbioru było przygotowanie metod lokalizujących cechy anatomiczne na zdjęciach dna oka oraz rozpoznających ich centrację i stronę oka. Zbiór wykorzystano zarówno do trenowania metod oraz do ich wewnętrznej walidacji.

Baza Fundacji Okulistyka 21 (BFO21) została pozyskana za pomocą urządzenia *Topcon camera Nw-400* przez specjalistów. Dla każdego pacjenta wykonano minimum cztery zdjęcia, po dwa na każde oko w standardzie pola widzenia 45 stopni. Liczba zdjęć była uzależniona od spełnienia standardów w postaci czytelności, jakości oraz centracji na dołeczek lub tarczę nerwu wzrokowego. W przypadku niespełnienia standardu, liczba zdjęć była większa celem uzyskania reprezentatywnego obrazu dna oka. W związku z tym każdy pacjent miał wykonane minimum:

- jedno zdjęcie dna prawego oka wycentrowane na dołeczek,
- jedno zdjęcie dna prawego oka wycentrowane na tarczę nerwu wzrokowego,
- jedno zdjęcie dna lewego oka wycentrowane na dołeczek,
- jedno zdjęcie dna lewego oka wycentrowane na tarczę nerwu wzrokowego.

Każdy z pacjentów został zdiagnozowany przez trzech specjalistów z zakresu okulistyki pod kątem obecności DR oraz DME na podstawie zdjęć dna oka. 370 pacjentów zostało ocenionych przez specjalistów z Polski, 509 przez

¹Założycielem fundacji jest prof. dr hab. med. Andrzej Grzybowski. Szczegółowe informacje dostępne są na stronie internetowej: <https://okulistyka21.pl>.

specjalistów z Indii, a 50 zarówno przez specjalistów polskich oraz indyjskich. Każdy z pacjentów został także zdiagnozowany przez pierwszy autonomiczny system wspierający diagnostykę DR – *IDx-DR* [ALB⁺18]. Na zdjęciach oznaczono w postaci adnotacji widoczne pozycje plamki żółtej oraz tarczy nerwu wzrokowego. Łącznie oznaczono 2289 plamek żółtych i 3557 tarcz nerwu wzrokowego. Przykładową zawartość bazy przedstawiono na Rysunku 4.1.

Na podstawie Tabeli 4.1 i 4.2 można zauważyć, że zbiór składa się z 3626 zdjęć o stałym rozmiarze pochodzących od 830 pacjentów. Liczba zdjęć prawego oka (1816 zdjęć) jest prawie wyrównana względem oka lewego (1814 zdjęć), jednak liczba zdjęć wycentrowanych na dołeczek (1788 zdjęć) przeważa nad liczbą zdjęć wycentrowanych na tarczę nerwu wzrokowego (1682 zdjęć), co można dostrzec w Tabeli 4.3. Jest także 160 zdjęć niewycentrowanych. Na podstawie Tabeli 4.4 można zauważyć, że zdjęcia wysokiej jakości ilościowo przeważają ponad dwukrotnie (2515 zdjęć) nad zdjęciami o niskiej jakości niezdatnymi do dalszej analizy (1017 zdjęć). Wystąpiły także 94 zdjęcia o zaniżonej jakości, lecz w dalszym ciągu zdatne do dalszej analizy. Na 298 zdjęciach odnotowano artefakty optyczne przysłaniające istotne obszary zdjęć, co uniemożliwiało ich wzrokową analizę pod kątem występowania chorób i mogło mieć wpływ na niską jakość. Zauważalna większość zdjęć miała jednak nieznaczące artefakty optyczne (1713 zdjęć) lub nie miała ich wcale (1619 zdjęć).

4.3 Indian Diabetic Retinopathy Image Dataset

Baza zdjęć dna oka *Indian Diabetic Retinopathy Image Dataset (IDRID)* [IDR23a, PPK⁺18] będąca przedmiotem analizy w tym podrozdziale powstała na potrzeby organizacji konkursu podzielonego na trzy etapy: segmentacji zmian chorobowych, wykrywania jednostek chorobowych (DR, DME) oraz lokalizacji cech anatomicznych. Przeznaczeniem bazy w omawianych badaniach były zewnętrzne testy prezentowanych metod lokalizacji cech anatomicznych dna oka. Rozmiar bazy wynosi 516 zdjęć, gdzie 413 stanowiło podzbiór treningowy, a 103 testowy. W badaniach wykorzystano jedynie podzbiór testowy.



(a) Zdjęcie o wysokiej jakości.



(b) Zdjęcie o niskiej jakości.



(c) Zdjęcie przedstawiające objawy DME.



(d) Zdjęcie z widocznymi artefaktami optycznymi.

Rysunek 4.1: Przykładowa zawartość bazy zdjęć dna oka Fundacji Okulistyka 21.
Wersja elektroniczna obrazu: <https://krzywicki.pro/fig-4-1>.

ROZDZIAŁ 4. BAZY DANYCH WYKORZYSTANE W BADANIACH 105

Baza *IDRID* została pozyskana w latach 2009-2017 za pomocą urządzenia *Kowa VX-10 alpha*. Każde ze zdjęć zostało oznaczone pod kątem występowania DR oraz DME. Na każdym zdjęciu zaznaczono także współrzędne środka dołeczka oraz tarczy nerwu wzrokowego, a także maski segmentacji tych obiektów (Rysunek 4.2).



(a) Przykładowe zdjęcie dna oka z widocznymi zmianami chorobowymi.



(b) Maska segmentacji tarczy nerwu wzrokowego dla zdjęcia 4.2a.

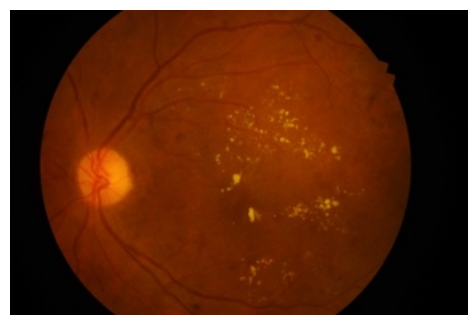
Rysunek 4.2: Maska segmentacji tarczy nerwu wzrokowego bazy *IDRID*. Wersja elektroniczna obrazu: <https://krzywicki.pro/fig-4-2>.

Współrzędne środka zostały oznaczone przez dwóch specjalistów, z czego wyciągnięto średnie współrzędne, a maski segmentacji zostały wykonane przez jednego specjalistę. Nie jest znana liczba specjalistów oceniających zdjęcia pod kątem występowania jednostek chorobowych. Nie jest także znana liczba pacjentów biorących udział w badaniach skutkujących powstaniem zbioru zdjęć. Przykładową zawartość bazy przedstawiono na Rysunku 4.3.

Na podstawie Tabeli 4.3 można zauważyć, że liczba zdjęć prawego oka (47 zdjęć) jest nieznacznie niższa od liczby zdjęć lewego oka (56 zdjęć). Niemal wszystkie zdjęcia były wycentrowane na dołeczek (101 zdjęć), a dwa pozostałe nie były wycentrowane z uwagi na prawie symetryczną pozycję plamki żółtej i tarczy nerwu wzrokowego. Warty jest także fakt, że żadne zdjęcie nie było wycentrowane na tarczę nerwu wzrokowego. W Tabeli 4.4 można dostrzec, że zdjęcia wysokiej jakości przeważają liczebnościowo (89 zdjęć) nad zdjęciami o jakości akceptowalnej (6 zdjęć) oraz niskiej (8 zdjęć). W związku z tym warto zwrócić uwagę na fakt, że nie wszystkie zdjęcia są czytelne, co przekłada się na widoczność cech anatomicznych, a mimo wszystko są do nich przypisane współrzędne ich środków. Interesującą kwestią jest brak odnotowanych przysłaniających artefaktów optycznych na zdjęciach. Zauważono natomiast 17 zdjęć z nieprzysłaniającymi artefaktami



(a) Zdjęcie wysokiej jakości.



(b) Zdjęcie niskiej jakości.



(c) Zdjęcie o obniżonej jakości z widocznymi objawami chorobowymi.



(d) Zdjęcie wysokiej jakości z widocznymi objawami chorobowymi.

Rysunek 4.3: Przykładowa zawartość bazy IDRID. Wersja elektroniczna obrazu: <https://krzywicki.pro/fig-4-3>.

i 86 zdjęć bez widocznych artefaktów optycznych.

4.4 Podsumowanie

W rozdziale przedstawiono charakterystyki dwóch zbiorów danych przeznaczonych do opracowania oraz przetestowania metod służących do rozpoznawania zdjęć dna oka. Godnym zwrócenia uwagi jest fakt, że artefakty optyczne widoczne na zdjęciach mogą istotnie zaburzać proces rozpoznawania zdjęć dna oka zarówno na etapie uczenia metody oraz testowania. Warto także mieć na względzie kwestię jakości zdjęć w bazach. W bazie *IDRID* jedynie 89 spośród 103 zdjęć ma czytelne cechy anatomiczne w postaci plamki żółtej oraz tarczy nerwu wzrokowego mimo przypisania współrzędnych ich środków na wszystkich zdjęciach. W bazie *BFO21* czytelnych było 1816 spośród 3626 zdjęć i wbrew nieczytelności części, udało się oznaczyć widoczne na nich cechy anatomiczne.

*ROZDZIAŁ 4. BAZY DANYCH WYKORZYSTANE W BADANIACH*107

Kolejnym istotnym faktem jest jakość etykiet i adnotacji przypisanych do zdjęć dna oka, co może stanowić miarę zaufania. Opisy powstałe przy pomocy kilku specjalistów będą bardziej wiarygodne, niż opisy wygenerowane przez jedną osobę [GBK⁺22]. W tej sytuacji warto zwrócić uwagę na kwestię dostępności specjalistów dokonujących oceny zdjęć. Nierzadko tego rodzaju czynności są związane z wysokimi kosztami.

Rozdział 5

Zastosowane metody

5.1 Przedstawienie problematyki poruszanej w rozdziale

W dziedzinie medycyny, szczególnie w okulistyce, badanie dna oka odgrywa kluczową rolę. Analiza cech anatomicznych dna oka może dostarczyć istotnych informacji diagnostycznych, takich jak kwalifikacja stanu plamki żółtej [Wan22] i tarczy nerwu wzrokowego [KYC23], wykrywanie ognisk chorobowych widocznych w obrębie siatkówki czy ocena ogólnego zdrowia pacjenta. W celu skutecznej analizy i interpretacji obrazów dna oka wspieranej metodami automatycznymi, istotne jest dokładne zlokalizowanie cech anatomicznych oraz prawidłowa centracja zdjęcia w stosunku do obszaru siatkówki. Z punktu widzenia klinicznego lokalizacja dołączka oraz tarczy nerwu wzrokowego może dostarczyć informacji na temat ich stanu i wyglądu wraz z okolicznymi strukturami anatomicznymi.

W Rozdziale 5. zaproponowano dwie nowatorskie metody, które opracowano w celu automatycznego lokalizowania wybranych cech anatomicznych dna oka. W ramach tego badania, przygotowano dwie metody dotyczące lokalizacji tarczy nerwu wzrokowego oraz dołączka.

Tradycyjne podejścia do tych zagadnień polegają na ręcznym wskazywaniu i pomiarze cech anatomicznych przez specjalistów, co jest czasochłonne i podatne na błędy ludzkie. W związku z tym rozwinięcie automatycznych

metod lokalizacji cech anatomicznych tylnego odcinka oka jest nie tylko wartościowe z punktu widzenia efektywności diagnostycznej, ale może również znacznie skrócić czas potrzebny do przeprowadzenia diagnostyki klinicznej.

Istniejące metody często wykorzystują zależności geometryczne struktur siatkówki, takie jak względne położenie plamki żółtej czy tarczy nerwu wzrokowego, a także odgórnie założoną centrację zdjęć. Proponowane metody lokalizacji cech anatomicznych niezależne od ich względnego położenia mogą przyczynić się do dalszej poprawy dokładności i precyzji analizy zdjęć dna oka.

Celem tego rozdziału jest zaproponowanie nowoczesnych metod, które zostały opracowane w celu automatycznego i jednoczesnego lokalizowania cech anatomicznych dna oka. W rozdziale przeanalizowano różne podejścia, takie jak techniki oparte na przetwarzaniu obrazów oraz uczeniu głębokim. Wyniki eksperymentów wraz z porównaniem jakości zaproponowanych metod w porównaniu do istniejących omówiono w Rozdziale 6.

5.2 Istniejące rozwiązania

Wśród rozwiązań lokalizujących jednocześnie dołeczek oraz tarczę nerwu wzrokowego na zdjęciach dna oka dużą rolę odgrywają głębokie sieci neuronowe [PPK⁺20, HAE⁺21, DK23, LSD18]. Ich główną zaletą jest niewątpliwie wysoka dokładność i zdolność do lokalizacji skomplikowanych wzorców, lecz mogą być obciążone także wadami. Jedną ze słabych stron jest wykorzystany podzbiór treningowy, który pochodził z tej samej populacji, co podzbiór wykorzystany podczas testów metod (baza *IDRID*). Co więcej, wszystkie zdjęcia wykorzystane podczas treningu rozwiązań, były podobnie wycentrowane i miały widoczne zarówno plamkę żółtą jak i tarczę nerwu wzrokowego. Tego rodzaju zależności w danych testowych mogą wpływać na zaniżoną zdolność modelu na lokalizację cech anatomicznych dna oka w sytuacji napotkania zdjęć wycentrowanych na tarczę nerwu wzrokowego lub nieprzedstawiających wszystkich wytrenowanych cech anatomicznych. Istotnymi wadami rozwiązań wykorzystujących sieci neuronowe są także konieczność posiadania obszernych oraz właściwie oznaczonych baz zdjęć, długi czas dokonywania predykcji oraz konieczna duża moc obliczeniowa na potrzeby treningu, co wiąże się z wysokimi kosztami zakupu sprzętu lub usług

chmurowych.

W literaturze naukowej popularne są także rozwiązania, wykorzystujące głębokie sieci neuronowe skoncentrowane na lokalizacji wybranej cechy anatomicznej dna oka – dołeczku [WWX⁺19, AWZ⁺20, FZL⁺22] lub tarczy nerwu wzrokowego [TTH22, LTX⁺22, BMS⁺19]. Istotną zaletą takich metod, podobnie jak w przypadku metod jednocześnie lokalizujących cechy anatomiczne, jest wysoka zdolność wyszukiwania wzorców. Ocena wad tego rodzaju metod jest procesem skomplikowanym z uwagi na lata, w których te prace były publikowane – nie wszystkie wyniki są ze sobą porównywalne z racji dostępnych metod, danych i sprzętu. Mimo wszystko funkcjonalność takich rozwiązań jest mniejsza ze względu na zorientowanie wokół tylko jednej cechy anatomicznej.

Warto zwrócić także uwagę na rozwiązania lokalizujące cechy anatomiczne dna oka wykorzystujące algorytmy przetwarzania obrazów. Wśród rozwiązań jednocześnie wykrywających dołeczek oraz tarczę nerwu wzrokowego godna uwagi jest praca Kanjanasurata i Satayaraka [KS21]. Zaproponowana metoda nie wykorzystuje sieci neuronowej, zatem nie wymaga wykorzystania dużych zasobów obliczeniowych, ani obszernych zbiorów danych treningowych. Pośród godnych uwagi jest także praca Abdullaha i innych [AFB16], w której została wykorzystana *Transformata Hougha* w wersji do wykrywania okręgów [IK87] wraz z metodą *grow-cut* [VK04]. Zaproponowane rozwiązanie dzięki wykorzystanej *Transformacie Hougha* jest w stanie wykrywać skomplikowane wzorce. Istotną wadą jest jednak konieczność – jak w przypadku wykorzystania sieci neuronowych – posiadania obszernego zbioru oznaczonych zdjęć w celu wytrenowania detektora okręgów. Z kolei praca Jianga i innych [JPS⁺20] dotyczy lokalizacji dołeczka za pomocą transformacji morfologicznych [Ser83, HSZ87] oraz segmentacji za pomocą binaryzacji, co czyni rozwiązanie szybkim w działaniu. Podczas analizy metody Jianga i innych zauważono jednak, że brakuje tam kilku istotnych kroków, co będzie stanowiło fundament pierwszego zaproponowanego rozwiązania omówionego w Podrozdziale 5.3.

5.3 Metoda Szybkiej i Równoległej Lokalizacji Współrzędnych Środków Dołeczka i Tarczy Nerwu Wzrokowego na Zdjęciach Dna Oka przy użyciu Segmentacji oraz Analizy Komponentów Połączonych

5.3.1 Motywacja

Rozpoczęcie prac nad *Metodą Szybkiej i Równoległej Lokalizacji Współrzędnych Środków Dołeczka i Tarczy Nerwu Wzrokowego na Zdjęciach Dna Oka przy użyciu Segmentacji oraz Analizy Komponentów Połączonych* (MSiRL) wynikało z potrzeby znalezienia szybkiego, nieskomplikowanego obliczeniowo oraz skutecznego rozwiązania lokalizującego cechy anatomiczne na dnie oka. Jak zauważono w Podrozdziale 5.2, poświęconym przeglądowi istniejących rozwiązań, współczesne rozwiązania lokalizujące cechy anatomiczne dna oka są skoncentrowane wokół wykorzystania głębokich sieci neuronowych, co wiąże się z potrzebą posiadania obszernych baz oznaczonych zdjęć dna oka oraz z wysokim kosztem obliczeniowym zarówno na etapie trenowania rozwiązania oraz podczas dokonywania predykcji. Rozwiązania kosztowne obliczeniowo do poprawnego działania wymagają wydajnych zasobów sprzętowych, co wiąże się z wysokim kosztem finansowym zakupu sprzętu lub usług chmurowych oraz z wysokim kosztem energii elektrycznej. Rozwiązania wykorzystujące algorytmy przetwarzania obrazów zwykle generują mniejszy koszt obliczeniowy, lecz są też mniej dokładne.

Punktem wyjścia do opracowania nowej metody *MSiRL* była praca Jianga i innych [JPS⁺20] dotycząca lokalizacji współrzędnych dołeczka na zdjęciu dna oka za pomocą transformacji morfologicznych oraz segmentacji za pomocą binaryzacji. Schemat działania algorytmu składał się z następujących kroków:

1. konwersja zdjęcia do skali szarości,
2. transformacje morfologiczne: *Top-hat* i *Bottom-hat* uwypuklające obszar dołeczka,
3. wyrównanie histogramu obrazu,

4. transformacje morfologiczne *erozja* i *dylacja* minimalizujące szum w postaci naczyń krwionośnych i tła plamki żółtej,
5. binaryzacja obrazu *metodą Otsu* [Ots79],
6. wybór finalnego obszaru plamki żółtej za pomocą wyznaczonej kolistości.

Podczas analizy rozwiązania zauważono, że niektóre kroki algorytmu można usprawnić z pozytywnym wpływem na jakość wyjścia. Zdjęcie dna oka skonwertowane do postaci skali szarości nie zawsze przedstawia wyraźną plamkę żółtą (Rysunek 5.1).

Zauważono także, że krok czwarty polegający na zastosowaniu erozji i dylacji nie przynosi oczekiwanego efektu w przypadku nierównomiernego oświetlenia siatkówki na zdjęciu, co w połączeniu z brakiem redukcji szumów powoduje segmentację przypadkowych regionów w kroku piątym.

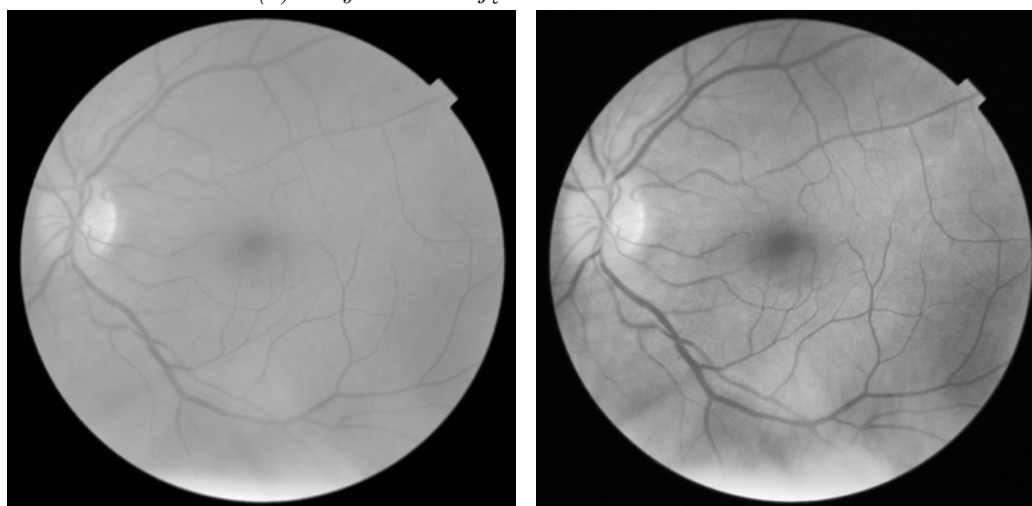
5.3.2 Zasada działania

Celem algorytmu *MSiRL* jest lokalizacja współrzędnych środków dołeczka oraz tarczy nerwu wzrokowego na zdjęciu dna oka. Na Rysunku 5.2 przedstawiono schemat działania proponowanej metody.

Można zauważyć, że część kroków w algorytmie odbywa się dwutorowo i niezależnie. Wejściowe zdjęcie dna oka jest przetwarzane jednocześnie pod kątem wyraźnego uwypuklenia obszarów będącymi wyraźnymi kandydatami na plamkę żółtą oraz tarczę nerwu wzrokowego. Ostatnie kroki algorytmu są wspólne dla obu obrazów, lecz każdy z nich nadal jest przetwarzany niezależnie. Każdy z obrazów jest binaryzowany w celu wyraźnego wyznaczenia regionów będących kandydatami na lokalizowany obiekt. Finalnie, zamknięte regiony na obrazach są etykietowane, a następnie wybierany jest najlepszy z nich na podstawie podobieństwa do oczekiwanego kształtu i obszaru. Wyjściem proponowanego algorytmu są geometryczne środki okręgów opisujących wybrane regiony obrazu. Szczegóły implementacji metody *MSiRL* znajdują się w Podrozdziale B.1 w dodatku B. W procesie implementacji wykorzystano język programowania *Python* [Pyt23] wraz z biblioteką *OpenCV* [Bra00].



(a) Przykładowe zdjęcie dna oka.



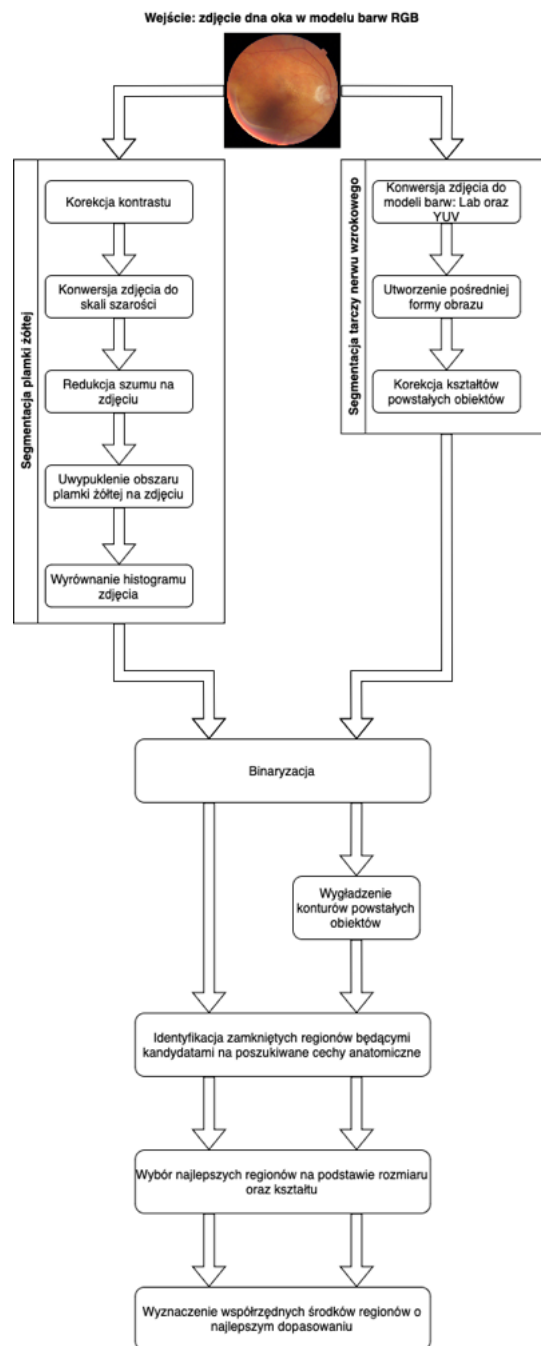
(b) Zdjęcie dna oka przekonwertowane do skali szarości bez skorygowanego kontrastu. Obszar plamki żółtej jest mało wyraźny.

(c) Zdjęcie dna oka z poprawionym kontrastem przekonwertowane do skali szarości. Obszar plamki żółtej jest wyraźniejszy.

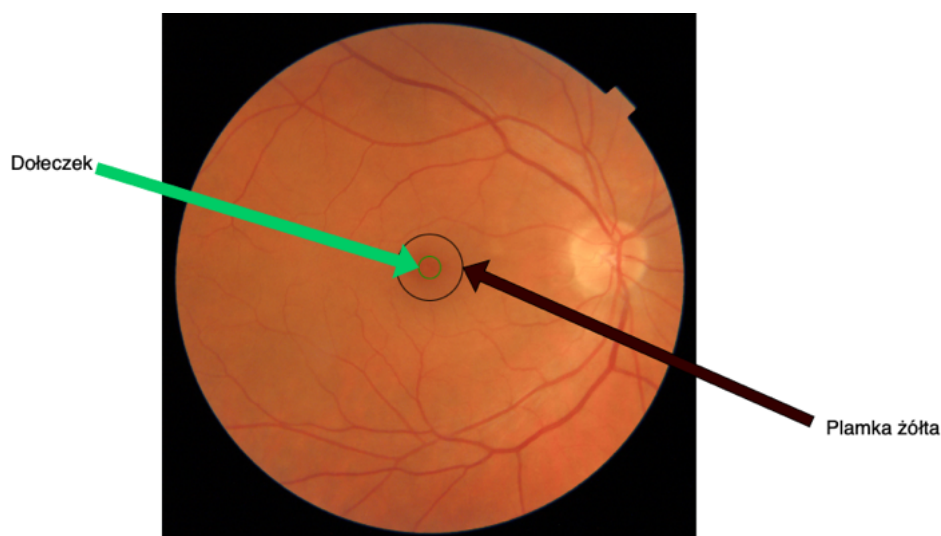
Rysunek 5.1: Przykład konwersji do skali szarości prawidłowo oświetlonego zdjęcia dna oka o wysokiej jakości. Bez przeprowadzenia operacji korekty kontrastu obszar plamki żółtej jest mało wyraźny. Wersja elektroniczna obrazu: <https://krzywicki.pro/fig-5-1>.

Lokalizacja dołeczka

Dołeczek to centralny region znajdujący się na plamce żółtej, który przedstawiono na Rysunku 5.3. Oba obiekty przyjmują kształt kolisty i są



Rysunek 5.2: Schemat działania proponowanej metody MSiRL. Zdjęcie jest przetwarzane niezależnie i równoległe pod kątem lokalizacji dołeczka i tarczy nerwu wzrokowego. Wersja elektroniczna obrazu: <https://krzywicki.pro/fig-5-2>.

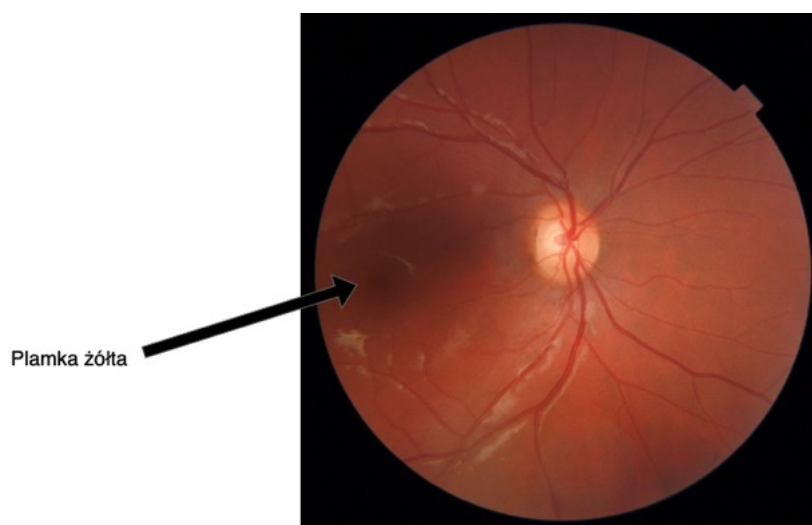


Rysunek 5.3: Plamka żółta oraz dołeczek na zdjęciu dna oka. Oba obiekty przyjmują kształt kołisty. Wersja elektroniczna obrazu: <https://krzywicki.pro/fig-5-3>.

ciemniejsze względem obszaru siatkówki, lecz na zdjęciach o zaniżonej jakości mogą różnić się kompozycją (Rysunek 5.4), gdzie obszar dołeczka jest już niedostrzegalny. Na zdjęciach o niskiej jakości obszar plamki żółtej jest całkowicie niewidoczny i niemożliwy do precyzyjnego ustalenia, co można zauważyć na Rysunku 5.5. Podobnie jest w przypadku pojawienia się zmian chorobowych w okolicach plamki żółtej (Rysunek 5.6) – wówczas obiekt jest przysłonięty częściowo lub całkowicie, co zmienia jego kompozycję i utrudnia lokalizację.

Lokalizacja dołeczka na zdjęciu dna oka jest ściśle uzależniona od wyniku procesu segmentacji plamki żółtej, którego opis przedstawiono w dalszej części podrozdziału. Segmentację plamki żółtej na zdjęciu dna oka można podzielić na trzy kluczowe etapy: przetwarzanie wstępne, przetwarzanie końcowe oraz wybór najlepiej dopasowanego regionu (Rysunek 5.7).

Etap przetwarzania wstępnego polega na przygotowaniu zdjęcia do procesu poszukiwania regionów będących kandydatami na plamkę żółtą i rozpoczyna się od operacji korekcji kontrastu na zdjęciu reprezentowanego domyślnie za pomocą modelu barw *RGB*, która uwypukla różnice między jasnymi i ciemnymi obszarami dając tym samym efekt w postaci poprawionej wyrazistości, oraz głębi (Rysunek 5.8). W celu poprawy kontrastu dokonano konwersji zdjęcia do modelu barw *CIELab*, następnie korzystając z metody *CLAHE* [Rez04] wyrównano w sposób adaptacyjny histogram



Rysunek 5.4: Plamka żółta o pogorszonej wyraźności i zmienionym kształcie. Strzałką wskazano orientacyjny obszar, który nie jest wyraźnie widoczny. Obszar dołączka jest niewidoczny. Wersja elektroniczna obrazu: <https://krzywicki.pro/fig-5-4>.

pierwszej warstwy, która jest odpowiedzialna za jasność i skonwertowano zdjęcie ponownie do modelu RGB . Kolejnym krokiem etapu przetwarzania wstępnego jest konwersja zdjęcia do skali szarości. W tym celu dla każdego piksela na zdjęciu w modelu barw RGB wyznaczono nową wartość za pomocą operacji przedstawionej w Równaniu 5.1, czego wynikiem jest obraz jednowarstwowy. W ostatnim kroku etapu przetwarzania wstępnego zdjęcie poddano operacji odsumowania poprzez rozmycie zdjęcia. Wykorzystano w tym celu filtr gaussowski o rozmiarze 5×5 (Równanie 5.2) w operacji splotu [CCP⁺21], która polega na przesuwaniu filtru po wszystkich pikselach obrazu źródłowego i przemnożeniu wartości pikseli obrazu przez odpowiadające im współczynniki w masce, a następnie zsumowaniu tych wartości. Wynikowy piksel w obrazie poddanym operacji splotu jest obliczany jako suma przemnożonych wartości pikseli i wartości filtru.

$$g(x, y) = 0.299 \cdot p(x, y, 0) + 0.587 \cdot p(x, y, 1) + 0.114 \cdot p(x, y, 2) \quad (5.1)$$

gdzie:

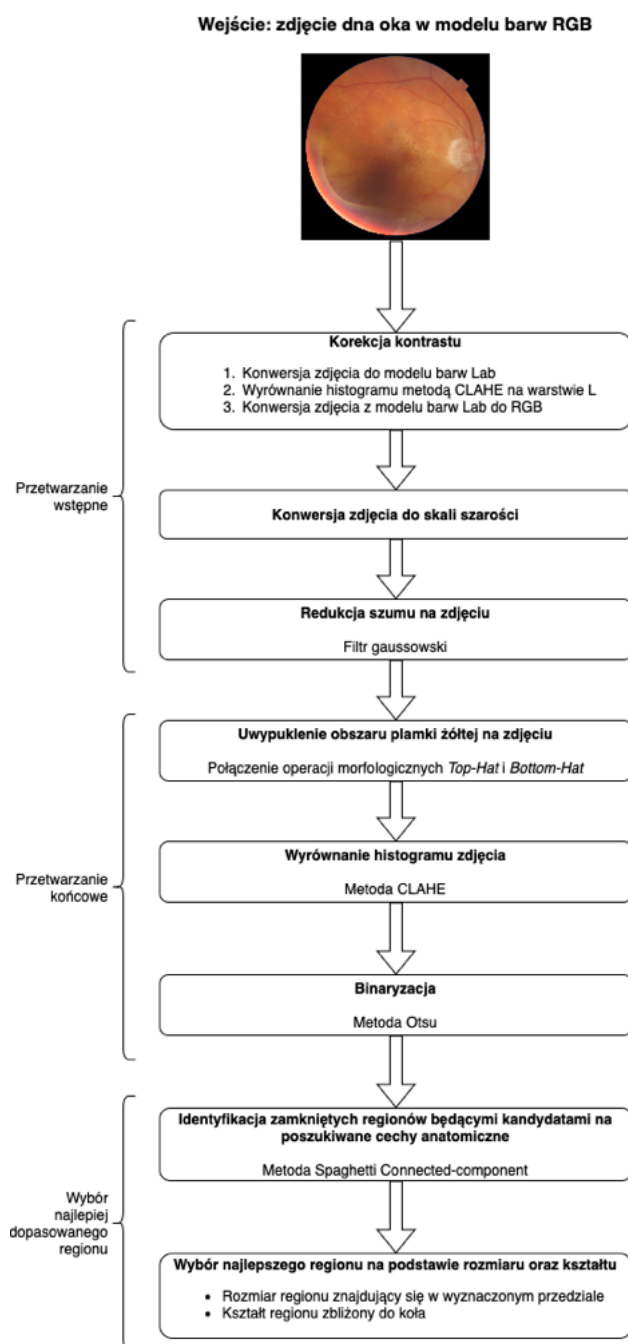
- $g(x, y)$ – wartość piksela na obrazie w skali szarości dla współrzędnych x i y ,



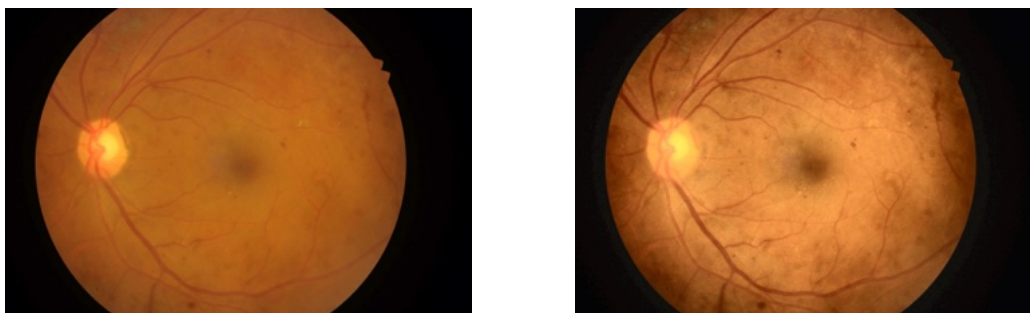
Rysunek 5.5: Zdjęcie dna oka o niskiej jakości, gdzie obszar plamki żółtej jest niewidoczny poprzez niedoświetlenie. Wersja elektroniczna obrazu: <https://krzywicki.pro/fig-5-5>.



Rysunek 5.6: Zdjęcie dna oka z objawami DME zaburzającymi kompozycję kształtu plamki żółtej. Wersja elektroniczna obrazu: <https://krzywicki.pro/fig-5-6>.



Rysunek 5.7: Szczegółowy schemat działania metody segmentującej region plamki żółtej na zdjęciu dna oka. Wersja elektroniczna obrazu: <https://krzywicki.pro/fig-5-7>.



(a) Przykładowe zdjęcie dna oka.

(b) Zdjęcie poprawionym kontrastem.

Rysunek 5.8: Przykład operacji korekty kontrastu na zdjęciu dna oka. Obszar plamki żółtej jest znacznie wyraźniejszy, niż na zdjęciu oryginalnym. Wersja elektroniczna obrazu: <https://krzywicki.pro/fig-5-8>.

- $p(x, y, i)$ – wartość piksela na obrazie w modelu barw RGB dla współrzędnych x i y oraz warstwy i .

$$h = \frac{1}{273} \begin{bmatrix} 1 & 4 & 7 & 4 & 1 \\ 4 & 16 & 26 & 16 & 4 \\ 7 & 26 & 41 & 26 & 7 \\ 4 & 16 & 26 & 16 & 4 \\ 1 & 4 & 7 & 4 & 1 \end{bmatrix} \quad (5.2)$$

Etap przetwarzania końcowego polega na wyznaczeniu finalnych regionów kandydatów na plamkę żółtą i rozpoczyna się od kroku polegającego na uwypukleniu obszaru plamki żółtej za pomocą połączenia efektów transformacji morfologicznych *Top-hat* i *Bottom-hat* (*Black-hat*) [Ser83, HSZ87] przeznaczonymi do podkreślania subtelnych detali obrazów oraz fragmentów ciemniejszych od tła, co przedstawiono w Równaniu 5.3.

$$p_{stage_{21}} = p_{stage_{13}} + p_{top} - p_{bottom} \quad (5.3)$$

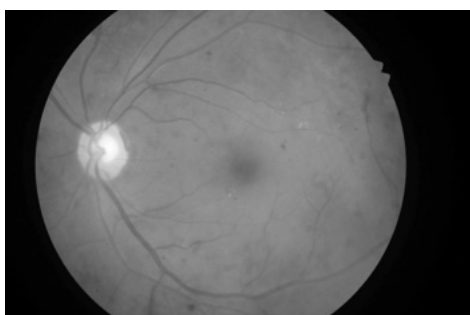
gdzie:

- $p_{stage_{21}}$ – obraz będący efektem połączonych transformacji morfologicznych (wynik kroku pierwszego etapu drugiego algorytmu segmentującego),

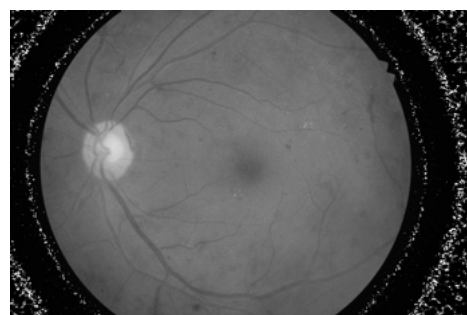
- $p_{stage_{13}}$ – obraz będący efektem redukcji szumu kończącego etap pierwszy algorytmu,
- p_{top} – obraz będący efektem transformacji *Top-hat* przy użyciu dyskowego elementu strukturalnego,
- p_{bottom} – obraz będący efektem transformacji *Bottom-hat* przy użyciu dyskowego elementu strukturalnego.

Efekt działania połączonych transformacji morfologicznych przedstawiono na Rysunku 5.9. Nie zauważono wpływu szumu widocznego wokół obszaru dna oka obrazu (Rysunek 5.9b) na powstawanie fałszywych regionów kandydackich na plamkę żółtą. W celu dalszego uwypuklenia obszaru plamki żółtej wykorzystano ponownie metodę *CLAHE*, której celem jest poprawa widoczności szczegółów obrazu o różnym oświetleniu poprzez adaptacyjne wyrównanie histogramu (Rysunek 5.11). Różnice w histogramach obrazów przed i po zastosowaniu metody *CLAHE* przedstawiono na Rysunku 5.10. Ostatnim krokiem etapu przetwarzania końcowego jest wyznaczenie za pomocą binaryzacji regionów kandydatów na plamkę żółtą. Wykorzystano w tym celu metodę *Otsu* [Ots79], która automatycznie dobiera wartość progową procesu binaryzacji. Efektem etapu przetwarzania końcowego jest obraz przedstawiony na Rysunku 5.12, gdzie z łatwością można dostrzec region plamki żółtej, który wyraźnie wyróżnia się na tle innych regionów kandydackich. Nie zdecydowano się na dokonanie dalszej korekcji kształtów regionów kandydackich za pomocą typowych kroków w postaci operacji *dylacji* i *erozji* morfologicznej z uwagi na zauważalne zaburzenia ich struktur na zdjęciach nieostrych.

Etap wyboru najlepiej dopasowanego regionu jest ostatnim w procesie segmentacji plamki żółtej na zdjęciu dna oka. W pierwszej kolejności dokonywany jest proces identyfikacji i oznaczania wszystkich regionów w postaci zamkniętych obszarów przy użyciu algorytmu *Spaghetti Labelling* [BABG20], w wyniku czego każdy region kandydacki na obrazie otrzymuje unikalny identyfikator (Rysunek 5.13). W następnym kroku każdy z oznaczonych regionów otrzymuje przypisany współczynnik kształtu (Równanie 5.4), który przyjmuje wartość w zakresie od 0 do 1, gdzie wartość bliska 1 oznacza kształt kolisty. Region o powierzchni znajdującej się w wyznaczonym przedziale oraz o największej wartości współczynnika kolistości zostaje wybrany jako najlepiej odzwierciedlający strukturę plamki żółtej.



(a) Obraz będący efektem etapu pierwszego (przetwarzanie wstępne).



(b) Efekt połączonych transformacji morfologicznych. Obszar plamki żółtej jest wyraźniejszy.

Rysunek 5.9: Przykład wpływu połączonych transformacji morfologicznych na wyraźność plamki żółtej na przetwarzanym obrazie. Wersja elektroniczna obrazu: <https://krzywicki.pro/fig-5-9>.

$$T = \frac{4A}{\pi P^2} \quad (5.4)$$

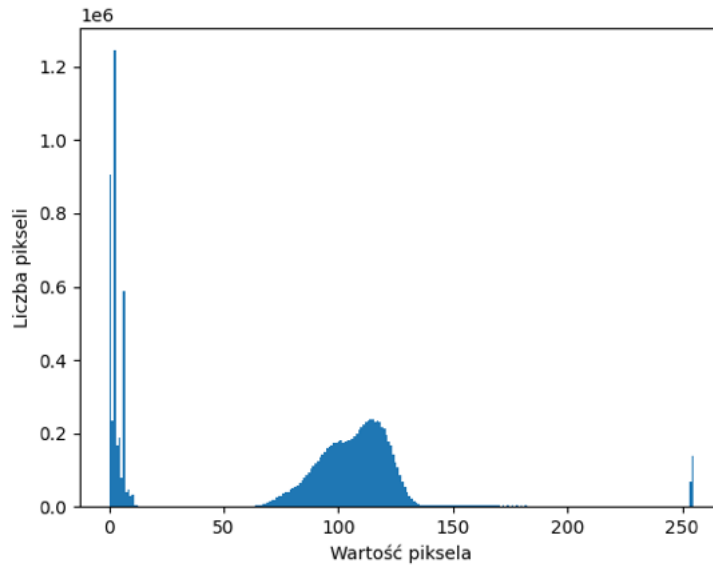
gdzie:

- A – powierzchnia regionu,
- P – najdłuższa przekątna regionu.

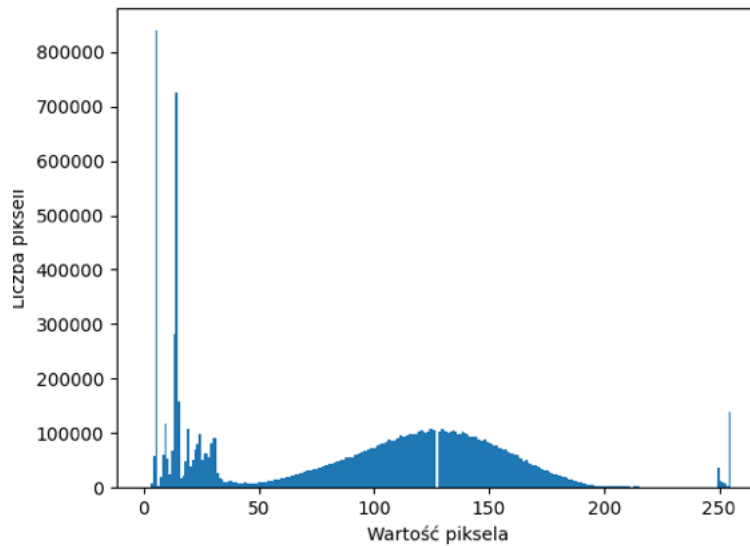
Proces lokalizacji współrzędnych środka dołeczka jest ściśle uzależniony od operacji segmentacji plamki żółtej na zdjęciu dna oka. Centrum dołeczka jest wyznaczane poprzez obliczenie współrzędnych środka koła opisanego na strukturze regionu plamki żółtej, o promieniu równym najdłuższej przekątnej, co przedstawiono na Rysunku 5.14. Końcowy efekt lokalizacji dołeczka zaznaczonego na oryginalnym zdjęciu dna oka przedstawiono na Rysunku 5.15.

Lokalizacja tarczy nerwu wzrokowego

Tarcza nerwu wzrokowego to jasny kolisty obiekt znajdujący się na siatkówce, który przedstawiono na Rysunku 5.16. W przypadku zdjęć o zaniżonej

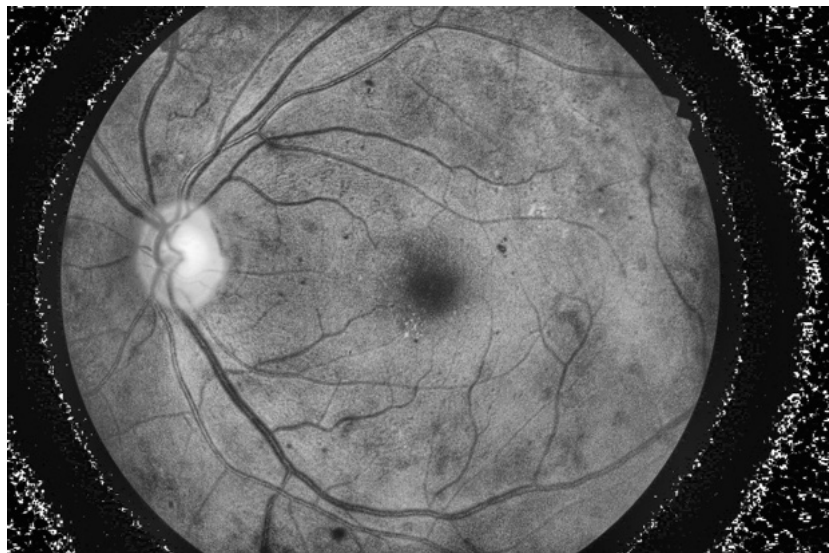


(a) Histogram obrazu b Rysunku 5.9.

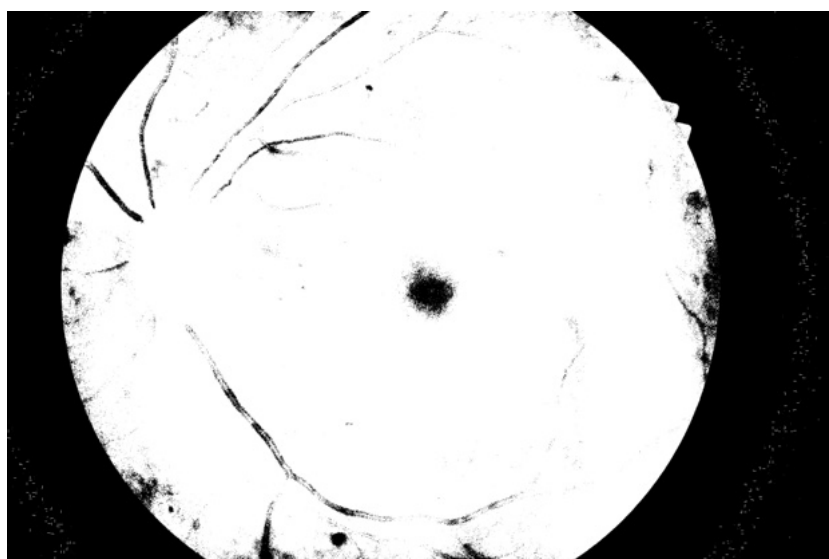


(b) Wyrównany histogram obrazu a.

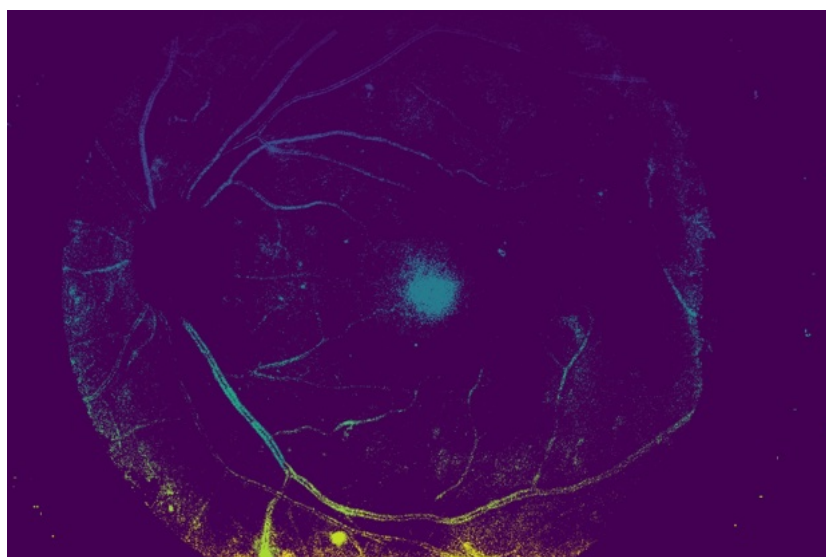
Rysunek 5.10: Porównanie histogramów obrazów przed i po wyrównaniu metodą CLAHE. Liczebność pikseli ciemnych jest nadal duża, co wynika z tła obszaru dna oka. Wersja elektroniczna obrazu: <https://krzywicki.pro/fig-5-10>.



Rysunek 5.11: Efekt korekty oświetlenia i kontrastu po dokonaniu łączonych transformacji morfologicznych. Wersja elektroniczna obrazu: <https://krzywicki.pro/fig-5-11>.



Rysunek 5.12: Obraz po przetworzeniu za pomocą kroku przetwarzania końcowego. Region plamki żółtej jest wyraźnie zauważalny. Wersja elektroniczna obrazu: <https://krzywicki.pro/fig-5-12>.



Rysunek 5.13: Oznaczone regiony będące kandydatami na plamkę żółtą. Każdy region ma przypisany inny kolor zgodny z wyznaczonym identyfikatorem. Wersja elektroniczna obrazu: <https://krzywicki.pro/fig-5-13>.



Rysunek 5.14: Wyznaczony środek regionu będącego plamką żółtą. Obraz w oryginalnym rozmiarze, będący końcowym efektem metody segmentującej plamkę żółtą. Wewnątrz okręgu znajduje się wyznaczona lokalizacja doteczka. Wersja elektroniczna obrazu: <https://krzywicki.pro/fig-5-14>.

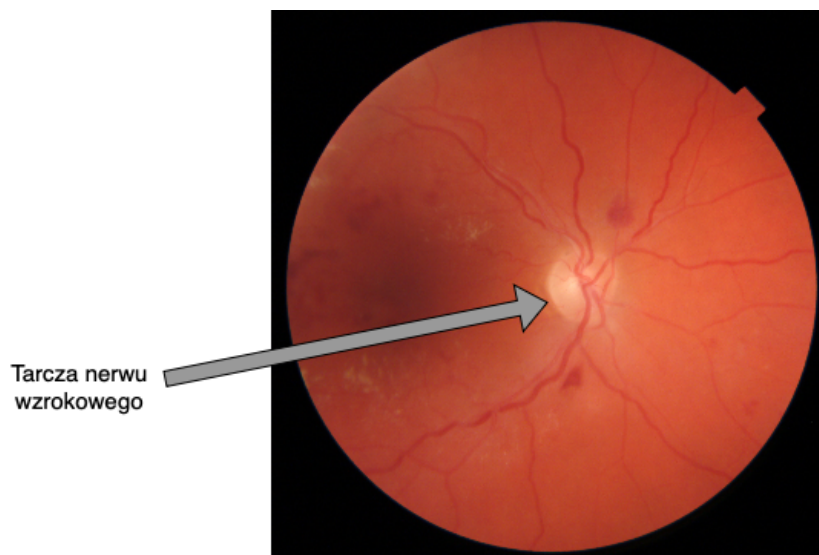


Rysunek 5.15: Wyznaczony przez algorytm region plamki żółtej na zdjęciu dna oka opisany okręgiem o promieniu najdłuższej przekątnej. Wewnątrz znajduje się wskazana przez algorytm lokalizacja dołeczka. Wersja elektroniczna obrazu: <https://krzywicki.pro/fig-5-15>.

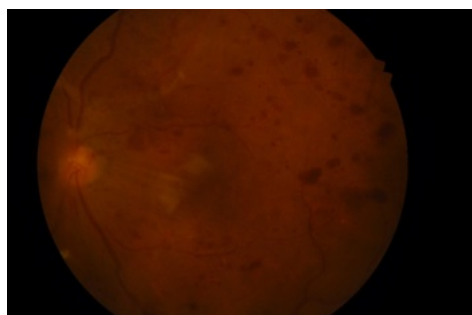
jakości, w szczególności nieprawidłowo oświetlonych, mogą występować różnice w kompozycji obiektu (Rysunek 5.17), gdzie obszar tarczy nerwu wzrokowego jest trudny do precyzyjnego ustalenia.

Lokalizacja tarczy nerwu wzrokowego na zdjęciu dna oka jest ściśle uzależniona od wyniku procesu segmentacji tarczy nerwu wzrokowego, którego opis przedstawiono w dalszej części podrozdziału. Segmentację tarczy nerwu wzrokowego na zdjęciu dna oka można podzielić na trzy kluczowe etapy: przetwarzanie wstępne, przetwarzanie końcowe oraz wybór najlepiej dopasowanego regionu (Rysunek 5.18).

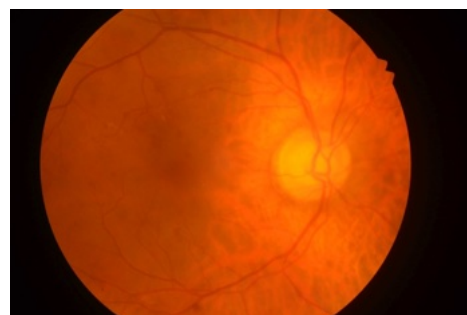
Etap przetwarzania wstępnego polega na przygotowaniu zdjęcia do procesu poszukiwania regionów kandydackich na tarczę nerwu wzrokowego i rozpoczyna się od konwersji oryginalnego zdjęcia reprezentowanego za pomocą modelu barw *RGB* do dwóch kolejnych modeli: *CIELab* oraz *YUV*. Oba modele barw uwzględniają różnice w jasności i kolorze, co czyni je wrażliwymi na zmiany tych czynników i może być korzystne w zadaniu segmentacji tarczy nerwu wzrokowego z uwagi na jej teksturę. Zauważono, że tarcza nerwu wzrokowego odznacza się wyraźnym kontrastem na warstwie pierwszej modelu *CIELab*, drugiej modelu *YUV* oraz drugiej modelu *RGB*, co



Rysunek 5.16: Tarcza nerwu wzrokowego na zdjęciu dna oka. Wersja elektroniczna obrazu: <https://krzywicki.pro/fig-5-16>.

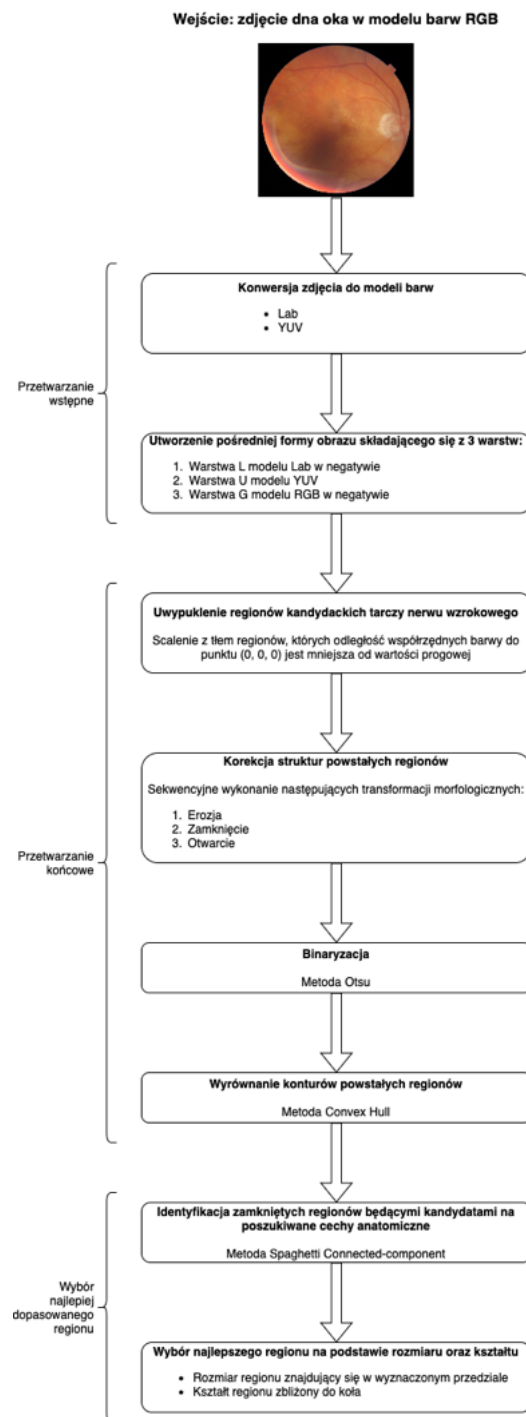


(a) Niedoświetlone zdjęcie dna oka. Tarcza nerwu wzrokowego jest zbyt ciemna do precyzyjnego ustalenia pełnego obszaru.

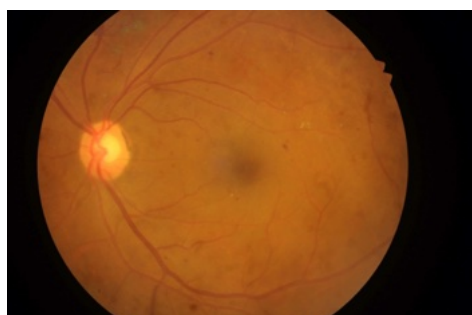


(b) Prześwietlone zdjęcie dna oka. Siatkówka w okolicy tarczy nerwu wzrokowego jest zbyt jasna, co czyni jej teksturę zbyt podobną do łą.

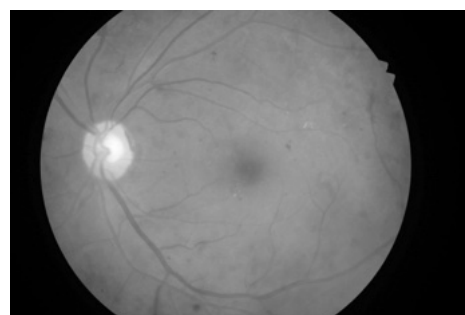
Rysunek 5.17: Wpływ nieprawidłowego oświetlenia dna oka na widoczność obszaru tarczy nerwu wzrokowego. Wersja elektroniczna obrazu: <https://krzywicki.pro/fig-5-17>.



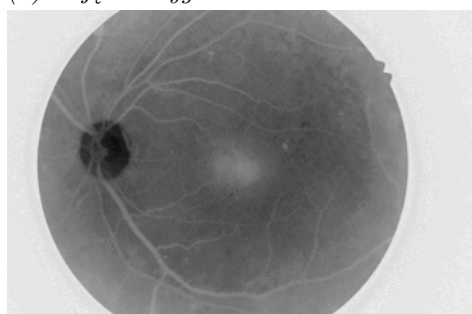
Rysunek 5.18: Szczegółowy schemat działania metody segmentującej region tarczy nerwu wzrokowego na zdjęciu dna oka. Wersja elektroniczna obrazu: <https://krzywicki.pro/fig-5-18>.



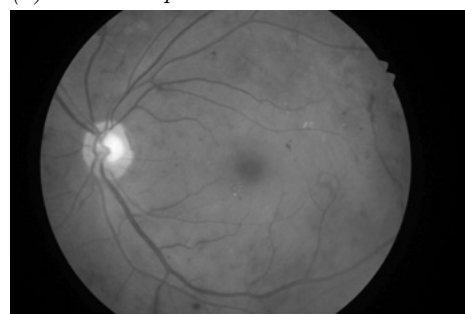
(a) Zdjęcie oryginalne.



(b) Warstwa pierwsza modelu CIE Lab.



(c) Warstwa druga modelu YUV.

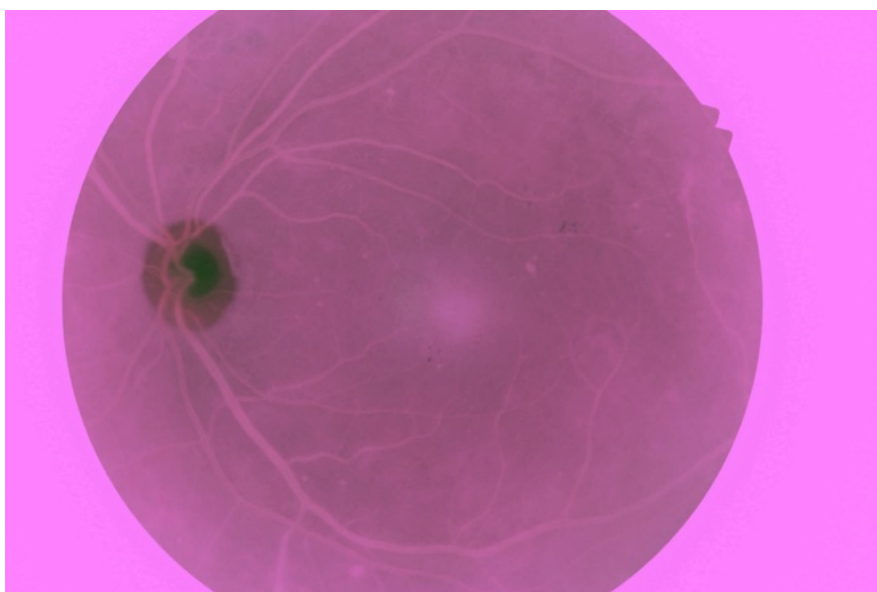


(d) Warstwa druga modelu RGB.

Rysunek 5.19: Tarcza nerwu wzrokowego reprezentowana za pomocą wybranych warstw modeli barw. Wersja elektroniczna obrazu: <https://krzywicki.pro/fig-5-19>.

przedstawiono na Rysunku 5.19. Wybrane warstwy reprezentują odpowiednio jasność, chrominację barwy niebieskiej oraz nasycenie barwy zielonej. Efektem końcowym kroku przetwarzania wstępnego jest pośredni obraz przedstawiony na Rysunku 5.20, który składa się z trzech wspomnianych warstw. Można zauważyć, że obszar tarczy nerwu wzrokowego jest wyraźnie ciemniejszy od innych struktur dna oka, co oznacza że wartości pikseli są bliskie współrzędnym punktu $(0, 0, 0)$ reprezentującego barwę czarną.

Celem etapu przetwarzania końcowego metody segmentacji obszaru tarczy nerwu wzrokowego jest wyznaczenie regionów kandydackich na tarczę nerwu wzrokowego. Zdjęcie dna oka będące efektem etapu przetwarzania wstępnego (będące w formie pośredniej) zostaje poddane operacji scalania z tłem szumu o teksturze podobnej do tarczy nerwu wzrokowego. W tym celu wszystkim pikselom mniejszym od wartości Th przypisano wartość 0. Jako Th przyjęto przeskalowany kontrast zdjęcia dna oka w formie oryginalnej, co przedstawiono w Równaniu 5.5. Efekt działania operacji można zauważyć na Rysunku 5.21. W kolejnym kroku następuje

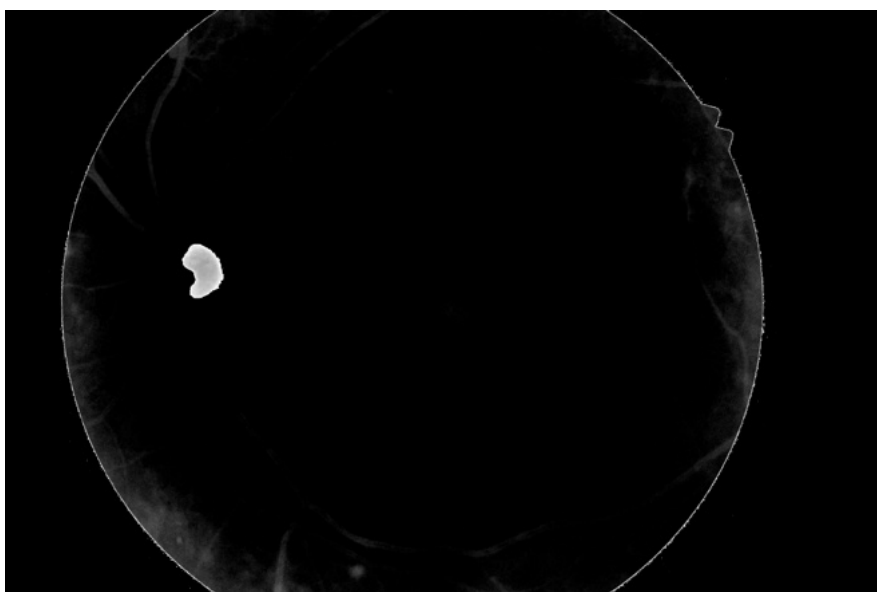


Rysunek 5.20: Pośrednia forma zdjęcia dna oka uwypuklająca tarczę nerwu wzrokowego. Wersja elektroniczna obrazu: <https://krzywicki.pro/fig-5-20>.

korekta struktur powstałych regionów poprzez zastosowanie sekwencyjnych transformacji morfologicznych [Ser83, HSZ87]: *erozji*, *domknięcia* i *otwarcia* odpowiadających za wygładzenie struktur regionów oraz za usuwanie szumów wewnątrz i wokół nich (Równanie 5.6). Przetworzony obraz będący efektem transformacji morfologicznych przedstawiono na Rysunku 5.22, na którym można dostrzec wyraźny region stanowiący część oryginalnej tarczy nerwu wzrokowego oraz mniejszą ilość szumów wokół. Następnie, obraz jest poddawany binaryzacji przy pomocy *metody Otsu* [Ots79] (Rysunek 5.23), co znacznie uwypukla regiony kandydackie na tarczę nerwu wzrokowego. Finalnie, wyznaczone zamknięte regiony kandydackie są wygładzane poprzez wyznaczenie wypukłych powłok na podstawie wierzchołków krawędzi metodą *Convex hull* [Skl82] oraz wyrównanie tekstury wewnątrz przy użyciu algorytmu *Flood fill* [Smi79]. Efekt końcowy operacji na etapie przetwarzania końcowego przedstawiono na Rysunku 5.24, gdzie można zauważyć, że dzięki operacji wygładzania, struktury regionów bardziej przypominają oryginalną tarczę nerwu wzrokowego.

$$contrast = \frac{\max' - \min'}{\max' + \min'}, \quad (5.5)$$

gdzie:



Rysunek 5.21: Pośrednia forma zdjęcia dna oka wypuklająca tarczę nerwu wzrokowego po dokonaniu spłaszczenia regionów przypominających teksturę tarczę nerwu wzrokowego. Wersja elektroniczna obrazu: <https://krzywicki.pro/fig-5-21>.

- $\max' = \max(p) - \text{margin}$,
- $\min' = \min(p) + \text{margin}$,
- p – pierwsza warstwa zdjęcia dna oka reprezentowanego za pomocą modelu barw RGB,
- \max , \min – funkcje zwracające odpowiednio największą i najmniejszą wartość piksela obrazu będącego parametrem,
- margin – margines zakresu barw na zdjęciu dobrany empirycznie.

$$p_{\text{stage}22} = \text{erode}(\text{close}(\text{open}(p_{\text{stage}21}))), \quad (5.6)$$

gdzie:

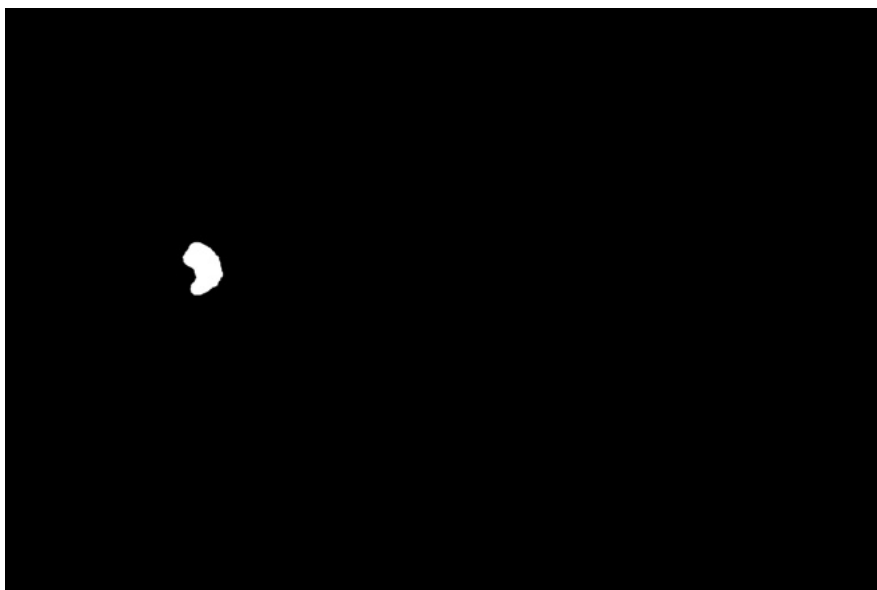
- $p_{\text{stage}21}$ – obraz będący formą pośrednią zdjęcia dna oka po dokonaniu spłaszczenia (wynik kroku pierwszego etapu drugiego algorytmu segmentującego)



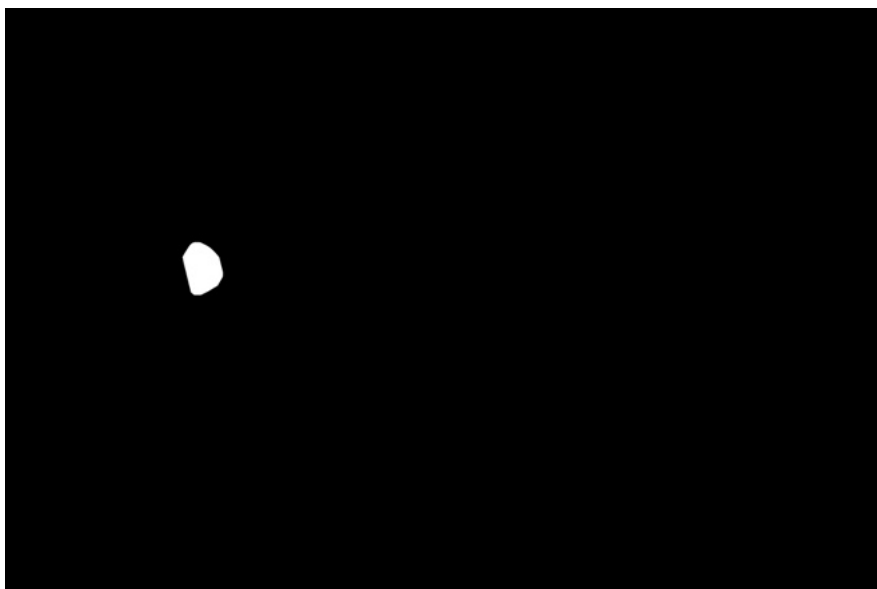
Rysunek 5.22: Odszumiona pośrednia forma zdjęcia dna oka uwypuklająca tarczę nerwu wzrokowego po dokonaniu spłaszczania regionów przypominających teksturę tarczę nerwu wzrokowego. Wersja elektroniczna obrazu: <https://krzywicki.pro/fig-5-22>.

- *erode* – funkcja erozji morfologicznej z wykorzystaniem dyskowego elementu strukturalnego,
- *close* – funkcja domknięcia morfologicznego z wykorzystaniem kwadratowego elementu strukturalnego,
- *open* – funkcja otwarcia morfologicznego z wykorzystaniem kwadratowego elementu strukturalnego.

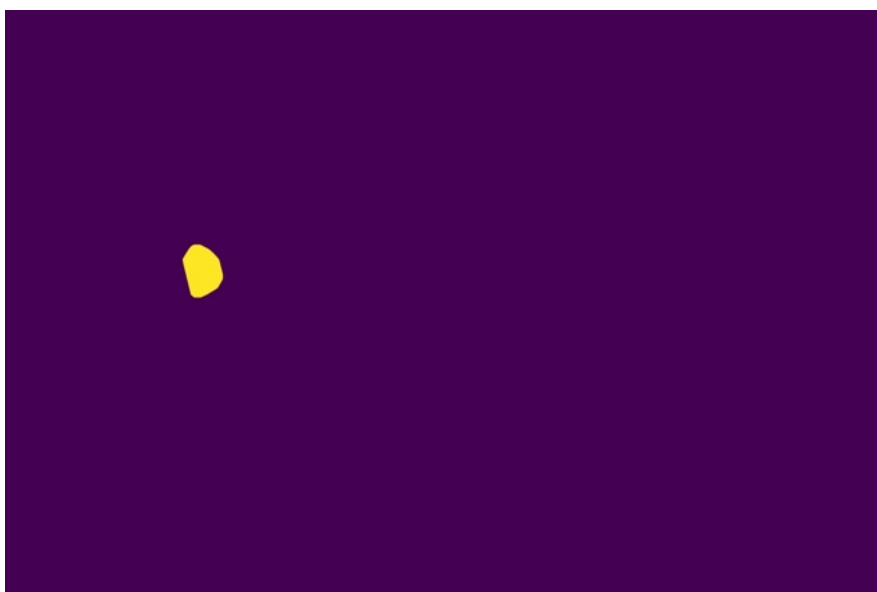
Wybór najlepiej dopasowanego regionu stanowi ostatni etap algorytmu segmentacji tarczy nerwu wzrokowego na zdjęciu dna oka. Upřednio dokonywany jest proces identyfikacji i oznaczania wszystkich regionów w postaci zamkniętych obszarów przy użyciu algorytmu *Spaghetti Labelling* [BABG20], w wyniku czego każdemu regionowi kandydackiemu na obrazie przydzielany jest unikalny identyfikator (Rysunek 5.25). Następnie każdy z oznaczonych regionów otrzymuje przypisany współczynnik kształtu (Równanie 5.4), który przyjmuje wartości w zakresie od 0 do 1, gdzie im bardziej wartość współczynnika zbliża się do 1 tym bardziej kształt kształt staje się zbliżony do kolistego. Region o powierzchni pasującej do wyznaczonego przedziału oraz o największej wartości współczynnika



Rysunek 5.23: Region kandydacki na tarczę nerwu wzrokowego po operacji binaryzacji metodą Otsu. Wersja elektroniczna obrazu: <https://krzywicki.pro/fig-5-23>.



Rysunek 5.24: Obraz będący efektem etapu przetwarzania końcowego algorytmu segmentującego region tarczy nerwu wzrokowego. Wersja elektroniczna obrazu: <https://krzywicki.pro/fig-5-24>.



Rysunek 5.25: Oznaczony region będący kandydatem na tarczę nerwu wzrokowego.
Wersja elektroniczna obrazu: <https://krzywicki.pro/fig-5-25>.

kolistości zostaje wybrany jako najlepiej odzwierciedlający tarczę nerwu wzrokowego.

Proces lokalizacji współrzędnych środka tarczy nerwu wzrokowego jest ściśle uzależniony od operacji segmentacji jej regionu na zdjęciu dna oka. Centrum tarczy nerwu wzrokowego jest wyznaczone poprzez obliczenie współrzędnych środka koła opisanego na strukturze regionu tarczy nerwu wzrokowego, o promieniu równym najdłuższej przekątnej, co przedstawiono na Rysunku 5.26. Końcowy efekt lokalizacji tarczy nerwu wzrokowego zaznaczonego na oryginalnym zdjęciu dna oka przedstawiono na Rysunku 5.27.

Parametry zewnętrzne algorytmu

Niektóre kroki w proponowanym algorytmie *MSiRL*, będącym przedmiotem analizy w tym podrozdziale, wymagają doprecyzowania sposobu działania poprzez zastosowanie zewnętrznych parametrów. W Tabeli 5.1 przedstawiono listę parametrów wraz z ich symbolami, opisami i proponowanymi wartościami domyślnymi dla bazy *IDRID*. Wartości wszystkich rozmiarów przedstawiono w pikselach. Warto zauważyć, że parametry *MT1* oraz



Rysunek 5.26: Wyznaczony środek regionu będącego tarczą nerwu wzrokowego. Obraz w oryginalnym rozmiarze, będący końcowym efektem metody segmentującej tarczę nerwu wzrokowego. Wewnątrz okręgu znajduje się wyznaczona lokalizacja środka tarczy nerwu wzrokowego. Wersja elektroniczna obrazu: <https://krzywicki.pro/fig-5-26>.



Rysunek 5.27: Wyznaczony przez algorytm region tarczy nerwu wzrokowego na zdjęciu dna oka opisany okręgiem o promieniu najdłuższej przekątnej. Wewnątrz znajduje się wskazana przez algorytm lokalizacja środka tarczy nerwu wzrokowego. Wersja elektroniczna obrazu: <https://krzywicki.pro/fig-5-27>.

MT_2 , odpowiadające za minimalną uznaną wartość współczynnika kolistości, mogą wpłynąć na wyniki pokrycia oraz dokładności działania algorytmu. Wysoka wartość MT_1 i MT_2 może zwiększyć dokładność lokalizacji środków cech anatomicznych na zdjęciu dna oka tym samym zmniejszając pokrycie poprzez zignorowanie regionów odbiegających kształtem od idealnego koła. Niska wartość MT_1 i MT_2 może wpłynąć pozytywnie na wynik pokrycia poprzez akceptację kształtów w mniejszym stopniu przypominających swą strukturą koło, lecz także negatywnie na dokładność lokalizacji środków cech anatomicznych za sprawą potencjalnie niepoprawnie wybranych najlepszych regionów.

Kolejnymi istotnymi parametrami, na które warto zwrócić uwagę, są \min_{P_1} , \max_{P_1} , \min_{P_2} i \max_{P_2} odpowiadające zakresowi rozmiarów regionów kandydackich, branych pod uwagę w wyborze najlepszego z nich. Podczas doboru ich wartości należy kierować się nie tylko rzeczywistym rozmiarem obrazu, ale także faktem że podczas segmentacji może zostać oznaczony jedynie fragment rzeczywistego obszaru obiektu, bądź obiekt wraz z otoczeniem, co może mieć źródło w nierównomiernym oświetleniu zdjęcia. Rekomendowaną metodą doboru tych parametrów jest metoda

LP	Nazwa	Symbol	Zadanie	Krok algorytmu	Wartość domyślna
1	Rozmiar filtru rozmycia gaussowskiego	$GF1$	FS	Redukcja szumu	5x5
2	Promień dyskowego elementu strukturalnego łączonej transformacji morfologicznej	$JMT1$	FS	Uwypuklanie obszaru plamki żółtej	9
3	Stopień wyrównania histogramu	$CL1$	FS	Wyrównanie histogramu zdjęcia	2
4	Rozmiar części zdjęcia podlegających adaptywnemu wyrównaniu histogramu	$TGS1$	FS	Wyrównanie histogramu zdjęcia	4x4
5	Minimalny rozmiar przekątnej regionu kandydackiego	\min_{P1}	FS	Wybór najlepiej dopasowanego regionu	200
6	Maksymalny obszar regionu kandydackiego	\max_{P1}	FS	Wybór najlepiej dopasowanego regionu	800 000
7	Minimalna akceptowalna wartość współczynnika kolistości	$MT1$	FS	Wybór najlepiej dopasowanego regionu	0.1
8	Wartość progu operacji scalania z tłem szumu o teksturze podobnej do tarczy nerwu wzrokowego	Th	ODS	Korekcja kształtów powstałych obiektów	Równanie 5.5
9	Wartość marginesu Równania 5.5	$Margin$	ODS	Korekcja kształtów	25
10	Promień dyskowego elementu strukturalnego sekwencyjnej transformacji morfologicznej (erozja)	$SMT1$	ODS	Korekcja kształtów powstałych obiektów	9
11	Bok kwadratowego elementu strukturalnego sekwencyjnej transformacji morfologicznej (domknięcie)	$SMT2$	ODS	Korekcja kształtów powstałych obiektów	5
12	Bok kwadratowego elementu strukturalnego sekwencyjnej transformacji morfologicznej (otwarcie)	$SMT3$	ODS	Korekcja kształtów powstałych obiektów	5
13	Minimalny rozmiar przekątnej regionu kandydackiego	\min_{P2}	ODS	Wybór najlepiej dopasowanego regionu	150
14	Maksymalny obszar regionu kandydackiego	\max_{P2}	ODS	Wybór najlepiej dopasowanego regionu	800 000
15	Minimalna akceptowalna wartość współczynnika kolistości	$MT2$	ODS	Wybór najlepiej dopasowanego regionu	0.3

Tabela 5.1: Parametry metody *MSiRL*. FS – proces segmentacji plamki żółtej, ODS – proces segmentacji tarczy nerwu wzrokowego.

przeszukiwania siatki polegająca na sprawdzeniu wszystkich możliwych kombinacji parametrów i wyborze tej, która zapewnia najlepsze wyniki na pewnej próbie w postaci bazy zawierającej testowe zdjęcia dna oka o zbliżonym rozmiarze oraz wykonane w podobnych warunkach oświetleniowych, jakie miały miejsce podczas wykonywania zdjęć właściwie analizowanych przez algorytm. Alternatywnie, celem minimalizacji zużycia zasobów oraz czasu, można skorzystać z metod optymalizacji procesu poszukiwania parametrów zewnętrznych, takich jak omówionych w pracy Bischla i innych [BBL⁺23].

5.4 Metoda Symultanicznej Lokalizacji Współrzędnych Środków Dołeczka i Tarczy Nerwu Wzrokowego na Zdjęciach Dna Oka przy użyciu Zespołu Sieci Neuronowych

5.4.1 Motywacja

Rozpoczęcie prac nad *Metodą Symultanicznej Lokalizacji Współrzędnych Środków Dołeczka i Tarczy Nerwu Wzrokowego na Zdjęciach Dna Oka przy użyciu Zespołu Sieci Neuronowych (MSLW)* wynikało z potrzeby znalezienia rozwiązania lokalizującego cechy anatomiczne na zdjęciu dna oka o znacznie wyższej skuteczności, niż metoda *MSiRL* (implementacja w Podrozdziale B.1 w Dodatku B). W kontrze do rozwiązań szybkich i wydajnych obliczeniowo konieczne są rozwiązania cechujące się wysoką jakością działania, co można zauważyć jako przewagę liczebnościową w obecnym stanie literatury naukowej dotyczącym zagadnień lokalizacji środków dołeczka oraz tarczy nerwu wzrokowego na zdjęciach dna oka. Możliwości obliczeniowe współczesnego sprzętu komputerowego są w zupełności wystarczające do uruchomienia kosztownych rozwiązań (przykładowo, wykorzystujących głębokie sieci neuronowe), których użycie może przyczynić się do dokładniejszej diagnostyki klinicznej. Sieci neuronowe do uzyskania zdolności dokładnej lokalizacji skomplikowanych wzorców wymagają jednak procesu treningu na obszernej i dokładnie oznaczonej bazie danych, co wiąże się z kosztownymi czynnikami w postaci:

- uzyskanie dużej ilości danych – w przypadku projektowania systemów pod kątem środowisk produkcyjnych i klinicznych – minimum 10000 zdjęć [ALE⁺16],
- oznaczenie danych wymagające zaangażowania specjalistów dziedzinowych,
- trening – proces wymagający wydajnych zasobów obliczeniowych, który może trwać długi czas i konsumować znaczne zasoby energetyczne.

Proponowane rozwiązanie w postaci metody *MSLW* wykorzystuje innowacyjne podejście polegające na wykorzystaniu uczenia zespołowego za pomocą metody *Pasting* [Bre99], która stanowi modyfikację popularnej metody *Bagging* *Bagging* [Bre96], w postaci agregacji lokalizacji wzorców przydzielonych przez 16 sieci neuronowych wytrenowanych niezależnie, co zaowocowało dokładniejszymi wynikami, w porównaniu z obecnym stanem literatury. Co więcej, metoda *MSLW* może być rozszerzana za pomocą innej architektury lub większej liczby sieci neuronowych biorących udział w agregacji lokalizacji cech, a także za pomocą innych metod przetwarzania wstępnego danych wejściowych.

5.4.2 Zasada działania

Celem metody *MSLW* jest lokalizacja współrzędnych środków dołączka oraz tarczy nerwu wzrokowego na zdjęciu dna oka. Działanie proponowanego rozwiązania jest procesem dwuetapowym polegającym na przygotowaniu rozwiązania na podstawie wyznaczonej bazy zawierającej zdjęcia dna oka oraz na jego wykorzystaniu.

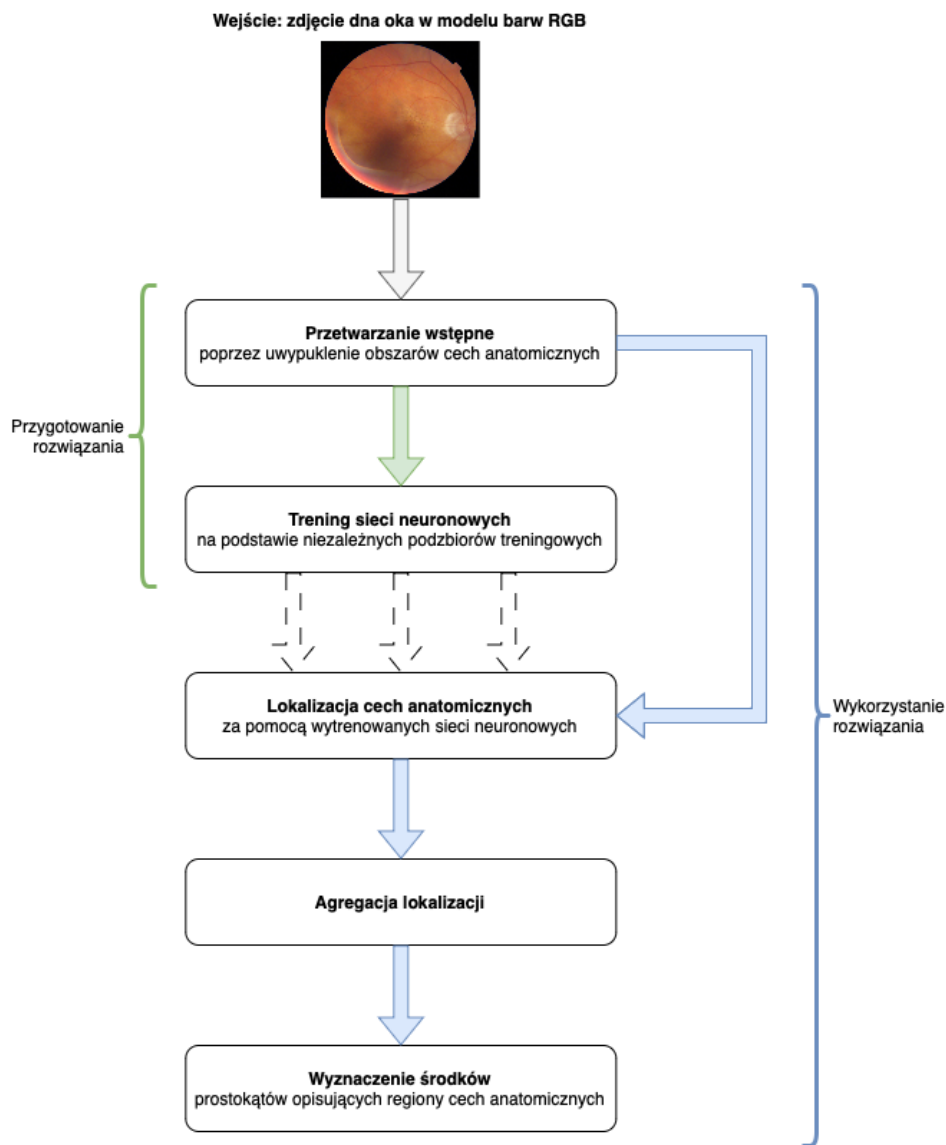
Etap przygotowania rozwiązania do późniejszego wykorzystania polega na przetworzeniu wstępne danych w postaci zdjęć dna oka, co ma na celu uwypuklenie cech anatomicznych oraz na wytrenowaniu modeli bazowych (sieci neuronowych, Podrozdział 3.5.2) na tych danych. Wykorzystanie przygotowanej metody *MSLW* polega na agregacji lokalizacji cech anatomicznych dna oka przydzielonych przez wytrenowane modele bazowe dla zdjęć wstępnie przetworzonych, a następnie na wyznaczeniu środków każdej z cech. Pełny schemat działania metody *MSLW* przedstawiono na Rysunku 5.28. Można zauważyć, że wszystkie kroki w przygotowaniu

oraz działaniu proponowanego rozwiązania są wspólne zarówno dla zadania lokalizacji środka dołączka, jak i tarczy nerwu wzrokowego. Etap wstępnego przetwarzania wejściowych zdjęć, uwypuklający cechy anatomiczne dna oka stanowiące obiekty zainteresowania, jest wspólny zarówno dla etapu przygotowania metody oraz jej wykorzystania. Wyznaczanie lokalizacji cech anatomicznych jest realizowane symultanicznie przez wytrenowane niezależnie modele bazowe. Wyjściem proponowanej metody są ważne geometryczne środki ramek opisujących regiony obrazu w postaci cech anatomicznych dna oka. Szczegóły implementacji metody *MSLW* znajdują się w Podrozdziale B.2 w Dodatku B. W procesie implementacji wykorzystano język programowania *Python* [Pyt23] wraz z bibliotekami *OpenCV* [Bra00] i *Ultralytics* [JCQ23]. W dalszej części podrozdziału omówiono szczegółowo etapy pracy metody *MSLW*.

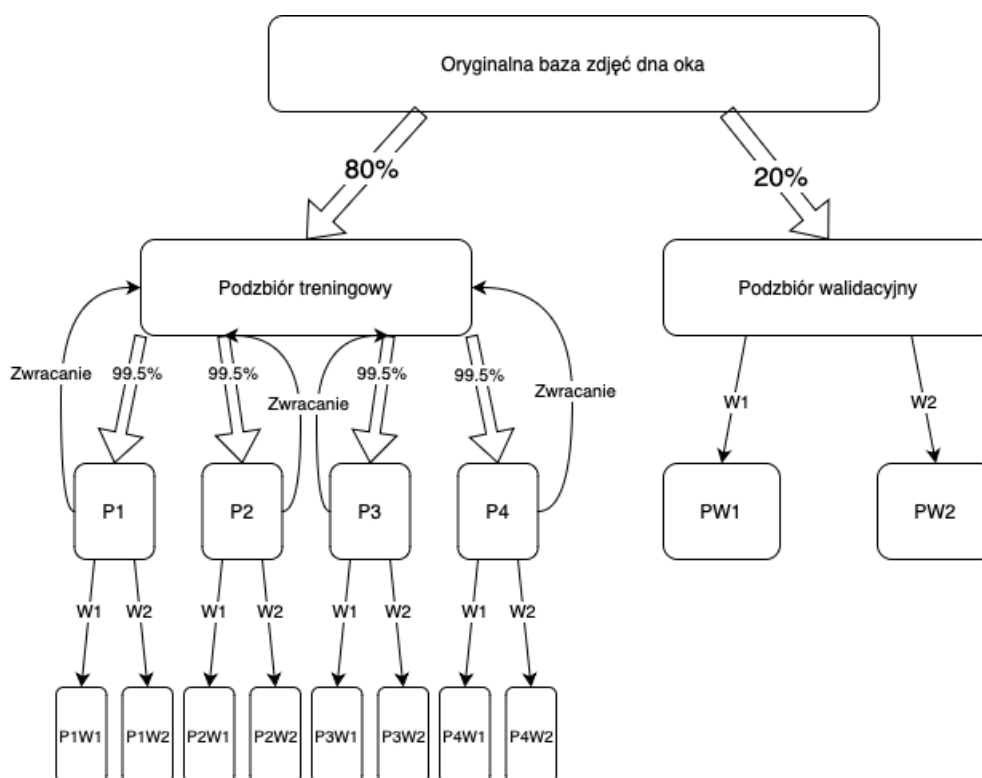
Wykorzystane dane

Na potrzeby etapu przygotowania metody *MSLW* wykorzystano bazę *BFO21*. Dla celów treningu oraz wewnętrznej oceny jakości wytrenowania (walidacji) modeli bazowych wykorzystano 3565, pochodzących z bazy *BFO21*, zdjęć dna oka z zaznaczonymi regionami plamki żółtej oraz tarczy nerwu wzrokowego w postaci ramek opisujących cechy anatomiczne będące przedmiotem zainteresowania. Mniejsza liczebność wykorzystanych zdjęć (w porównaniu z pełną liczebnością bazy *BFO21*) wynikała z niespełnionych kryteriów jakości części obrazów znajdujących się w bazie oraz z całkowitego braku widoczności cech anatomicznych. Warto także zwrócić uwagę na fakt, że wykorzystana baza zdjęć dna oka była mniej liczna, niż wspomniane minimum stosowane w rozwiązaniach produkcyjnych i klinicznych, czego przyczyną były w trudności w pozyskaniu zdjęć spełniających kryteria jakości, centracji, a także czasochłonność procesu oznaczania regionów cech anatomicznych.

Wybrane zdjęcia z bazy *BFO21* podzielono pseudolosowo bez zwracania (rozłącznie) w stosunku 80% : 20%, który odpowiadał odpowiednio podzbirowi treningowemu przeznaczonemu do treningu modeli bazowych oraz walidacyjnemu przeznaczonemu do przeprowadzenia wewnętrznych testów podczas treningu. Z podzbioru treningowego wydzielono pseudolosowo ze zwracaniem cztery podzbiory o proporcjach wynoszących 99.5% liczebności podzbioru treningowego, czego celem był trening modeli bazowych na podstawie niezależnych podzbiorów niezawierających powtórzeń. W kroku



Rysunek 5.28: Schemat działania proponowanej metody MSLW. Zdjęcie jest przetwarzane i analizowane pod kątem lokalizacji doteczka i tarczy nerwu wzrokowego za pomocą wspólnych operacji. Wersja elektroniczna obrazu: <https://krzywicki.pro/fig-5-28>.



Rysunek 5.29: Metoda podziału danych treningowych na potrzeby treningu metody MSLW. Podzbiór treningowy jest wielokrotnie dzielony na mniejsze podzbiory celem wytrenowania niezależnych sieci neuronowych. P – podzbiór, W – wariant przetwarzania, P_nW_m – n -ty podzbiór przetworzony m -tym wariantem. Wersja elektroniczna obrazu: <https://krzywicki.pro/fig-5-29>.

przetwarzania wstępnego (omówionego w dalszej części podrozdziału) zdjęcia treningowe przetworzono na dwa sposoby, celem ukazania cech anatomicznych będących przedmiotem zainteresowania pod różnymi kątami, czego efektem było podwojenie każdego z podzbiorów. Pełny proces podziału danych treningowych zaprezentowano schematycznie Rysunku 5.29 oraz ilościowo w Tabeli 5.2. Warto zauważyć, że spośród powstałych 13 podzbiorów z przeznaczeniem treningowym, jedynie osiem posłużyło do właściwego wytrenowania 16 modeli bazowych w postaci sieci neuronowych. Pozostałe pięć podzbiorów posłużyło jako baza do utworzenia podzbiorów użytych podczas uczenia modeli bazowych. Do procesu walidacji przeznaczono trzy podzbiory, z czego dwa służyły do bezpośredniej oceny jakości treningu modeli bazowych trenowanych na dwóch wariantach przetwarzania wstępnego zdjęć dna oka oraz jeden jako baza do utworzenia dwóch docelowych podzbiorów.

LP	Nazwa / symbol podzbioru	Sposób utworzenia	Liczba zdjęć	Właściwe przeznaczenie
1	Oryginalna baza	Wyodrębnienie zdjęć z bazy BFO21 z oznaczonym minimum jednym obiektem	3565	Baza do podziału na podzbiór treningowy oraz walidacyjny
2	Podzbiór treningowy	80% obiektów z oryginalnej bazy	2852	Baza do podziału na cztery podzbiory treningowe
3	Podzbiór walidacyjny	20% obiektów z oryginalnej bazy	713	Baza do podziału na dwa podzbiory walidacyjne dla każdego wariantu przetwarzania wstępnego
4	Podzbiór 1 (P1)	99.5% obiektów z podzbioru treningowego	2837	Baza do duplikowania za pomocą dwóch wariantów przetwarzania wstępnego
5	Podzbiór 2 (P2)	99.5% obiektów z podzbioru treningowego	2837	Baza do duplikowania za pomocą dwóch wariantów przetwarzania wstępnego
5	Podzbiór 3 (P3)	99.5% obiektów z podzbioru treningowego	2837	Baza do duplikowania za pomocą dwóch wariantów przetwarzania wstępnego
6	Podzbiór 4 (P4)	99.5% obiektów z podzbioru treningowego	2837	Baza do duplikowania za pomocą dwóch wariantów przetwarzania wstępnego
7	Podzbiór walidacyjny wariant 1 (PW1)	Podzbiór walidacyjny przetworzony wariantem 1.	713	Walidacja sieci trenowanych. podzbiorymi o wariacie 1
8	Podzbiór walidacyjny wariant 2 (PW2)	Podzbiór walidacyjny przetworzony wariantem 2.	713	Walidacja sieci trenowanych. podzbiorymi o wariacie 2
9	Podzbiór treningowy 1 wariant 1 (P1W1)	Podzbiór treningowy 1 przetworzony wariantem 1.	2837	Trening sieci za pomocą podzbioru 1 o wariacie 1
10	Podzbiór treningowy 1 wariant 2 (P1W2)	Podzbiór treningowy 1 przetworzony wariantem 2.	2837	Trening sieci za pomocą podzbioru 1 o wariacie 2.
11	Podzbiór treningowy 2 wariant 1 (P2W1)	Podzbiór treningowy 2 przetworzony wariantem 1.	2837	Trening sieci za pomocą podzbioru 2 o wariacie 1
12	Podzbiór treningowy 2 wariant 2 (P2W2)	Podzbiór treningowy 2 przetworzony wariantem 2.	2837	Trening sieci za pomocą podzbioru 2 o wariacie 2
12	Podzbiór treningowy 3 wariant 1 (P3W1)	Podzbiór treningowy 3 przetworzony wariantem 1.	2837	Trening sieci za pomocą podzbioru 3 o wariacie 1
13	Podzbiór treningowy 3 wariant 2 (P3W2)	Podzbiór treningowy 3 przetworzony wariantem 2.	2837	Trening sieci za pomocą podzbioru 3 o wariacie 2
14	Podzbiór treningowy 4 wariant 1 (P4W1)	Podzbiór treningowy 4 przetworzony wariantem 1.	2837	Trening sieci za pomocą podzbioru 4 o wariacie 1
15	Podzbiór treningowy 4 wariant 2 (P4W2)	Podzbiór treningowy 4 przetworzony wariantem 2.	2837	Trening sieci za pomocą podzbioru 4 o wariacie 2

Tabela 5.2: Szczegóły podzbiorów bazy BFO21 wykorzystywanych na etapie przygotowywania metody MSLW.

W celu dokonania zewnętrznych testów zaproponowanego rozwiązania wykorzystano testowy podzbiór bazy IDRID [PPK⁺18] składający się ze 103 zdjęć dna oka, dzięki czemu obrazy wykorzystywane podczas treningu nie rzutowały bezpośrednio na wyniki uzyskane przez metodę MSLW. Warto również mieć na uwadze fakt, że dane treningowe oraz testowe pochodziły z całkowicie rozłącznych populacji, dzięki czemu wyniki testów proponowanego rozwiązania są wiarogodne – zdjęcia w obydwu bazach oraz

ich cechy nie były od siebie zależne.

Przetwarzanie wstępne zdjęć

Celem wszechstronnego uwydatnienia cech anatomicznych dna oka zastosowano dwa warianty przetwarzania zdjęć wykorzystywanych zarówno podczas treningu modeli bazowych, jak i podczas wykorzystania gotowego rozwiązania:

1. poprawa kontrastu i oświetlenia za pomocą korekty kontrastu i oświetlenia (zastosowane w metodzie *MSiRL*, Podrozdział 5.3.2),
2. kanał zielony wydobyty ze zdjęcia przetworzonego wariantem 1.

W celu dokonania operacji korekty kontrastu oraz oświetlenia na zdjęciu dna oka dokonano konwersji zdjęcia do modelu barw *CIELab*, następnie korzystając z metody *CLAHE* wyrównano w sposób adaptacyjny histogram pierwszej warstwy, która jest odpowiedzialna za jasność i skonwertowano zdjęcie ponownie do modelu *RGB* (wariant 1). Wariant 2. polegał na ekstrakcji ze zdjęcia przetworzonego za pomocą wariantu 1. kanału zielonego, który odzwierciedla najwyraźniejsze właściwości kontrastowe zarówno tarczy nerwu wzrokowego, jak i plamki żółtej na tle siatkówki. Efekt zastosowanych metod przetwarzania wstępnego na zdjęciu dna oka przedstawiono na Rysunku 5.30. Można zauważyć, że na przetworzonych zdjęciach 5.30b i 5.30c obszary plamki żółtej i tarczy nerwu wzrokowego są wyraźniejsze, niż na zdjęciu oryginalnym 5.30a. Zdjęcie 5.30c z uwagi na celową jednowarstwowość prezentuje cechy anatomiczne wyżej kontrastujące z tłem (siatkówką), niż zdjęcie kolorowe 5.30b. Na zdjęciu 5.30b można jednak dostrzec większe różnice w samej teksturze cech anatomicznych względem siatkówki, co stanowi zasługę zachowanej informacji o kolorach.

Architektura zastosowanych sieci neuronowych

W metodzie *MSLW* wykorzystano sieci neuronowe o architekturze *YOLOv8* [RKHD23] jako modele bazowe, których przeznaczeniem jest detekcja obiektów na obrazach. Architektura *YOLOv8* jest dostępna w pięciu wariantach [JCQ23]:



(a) Zdjęcie oryginalne.



(b) Zdjęcie przetworzone wariantem 1.



(c) Zdjęcie przetworzone wariantem 2.

Rysunek 5.30: Proces przetwarzania wstępnego zdjęć dna oka w dwóch wariantach.
Wersja elektroniczna obrazu: <https://krzywicki.pro/fig-5-30>.

- n – nano,
- s – small,
- m – medium,
- l – large,
- x – extra large,

które różnią się od siebie złożonością, liczbą parametrów, szybkością działania i dokładnością detekcji.

Zastosowana architektura *YOLOv8* przyjmuje wejście w postaci obrazu kolorowego o rozmiarach 640x640 px, co wymaga uprzedniego przeskalowania danych. W metodzie *MSLW* zastosowano 16 sieci neuronowych w wariancie x , które charakteryzowały się największą liczbą parametrów, co rzutowało na wysoką złożoność rozwiązania, długi czas treningu i zastosowania oraz wysoką konsumpcję zasobów obliczeniowych i energii. Złożoność pojedynczej sieci wynosi 257.8 miliardów FLOPs¹ oraz 68.2 miliona parametrów.

Na zastosowanie architektury *YOLOv8* w metodzie *MSLW* wpłynął szereg czynników warunkujących przewagę rodziny architektur *YOLO* nad innymi architekturami sieci neuronowych przeznaczonymi do detekcji obiektów. Architekturę *YOLOv8* została zaprezentowana w styczniu 2023 roku, co sprawia że jest jedną z najnowszych wersji architektur z rodziny *YOLO*. Mimo, że przeprowadzone testy dokładności i wydajności [RKHD23, TCE23] architektury *YOLOv8* nie są jednoznaczne, zdecydowano się na zastosowanie jej z uwagi na wykorzystany szereg nowych rozwiązań i usprawnień w postaci sieci piramidy cech (*Feature Pyramid Network*) i sieci agregacji ścieżek (*Path Aggregation Network*), co może mieć znaczny wpływ na skuteczność lokalizacji wzorców w różnych skalach, oraz poprawioną wydajność obliczeniową względem wersji poprzednich. Rodzina architektur *YOLO* jest także przystosowana do symultanicznej lokalizacji na obrazach obiektów różnych klas, co znacznie przyspiesza proces jednoczesnej lokalizacji współrzędnych dołączka oraz tarczy nerwu wzrokowego. Jednym z czynników decydujących była także popularność architektur z rodziny *YOLO*, czego efektem jest ich duża liczba, liczne artykuły proponujące wszechstronne

¹FLOPs (Floating Point Operations) – jednostka miary używana do określania liczby operacji arytmetycznych na liczbach zmiennoprzecinkowych wykonywanych przez system lub urządzenie w jednostce czasu.

zastosowania, badające ich jakość i wydajność oraz szereg narzędzi w postaci bibliotek i implementacje rozwiązujące nietypowe problemy, z czym wiąże się także duże techniczne wsparcie społeczności.

Trening

Do procesu treningu modeli bazowych w postaci sieci neuronowych o architekturze *YOLOv8* wykorzystano metodę *Train & Test*, która polega na wyuczeniu modelu na podstawie danych treningowych, a następnie na dokonaniu testu na podstawie odrębnego zbioru danych testowych. Z uwagi na dużą czasochłonność procesu treningu i dostępne zasoby obliczeniowe, nie zdecydowano się na wykonanie często stosowanej walidacji krzyżowej [Sto74], która polega na wielokrotnym podziale danych treningowych oraz na wielokrotnym procesie treningu i testu każdego z modeli.

Na podstawie podzbiorów $P1W1$, $P1W2$, $P2W1$, $P2W2$, $P3W1$, $P3W2$, $P4W1$, $P4W2$ wytrenowano 16 modeli bazowych, z których każdy był przystosowany do lokalizacji wzorców charakteryzujących się wysokim kontrastem lub teksturą na tle siatkówki. Do walidacji procesu treningu zastosowano podzbiory $PW1$ i $PW2$, które przeznaczono dla sieci trenowanych podzbiorymi odpowiednio: $P1W1$, $P2W1$, $P3W1$, $P4W1$ oraz $P1W2$, $P2W2$, $P3W2$, $P4W2$. Powstałe modele bazowe wraz ze zbiorami wykorzystanymi w procesie treningu oraz z przeznaczeniem przedstawiono w Tabeli 5.3.

Podczas treningu 8 spośród 16 modeli bazowych zastosowano *uczenie transferowe* [YXL22] wraz z metodą *fine-tuning*, gdzie trenowane sieci neuronowe były wyuczone pierwotnie na zbiorze danych *COCO* [LMB⁺14] przeznaczonym do detekcji obiektów, składającym się z 1.5 miliona zdjęć oraz obiektów należących do 80 klas, a następnie dostosowane do lokalizacji cech anatomicznych dna oka. Parametry procesu trenowania każdego z modeli bazowych przedstawiono w Tabeli 5.4, które zaczerpnięto od twórców pierwszej publicznej implementacji architektury *YOLOv8* [YOL23].

Do procesu treningu wykorzystano serwer obliczeniowy znajdujący się na Wydziale Matematyki i Informatyki Uniwersytetu Warmińsko-Mazurskiego w Olsztynie o następującej konfiguracji:

- system operacyjny: Arch Linux [Arc23] z jądrem w wersji 6.2.2,

- procesor centralny (CPU): AMD Ryzen 9 3950X [AMD23a],
- rozmiar pamięci RAM: 64 GB,
- procesor graficzny (GPU): NVIDIA Tesla T4 [AMD23b] z pamięcią o rozmiarze 16 GB.

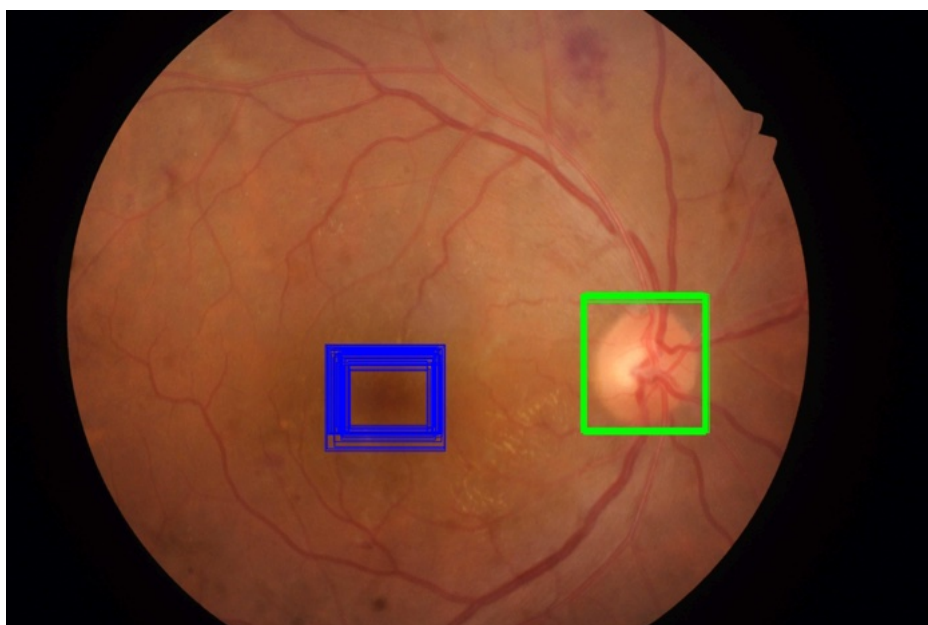
Sieci neuronowe będące przedmiotem analizy trenowano w środowisku uruchomieniowym o następującej konfiguracji:

- interpreter języka Python [Pyt23] w wersji 3.11,
- biblioteka OpenCV [Bra00] w wersji 4.7.0 przeznaczona do wstępnego przetwarzania zdjęć dna oka,
- biblioteka Ultralytics [JCQ23] w wersji 8.0.76 przeznaczona do wykorzystania sieci neuronowej o architekturze *YOLOv8*,
- zestaw bibliotek *Cuda Toolkit* [Cud23a] w wersji 12.0 przeznaczonych do wsparcia wykonywania obliczeń na procesorach graficznych zgodnych z technologią NVIDIA Cuda [NVI23],
- biblioteka *CuDNN* w wersji 8.9.3 przeznaczona do obsługi sieci neuronowych na procesorach graficznych zgodnych z technologią NVIDIA Cuda [NVI23].

Wszystkie modele bazowe wytrenowano korzystając ze wsparcia zrównoleglania obliczeń za pomocą GPU i technologii NVIDIA Cuda [NVI23]. Z uwagi na możliwości pamięciowe GPU, sieci neuronowe trenowano sekwencyjnie, co oznacza że w jednym momencie była trenowana tylko jedna sieć za pomocą równoległej optymalizacji obliczeń na GPU. Łączny czas ciągłego treningu 16 sieci neuronowych o architekturze *YOLOv8* wyniósł 228 godzin i 50 minut.

Agregacja lokalizacji i wyznaczenie finalnych współrzędnych środka

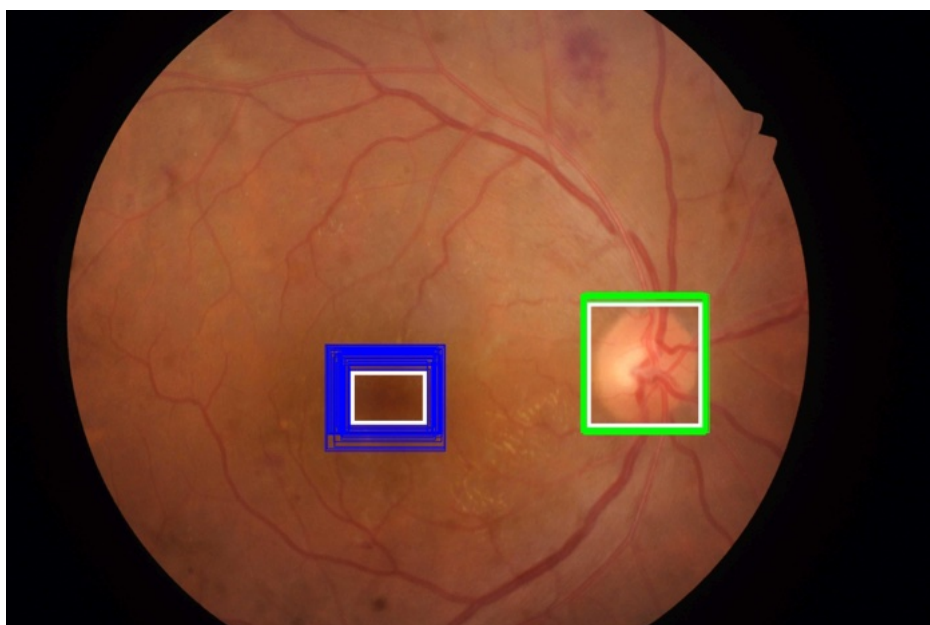
Agregacja lokalizacji oraz wyznaczenie finalnych współrzędnych środka to ostatnie kroki metody *MSLR*, których celem jest zwrócenie lokalizacji cech



Rysunek 5.31: Obszary plamki żółtej i tarczy nerwu wzrokowego oznaczone za pomocą ramek wyznaczonych przez 16 sieci neuronowych o architekturze YOLOv8. Kolorem niebieskim oznaczono plamkę żółtą, zielonym – tarczę nerwu wzrokowego. Wszystkie obszary wyznaczone przez sieci są do siebie zbliżone. Wersja elektroniczna obrazu: <https://krzywicki.pro/fig-5-31>.

anatomicznych dna oka. Za pomocą każdego z szesnastu wytrenowanych modeli bazowych w postaci sieci neuronowych o architekturze YOLOv8 wyznaczono lokalizacje plamki żółtej oraz tarczy nerwu wzrokowego na parach zdjęć dna oka przetworzonych za pomocą dwóch wariantów przetwarzania wstępnego (Rysunek 5.31). Następnie przefiltrowano wyznaczone ramki otaczające cechy zostawiając tylko te, dla których przypisany współczynnik ufności wynosił przydzieloną minimalną wartość progową Box_{Th_c} w obrębie każdej cechy i wyznaczono największe skupienie ramek dla każdej ze lokalizowanych cech anatomicznych, dla których można wyznaczyć część wspólną na obszarze dna oka (Rysunek 5.32). W przypadku gdy w wyniku przyjętej zbyt wysokiej wartości progu Box_{Th_c} żadna z ramek wyznaczonych przez model bazowy nie posiada przypisanego współczynnika ufności o wymaganym stopniu, brana jest z puli największego skupienia (jeżeli istnieje) jedna ramka o najwyższym współczynniku ufności.

Wyznaczenie finalnych współrzędnych środków cech anatomicznych na zdjęciu dna oka polega na obliczeniu ważonego środka masy ramki otaczającej cechę anatomiczną. W tym celu dla każdej ramki ze znalezionej



Rysunek 5.32: Części wspólne (zaznaczone białą ramką) obszarów cech anatomicznych dna oka wyznaczonych przez modele bazowe. Wersja elektroniczna obrazu: <https://krzywicki.pro/fig-5-32>.

największego skupienia wyznaczana jest waga za pomocą przypisanego współczynnika ufności i pola powierzchni (Równanie 5.7) w postaci sumy wartości współczynnika ufności i połowy odsetka obszaru prostokąta na zdjęciu. Finalnie, geometryczny ważony środek masy ramki jest wyznaczany jako średnia ważona w obrębie współrzędnych x i y środków wszystkich ramek największego skupienia przy użyciu wag wyznaczonych dla każdego z tych prostokątów (Równanie 5.8). Na Rysunku 5.33 przedstawiono zlokalizowane środki dołeczka oraz tarczy nerwu wzrokowego. Warto zauważyć, że zaznaczone lokalizacje zostały wyznaczone na podstawie rozmiaru i umiejscowienia ramek otaczających cechy anatomiczne.

$$box_{weight} = conf + \frac{box_{area}}{photo_{area}} \cdot 0.5 \quad (5.7)$$

gdzie:

- $conf$ – współczynnik ufności ramki wyznaczonej przez model bazowy, opisującej obszar cechy anatomicznej na zdjęciu dna oka,

- box_{area} – pole powierzchni ramki opisującej obszar cechy anatomicznej,
- $photo_{area}$ – pole powierzchni zdjęcia.

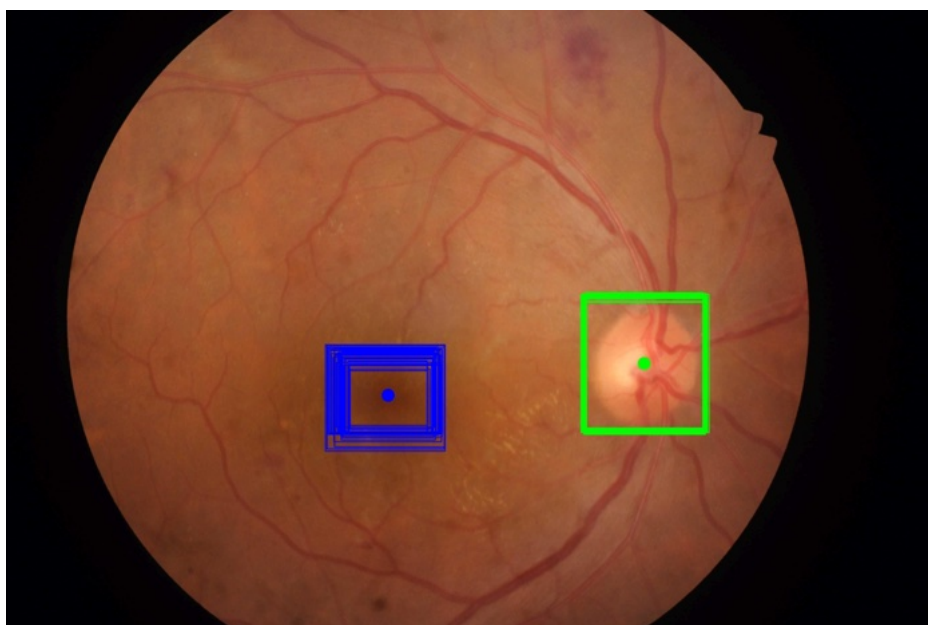
$$\begin{aligned}x_w &= \frac{\sum_{i=1}^n box_{weight_i} x_i}{\sum_{i=1}^n box_{weight_i}} \\y_w &= \frac{\sum_{i=1}^n box_{weight_i} y_i}{\sum_{i=1}^n box_{weight_i}}\end{aligned}\tag{5.8}$$

gdzie:

- x_w, y_w – współrzędne geometrycznego ważonego środka masy ramki opisującej obszar cechy anatomicznej,
- n – liczba ramek wchodzących w skład skupienia,
- x_i, y_i – współrzędne środka i -tej ramki wchodzącej w skład skupienia,
- box_{weight_i} – waga ramki opisującej obszar cechy anatomicznej.

Parametry zewnętrzne metody

W metodzie *MSLW*, będącej przedmiotem analizy w tym podrozdziale pojawił się jeden parametr – Box_{Th_c} stanowiący próg minimalnego współczynnika ufności przypisanego do branej pod uwagę w dalszych krokach – ramki opisującej lokalizację i obszar cechy anatomicznej na zdjęciu dna oka. Wśród pozostałych kroków metody jako parametr można wskazać także liczbę modeli bazowych w postaci sieci neuronowych biorących udział w wyznaczaniu współrzędnych środków cech anatomicznych. Biorąc pod uwagę główny charakter metody, jakim jest zespołowe uczenie i przydzielanie predykcji za pomocą metody *Pasting* [Bre99], jako parametr można wyróżnić także odsetek zdjęć trafiających do podzbiorów treningowych *P1W1* – *P4W2*, a także odsetek zdjęć stanowiących zbiór treningowy i walidacyjny. Nie sposób także nie wspomnieć o wszystkich parametrach treningu sieci znajdujących się w Tabeli 5.4 wraz z wykorzystaną architekturą sieci neuronowej jako modelu bazowego. Kompletny zestaw parametrów proponowanej *MSLW* przedstawiono w Tabeli 5.5. Warto zwrócić uwagę na fakt, że przytoczone parametry dotyczą zarówno etapu przygotowania, jak i wykorzystania metody *MSLW*.



Rysunek 5.33: Zlokalizowane metodą MSLW i zaznaczone punktami środki dołeczka (niebieski) oraz tarczy nerwu wzrokowego (zielony). Wersja elektroniczna obrazu: <https://krzywicki.pro/fig-5-33>.

5.5 Podsumowanie

W rozdziale przedstawiono charakterystyki dwóch metod przeznaczonych do lokalizacji współrzędnych środków dołeczka oraz tarczy nerwu wzrokowego na zdjęciu dna oka. Przedstawione rozwiązania różnią się od siebie charakterem działania, zużyciem zasobów obliczeniowych, a także sposobem przygotowania do wykorzystania w warunkach klinicznych. Wspomniane różnice mogą mieć także wpływ na jakość wyników uzyskanych przez oba podejścia.

Pierwszym z zaproponowanych rozwiązań był algorytm *MSiRL*, którego podstawą działania są metody przetwarzania obrazów, takie jak konwersja do innych modeli barw, transformacje morfologiczne, operacje splotu, wyrównywanie histogramu, binaryzacja, łączenie wierzchołków krawędzi regionów i ich etykietowanie oraz analiza struktury kształtu. Celem ich połączenia jest wydobycie ze zdjęcia dna oka jednoznacznego regionu plamki żółtej lub tarczy nerwu wzrokowego. Najlepszy region jest wybierany na podstawie wyznaczonego zakresu rozmiarów, zatem metoda powinna być stosowana na próbie zdjęć o zbliżonym rozmiarze. Warto zwrócić także uwagę

na zewnętrzne parametry $MT1$ i $MT2$ algorytmu MSiRL, które określają minimalną akceptowalną wartość współczynnika kolistości znalezionych regionów. W przypadku procesu segmentacji plamki żółtej proponowana wartość domyślna wynosi 0.1, a w przypadku tarczy nerwu wzrokowego 0.3. Kierując się własną metodą doboru wartości tych parametrów sugerowane jest zachowanie zależności, gdzie wartość $MT2$ będzie większa od $MT1$ co wynika z widoczności oraz wyraźności tych obiektów na zdjęciach dna oka. Tarcza nerwu wzrokowego jest znacznie wyraźniejsza i jaśniejsza od plamki żółtej na większości zdjęć dna oka dostępnych w publicznych bazach, co ma wpływ na kształt regionu powstałego w procesie segmentacji. Co więcej, z uwagi na mniej wyraźny szum, tarcza nerwu wzrokowego może zostać z powodzeniem poddana procesowi wyrównywania krawędzi struktury regionu, co może mieć pozytywny wpływ na jej finalną kolistość i uzyskaną za pomocą Równania 5.4 wartość współczynnika kolistości.

Fundament drugiej z zaproponowanych metod stanowią sieci neuronowe o architekturze *YOLOv8* [RKHD23] oraz uczenie zespołowe za pomocą metody *Pasting* [Bre99], która stanowi modyfikację popularnej metody Bagging [Bre96]. Celem połączenia modeli bazowych (sieci neuronowych o architekturze *YOLOv8*) w zespół za pomocą metody *Pasting* było poprawienie ogólnej jakości predykcji poprzez wykorzystanie wielu modeli uczonych na różnych podzbiorach danych treningowych. Metoda ta jest często stosowana w problemach uczenia maszynowego, lecz w połączeniu z zaproponowanym podejściem do agregacji lokalizacji, stanowi nowość. W metodzie *MSLW* udział w symultanicznej lokalizacji dołączka oraz tarczy nerwu wzrokowego na zdjęciu dna oka bierze 16 sieci neuronowych, z których każda jest wytrenowana na rozłącznym (podobnym) podzbiore zdjęć dna oka. Po wyznaczeniu lokalizacji każdej z cech anatomicznych, następuje ich agregacja w postaci operacji wyznaczenia ważonego środka masy otaczającej ramki. Mając na uwadze fakt, że zastosowane sieci neuronowe o architekturze *YOLOv8*, jako modele bazowe metody *MSLW*, dokonują transformacji zdjęcia do rozmiaru 640x640 px, zgodność rozmiarów zdjęć w przetwarzanej bazie nie ma istotnego znaczenia. Zdjęcia nie powinny być jednak mniejsze niż wejście sieci neuronowej. Warto zwrócić także uwagę na zewnętrzny parametr metody Box_{Th_c} , który określa minimalną wartość współczynnika ufności, który musi posiadać ramka otaczająca wybraną cechę anatomiczną, aby była brana pod uwagę podczas agregacji lokalizacji. Wartość parametru może pływać na wyniki dokładności lokalizacji oraz pokrycia. Wysoka wartość może pozostawić jedynie te ramki, których sieć neuronowa jest bardziej pewna, co może przyczynić się do mniejszej liczby wyznaczonych lokalizacji cech anatomicznych oraz do wyznaczenia dokładniejszych środków. W przypadku

małej wartości podczas agregacji lokalizacji mogą zostać wzięte pod uwagę ramki, których sieć jest mniej pewna, co może przyczynić się do mniej dokładnych lokalizacji cech anatomicznych, lecz w większej liczbie.

LP	Symbol modelu do treningu	Wykorzystany podzbiór	Przeznaczenie
1	SNP1W1	P1W1	Lokalizacja wzorców o wysokim kontraście
2	SNP1W2	P1W2	Lokalizacja wzorców o wyraźnej barwie i teksturze
3	SNP2W1	P2W1	Lokalizacja wzorców o wysokim kontraście
4	SNP2W2	P2W2	Lokalizacja wzorców o wyraźnej barwie i teksturze
5	SNP3W1	P3W1	Lokalizacja wzorców o wysokim kontraście
6	SNP3W2	P3W2	Lokalizacja wzorców o wyraźnej barwie i teksturze
7	SNP4W1	P4W1	Lokalizacja wzorców o wysokim kontraście
8	SNP4W2	P4W2	Lokalizacja wzorców o wyraźnej barwie i teksturze
9	SNP1W1T	P1W1	Lokalizacja wzorców o wysokim kontraście
10	SNP1W2T	P1W2	Lokalizacja wzorców o wyraźnej barwie i teksturze
11	SNP2W1T	P2W1	Lokalizacja wzorców o wysokim kontraście
12	SNP2W2T	P2W2	Lokalizacja wzorców o wyraźnej barwie i teksturze
13	SNP3W1T	P3W1	Lokalizacja wzorców o wysokim kontraście
14	SNP3W2T	P3W2	Lokalizacja wzorców o wyraźnej barwie i teksturze
15	SNP4W1T	P4W1	Lokalizacja wzorców o wysokim kontraście
16	SNP4W2T	P4W2	Lokalizacja wzorców o wyraźnej barwie i teksturze

Tabela 5.3: Szczegóły wytrenowanych modeli bazowych w postaci sieci neuronowych o architekturze YOLOv8. Sieci z przyrostkiem "T" w symbolu były trenowane transferowo.

LP	Nazwa parametru	Wartość	Szczegóły
1	Maksymalna liczba epok (ang. max epochs)	1000	Maksymalna liczba epok treningowych
2	Liczba epok oczekujących na poprawę (ang. patience)	50	Maksymalna liczba epok, po których brak poprawy jakości treningu zatrzyma uczenie
3	Rozmiar wsadu (ang. batch size)	16	Liczba obiektów przeznaczona do korekty wag podczas treningu
4	Optymalizator	Adam [KB14]	Zastosowany wariant optymalizatora podczas treningu
5	Początkowa wartość współczynnika uczenia (ang. learning rate)	0.0001	Pierwotna wartość współczynnika uczenia zmniejszana sukcesywnie podczas dalszych etapów treningu
6	Minimalna końcowa wartość współczynnika uczenia	0.00001	Najmniejsza wartość, jaką może osiągnąć współczynnik uczenia
7	β_1	0.937	Parametr β_1 optymalizatora ADAM
8	Regularyzacja powiązana z wartościami wag (ang. weight decay)	0.0005	Dodanie kary do funkcji kosztu podczas uczenia sieci w celu zmniejszenia nadmiernego dopasowania

Tabela 5.4: Parametry treningu modeli bazowych w postaci sieci neuronowych o architekturze YOLOv8.

LP	Nazwa	Symbol	Etap	Wartość domyślna
1	Próg minimalnego współczynnika ufności przypisanego do branej pod uwagę ramki	$BoxTh_c$	Zastosowanie	0.3 dla dołączka 0.6 dla tarczy nerwu wzrokowego
2	Liczba modeli bazowych	n_{models}	Trening i zastosowanie	16
3	Odsetek zdjęć trafiających do podzbiorów treningowych korzystając z losowania ze zwracaniem	$TW\%$	Trening	99.5
4	Odsetek zdjęć trafiających do zbioru treningowego i walidacyjnego	$Train\%, Val\%$	Trening	80, 20
5	Architektura sieci neuronowej w modelach bazowych	NN_{arch}	Trening i zastosowanie	YOLOv8
6	Parametry treningu modeli bazowych	–	Trening	Tabela 5.4

Tabela 5.5: Parametry metody MSLW.

Rozdział 6

Wyniki własne i komparatystyka

6.1 Przedstawienie problematyki poruszanej w rozdziale

Niniejszy rozdział poświęcono przedstawieniu oraz omówieniu wyników uzyskanych przez autorskie nowo opracowane metody lokalizacji współrzędnych środka dołeczka oraz tarczy nerwu wzrokowego, które zaproponowano w Rozdziale 5, oraz porównaniu tych wyników z innymi rozwiązaniami dostępnymi w literaturze naukowej. W celu oceny jakości opracowanych algorytmów przeprowadzono badania na publicznych danych klinicznych zawartych w bazie *IDRID* [IDR23a, PPK⁺18] (opisanej w Podrozdziale 4.3 Rozdziału 4), które obejmowały przypadki oczne o zróżnicowanej patologii oraz jakości.

Przedmiot analizy w tym rozdziale stanowią miary jakości uzyskane przez zaproponowane metody. Przedstawione rozwiązania poddano w tym celu analizie pod kątem uzyskania precyzji i pokrycia wyznaczonych środków cech anatomicznych za pomocą miar odległości między lokalizacjami wyznaczonymi przez omówione metody i twórców baz zawierających oznaczone zdjęcia dna oka. Osiągnięte wyniki porównano z literaturą naukową, co pozwoliło na ocenę przewagi lub słabych stron wraz z potencjalnymi drogami ulepszenia zaproponowanych metod w stosunku do innych rozwiązań. Otrzymane wyniki wraz z porównaniami poddano także dyskusji, która ma na celu ocenę użyteczności klinicznej omówionych

rozwiązań oraz propozycje dalszych dróg rozwoju.

6.2 Metodologia uzyskania wyników oraz ich porównania

Dla każdego zdjęcia dna oka zawartego w próbie testowej bazy *IDRID* wyznaczono współrzędne środków dołeczka oraz tarczy nerwu wzrokowego i porównano je ze środkami wyznaczonymi przez twórców bazy za pomocą odległości Minkowskiego (Równanie 6.1) w n -wymiarowej przestrzeni metrycznej dla trzech wariantów wartości parametru wynoszących kolejno: 1, 2 i ∞ . Dla przedstawionych rozwiązań wyznaczono także miarę pokrycia (Równanie 6.2) określającą odsetek zdjęć, które otrzymały lokalizację środka cechy anatomicznej dna oka za pomocą testowanej metody. Wydajność działania zaproponowanych rozwiązań określono za pomocą czasu wyznaczania pojedynczej lokalizacji oraz rozmiaru zużytych zasobów pamięciowych w systemie komputerowym.

$$d(x, y) = \left(\sum_{i=1}^n |x_i - y_i|^p \right)^{\frac{1}{p}}, \quad (6.1)$$

gdzie:

- $x = (x_1, x_2, \dots, x_n)$ i $y = (y_1, y_2, \dots, y_n)$ – współrzędne punktów w przestrzeni metrycznej,
- $p \geq 1$ – parametr charakteryzujący sposób wyznaczania odległości. Poszczególne warianty odległości Minkowskiego omówiono w Podrozdziale C.2.1 znajdującej się w Dodatku C.

$$coverage = \frac{|RFP_{xy}|}{|RFP_{DB}|}, \quad (6.2)$$

gdzie:

- RFP_{xy} – zbiór zdjęć dna oka z wyznaczonymi lokalizacjami środków wybranej cechy anatomicznej,
- RFP_{DB} – pełny zbiór zdjęć dna oka,
- $|\cdot|$ – operator liczebności zbioru.

Do oceny wydajności wykorzystano komputer przenośny *Acer ConceptD CN516-72G* [Ace23] o następujących parametrach, czego celem była obserwacja możliwości zaproponowanych rozwiązań na powszechnie dostępnym urządzeniu:

- system operacyjny: *Windows 11 Pro* [Win23], wersja *22H2*,
- procesor centralny (*CPU*): 64-bitowy *Intel Core i7-11800H* [Int23] o taktowaniu 2.30 GHz,
- rozmiar pamięci RAM: 32 GB,
- procesor graficzny (*GPU*): *NVIDIA GeForce RTX 3060 Mobile* [rtx23] z pamięcią o rozmiarze 6144 MB.

Zaproponowane rozwiązania testowano w środowisku uruchomieniowym o następującej konfiguracji:

- interpreter języka *Python* [Pyt23] w wersji 3.11.4,
- biblioteka *OpenCV* [Bra00] w wersji 4.7.0 skompilowana lokalnie ze wsparciem technologii *NVIDIA Cuda* [NVI23] przeznaczona do zrównoleglania obliczeń na *GPU*,
- biblioteka *Ultralytics* [JCQ23] w wersji 8.0.124 przeznaczona do wykorzystania sieci neuronowej o architekturze *YOLO v8* [RKHD23],
- biblioteka *Memory Profiler* [Mem23] w wersji 0.61 przeznaczona do pomiaru zużycia pamięci tymczasowej systemu,
- zestaw bibliotek *Cuda Toolkit* [Cud23a] w wersji 12.1 przeznaczonych do wsparcia wykonywania obliczeń na procesorach graficznych zgodnych z technologią *NVIDIA Cuda* [NVI23],

- biblioteka *CuDNN* w wersji 8.9.3 przeznaczona do obsługi sieci neuronowych na procesorach graficznych zgodnych z technologią NVIDIA Cuda [NVI23].

Zaproponowane metody zaimplementowano w dwóch wariantach: sekwencyjnie, gdzie operacje wykonywane są synchronicznie i jednowątkowo na *CPU* oraz równoległe, gdzie operacje wykonywane są asynchronicznie i wielowątkowo na *GPU*, co umożliwiła redukcję czasu potrzebną na dokonanie lokalizacji cech anatomicznych na zdjęciu dna oka. Z uwagi na dostępność równoległych implementacji jedynie niektórych algorytmów używanych w metodzie *MSiRL*, szczegóły działania wersji równoległej omówiono w Podrozdziale 6.6. Metoda *MSLW* dzięki wykorzystaniu biblioteki *Ultralytics* [JCQ23] wraz z biblioteką *PyTorch* [PGM⁺19] wykonuje się zarówno sekwencyjnie na *CPU*, jak i w pełni równoległe na *GPU*.

Do celów porównania czasu wyznaczania lokalizacji cech anatomicznych zmierzono czas działania funkcji zawierających główną implementację algorytmu, zamiast czasu działania całego procesu realizującego działanie metody. W celu porównania zużycia pamięci operacyjnej zmierzono pełne zużycie pamięci przez cały czas działania pełnego procesu realizującego działanie metody, a następnie wyznaczono średnie zużycie pamięci z całego cyklu życia procesu.

6.3 Wyniki własne

Wyniki w postaci zagregowanej, uzyskane przez zaproponowane metody, przedstawiono w Tabeli 6.1. Jako parametry pracy metody *MSiRL* przyjęto proponowane wartości domyślne przedstawione w Tabeli 5.1. Pełne wyniki w postaci lokalizacji obiektów przydzielonych dla każdego zdjęcia wraz z wyznaczonymi odległościami stanowiącymi różnice między lokalizacją wskazaną przez eksperta, a lokalizacją wskazaną przez algorytm, przedstawiono w Tabeli A.1 znajdującej się w Podrozdziale A.1 Dodatku A.

Dla metody *MSLW* przyjęto parametry w postaci proponowanych wartości domyślnych znajdujących się w Tabeli 5.5. Pełne wyniki w postaci lokalizacji cech anatomicznych dna oka przydzielonych dla każdego zdjęcia z bazy *IDRID* wraz z wyznaczonymi odległościami stanowiącymi różnice między

lokalizacjami wskazanymi przez eksperta i przez algorytm przedstawiono w Tabeli A.3 znajdującej się w Podrozdziale A.2 Dodatku A.

Nazwa parametru	MSiRL		MSLW	
	F	OD	F	OD
Średnia odległość dla $p = 1$	367.09	151.06	47.67	30.44
Średnia odległość dla $p = 2$	278.03	123.60	37.31	24.21
Średnia odległość dla $p = \infty$	258.78	118.95	35.27	23.04
Minimalna odległość dla $p = 1$	10.00	22.00	8.00	4.00
Minimalna odległość dla $p = 2$	7.62	16.55	5.83	3.16
Minimalna odległość dla $p = \infty$	7.18	15.49	5.34	3.04
Maksymalna odległość dla $p = 1$	2642.00	1040.00	471.00	87.00
Maksymalna odległość dla $p = 2$	1870.30	941.76	358.34	68.73
Maksymalna odległość dla $p = \infty$	1668.13	936.43	337.59	68.07
$Max - Min$ dla $p = 1$	2632.00	1018.00	463.00	83.00
$Max - Min$ dla $p = 2$	1862.68	925.21	352.51	65.57
$Max - Min$ dla $p = \infty$	1660.95	920.94	332.25	65.04
Liczba wyznaczonych lokalizacji	92	99	92	103

Tabela 6.1: Zagregowane wyniki uzyskane przez metody zaproponowane w Rozdziale 5. F – proces lokalizacji środka dołączka, OD – proces lokalizacji środka tarczy nerwu wzrokowego.

Zagregowane wyniki wydajnościowe zaproponowanych metod uzyskane podczas testów, których wyniki znajdują się w Tabelach 6.1 i A.1, przedstawiono w Tabelach 6.2 i 6.3. Pełne wyniki w postaci czasu wykonania oraz zużytej pamięci dla każdej operacji lokalizacji środków obiektów na zdjęciach dna oka przedstawiono w Tabeli A.2 znajdującej się w Podrozdziale A.1 Dodatku A.

Wyniki w postaci odległości Minkowskiego dla parametru $p = 2$ uzyskane przez poszczególne modele bazowe (sieci neuronowe o architekturze *YOLOv8*) metody *MSLW* przedstawiono w Tabelach 6.4 i 6.5. Pełne wyniki dla parametrów p wynoszących odpowiednio: 1, 2 i ∞ przedstawiono w Tabelach A.5 i A.6 znajdujących się w Podrozdziale A.1 Dodatku A.

LP	Nazwa parametru	MSiRL		MSLW	
		CPU	GPU	CPU	GPU
1	Średni czas wyznaczania pojedynczych lokalizacji [s] bez uwzględniania liczby wykrytych obiektów	0.68	0.38	16.85	2.29
2	Minimalny czas wyznaczania pojedynczych lokalizacji [s] bez uwzględniania liczby wykrytych obiektów	0.62	0.31	16.03	12.38
3	Maksymalny czas wyznaczania pojedynczych lokalizacji [s] bez uwzględniania liczby wykrytych obiektów	0.84	0.55	29.44	12.38
4	Średni czas wyznaczania pojedynczych lokalizacji [s] przy wykrytych obydwu obiektach	0.69	0.39	16.78	2.20
5	Minimalny czas wyznaczania pojedynczych lokalizacji [s] przy wykrytych obydwu obiektach	0.64	0.33	16.03	1.83
6	Maksymalny czas wyznaczania pojedynczych lokalizacji [s] przy wykrytych obydwu obiektach	0.84	0.55	18.61	2.42
7	Czas wyznaczania pojedynczej lokalizacji: $Max - Min$ bez uwzględniania liczby wykrytych obiektów	0.22	0.24	13.41	0.00
8	Czas wyznaczania pojedynczej lokalizacji: $Max - Min$ przy wykrytych obydwu obiektach	0.20	0.22	2.58	0.59

Tabela 6.2: Czas wyznaczania cech anatomicznych na zdjęciu dna oka przez metody zaproponowane w Rozdziale 5.

LP	Nazwa parametru	MSiRL		MSLW	
		CPU	GPU	CPU	GPU
1	Średnie zużycie pamięci [MiB] bez uwzględniania liczby wykrytych obiektów	242.65	313.17	3505.89	3531.75
2	Minimalne zużycie pamięci [MiB] bez uwzględniania liczby wykrytych obiektów	229.00	293.62	3422.61	2887.86
3	Maksymalne zużycie pamięci [MiB] bez uwzględniania liczby wykrytych obiektów	266.18	346.38	3670.21	3617.88
4	Średnie zużycie pamięci [MiB] przy wykrytych obydwu obiektach	242.90	314.01	3507.24	3533.12
5	Minimalne zużycie pamięci [MiB] przy wykrytych obydwu obiektach	229.00	298.18	3422.61	2887.86
6	Maksymalne zużycie pamięci [MiB] przy wykrytych obydwu obiektach	266.18	346.38	3670.21	3617.88
7	Zużycie pamięci $Max - Min$ bez uwzględniania liczby wykrytych obiektów	37.18	52.76	247.60	730.02
8	Zużycie pamięci $Max - Min$ przy wykrytych obydwu obiektach	37.18	48.2	247.70	730.02

Tabela 6.3: Średnie zużycie pamięci podczas wyznaczania cech anatomicznych na zdjęciu dna oka za pomocą metod zaproponowanych w Rozdziale 5.

Cecha anatomiczna / symbol modelu bazowego	Średnia	Maksymalna	Minimalna	Max-Min	Liczba wykrytych lokalizacji cech
D SNP1W1	30.59	146.00	7.00	139.00	76
TNW SNP1W1	26.02	73.01	2.00	71.01	101
D SNP1W2	32.04	105.12	4.12	101.00	80
TNW SNP1W2	23.76	62.51	1.00	61.51	100
D SNP2W1	32.80	143.28	2.24	141.04	76
TNW SNP2W1	26.02	73.93	5.00	68.93	102
D SNP2W2	35.05	108.00	4.47	103.53	73
TNW SNP2W2	27.48	69.87	6.40	63.47	100
D SNP3W1	33.89	183.17	5.39	177.79	74
TNW SNP3W1	26.24	87.32	6.71	80.61	102
D SNP3W2	40.97	358.34	0.00	358.34	80
TNW SNP3W2	24.88	65.39	3.00	62.39	99
D SNP4W1	31.47	139.18	3.16	136.01	78
TNW SNP4W1	26.91	71.40	4.47	66.93	100
D SNP4W2	33.88	122.41	3.61	118.81	76
TNW SNP4W2	27.08	68.66	2.83	65.83	101
D SNP1W1T	32.16	122.33	4.00	118.33	76
TNW SNP1W1T	25.10	86.68	3.16	83.52	102
D SNP1W2T	29.54	135.67	6.00	129.67	67
TNW SNP1W2T	27.59	90.80	1.00	89.80	103
D SNP2W1T	34.26	105.68	1.00	104.68	76
TNW SNP2W1T	25.40	175.84	3.00	172.84	103
D SNP2W2T	29.39	62.20	2.00	60.20	73
TNW SNP2W2T	27.69	100.40	0.00	100.40	99
D SNP3W1T	31.47	118.79	1.41	117.38	80
TNW SNP3W1T	27.30	70.84	5.83	65.01	102
D SNP3W2T	32.46	92.44	4.47	87.97	68
TNW SNP3W2T	26.03	88.46	2.24	86.22	101
D SNP4W1T	30.38	120.17	5.00	115.17	78
TNW SNP4W1T	23.90	77.67	2.24	75.43	102
D SNP4W2T	37.95	137.28	4.47	132.80	77
TNW SNP4W2T	27.22	73.68	4.12	69.56	102

Tabela 6.4: Odległość Minkowskiego dla $p = 2$ między lokalizacjami cech anatomicznych, wyznaczonymi przez modele bazowe metody MSLW oraz ekspertów, dla wszystkich zdjęć z bazy IDRID. D – lokalizacja dołeczka, TNW – lokalizacja tarczy nerwu wzrokowego.

6.4 Porównanie z innymi rozwiązaniami

Do porównania z innymi pracami posłużą wartości średniej *odległości Minkowskiego* dla parametru $p = 2$ (odległość euklidesowa) między środkami obiektów wyznaczonych przez ekspertów i przez proponowane metody. Na potrzeby porównania wyników wartości średniej *odległości Minkowskiego* dla parametru $p = 2$ przyjęto uproszczoną nazwę *Średniego Błędu Euklidesowego*, *SBE*. W Tabeli 6.6 przedstawiono wyniki własne oraz uzyskane przez rozwiązania zaproponowane w pracach [PPK⁺20, HAE⁺21, LSD18], stanowiące przedmiot analizy w Podrozdziale 5.2. Wszystkie wyniki

Cecha anatomiczna / symbol modelu bazowego	Średnia	Maksymalna	Minimalna	Max-Min	Liczba wykrytych cech
D SNP1W1	30.59	146.00	7.00	139.00	76
TNW SNP1W1	26.52	73.01	2.00	71.01	92
D SNP1W2	32.04	105.12	4.12	101.00	80
TNW SNP1W2	23.62	62.51	1.00	61.51	91
D SNP2W1	32.80	143.28	2.24	141.04	76
TNW SNP2W1	25.65	70.71	5.00	65.71	92
D SNP2W2	35.05	108.00	4.47	103.53	73
TNW SNP2W2	27.15	69.87	6.40	63.47	91
D SNP3W1	33.89	183.17	5.39	177.79	74
TNW SNP3W1	25.43	70.41	8.25	62.16	92
D SNP3W2	40.97	358.34	0.00	358.34	80
TNW SNP3W2	24.73	65.39	3.00	62.39	91
D SNP4W1	31.47	139.18	3.16	136.01	78
TNW SNP4W1	27.03	71.40	4.47	66.93	92
D SNP4W2	33.88	122.41	3.61	118.81	76
TNW SNP4W2	26.93	68.66	2.83	65.83	92
D SNP1W1T	32.16	122.33	4.00	118.33	76
TNW SNP1W1T	25.51	86.68	3.16	83.52	93
D SNP1W2T	29.54	135.67	6.00	129.67	67
TNW SNP1W2T	27.26	90.80	1.41	89.38	93
D SNP2W1T	34.26	105.68	1.00	104.68	76
TNW SNP2W1T	25.31	175.84	3.00	172.84	93
D SNP2W2T	29.39	62.20	2.00	60.20	73
TNW SNP2W2T	28.28	100.40	0.00	100.40	91
D SNP3W1T	31.47	118.79	1.41	117.38	80
TNW SNP3W1T	27.28	70.84	5.83	65.01	93
D SNP3W2T	32.46	92.44	4.47	87.97	68
TNW SNP3W2T	25.99	88.46	2.24	86.22	92
D SNP4W1T	30.38	120.17	5.00	115.17	78
TNW SNP4W1T	23.92	77.67	2.24	75.43	93
D SNP4W2T	37.73	137.28	4.47	132.80	76
TNW SNP4W2T	26.92	73.68	4.12	69.56	92

Tabela 6.5: Odległość Minkowskiego dla $p = 2$ między lokalizacjami cech anatomicznych, wyznaczonymi przez modele bazowe metody MSLW oraz ekspertów, dla zdjęć na których metoda MSLW zlokalizowała obydwie cechy anatomiczne. D – lokalizacja dołeczka, TNW – lokalizacja tarczy nerwu wzrokowego.

zostały uzyskane na podstawie bazy *IDRID*, która jako jedyna ma oznaczone środki cech anatomicznych będących przedmiotem analizy w niniejszej rozprawie.

Podczas analizy prac omawiających porównywane rozwiązania nie pojawiły się informacje dotyczące pokrycia wynikami zbioru *IDRID*, ani informacje dotyczące zużycia zasobów obliczeniowych oraz czasu poświęconego na zlokalizowanie obiektów na zdjęciach i treningu. Nie będą zatem wykorzystane w dalszych porównaniach.

LP	Metoda	SBE w lokalizacji środką dołeczka	SBE w lokalizacji środką tarczy nerwu wzrokowego
1	Zwycięzca rywalizacji <i>online</i> [IDR23b]	45.90	25.62
2	Zwycięzca rywalizacji <i>onsite</i> [PPK ⁺ 20]	64.49	21.07
3	Region Proposal Network [LSD18]	52.00	32.60
4	DRNet [HAE ⁺ 21]	41.87	20.23
5	MSiRL	278.03	123.60
6	MSLW	37.31	24.21

Tabela 6.6: Porównanie wyników uzyskanych przez rozwiązania lokalizujące środki dołeczka oraz tarczy nerwu wzrokowego na zdjęciu dna oka.

W literaturze często wykorzystywany jest także zbiór zdjęć dna oka *HRF* (*High-Resolution Fundus*) [HRF23], który ma (podobnie jak *IDRID*) oznaczone środki tarczy nerwu wzrokowego. Brak oznaczeń środków dołeczków dyskwalifikuje go jednak z wykorzystania w porównaniach zaproponowanych metod z uwagi na niepełny obraz wyników.

6.5 Dyskusja

Na podstawie Tabeli A.1 oraz przyjętego progu wynoszącego 500 pikseli według odległości Minkowskiego z parametrem $p = 2$, można zauważyć, że 12 spośród 92 lokalizacji dołeczka wyznaczona przez metodę *MSiRL* i jedna spośród 99 lokalizacji tarczy nerwu wzrokowego znajdowały się poza rzeczywistym regionem. Według Tabeli A.3 i takich samych założeń wszystkie wyznaczone lokalizacje znajdowały się w obrębie rzeczywistych regionów cech anatomicznych dna oka. Na podstawie Tabeli 6.1 można zauważyć, że zarówno pokrycie, jak i trafność zadania lokalizacji środka tarczy nerwu wzrokowego przebiegło sprawniej, niż lokalizacja dołeczka, czego przyczyną może być wyraźniejsza struktura tarczy nerwu wzrokowego na zdjęciu dna oka.

W Tabelach 6.2 i 6.3 można dostrzec różnice w czasie wyznaczania lokalizacji cech anatomicznych oraz w zużyciu pamięci przy wykonaniu

zaproponowanych metod na *CPU* oraz *GPU*. Można także zauważyć, że spośród zaproponowanych rozwiązań, metoda *MSiLR* notuje krótszy czas wyznaczania lokalizacji, niż metoda *MSLW*. Podobną zależność można dostrzec również w przypadku zużycia pamięci operacyjnej. Można zatem wnioskować, że wykorzystanie metody *MSiLR* jest bardziej wydajne i mniej kosztowne pod względem zużycia zasobów, w porównaniu do metody *MSLW*, dzięki czemu metoda *MSiLR* może być przydatna do zastosowania w małych i przenośnych urządzeniach. W przytoczonych tabelach można zauważyć także, że wykorzystanie *GPU* do zrównoleglania obliczeń optymalizuje czas ich wykonania, lecz zwiększa zużycie pamięci. Warto również zwrócić uwagę na interesującą zależność, że liczba wykrytych cech na zdjęciu może mieć wpływ na wyższy czas przydzielania lokalizacji oraz na wyższe zużycie pamięci.

Na podstawie Tabel 6.1 i 6.6 można także zauważyć, że metoda *MSLW* odnotowuje dokładniejsze wyniki w lokalizacji obydwu cech anatomicznych dna oka oraz wyższe pokrycie, niż metoda *MSiLR*. Metoda *MSLW* odnotowuje także najdokładniejszy wynik w zadaniu lokalizacji środka dołeczka spośród bieżącego stanu literatury naukowej oraz nieznacznie gorszy wynik w zadaniu lokalizacji środka tarczy nerwu wzrokowego, niż dwa najlepsze rozwiązania w obecnym stanie wiedzy. SBE wyższy o niemal cztery piksele od najlepszego rozwiązania (na zdjęciu o rozmiarze 4288×2848 px) może wynikać z dwóch czynników: uczenie metody *MSLW* na próbie całkowicie odmiennej od treningowej części zbioru *IDRID* oraz kierunkowanie inteligencji modeli bazowych (sieci neuronowych o architekturze *YOLOv8*) na detekcję pełnego obszaru obiektu w postaci cechy anatomicznej zamiast wskazywania pojedynczego punktu środka cechy.

W Tabelach 6.4 i 6.5 można zauważyć, że wyniki uzyskane przez pojedyncze modele bazowe metody *MSLW* uzyskały podobne (nieznacznie niższe) wyniki do zagregowanych, lecz przy niższym pokryciu. Nie dostrzeżono wpływu zastosowania uczenia transferowego przy treningu modeli bazowych na średnią odległość stanowiącą różnicę w wyznaczonych lokalizacjach przez metodę *MSLW* oraz eksperta.

W innych rozwiązaniach lokalizujących środki dołeczka oraz tarczy nerwu wzrokowego, przedstawionych w literaturze naukowej, można dostrzec pewne potencjalne wady przede wszystkim w postaci tendencji do trenowania ich pod kątem realizacji zadania regresji polegającej na wskazaniu na zdjęciu współrzędnych punktów stanowiących obiekt zainteresowania. W bazie *IDRID* można dostrzec, że środki cech anatomicznych zostały wyznaczone

przez twórców zarówno w sytuacjach gdy te cechy są i nie są widoczne, co sprawia że rozwiązania były uczone realizacji zadania w warunkach, w których nie istnieje możliwość wizualnej oceny rezultatu. Co więcej, rozwiązania obecne w literaturze naukowej były trenowane na niewielkiej próbie wynoszącej 413 niewycentrowanych zdjęć dna oka, co czyni je potencjalnie zależnymi od przyjętego wyśrodkowania kadru.

Zaproponowane w pracy rozwiązania są ściśle uzależnione od możliwości wizualnej oceny i lokalizacji cech anatomicznych na dnie oka. Metoda *MSiLR*, z uwagi wykorzystanie algorytmów przetwarzania obrazów dla celów rozpoznawania finalnych wzorców, może być bardziej wrażliwa na zmiany oświetlenia zdjęcia niż metoda *MSLW*. Podczas trenowania metody *MSLW* wykorzystano bazę zdjęć *BFO21* samodzielnie oznaczoną pod kątem jedynie tych cech które były widoczne, co oznacza że metoda wskaże lokalizację tylko tych cech anatomicznych, które są widoczne. Dzięki takiemu podejściu, metoda *MSLW* może być odnotować jeszcze dokładniejsze wyniki, w porównaniu ze stanem literatury, dla zdjęć dna oka wykonanych w różnych wariantach centracji i w zmiennym oświetleniu.

6.6 Uwagi o zrównoleganiu metod na procesorach graficznych

Wprowadzenie implementacji algorytmów zrównoleglonych na procesorach graficznych (GPU) może przynieść znaczący wzrost wydajności. Jednak w niektórych przypadkach, jak w proponowanej metodzie *MSiRL*, istnieje ograniczenie związane z brakiem dostępnych równoległych implementacji wszystkich wykorzystanych metod. W celu zwiększenia wydajności algorytmu, zdecydowano się na zrównoleglenie na GPU tych kroków metody *MSiLR*, które miały dostępne implementacje zgodne z platformą NVIDIA Cuda [NVI23]. Dzięki temu osiągnięto przyspieszenie działania algorytmu co wpłynęło na krótszy czas potrzebny do przetworzenia zdjęć. W Tabeli 6.7 przedstawiono kroki metody *MSiLR*, które zostały zrównoleglone na GPU, dzięki dostępnym implementacjom zgodnym z platformą NVIDIA Cuda.

Na podstawie Tabeli 6.7 oraz Rysunków 5.2 i 5.16 można zauważyć, że pomyślnie zrównoleglone zostały kroki metody *MSiLR* wykonywane przed operacją adaptywnej binaryzacji obrazu za pomocą metody *Otsu*

LP	Krok algorytmu	Operacja	Zadanie
1	Korekcja kontrastu	Transformacja do modelu CIELab, wyrównanie histogramu warstwy L metodą CLAHE [Rez04]	FS
2	Konwersja do skali szarości	Uśrednienie wartości każdego piksela	FS
3	Redukcja szumu	Splot przy użyciu filtru gaussowskiego	FS
4	Uwypuklenie obszaru plamki żółtej	Sekwencja operacji morfologicznych	FS
5	Wyrównanie histogramu	CLAHE [Rez04]	FS
6	Konwersja do modeli barw	Konwersja do modeli CIELab oraz YUV	ODS
7	Utworzenie pośredniej formy obrazu	Wyrównanie i wygaszenie tekstury siatkówki w pośredniej formie obrazu	ODS
8	Korekcja struktury plamki żółtej	Łączone transformacje morfologiczne	ODS

Tabela 6.7: Zrównoleglone na GPU kroki algorytmu MSiLR. FS – segmentacja regionu plamki żółtej, ODS – segmentacja regionu tarczy nerwu wzrokowego.

[Ots79], której równoległa implementacja dla procesorów *GPU* nie jest jeszcze dostępna w bibliotece *OpenCV* [Bra00]. Niewątpliwie, brak pełnego zrównoleglenia wszystkich algorytmów wpływa na ostateczną wydajność algorytmu. Mimo to, podejście częściowego zrównoleglenia pozwoliło osiągnąć przyspieszenie i poprawić efektywność obliczeń. Jest to kompromis, który uwzględniono ze uwagi na ograniczenia wynikające z bieżącego stanu wiedzy i dostępność równoległych implementacji poszczególnych metod.

6.7 Podsumowanie

W Rozdziale 6. przedstawiono wyniki uzyskane przez metody zaproponowane w Rozdziale 5. Metody poddano testom pod kątem dokładności wskazywania lokalizacji środków cech anatomicznych na zdjęciach dna oka w postaci dołeczka oraz tarczy nerwu wzrokowego.

Pierwsza z zaproponowanych metod – *MSiLR* odnotowała wyniki dokładności wskazywanych lokalizacji wynoszące średnio 278.03 piksele w zadaniu lokalizacji dołeczka oraz 126.60 piksela w lokalizacji tarczy nerwu wzrokowego według odległości euklidesowej. Zaproponowane metody poddano także testom zużycia zasobów w postaci czasu poświęconego obliczeniom i zużycia pamięci operacyjnej.

Oprócz dobrych stron, warto także zwrócić uwagę na potencjalne słabe strony proponowanych rozwiązań. Metoda *MSiLR*, nie została całkowicie zrównoleglona z uwagi na brak dostępnych implementacji części z algorytmów wykorzystywanych przez metodę, jednak istnieje możliwość, że takie implementacje pojawią się w przyszłości. Zarówno metoda *MSiLR*, jak i *MSLW* były testowane na jednej publicznej bazie zdjęć dna oka – *IDRID* [PPK⁺18], co stanowiło wymóg wiarygodnego porównania zaproponowanych metod ze stanem literatury naukowej. Brak innych publicznych baz zawierających opisane zdjęcia dna oka pod kątem lokalizacji środków dołeczka i tarczy nerwu wzrokowego może być potencjalną przyczyną niepełnej weryfikacji rozwiązań. Zaproponowana metoda *MSLW* nie została również przygotowana korzystając z popularnej praktyki, jaką jest stosowanie walidacji krzyżowej [Sto74], czego przyczyną jest brak dostępu do wystarczających zasobów obliczeniowych. Jak wspomniano w Podrozdziale 5, trening modeli bazowych metody *MSLW* na GPU *NVIDIA Tesla T4* [AMD23b], który jest przeznaczony do trenowania sieci neuronowych, trwał

228 godzin i 50 minut. Przy zastosowaniu 5-częściowej walidacji krzyżowej, przygotowanie metody mogłoby trwać około 5-krotnie dłużej, czyli ponad 1144 godziny i 10 minut, co daje 47 dni, 16 godzin i 10 minut.

Rozdział 7

Podsumowanie

W niniejszej rozprawie zawarto rozważania dotyczące metod wsparcia procesu diagnostycznego w okulistyce. Celem rozprawy było opracowanie dwóch nowych metod lokalizacji środka dołączka oraz środka tarczy nerwu wzrokowego na zdjęciach dna oka.

Fundament pierwszej z proponowanych metod (*MSiRL*) stanowią techniki przetwarzania obrazów, takie jak operacje morfologiczne, splot, binaryzacja, wyrównywanie histogramów oraz techniki analizy i obróbki kształtu, takie jak uwypuklanie powłok konturów obiektów czy analiza komponentów połączonych.

Druga z zaproponowanych metod (*MSLW*) oparta jest na nowoczesnych metodach uczenia maszynowego, takich jak uczenie transferowe, uczenie zespołowe i zastosowanie sieci neuronowych w detekcji obiektów.

Każdą z opracowanych metod poddano testom pod kątem dokładności i pokrycia wyznaczonych lokalizacji oraz zużycia pamięci i czasu obliczeń w wariantach implementacji sekwencyjnej wykonywanej na procesorze centralnym oraz w wariantach implementacji częściowo równoległej wykonywanej częściowo na procesorze graficznym. Testom poddano również dokładność i pokrycie lokalizacji wyznaczonej przez pojedyncze sieci neuronowe oraz przez zespół sieci neuronowych.

Na podstawie tez postawionych w Podrozdziale 1.2 wkład badawczy przedstawiony w niniejszej rozprawie można określić następująco.

1. Utworzono dwie nowe metody lokalizujące na zdjęciu dna oka środek dołeczka i środek tarczy nerwu wzrokowego.
2. Sprawdzono dokładność i pokrycie wyznaczonych lokalizacji cech anatomicznych na zdjęciach dna oka za pomocą obydwu zaproponowanych metod.
3. Sprawdzono czas wykonania obliczeń i zużycie pamięci operacyjnej za pomocą obydwu zaproponowanych metod.
4. Zastosowano przyspieszenie sprzętowe w implementacjach obydwu metod za pomocą zrównoleglania obliczeń na procesorze graficznym.
5. Sprawdzono dokładność i pokrycie lokalizacji cech anatomicznych na zdjęciach dna oka przy użyciu zespołu sieci neuronowych oraz przy użyciu pojedynczych sieci neuronowych.

Metoda *MSLW* osiągnęła wyższą efektywność w zadaniu lokalizacji środka dołeczka w porównaniu z innymi rozwiązaniami znanymi z literatury naukowej i nieznacznie niższą w zadaniu lokalizacji środka tarczy nerwu wzrokowego. Należy jednak podkreślić, że nie jest obecnie dostępna pełna informacja o pokryciu, czyli liczbie wyznaczonych lokalizacji, przez metody ujęte w literaturze naukowej. Takie braki danych mogą stanowić czynnik wprowadzający pewien stopień niepewności w dokładności porównania metody *MSLW* z innymi rozwiązaniami prezentowanymi w publikacjach naukowych.

Wszystkie prace implementacyjne, obliczeniowe i analityczne zostały dokonane samodzielnie przez autora rozprawy.

Dodatek A

Wyniki testów metod

A.1 Metoda MSiRL

Nazwa pliku z rozszerzeniem .jpg	Współrzędne środka dołączka wyznaczone przez ekspertów	Współrzędne środka tarczy wzrokowego wyznaczone przez ekspertów	Współrzędne środka dołączka wyznaczone przez MSiRL	Współrzędne środka tarczy wzrokowego wyznaczone przez MSiRL	Różnica między środkami dla $p = 1$	Różnica między środkami dla $p = 2$	Różnica między środkami dla $p = \infty$
IDRiD_001	1914,1617	651,1453	1888,1655	-	64.0, -	46.04, -	41.69, -
IDRiD_002	1922,1468	3158,1280	1899,1400	3069,1267	91.0, 102.0	71.78, 89.94	68.87, 89.09
IDRiD_003	2566,1591	1101,1374	-	816,1134	-, 525.0	-, 372.59	-, 333.14
IDRiD_004	1562,1550	2881,1430	1569,1513	2796,1385	44.0, 130.0	37.66, 96.18	37.08, 89.01
IDRiD_005	1757,1310	2840,1175	1571,189	2784,1157	1307.0, 74.0	1136.33, 58.82	1122.7, 56.61
IDRiD_006	1745,1580	2963,1501	1743,1495	2844,1464	87.0, 156.0	85.02, 124.62	85.0, 120.18
IDRiD_007	2315,1505	842,1374	2940,1908	677,1324	1028.0, 215.0	743.66, 172.41	676.49, 166.52
IDRiD_008	2203,1483	895,1299	2154,1443	979,1266	89.0, 117.0	63.25, 90.25	56.63, 85.66
IDRiD_009	2011,1670	3241,1591	2050,1690	3137,1575	59.0, 120.0	43.83, 105.22	40.68, 104.13
IDRiD_010	1768,1771	2948,1655	1717,1777	2826,1600	57.0, 177.0	51.35, 133.82	51.03, 125.62
IDRiD_011	2019,1629	872,1385	1758,1369	1000,1374	521.0, 139.0	368.4, 128.47	328.21, 128.03
IDRiD_012	1865,1449	647,1149	1735,1320	785,1131	259.0, 156.0	183.14, 139.17	163.16, 138.1
IDRiD_013	2150,1389	850,1224	2112,1379	979,1231	48.0, 136.0	39.29, 129.19	38.23, 129.01
IDRiD_014	2086,1917	816,1636	2081,1975	-	63.0, -	58.22, -	58.01, -
IDRiD_015	1955,1576	3076,1483	1587,753	2978,1405	1191.0, 176.0	901.53, 125.25	846.83, 112.29
IDRiD_016	1937,1288	3241,1325	1439,1119	3145,1289	667.0, 132.0	525.89, 102.53	504.4, 97.66
IDRiD_017	2026,1644	3286,1321	1869,1503	3172,1318	298.0, 117.0	211.02, 114.04	188.27, 114.0
IDRiD_018	1704,1573	520,1340	-	627,1320	-, 127.0	-, 108.85	-, 107.23
IDRiD_019	2161,1621	962,1303	832,2382	1072,1295	2090.0, 118.0	1531.46, 110.29	1407.45, 110.01
IDRiD_020	1723,1842	2933,1786	1691,1852	2834,1718	42.0, 167.0	33.53, 120.1	32.32, 108.71

Tabela A.1: Pełne wyniki uzyskane przez metodę MSiRL na bazie IDRiD. W kolumnach zawierających różnice między środkami znajdują się odległości wyrażone odległością Minkowskiego dla podanych wartości parametru p w kolejności: różnica odległości środka dołączka, różnica odległości środka tarczy nerwu wzrokowego. Wersja elektroniczna tabeli: <https://krzywicki.pro/tab-a-1>.

Nazwa pliku z rozszerzeniem .jpg	Współrzędne środka dołączka wyznaczone przez ekspertów	Współrzędne środka tarczy wzrokowego wyznaczone przez ekspertów	Współrzędne środka dołączka wyznaczone przez MSiRL	Współrzędne środka tarczy wzrokowego wyznaczone przez MSiRL	Różnica między środkami dla $p = 1$	Różnica między środkami dla $p = 2$	Różnica między środkami dla $p = \infty$
IDRiD_021	2011,1670	1018,1415	-	1141,1388	-, 150.0	-, 125.93	-, 123.43
IDRiD_022	2000,1385	3136,1325	1386,1041	3028,1305	958.0, 128.0	703.8, 109.84	648.07, 108.23
IDRiD_023	2124,1359	865,1059	2045,1288	903,1054	150.0, 43.0	106.22, 38.33	94.76, 38.03
IDRiD_024	2266,1314	902,1138	1399,921	973,1121	1260.0, 88.0	951.91, 73.01	893.12, 71.32
IDRiD_025	1543,1573	2922,1479	1553,1612	2798,1454	49.0, 149.0	40.26, 126.5	39.22, 124.34
IDRiD_026	2367,1550	962,1543	1645,2680	1079,1480	1852.0, 180.0	1340.96, 132.88	1220.76, 122.8
IDRiD_027	1794,1419	3113,1370	1772,1390	2990,1361	51.0, 132.0	36.4, 123.33	32.72, 123.02
IDRiD_028	1730,1393	2975,1359	-	2868,1344	-, 122.0	-, 108.05	-, 107.1
IDRiD_029	2397,1576	1030,1411	2386,1593	1048,1380	28.0, 49.0	20.25, 35.85	18.41, 32.9
IDRiD_030	2337,1588	1078,1404	2311,1557	1170,1392	57.0, 104.0	40.46, 92.78	36.18, 92.07
IDRiD_031	2367,1419	1045,1291	2411,1421	1145,1297	46.0, 106.0	44.05, 100.18	44.0, 100.01
IDRiD_032	1502,1543	2941,1524	1494,1402	2836,1510	149.0, 119.0	141.23, 105.93	141.01, 105.08
IDRiD_033	1933,1505	659,1164	3245,2048	-	1855.0, -	1419.93, -	1342.3, -
IDRiD_034	2255,1153	951,1044	2021,1003	1089,1043	384.0, 139.0	277.95, 138.0	252.97, 138.0
IDRiD_035	1532,1707	2847,1734	1447,1459	2741,1678	333.0, 162.0	262.16, 119.88	251.28, 110.97
IDRiD_036	1700,1644	3068,1573	1634,1586	2955,1542	124.0, 144.0	87.86, 117.18	78.44, 113.77
IDRiD_037	2105,1782	3312,1617	2098,1785	3217,1595	10.0, 117.0	7.62, 97.51	7.18, 95.39
IDRiD_038	2293,1535	962,1434	2216,1510	1043,1417	102.0, 98.0	80.96, 82.76	77.87, 81.25
IDRiD_039	1494,1895	2858,1771	1536,1940	2762,1726	87.0, 141.0	61.55, 106.02	54.87, 99.19
IDRiD_040	1502,1385	2847,1138	1460,1250	2819,1146	177.0, 36.0	141.38, 29.12	136.34, 28.22
IDRiD_041	2379,1629	1161,1505	2403,1644	1180,1492	39.0, 32.0	28.3, 23.02	25.81, 20.84
IDRiD_042	1708,1584	2922,1535	1509,1325	2810,1501	458.0, 146.0	326.62, 117.05	293.39, 113.03
IDRiD_043	1322,1198	2667,1153	-	2562,1118	-, 140.0	-, 110.68	-, 106.28
IDRiD_044	2188,1321	962,1097	2110,1331	1090,1093	88.0, 132.0	78.64, 128.06	78.05, 128.0
IDRiD_045	2154,1408	846,1284	2053,1371	968,1262	138.0, 144.0	107.56, 123.97	102.63, 122.24
IDRiD_046	1787,1396	3222,1340	1726,1396	3090,1318	61.0, 154.0	61.0, 133.82	61.0, 132.2
IDRiD_047	2169,1411	988,1168	2113,1389	1079,1163	78.0, 96.0	60.17, 91.14	57.11, 91.01
IDRiD_048	2214,1456	966,1314	2100,1426	994,1343	144.0, 57.0	117.88, 40.31	114.69, 35.92
IDRiD_049	1963,1617	3199,1644	1341,1096	3105,1605	1143.0, 133.0	811.37, 101.77	725.62, 96.19
IDRiD_050	2015,1314	760,1160	1523,1220	788,1150	586.0, 38.0	500.9, 29.73	493.14, 28.42
IDRiD_051	1775,1917	3087,1674	1872,2107	3006,1649	287.0, 106.0	213.33, 84.77	198.08, 81.79
IDRiD_052	2120,1355	884,1378	2045,1360	965,1350	80.0, 109.0	75.17, 85.7	75.01, 82.1
IDRiD_053	2337,1677	1007,1505	1982,2288	1104,1413	966.0, 189.0	706.64, 133.69	648.59, 119.15
IDRiD_054	1521,1543	2851,1408	-	2732,1369	-, 158.0	-, 125.23	-, 120.38
IDRiD_055	1940,1869	647,1644	1661,1848	741,1652	300.0, 102.0	279.79, 94.34	279.04, 94.02
IDRiD_056	1547,1543	2907,1333	1421,1372	2798,1287	297.0, 155.0	212.41, 118.31	191.3, 111.67
IDRiD_057	2011,1359	719,1381	1898,1382	834,1366	136.0, 130.0	115.32, 115.97	113.32, 115.09
IDRiD_058	1644,1441	2810,1370	1557,1358	3008,1171	170.0, 397.0	120.24, 280.72	107.15, 250.1
IDRiD_059	2060,1295	854,1172	1992,1292	927,1164	71.0, 81.0	68.07, 73.44	68.0, 73.03
IDRiD_060	2491,1306	1153,1142	2532,1310	1243,1123	45.0, 109.0	41.19, 91.98	41.01, 90.28
IDRiD_061	2161,1812	3316,1576	2167,1857	3248,1557	51.0, 87.0	45.4, 70.6	45.04, 68.49
IDRiD_062	2150,1258	884,1198	2051,1192	981,1192	165.0, 103.0	118.98, 97.19	107.95, 97.01
IDRiD_063	1506,1554	2825,1479	1473,1454	2611,907	133.0, 786.0	105.3, 610.72	101.18, 581.82
IDRiD_064	1468,1576	2750,1355	1436,1360	2629,1344	248.0, 132.0	218.36, 121.5	216.23, 121.03
IDRiD_065	1337,1629	2720,1385	2595,245	-	2642.0, -	1870.3, -	1668.13, -
IDRiD_066	2049,1509	801,1374	1623,1576	912,1348	493.0, 137.0	431.24, 114.0	426.55, 111.47
IDRiD_067	1757,1790	3166,1677	1687,1713	3007,1518	147.0, 318.0	104.06, 224.86	92.81, 200.33
IDRiD_068	2116,1719	3301,1438	2123,1765	3198,1422	53.0, 119.0	46.53, 104.24	46.05, 103.13
IDRiD_069	2259,1932	977,1797	2165,1997	1068,1776	159.0, 112.0	114.28, 93.39	103.39, 91.37
IDRiD_070	1547,1558	2780,1475	1512,1497	2733,1438	96.0, 84.0	70.33, 59.82	64.62, 53.66
IDRiD_071	2604,1670	1337,1460	-	1201,1439	-, 157.0	-, 137.61	-, 136.17
IDRiD_072	1817,1865	3098,1782	1763,1843	3008,1737	76.0, 135.0	58.31, 100.62	55.19, 93.6
IDRiD_073	2045,1561	805,1509	2012,1551	936,1431	43.0, 209.0	34.48, 152.46	33.3, 139.64
IDRiD_074	2274,1696	992,1610	2233,1739	1095,1602	84.0, 111.0	59.41, 103.31	52.95, 103.02
IDRiD_075	2308,1505	1022,1408	2247,1477	1046,1389	89.0, 43.0	67.12, 30.61	62.91, 27.45
IDRiD_076	2000,1591	707,1423	2474,2176	835,1398	1059.0, 153.0	752.93, 130.42	674.38, 128.32
IDRiD_077	1704,1550	2986,1644	1484,1219	2899,1573	551.0, 158.0	397.44, 112.29	360.66, 100.54
IDRiD_078	2244,1666	925,1509	2108,1625	1040,1478	177.0, 146.0	142.05, 119.1	137.23, 115.75
IDRiD_079	2367,1644	932,1449	2387,1653	1067,1433	29.0, 151.0	21.93, 135.94	20.59, 135.07
IDRiD_080	2367,1606	977,1374	2440,1611	836,1279	78.0, 236.0	73.17, 170.02	73.01, 154.12

Tabela A.1: Pełne wyniki uzyskane przez metodę MSiRL na bazie IDRiD (kontynuacja).

Nazwa pliku z rozszerzeniem .jpg	Współrzędne środka dołączka wyznaczone przez ekspertów	Współrzędne środka tarczy wzrokowego wyznaczone przez ekspertów	Współrzędne środka dołączka wyznaczone przez MSiRL	Współrzędne środka tarczy wzrokowego wyznaczone przez MSiRL	Różnica między środkami dla $p = 1$	Różnica między środkami dla $p = 2$	Różnica między środkami dla $p = \infty$
IDRiD_081	1952,1363	730,1228	-	852,1200	-, 150.0	-, 125.17	-, 122.49
IDRiD_082	2015,1681	3203,1561	1931,1709	3092,1547	112.0, 125.0	88.54, 111.88	85.02, 111.07
IDRiD_083	1532,1456	2780,1453	1445,1398	2671,1413	145.0, 149.0	104.56, 116.11	94.86, 110.77
IDRiD_084	2293,1505	1067,1366	2267,1492	1074,1381	39.0, 22.0	29.07, 16.55	27.04, 15.49
IDRiD_085	1884,1606	3079,1610	1411,1104	2979,1545	975.0, 165.0	689.73, 119.27	614.75, 108.42
IDRiD_086	1918,1411	685,1149	1708,1359	789,1120	262.0, 133.0	216.34, 107.97	211.06, 104.75
IDRiD_087	1903,1325	3020,1261	1734,1364	2942,1250	208.0, 89.0	173.44, 78.77	169.69, 78.07
IDRiD_088	1562,1546	2903,1588	2639,127	2764,1568	2496.0, 159.0	1781.43, 140.43	1601.36, 139.14
IDRiD_089	2420,1580	1120,1303	2357,1581	1237,1332	64.0, 146.0	63.01, 120.54	63.0, 117.59
IDRiD_090	2435,1580	1120,1299	-	1250,1283	-, 146.0	-, 130.98	-, 130.08
IDRiD_091	1712,1468	3169,1299	-	3074,1279	-, 115.0	-, 97.08	-, 95.29
IDRiD_092	2083,1614	786,1438	-	888,1402	-, 138.0	-, 108.17	-, 103.47
IDRiD_093	1914,1632	726,1404	1784,1604	817,1365	158.0, 130.0	132.98, 99.01	130.43, 93.33
IDRiD_094	2105,1670	835,1434	2031,1691	952,1413	95.0, 138.0	76.92, 118.87	74.56, 117.23
IDRiD_095	1858,1902	3042,1884	1832,1938	3017,1831	62.0, 78.0	44.41, 58.6	40.05, 54.79
IDRiD_096	2146,1554	794,1284	2065,1526	900,1272	109.0, 118.0	85.7, 106.68	82.1, 106.05
IDRiD_097	1918,1558	3188,1520	1619,1397	3075,1448	460.0, 185.0	339.59, 133.99	313.81, 122.01
IDRiD_098	2071,1640	857,1385	1946,1647	952,1374	132.0, 106.0	125.2, 95.63	125.01, 95.05
IDRiD_099	2495,1464	1116,1321	2462,1445	1210,1310	52.0, 105.0	38.08, 94.64	34.98, 94.05
IDRiD_100	2180,1531	913,1276	2165,1493	1011,1253	53.0, 121.0	40.85, 100.66	38.76, 98.42
IDRiD_101	1674,1846	2900,1696	1522,1821	2791,1635	177.0, 170.0	154.04, 124.91	152.23, 115.03
IDRiD_102	2356,1745	992,1520	2954,1887	1928,1624	740.0, 1040.0	614.63, 941.76	600.66, 936.43
IDRiD_103	1562,1715	2806,1640	1478,1503	2708,1587	296.0, 151.0	228.04, 111.41	216.31, 102.92

Tabela A.1: Pełne wyniki uzyskane przez metodę MSiRL na bazie IDRiD (kontynuacja).

Nazwa pliku z rozszerzeniem .jpg	Czas [s]		Zużyta pamięć [MiB]	
	CPU	GPU	CPU	GPU
IDRiD_001	0.78	0.39	233.04	308.24
IDRiD_002	0.78	0.37	250.42	316.19
IDRiD_003	0.69	0.38	240.15	318.31
IDRiD_004	0.74	0.44	245.33	317.29
IDRiD_005	0.68	0.42	238.98	314.24
IDRiD_006	0.66	0.36	239.13	300.14
IDRiD_007	0.69	0.35	242.77	308.44
IDRiD_008	0.65	0.35	241.37	303.11
IDRiD_009	0.65	0.38	241.67	306.23
IDRiD_010	0.66	0.35	246.13	313.63

Tabela A.2: Pełne wyniki wydajnościowe uzyskane przez metodę MSiRL na bazie IDRiD. Wersja elektroniczna tabeli: <https://krzywicki.pro/tab-a-2>.

Nazwa pliku z rozszerzeniem .jpg	Czas [s]		Zużyta pamięć [MiB]	
	CPU	GPU	CPU	GPU
IDRiD_011	0.70	0.43	248.64	318.06
IDRiD_012	0.66	0.35	248.17	299.15
IDRiD_013	0.66	0.35	247.30	301.72
IDRiD_014	0.67	0.40	242.48	307.50
IDRiD_015	0.70	0.38	250.53	311.64
IDRiD_016	0.65	0.39	246.71	298.18
IDRiD_017	0.76	0.47	253.93	319.78
IDRiD_018	0.62	0.32	250.14	302.71
IDRiD_019	0.72	0.46	255.08	326.20
IDRiD_020	0.66	0.34	246.93	311.88
IDRiD_021	0.66	0.35	241.16	302.75
IDRiD_022	0.66	0.34	230.88	298.88
IDRiD_023	0.69	0.38	243.20	318.33
IDRiD_024	0.67	0.33	238.04	310.32
IDRiD_025	0.74	0.43	246.54	327.16
IDRiD_026	0.75	0.44	244.42	316.58
IDRiD_027	0.73	0.41	254.80	323.06
IDRiD_028	0.63	0.31	239.61	293.62
IDRiD_029	0.66	0.35	238.82	311.28
IDRiD_030	0.66	0.35	229.00	311.26
IDRiD_031	0.66	0.39	234.43	303.77
IDRiD_032	0.69	0.39	244.16	319.39
IDRiD_033	0.72	0.41	244.65	324.43
IDRiD_034	0.68	0.38	239.21	317.12
IDRiD_035	0.66	0.40	231.58	302.78
IDRiD_036	0.70	0.40	249.69	312.61
IDRiD_037	0.66	0.35	235.19	308.16
IDRiD_038	0.65	0.34	233.61	313.05
IDRiD_039	0.69	0.37	236.72	315.05
IDRiD_040	0.66	0.36	243.44	305.92
IDRiD_041	0.65	0.35	230.27	307.74
IDRiD_042	0.66	0.38	233.81	321.02
IDRiD_043	0.62	0.35	232.82	302.63
IDRiD_044	0.67	0.34	244.80	305.09
IDRiD_045	0.73	0.41	251.98	325.44

Tabela A.2: Pełne wyniki wydajnościowe uzyskane przez metodę MSiRL na bazie IDRiD (kontynuacja).

Nazwa pliku z rozszerzeniem .jpg	Czas [s]		Zużyta pamięć [MiB]	
	CPU	GPU	CPU	GPU
IDRiD_045	0.73	0.41	251.98	325.44
IDRiD_046	0.69	0.47	236.97	320.97
IDRiD_047	0.66	0.35	239.86	302.00
IDRiD_048	0.69	0.38	250.05	320.53
IDRiD_049	0.78	0.38	240.75	312.03
IDRiD_050	0.72	0.41	255.67	326.34
IDRiD_051	0.72	0.49	240.35	324.58
IDRiD_052	0.65	0.38	239.98	299.98
IDRiD_053	0.68	0.40	240.47	324.84
IDRiD_054	0.63	0.32	233.13	309.78
IDRiD_055	0.65	0.35	234.39	305.71
IDRiD_056	0.68	0.37	247.63	308.75
IDRiD_057	0.69	0.42	238.31	320.36
IDRiD_058	0.69	0.40	243.27	310.66
IDRiD_059	0.69	0.45	257.80	331.42
IDRiD_060	0.68	0.49	250.78	321.44
IDRiD_061	0.70	0.39	241.74	310.94
IDRiD_062	0.74	0.44	250.40	315.17
IDRiD_063	0.69	0.38	238.59	315.19
IDRiD_064	0.69	0.37	233.89	317.00
IDRiD_065	0.65	0.34	244.16	308.94
IDRiD_066	0.69	0.38	229.86	315.02
IDRiD_067	0.84	0.55	266.18	343.09
IDRiD_068	0.68	0.37	236.67	309.45
IDRiD_069	0.71	0.40	257.93	319.63
IDRiD_070	0.69	0.38	245.40	320.55
IDRiD_071	0.65	0.34	248.05	307.35
IDRiD_072	0.74	0.43	249.64	326.80
IDRiD_073	0.65	0.34	237.82	313.03
IDRiD_074	0.66	0.34	246.80	303.18
IDRiD_075	0.82	0.50	248.08	331.59
IDRiD_076	0.69	0.38	240.10	309.90
IDRiD_077	0.66	0.37	232.73	311.79
IDRiD_078	0.66	0.34	234.83	313.20
IDRiD_079	0.65	0.34	246.73	302.29
IDRiD_080	0.68	0.38	240.87	303.30

Tabela A.2: Pełne wyniki wydajnościowe uzyskane przez metodę MSiRL na bazie IDRiD (kontynuacja).

Nazwa pliku z rozszerzeniem .jpg	Czas [s]		Zużyta pamięć [MiB]	
	CPU	GPU	CPU	GPU
IDRiD_081	0.62	0.40	234.31	311.73
IDRiD_082	0.68	0.37	237.93	316.81
IDRiD_083	0.69	0.35	251.59	313.82
IDRiD_084	0.66	0.34	230.75	303.06
IDRiD_085	0.76	0.43	235.74	309.41
IDRiD_086	0.66	0.34	242.11	302.06
IDRiD_087	0.71	0.40	238.57	323.75
IDRiD_088	0.68	0.37	237.18	311.50
IDRiD_089	0.71	0.40	254.37	323.84
IDRiD_090	0.62	0.31	231.77	296.29
IDRiD_091	0.62	0.31	250.79	310.53
IDRiD_092	0.62	0.31	249.82	307.03
IDRiD_093	0.68	0.37	239.65	309.20
IDRiD_094	0.68	0.48	245.41	322.05
IDRiD_095	0.68	0.40	251.35	305.39
IDRiD_096	0.64	0.36	243.76	311.55
IDRiD_097	0.69	0.38	248.04	317.75
IDRiD_098	0.65	0.35	242.02	303.17
IDRiD_099	0.65	0.40	229.94	311.58
IDRiD_100	0.69	0.37	247.02	314.48
IDRiD_101	0.65	0.42	232.77	314.61
IDRiD_102	0.74	0.45	252.83	346.38
IDRiD_103	0.65	0.44	243.62	333.27

Tabela A.2: Pełne wyniki wydajnościowe uzyskane przez metodę MSiRL na bazie IDRiD (kontynuacja).

A.2 Metoda MSLW

Nazwa pliku z rozszerzeniem .jpg	Współrzędne środka dołączka wyznaczone przez ekspertów	Współrzędne środka tarczy nerwu wzrokowego wyznaczone przez ekspertów	Współrzędne środka dołączka wyznaczone przez MSLW	Współrzędne środka tarczy nerwu wzrokowego wyznaczone przez MSLW	Różnica między środkami dla $p = 1$	Różnica między środkami dla $p = 2$	Różnica między środkami dla $p = \infty$
IDRiD_001	1914, 1617	651, 1453	-	679, 1394	- 87.0	- 65.31	- 61.03
IDRiD_002	1922, 1468	3158, 1280	1937, 1403	3153, 1264	80.0, 21.0	66.71, 16.76	65.27, 16.16
IDRiD_003	2566, 1591	1101, 1374	2527, 1558	1098, 1376	72.0, 5.0	51.09, 3.61	45.67, 3.27
IDRiD_004	1562, 1550	2881, 1430	1556, 1531	2875, 1392	25.0, 44.0	19.92, 38.47	19.2, 38.05
IDRiD_005	1757, 1310	2840, 1175	1801, 1328	2851, 1138	62.0, 48.0	47.54, 38.6	44.98, 37.32
IDRiD_006	1745, 1580	2963, 1501	1633, 1657	2967, 1482	189.0, 23.0	135.92, 19.42	123.01, 19.06
IDRiD_007	2315, 1505	842, 1374	2249, 1472	857, 1356	99.0, 33.0	73.79, 23.43	68.64, 20.96
IDRiD_008	2203, 1483	895, 1299	2191, 1446	897, 1290	49.0, 11.0	38.9, 9.22	37.42, 9.03
IDRiD_009	2011, 1670	3241, 1591	-	3232, 1571	- 29.0	- 21.93	- 20.59
IDRiD_010	1768, 1771	2948, 1655	1754, 1767	2930, 1621	18.0, 52.0	14.56, 38.47	14.11, 35.6
IDRiD_011	2019, 1629	872, 1385	2013, 1601	888, 1369	34.0, 32.0	28.64, 22.63	28.09, 20.16
IDRiD_012	1865, 1449	647, 1149	1857, 1421	685, 1134	36.0, 53.0	29.12, 40.85	28.22, 38.76
IDRiD_013	2150, 1389	850, 1224	2149, 1364	856, 1198	26.0, 32.0	25.02, 26.68	25.0, 26.11
IDRiD_014	2086, 1917	816, 1636	2060, 1917	782, 1626	26.0, 44.0	26.0, 35.44	26.0, 34.29
IDRiD_015	1955, 1576	3076, 1483	-	3073, 1449	- 37.0	- 34.13	- 34.01
IDRiD_016	1937, 1288	3241, 1325	1930, 1272	3244, 1324	23.0, 4.0	17.46, 3.16	16.43, 3.04
IDRiD_017	2026, 1644	3286, 1321	-	3277, 1297	- 33.0	- 25.63	- 24.41
IDRiD_018	1704, 1573	520, 1340	1714, 1552	588, 1330	31.0, 78.0	23.26, 68.73	21.73, 68.07
IDRiD_019	2161, 1621	962, 1303	2137, 1579	994, 1289	66.0, 46.0	48.37, 34.93	44.46, 32.87
IDRiD_020	1723, 1842	2933, 1786	1695, 1844	2925, 1758	30.0, 36.0	28.07, 29.12	28.0, 28.22

Tabela A.3: Pełne wyniki uzyskane przez metodę MSLW na bazie IDRiD. W kolumnach zawierających różnice między środkami znajdują się odległości wyrażone odległością Minkowskiego dla podanych wartości parametru p w kolejności: różnica odległości środka dołączka, różnica odległości środka tarczy nerwu wzrokowego. Wersja elektroniczna tabeli: <https://krzywicki.pro/tab-a-3>.

Nazwa pliku z rozszerzeniem .jpg	Współrzędne środka dołączka wyznaczone przez ekspertów	Współrzędne środka tarczy wzrokowego wyznaczone przez ekspertów	Współrzędne środka dołączka wyznaczone przez MSLW	Współrzędne środka tarczy wzrokowego wyznaczone przez MSLW	Różnica między środkami dla $p = 1$	Różnica między środkami dla $p = 2$	Różnica między środkami dla $p = \infty$
IDRiD_021	2011, 1670	1018, 1415	-	1025, 1388	- 34.0	- 27.89	- 27.16
IDRiD_022	2000, 1385	3136, 1325	2329, 1527	3130, 1309	471.0, 22.0	358.34, 17.09	337.59, 16.28
IDRiD_023	2124, 1359	865, 1059	2108, 1322	898, 1056	53.0, 36.0	40.31, 33.14	37.97, 33.01
IDRiD_024	2266, 1314	902, 1138	2263, 1289	912, 1132	28.0, 16.0	25.18, 11.66	25.01, 10.67
IDRiD_025	1543, 1573	2922, 1479	1527, 1554	2931, 1458	35.0, 30.0	24.84, 22.85	22.21, 21.54
IDRiD_026	2367, 1550	962, 1543	2383, 1537	972, 1527	29.0, 26.0	20.62, 18.87	18.46, 17.21
IDRiD_027	1794, 1419	3113, 1370	1755, 1404	3102, 1357	54.0, 24.0	41.79, 17.03	39.73, 15.22
IDRiD_028	1730, 1393	2975, 1359	1698, 1389	2987, 1348	36.0, 23.0	32.25, 16.28	32.02, 14.52
IDRiD_029	2397, 1576	1030, 1411	2390, 1560	1039, 1395	23.0, 25.0	17.46, 18.36	16.43, 16.9
IDRiD_030	2337, 1588	1078, 1404	2351, 1556	1098, 1372	46.0, 52.0	34.93, 37.74	32.87, 34.42
IDRiD_031	2367, 1419	1045, 1291	2438, 1412	1055, 1276	78.0, 25.0	71.34, 18.03	71.02, 16.36
IDRiD_032	1502, 1543	2941, 1524	1517, 1421	2936, 1499	137.0, 30.0	122.92, 25.5	122.08, 25.07
IDRiD_033	1933, 1505	659, 1164	1900, 1521	668, 1139	49.0, 34.0	36.67, 26.57	34.21, 25.38
IDRiD_034	2255, 1153	951, 1044	2267, 1154	962, 1036	13.0, 19.0	12.04, 13.6	12.0, 12.26
IDRiD_035	1532, 1707	2847, 1734	1522, 1709	2844, 1711	12.0, 26.0	10.2, 23.19	10.03, 23.02
IDRiD_036	1700, 1644	3068, 1573	1689, 1640	3052, 1560	15.0, 29.0	11.7, 20.62	11.17, 18.46
IDRiD_037	2105, 1782	3312, 1617	2088, 1754	3301, 1594	45.0, 34.0	32.76, 25.5	29.95, 23.81
IDRiD_038	2293, 1535	962, 1434	2283, 1516	971, 1423	29.0, 20.0	21.47, 14.21	19.88, 12.72
IDRiD_039	1494, 1895	2858, 1771	1511, 1890	2858, 1739	22.0, 32.0	17.72, 32.0	17.14, 32.0
IDRiD_040	1502, 1385	2847, 1138	1490, 1378	2843, 1154	19.0, 20.0	13.89, 16.49	12.75, 16.08
IDRiD_041	2379, 1629	1161, 1505	2375, 1609	1167, 1477	24.0, 34.0	20.4, 28.64	20.05, 28.09
IDRiD_042	1708, 1584	2922, 1535	1710, 1636	2928, 1530	54.0, 11.0	52.04, 7.81	52.0, 6.99
IDRiD_043	1322, 1198	2667, 1153	1294, 1184	2670, 1138	42.0, 18.0	31.3, 15.3	29.12, 15.04
IDRiD_044	2188, 1321	962, 1097	2173, 1319	973, 1082	17.0, 26.0	15.13, 18.6	15.01, 16.76
IDRiD_045	2154, 1408	846, 1284	2159, 1361	862, 1258	52.0, 42.0	47.27, 30.53	47.02, 27.88
IDRiD_046	1787, 1396	3222, 1340	1788, 1375	3224, 1322	22.0, 20.0	21.02, 18.11	21.0, 18.01
IDRiD_047	2169, 1411	988, 1168	2154, 1406	983, 1152	20.0, 21.0	15.81, 16.76	15.18, 16.16
IDRiD_048	2214, 1456	966, 1314	2205, 1438	980, 1314	27.0, 14.0	20.12, 14.0	18.72, 14.0
IDRiD_049	1963, 1617	3199, 1644	1948, 1621	3197, 1619	19.0, 27.0	15.52, 25.08	15.09, 25.0
IDRiD_050	2015, 1314	760, 1160	2008, 1330	758, 1149	23.0, 13.0	17.46, 11.18	16.43, 11.02
IDRiD_051	1775, 1917	3087, 1674	1759, 1889	3076, 1651	44.0, 34.0	32.25, 25.5	29.64, 23.81
IDRiD_052	2120, 1355	884, 1378	2113, 1371	934, 1358	23.0, 70.0	17.46, 53.85	16.43, 51.04
IDRiD_053	2337, 1677	1007, 1505	-	1007, 1497	- 8.0	- 8.0	- 8.0
IDRiD_054	1521, 1543	2851, 1408	1518, 1535	2860, 1385	11.0, 32.0	8.54, 24.7	8.14, 23.45
IDRiD_055	1940, 1869	647, 1644	1944, 1860	658, 1623	13.0, 32.0	9.85, 23.71	9.26, 21.96
IDRiD_056	1547, 1543	2907, 1333	1510, 1547	2912, 1328	41.0, 10.0	37.22, 7.07	37.02, 6.3
IDRiD_057	2011, 1359	719, 1381	-	713, 1372	- 15.0	- 10.82	- 9.81
IDRiD_058	1644, 1441	2810, 1370	1652, 1408	2812, 1331	41.0, 41.0	33.96, 39.05	33.16, 39.0
IDRiD_059	2060, 1295	854, 1172	2056, 1269	824, 1145	30.0, 57.0	26.31, 40.36	26.03, 36.01
IDRiD_060	2491, 1306	1153, 1142	2486, 1295	1163, 1125	16.0, 27.0	12.08, 19.72	11.33, 18.08
IDRiD_061	2161, 1812	3316, 1576	2156, 1809	3321, 1579	8.0, 8.0	5.83, 5.83	5.34, 5.34
IDRiD_062	2150, 1258	884, 1198	2116, 1252	901, 1194	40.0, 21.0	34.53, 17.46	34.06, 17.07
IDRiD_063	1506, 1554	2825, 1479	1480, 1539	2816, 1474	41.0, 14.0	30.02, 10.3	27.57, 9.49
IDRiD_064	1468, 1576	2750, 1355	1453, 1587	2760, 1356	26.0, 11.0	18.6, 10.05	16.76, 10.0
IDRiD_065	1337, 1629	2720, 1385	-	2719, 1375	- 11.0	- 10.05	- 10.0
IDRiD_066	2049, 1509	801, 1374	2133, 1575	808, 1364	150.0, 17.0	106.83, 12.21	95.84, 11.03
IDRiD_067	1757, 1790	3166, 1677	1722, 1765	3149, 1667	60.0, 27.0	43.01, 19.72	38.82, 18.08
IDRiD_068	2116, 1719	3301, 1438	2090, 1704	3293, 1414	41.0, 32.0	30.02, 25.3	27.57, 24.29
IDRiD_069	2259, 1932	977, 1797	2251, 1915	979, 1772	25.0, 27.0	18.79, 25.08	17.57, 25.0
IDRiD_070	1547, 1558	2780, 1475	1554, 1542	2775, 1467	23.0, 13.0	17.46, 9.43	16.43, 8.6
IDRiD_071	2604, 1670	1337, 1460	2564, 1657	1347, 1431	53.0, 39.0	42.06, 30.68	40.45, 29.39
IDRiD_072	1817, 1865	3098, 1782	-	3113, 1760	- 37.0	- 26.63	- 24.11
IDRiD_073	2045, 1561	805, 1509	2033, 1536	815, 1482	37.0, 37.0	27.73, 28.79	25.89, 27.45
IDRiD_074	2274, 1696	992, 1610	2273, 1689	999, 1593	8.0, 24.0	7.07, 18.38	7.01, 17.39
IDRiD_075	2308, 1505	1022, 1408	2305, 1483	1042, 1384	25.0, 44.0	22.2, 31.24	22.02, 27.95
IDRiD_076	2000, 1591	707, 1423	1980, 1592	735, 1398	21.0, 53.0	20.02, 37.54	20.0, 33.49
IDRiD_077	1704, 1550	2986, 1644	1651, 1536	2979, 1598	67.0, 53.0	54.82, 46.53	53.32, 46.05
IDRiD_078	2244, 1666	925, 1509	2225, 1649	932, 1485	36.0, 31.0	25.5, 25.0	22.75, 24.2
IDRiD_079	2367, 1644	932, 1449	2359, 1643	946, 1429	9.0, 34.0	8.06, 24.41	8.01, 22.07
IDRiD_080	2367, 1606	977, 1374	2371, 1594	965, 1317	16.0, 69.0	12.65, 58.25	12.15, 57.18

Tabela A.3: Pełne wyniki uzyskane przez metodę MSLW na bazie IDRiD (kontynuacja).

Nazwa pliku z rozszerzeniem .jpg	Współrzędne środka dołączka wyznaczone przez ekspertów	Współrzędne środka tarczy wzrokowego wyznaczone przez ekspertów	Współrzędne środka dołączka wyznaczone przez MSLW	Współrzędne środka tarczy wzrokowego wyznaczone przez MSLW	Różnica między środkami dla $p = 1$	Różnica między środkami dla $p = 2$	Różnica między środkami dla $p = \infty$
IDRiD_081	1952, 1363	730, 1228	1876, 1312	750, 1204	127.0, 44.0	91.53, 31.24	82.99, 27.95
IDRiD_082	2015, 1681	3203, 1561	1991, 1672	3206, 1535	33.0, 29.0	25.63, 26.17	24.41, 26.01
IDRiD_083	1532, 1456	2780, 1453	1498, 1475	2788, 1419	53.0, 42.0	38.95, 34.93	35.87, 34.15
IDRiD_084	2293, 1505	1067, 1366	2313, 1488	1071, 1401	37.0, 39.0	26.25, 35.23	23.46, 35.02
IDRiD_085	1884, 1606	3079, 1610	1904, 1608	3095, 1583	22.0, 43.0	20.1, 31.38	20.01, 28.76
IDRiD_086	1918, 1411	685, 1149	1894, 1423	679, 1136	36.0, 19.0	26.83, 14.32	24.96, 13.41
IDRiD_087	1903, 1325	3020, 1261	1886, 1311	3020, 1226	31.0, 35.0	22.02, 35.0	19.71, 35.0
IDRiD_088	1562, 1546	2903, 1588	1586, 1537	2908, 1583	33.0, 10.0	25.63, 7.07	24.41, 6.3
IDRiD_089	2420, 1580	1120, 1303	2384, 1568	1126, 1296	48.0, 13.0	37.95, 9.22	36.44, 8.24
IDRiD_090	2435, 1580	1120, 1299	2421, 1576	1131, 1291	18.0, 19.0	14.56, 13.6	14.11, 12.26
IDRiD_091	1712, 1468	3169, 1299	-	3153, 1295	- 20.0	- 16.49	- 16.08
IDRiD_092	2083, 1614	786, 1438	1992, 1480	794, 1420	225.0, 26.0	161.98, 19.7	146.74, 18.51
IDRiD_093	1914, 1632	726, 1404	1894, 1604	720, 1384	48.0, 26.0	34.41, 20.88	31.06, 20.18
IDRiD_094	2105, 1670	835, 1434	2076, 1629	846, 1407	70.0, 38.0	50.22, 29.15	45.36, 27.6
IDRiD_095	1858, 1902	3042, 1884	1844, 1901	3035, 1845	15.0, 46.0	14.04, 39.62	14.0, 39.08
IDRiD_096	2146, 1554	794, 1284	2108, 1531	800, 1267	61.0, 23.0	44.42, 18.03	40.62, 17.25
IDRiD_097	1918, 1558	3188, 1520	1912, 1539	3196, 1503	25.0, 25.0	19.92, 18.79	19.2, 17.57
IDRiD_098	2071, 1640	857, 1385	1998, 1629	856, 1363	84.0, 23.0	73.82, 22.02	73.08, 22.0
IDRiD_099	2495, 1464	1116, 1321	2462, 1439	1128, 1300	58.0, 33.0	41.4, 24.19	37.22, 22.23
IDRiD_100	2180, 1531	913, 1276	2167, 1511	928, 1254	33.0, 37.0	23.85, 26.63	21.68, 24.11
IDRiD_101	1674, 1846	2900, 1696	1623, 1847	2895, 1678	52.0, 23.0	51.01, 18.68	51.0, 18.13
IDRiD_102	2356, 1745	992, 1520	-	990, 1495	- 27.0	- 25.08	- 25.0
IDRiD_103	1562, 1715	2806, 1640	1555, 1710	2811, 1609	12.0, 36.0	8.6, 31.4	7.76, 31.04

Tabela A.3: Pełne wyniki uzyskane przez metodę MSLW na bazie IDRiD (kontynuacja).

Nazwa pliku z rozszerzeniem .jpg	Czas [s]		Zużyta pamięć [MiB]	
	CPU	GPU	CPU	GPU
IDRiD_001.jpg	23.44	12.38	3506.30	3453.64
IDRiD_002.jpg	16.03	2.06	3575.03	3464.88
IDRiD_003.jpg	16.44	2.34	3550.88	3453.50
IDRiD_004.jpg	16.64	2.25	3608.61	3454.27
IDRiD_005.jpg	16.52	2.22	3523.94	3436.04
IDRiD_006.jpg	16.53	2.30	3463.01	3446.74
IDRiD_007.jpg	16.39	2.36	3484.36	3476.72
IDRiD_008.jpg	16.96	2.33	3480.08	3442.03
IDRiD_009.jpg	16.73	2.29	3518.58	3447.72
IDRiD_010.jpg	16.86	2.22	3627.67	3471.58

Tabela A.4: Pełne wyniki wydajnościowe uzyskane przez metodę MSLW na bazie IDRiD. Wersja elektroniczna tabeli: <https://krzywicki.pro/tab-a-4>.

Nazwa pliku z rozszerzeniem .jpg	Czas [s]		Zużyta pamięć [MiB]	
	CPU	GPU	CPU	GPU
IDRiD_011.jpg	16.53	2.29	3562.24	3433.12
IDRiD_012.jpg	16.52	2.23	3589.46	3417.92
IDRiD_013.jpg	16.68	2.26	3581.10	3456.96
IDRiD_014.jpg	16.78	2.35	3590.44	3463.36
IDRiD_015.jpg	16.51	2.30	3575.40	3470.18
IDRiD_016.jpg	16.67	2.29	3576.65	3470.28
IDRiD_017.jpg	16.66	2.19	3588.28	3448.47
IDRiD_018.jpg	16.65	2.07	3656.96	3475.86
IDRiD_019.jpg	16.42	1.89	3446.81	2887.86
IDRiD_020.jpg	18.61	1.85	3431.84	3523.40
IDRiD_021.jpg	17.17	1.83	3432.45	3538.68
IDRiD_022.jpg	17.03	1.84	3549.39	3547.47
IDRiD_023.jpg	16.80	1.88	3446.22	3563.62
IDRiD_024.jpg	16.85	2.05	3583.36	3540.85
IDRiD_025.jpg	16.69	2.05	3491.00	3497.65
IDRiD_026.jpg	16.63	2.18	3525.95	3544.57
IDRiD_027.jpg	16.88	2.20	3427.43	3523.79
IDRiD_028.jpg	16.77	2.27	3482.47	3543.18
IDRiD_029.jpg	16.58	2.26	3485.35	3532.85
IDRiD_030.jpg	16.68	2.31	3603.62	3541.98
IDRiD_031.jpg	16.90	2.27	3553.48	3554.78
IDRiD_032.jpg	16.68	2.18	3534.01	3540.74
IDRiD_033.jpg	16.63	2.31	3507.01	3552.96
IDRiD_034.jpg	16.55	2.27	3597.09	3545.68
IDRiD_035.jpg	16.81	2.41	3614.44	3521.15
IDRiD_036.jpg	16.56	2.30	3631.54	3558.46
IDRiD_037.jpg	16.52	2.32	3661.67	3598.88
IDRiD_038.jpg	16.61	2.30	3530.34	3564.22
IDRiD_039.jpg	17.08	2.36	3570.06	3541.04
IDRiD_040.jpg	16.88	2.42	3639.54	3538.14
IDRiD_041.jpg	16.54	2.11	3607.98	3548.57
IDRiD_042.jpg	16.63	1.93	3622.27	3549.45
IDRiD_043.jpg	16.71	1.86	3648.17	3558.65
IDRiD_044.jpg	16.43	1.83	3669.96	3544.76
IDRiD_045.jpg	16.48	1.86	3670.21	3547.92

Tabela A.4: Pełne wyniki wydajnościowe uzyskane przez metodę MSLW na bazie IDRiD (kontynuacja).

Nazwa pliku z rozszerzeniem .jpg	Czas [s]		Zużyta pamięć [MiB]	
	CPU	GPU	CPU	GPU
IDRiD_046.jpg	16.60	1.87	3589.43	3555.60
IDRiD_047.jpg	16.52	2.03	3541.41	3575.52
IDRiD_048.jpg	16.42	2.16	3515.64	3569.47
IDRiD_049.jpg	16.66	2.26	3442.81	3561.89
IDRiD_050.jpg	17.30	2.31	3535.68	3556.92
IDRiD_051.jpg	16.42	2.22	3470.61	3550.95
IDRiD_052.jpg	16.61	2.37	3463.55	3601.30
IDRiD_053.jpg	16.88	2.21	3462.42	3559.29
IDRiD_054.jpg	16.65	2.35	3465.60	3560.83
IDRiD_055.jpg	16.67	2.31	3505.50	3565.88
IDRiD_056.jpg	16.45	2.26	3455.57	3550.89
IDRiD_057.jpg	16.63	2.28	3465.34	3570.50
IDRiD_058.jpg	16.86	2.24	3441.74	3549.14
IDRiD_059.jpg	16.43	2.19	3455.81	3532.85
IDRiD_060.jpg	16.64	2.35	3478.06	3542.94
IDRiD_061.jpg	16.47	2.22	3440.04	3533.66
IDRiD_062.jpg	16.53	2.33	3454.90	3552.80
IDRiD_063.jpg	16.74	2.38	3431.79	3533.51
IDRiD_064.jpg	16.93	2.41	3498.10	3548.85
IDRiD_065.jpg	16.77	2.21	3462.71	3555.55
IDRiD_066.jpg	16.66	1.95	3440.30	3532.68
IDRiD_067.jpg	16.49	1.86	3456.58	3537.33
IDRiD_068.jpg	16.80	1.85	3443.39	3527.48
IDRiD_069.jpg	17.42	1.89	3512.51	3542.87
IDRiD_070.jpg	17.02	1.94	3484.41	3546.85
IDRiD_071.jpg	16.56	2.19	3458.22	3559.55
IDRiD_072.jpg	16.71	2.08	3467.68	3556.21
IDRiD_073.jpg	16.63	2.30	3436.87	3585.56
IDRiD_074.jpg	16.60	2.27	3481.62	3579.22
IDRiD_075.jpg	16.91	2.31	3443.42	3614.80
IDRiD_076.jpg	16.80	2.36	3462.14	3563.85
IDRiD_077.jpg	16.63	2.31	3453.35	3553.57
IDRiD_078.jpg	17.26	2.38	3433.65	3547.40
IDRiD_079.jpg	16.70	2.34	3451.97	3556.60
IDRiD_080.jpg	16.84	2.36	3456.55	3572.25

Tabela A.4: Pełne wyniki wydajnościowe uzyskane przez metodę MSLW na bazie IDRiD (kontynuacja).

Nazwa pliku z rozszerzeniem .jpg	Czas [s]		Zużyta pamięć [MiB]	
	CPU	GPU	CPU	GPU
IDRiD_081.jpg	16.75	2.26	3422.61	3552.08
IDRiD_082.jpg	16.97	2.31	3515.73	3572.30
IDRiD_083.jpg	16.74	2.37	3446.83	3559.60
IDRiD_084.jpg	16.80	2.38	3434.78	3547.22
IDRiD_085.jpg	17.54	2.26	3438.86	3551.82
IDRiD_086.jpg	17.01	2.37	3445.42	3542.68
IDRiD_087.jpg	17.32	2.29	3434.01	3560.10
IDRiD_088.jpg	17.31	2.33	3445.53	3568.74
IDRiD_089.jpg	17.29	2.01	3451.79	3535.52
IDRiD_090.jpg	17.25	1.92	3500.95	3552.13
IDRiD_091.jpg	16.77	1.87	3454.65	3589.11
IDRiD_092.jpg	16.82	1.86	3456.54	3585.72
IDRiD_093.jpg	17.11	1.90	3459.75	3617.88
IDRiD_094.jpg	16.84	2.06	3464.87	3553.06
IDRiD_095.jpg	16.62	2.11	3461.10	3571.75
IDRiD_096.jpg	17.85	2.34	3442.93	3581.26
IDRiD_097.jpg	17.17	2.31	3499.10	3579.08
IDRiD_098.jpg	16.85	2.21	3437.95	3565.75
IDRiD_099.jpg	16.97	2.22	3539.86	3568.86
IDRiD_100.jpg	16.82	2.35	3553.02	3569.98
IDRiD_101.jpg	16.90	2.23	3423.63	3576.41
IDRiD_102.jpg	16.69	2.24	3453.21	3560.26
IDRiD_103.jpg	16.91	2.24	3441.94	3569.44

Tabela A.4: Pełne wyniki wydajnościowe uzyskane przez metodę MSLW na bazie IDRiD (kontynuacja).

Cecha anatomiczna / symbol modelu bazowego	$p = 1$			$p = 2$			$p = \infty$		
	Średnia	Max	Min	Średnia	Max	Min	Średnia	Max	Min
D SNP1W1	39.07	206.00	7.00	30.59	146.00	7.00	28.84	130.37	6.54
TNW SNP1W1	32.55	77.00	2.00	26.02	73.01	2.00	24.79	73.00	2.00
D SNP1W2	40.55	124.00	5.00	32.04	105.12	4.12	30.43	103.29	4.02
TNW SNP1W2	29.85	72.00	1.00	23.76	62.51	1.00	22.62	62.04	1.00
D SNP2W1	41.43	183.00	3.00	32.80	143.28	2.24	31.14	136.99	2.08
TNW SNP2W1	32.91	101.00	7.00	26.02	73.93	5.00	24.71	70.07	4.50
D SNP2W2	43.41	139.00	6.00	35.05	108.00	4.47	33.35	108.00	4.16
TNW SNP2W2	34.18	80.00	9.00	27.48	69.87	6.40	26.25	69.09	5.74
D SNP3W1	43.31	252.00	7.00	33.89	183.17	5.39	32.02	167.28	5.10
TNW SNP3W1	33.12	115.00	9.00	26.24	87.32	6.71	24.93	82.17	6.24
D SNP3W2	52.12	471.00	0.00	40.97	358.34	0.00	38.72	337.59	0.00
TNW SNP3W2	30.74	86.00	3.00	24.88	65.39	3.00	23.80	64.00	3.00
D SNP4W1	40.00	196.00	4.00	31.47	139.18	3.16	29.70	124.50	3.04
TNW SNP4W1	33.88	100.00	6.00	26.91	71.40	4.47	25.58	64.21	4.16
D SNP4W2	43.13	171.00	4.00	33.88	122.41	3.61	31.99	110.34	3.27
TNW SNP4W2	34.56	90.00	4.00	27.08	68.66	2.83	25.58	67.25	2.52
D SNP1W1T	40.97	153.00	4.00	32.16	122.33	4.00	30.41	121.13	4.00
TNW SNP1W1T	31.45	108.00	4.00	25.10	86.68	3.16	23.89	83.75	3.04
D SNP1W2T	37.03	171.00	6.00	29.54	135.67	6.00	28.05	130.47	6.00
TNW SNP1W2T	34.80	102.00	1.00	27.59	90.80	1.00	26.20	90.07	1.00
D SNP2W1T	42.78	147.00	1.00	34.26	105.68	1.00	32.63	95.63	1.00
TNW SNP2W1T	32.58	244.00	3.00	25.40	175.84	3.00	23.96	159.44	3.00
D SNP2W2T	37.34	87.00	2.00	29.39	62.20	2.00	27.80	60.03	2.00
TNW SNP2W2T	34.94	139.00	0.00	27.69	100.40	0.00	26.27	91.22	0.00
D SNP3W1T	39.86	168.00	2.00	31.47	118.79	1.41	29.84	105.83	1.26
TNW SNP3W1T	34.61	90.00	8.00	27.30	70.84	5.83	25.94	67.89	5.34
D SNP3W2T	40.53	110.00	6.00	32.46	92.44	4.47	30.92	92.03	4.16
TNW SNP3W2T	32.57	103.00	3.00	26.03	88.46	2.24	24.79	87.18	2.08
D SNP4W1T	37.88	152.00	7.00	30.38	120.17	5.00	29.00	115.39	4.50
TNW SNP4W1T	30.05	92.00	3.00	23.90	77.67	2.24	22.72	76.24	2.08
D SNP4W2T	49.29	167.00	6.00	37.95	137.28	4.47	35.61	133.74	4.16
TNW SNP4W2T	34.35	98.00	5.00	27.22	73.68	4.12	25.81	73.06	4.02

Tabela A.5: Błąd między współrzędnymi środków dołeczka oraz tarczy nerwu wzrokowego wyznaczonymi przez modele bazowe metody MSLW dla wszystkie zdjęć bazy IDRID za pomocą trzech wariantów odległości Minkowskiego. D – lokalizacja dołeczka, TNW – lokalizacja tarczy nerwu wzrokowego, Max – wartość największa, Min – wartość najmniejsza. Wersja elektroniczna tabeli: <https://krzywicki.pro/tab-a-5>.

Cecha anatomiczna / symbol modelu bazowego	$p = 1$			$p = 2$			$p = \infty$		
	Średnia	Max	Min	Średnia	Max	Min	Średnia	Max	Min
D SNP1W1	39.07	206.00	7.00	30.59	146.00	7.00	28.84	130.37	6.54
TNW SNP1W1	33.05	77.00	2.00	26.52	73.01	2.00	25.30	73.00	2.00
D SNP1W2	40.55	124.00	5.00	32.04	105.12	4.12	30.43	103.29	4.02
TNW SNP1W2	29.73	72.00	1.00	23.62	62.51	1.00	22.48	62.04	1.00
D SNP2W1	41.43	183.00	3.00	32.80	143.28	2.24	31.14	136.99	2.08
TNW SNP2W1	32.26	80.00	7.00	25.65	70.71	5.00	24.41	70.07	4.50
D SNP2W2	43.41	139.00	6.00	35.05	108.00	4.47	33.35	108.00	4.16
TNW SNP2W2	33.84	80.00	9.00	27.15	69.87	6.40	25.91	69.09	5.74
D SNP3W1	43.31	252.00	7.00	33.89	183.17	5.39	32.02	167.28	5.10
TNW SNP3W1	31.95	83.00	9.00	25.43	70.41	8.25	24.19	69.19	8.04
D SNP3W2	52.12	471.00	0.00	40.97	358.34	0.00	38.72	337.59	0.00
TNW SNP3W2	30.55	86.00	3.00	24.73	65.39	3.00	23.66	64.00	3.00
D SNP4W1	40.00	196.00	4.00	31.47	139.18	3.16	29.70	124.50	3.04
TNW SNP4W1	34.07	100.00	6.00	27.03	71.40	4.47	25.69	64.21	4.16
D SNP4W2	43.13	171.00	4.00	33.88	122.41	3.61	31.99	110.34	3.27
TNW SNP4W2	34.59	90.00	4.00	26.93	68.66	2.83	25.38	67.25	2.52
D SNP1W1T	40.97	153.00	4.00	32.16	122.33	4.00	30.41	121.13	4.00
TNW SNP1W1T	31.86	108.00	4.00	25.51	86.68	3.16	24.29	83.75	3.04
D SNP1W2T	37.03	171.00	6.00	29.54	135.67	6.00	28.05	130.47	6.00
TNW SNP1W2T	34.25	102.00	2.00	27.26	90.80	1.41	25.94	90.07	1.26
D SNP2W1T	42.78	147.00	1.00	34.26	105.68	1.00	32.63	95.63	1.00
TNW SNP2W1T	32.34	244.00	3.00	25.31	175.84	3.00	23.91	159.44	3.00
D SNP2W2T	37.34	87.00	2.00	29.39	62.20	2.00	27.80	60.03	2.00
TNW SNP2W2T	35.79	139.00	0.00	28.28	100.40	0.00	26.80	91.22	0.00
D SNP3W1T	39.86	168.00	2.00	31.47	118.79	1.41	29.84	105.83	1.26
TNW SNP3W1T	34.56	90.00	8.00	27.28	70.84	5.83	25.91	67.89	5.34
D SNP3W2T	40.53	110.00	6.00	32.46	92.44	4.47	30.92	92.03	4.16
TNW SNP3W2T	32.66	103.00	3.00	25.99	88.46	2.24	24.72	87.18	2.08
D SNP4W1T	37.88	152.00	7.00	30.38	120.17	5.00	29.00	115.39	4.50
TNW SNP4W1T	30.05	92.00	3.00	23.92	77.67	2.24	22.73	76.24	2.08
D SNP4W2T	49.07	167.00	6.00	37.73	137.28	4.47	35.38	133.74	4.16
TNW SNP4W2T	33.78	85.00	5.00	26.92	73.68	4.12	25.58	73.06	4.02

Tabela A.6: Błąd między współrzędnymi środków doteczka oraz tarczy nerwu wzrokowego wyznaczonymi przez modele bazowe metody MSLW, dla zdjęć IDRID na których metoda MSLW wykryła obie cechy, za pomocą trzech wariantów odległości Minkowskiego. *D* – lokalizacja doteczka, *TNW* – lokalizacja tarczy nerwu wzrokowego, *Max* – wartość największa, *Min* – wartość najmniejsza. Wersja elektroniczna tabeli: <https://krzywicki.pro/tab-a-6>.

Dodatek B

Implementacje metod i narzędzi

B.1 Metoda MSiRL

Implementację metody *MSiLR* wraz z najbardziej aktualną instrukcją instalacji i uruchomienia umieszczono w repozytorium w serwisie *Github* [MSI23], które można pobrać na dysk za pomocą przeglądarki internetowej lub za pomocą systemu kontroli wersji *Git* [CS14].

W celu uruchomienia rozwiązania należy zainstalować interpreter języka *Python* [Pyt23] w wersji przynajmniej 3.10. Po uzyskaniu zawartości repozytorium należy dokonać instalacji bibliotek zewnętrznych za pomocą powłoki systemu operacyjnego i narzędzia *pip* [PIP23] wydając następujące polecenie:

```
> pip install -r /path/to/file/requirements.txt
```

Alternatywną drogą jest uprzednie utworzenie środowiska wirtualnego umożliwiającego separację zależności projektowych interpretera języka *Python*, a następnie dokonanie instalacji właściwych bibliotek. Proces obsługi i zarządzania środowiskami wirtualnymi w różnych systemach operacyjnych został opisany na portalu *RealPython* [VEN23].

Po zainstalowaniu wymaganych bibliotek zewnętrznych implementacja jest gotowa do uruchomienia. W tym celu należy w powłoce systemu operacyjnego wydać następujące polecenie:

```
> python main.py
```

Wynikiem działania wywołanego polecenia będą wyznaczone lokalizacje środka dołączka oraz środka tarczy nerwu wzrokowego na przekazanym zdjęciu dna oka, zgodnie ze wskazanymi parametrami metody.

B.2 Metoda MSLW

Implementację metody *MSLW* wraz z najbardziej aktualną instrukcją instalacji i uruchomienia umieszczono w repozytorium w serwisie *Github* [MSL23], które można pobrać na dysk za pomocą przeglądarki internetowej lub za pomocą systemu kontroli wersji *Git*.

Proces instalacji narzędzi oraz bibliotek zewnętrznych niezbędnych do uruchomienia implementacji omówiono w Podrozdziale B.1.

Po zainstalowaniu wymaganych bibliotek zewnętrznych implementacja jest gotowa do uruchomienia. W tym celu należy w powłoce systemu operacyjnego wydać następujące polecenie:

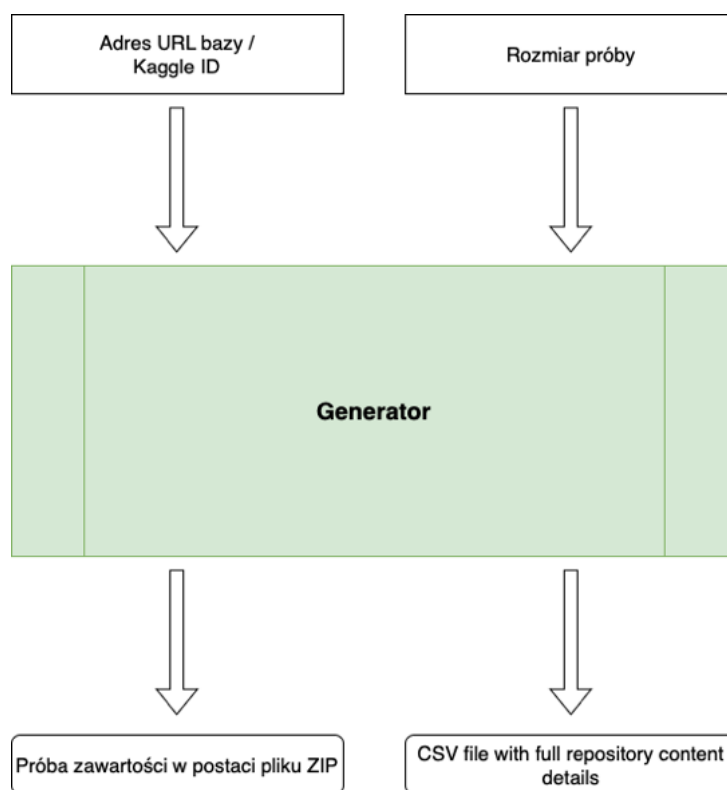
```
> python main.py
```

Wynikiem działania wywołanego polecenia będą wyznaczone lokalizacje środka dołączka oraz środka tarczy nerwu wzrokowego na przekazanym zdjęciu dna oka, zgodnie ze wskazanymi parametrami metody.

B.3 Narzędzie ORSG

Ophthalmic Repository Sample Generator (ORSG) to narzędzie pomocnicze, którego celem jest generowanie pseudolosowych reprezentatywnych prób zawartości obszernych baz zdjęć dna oka oraz generowanie podsumowań w postaci atrybutów technicznych, co przedstawiono na Rysunku B.1. Na Rysunku B.1 przedstawiono schemat działania narzędzia *ORSG*, gdzie można zauważyć że minimalnym wejściem jest adres sieciowy repozytorium zawierającego zdjęcia dna oka i rozmiar

próby, a wyjściem reprezentatywna próba zawartości wraz z plikiem CSV zawierającym atrybuty techniczne.



Rysunek B.1: Schemat ORSG - narzędzia generującego próby zawartości baz zdjęć dna oka oraz atrybuty techniczne. ORSG jako wejście przyjmuje adres URL pliku z bazą lub jego identyfikator na platformie Kaggle [KAG23] oraz rozmiar próby. Wersja elektroniczna obrazu: <https://krzywicki.pro/fig-b-1>.

Implementację narzędzia ORSG wraz z najbardziej aktualną instrukcją instalacji i uruchomienia umieszczono w repozytorium w serwisie *GitHub* [ORS23], które można pobrać na dysk za pomocą przeglądarki internetowej lub za pomocą systemu kontroli wersji *Git*.

Proces instalacji narzędzi oraz bibliotek zewnętrznych niezbędnych do uruchomienia narzędzia omówiono w Podrozdziale B.1.

Po zainstalowaniu wymaganych bibliotek zewnętrznych narzędzie jest gotowe do uruchomienia. Z uwagi na bogaty charakter parametrów wejściowych rekomendowane jest zapoznanie się z systemem pomocy wywołując polecenie:

```
> python main.py -h
```

B.4 Narzędzie FPL

Fundus Photo Labeler (FPL) to narzędzie pomocnicze umożliwiające oznaczanie zdjęć dna oka za pomocą atrybutów anatomicznych i jakościowych, takich jak:

- strona oka (lewe / prawe),
- jakość (dobra / akceptowalna / zła),
- centracja (dołeczek / tarcza nerwu wzrokowego / brak),
- artefakty optyczne (brak / nieprzysłaniające / przysłaniające),
- pole widzenia dna oka wyrażane w stopniach.

Narzędzie *FPL* zawiera graficzny interfejs użytkownika przystosowany do pracy w dwóch środowiskach: lokalnym i chmurowym w przypadku oznaczania obszernych baz zawierających zdjęcia dna oka.

Implementację narzędzia *FPL* wraz z najbardziej aktualną instrukcją instalacji i uruchomienia zarówno w środowisku lokalnym, jak i chmurowym, umieszczono w repozytorium w serwisie *GitHub* [Fun23], które można pobrać na dysk za pomocą przeglądarki internetowej lub za pomocą systemu kontroli wersji *Git*.

Z uwagi na wysoką złożoność procesu uruchomienia narzędzia w obydwu środowiskach oraz potencjalne zmiany interfejsów środowisk chmurowych, zalecane jest zastosowanie instrukcji zamieszczonej w repozytorium celem sprawnej i bezproblemowej instalacji i uruchomienia narzędzia.

Dodatek C

Formuły z komentarzem

C.1 Formuły konwersji obrazów między modelami barw

C.1.1 CIELab

Proces konwersji obrazu reprezentowanego za pomocą modelu barw *RGB* do *CIELab* polega na wykonaniu dwóch konwersji obrazu: modelu barw *CIEXYZ* [Ker10], a następnie z modelu *CIEXYZ* do *CIELab*.

W Równaniu C.1 przedstawiono metodę konwersji obrazu z modelu barw *RGB* do modelu *CIEXYZ*. Konwersję z modelu barw *CIEXYZ* do *CIELab* zaprezentowano w Równaniu C.2.

$$\begin{aligned}
&\text{jeżeli } \frac{R}{95.047} > 0.008856, \text{ to } X = X^{1/3} \\
&\text{w przeciwnym wypadku } X = 7.787 \cdot X + \frac{16}{116} \\
&\text{jeżeli } \frac{G}{100} > 0.008856, \text{ to } Y = Y^{1/3} \\
&\text{w przeciwnym wypadku } Y = 7.787 \cdot Y + \frac{16}{116} \\
&\text{jeżeli } \frac{B}{108.883} > 0.008856, \text{ to } Z = Z^{1/3} \\
&\text{w przeciwnym wypadku } Z = 7.787 \cdot Z + \frac{16}{116}
\end{aligned} \tag{C.1}$$

gdzie:

- X, Y, Z – wartości pikseli obrazu w modelu *CIEXYZ*,
- R, G, B – wartości pikseli obrazu modelu *RGB*.

$$\begin{aligned}
L &= 116 \cdot Y - 16 \\
a &= 500 \cdot (X - Y) \\
b &= 200 \cdot (Y - Z)
\end{aligned} \tag{C.2}$$

gdzie:

- X, Y, Z – wartości pikseli obrazu modelu *CIEXYZ*,
- L, a, b – wartości pikseli obrazu modelu *CIELab*.

Warto mieć na uwadze, że operacje przedstawione w Równaniach C.1 i C.2 należy wykonać dla każdego piksela każdej składowej warstwy. W przypadku

uzyskania w wyniku konwersji wartości piksela typu niecałkowitego, należy ją zaokrąglić do postaci całkowitej.

Można zauważyć, że konwersja z modelu barw *RGB* do *CIEXYZ* jest bardziej złożona niż, przykładowo, z modelu *CIEXYZ* do *CIELab* lub z *RGB* do *YUV*. Model barw *CIEXYZ* jest formatem pośrednim, który ma na celu zachowanie wspólnego punktu odniesienia do innych modeli barw stworzonych przez Międzynarodową Komisję Oświetlenia (*fr. Commission Internationale de l'Eclairage – CIE*).

C.1.2 YUV

Proces konwersji obrazu reprezentowanego za pomocą modelu barw *RGB* do modelu *YUV* polega na zastosowaniu Równania C.3 dla każdego piksela znajdującego się w każdej składowej warstwie.

$$\begin{aligned} Y &= 0.299 \cdot R + 0.587 \cdot G + 0.114 \cdot B \\ U &= 0.492 \cdot (B - Y) \\ V &= 0.877 \cdot (R - Y) \end{aligned} \tag{C.3}$$

gdzie:

- Y, U, V – wartości pikseli obrazu w modelu *YUV*,
- R, G, B – wartości pikseli obrazu modelu *RGB*.

Warto mieć na uwadze fakt, że w przypadku uzyskania w wyniku konwersji wartości piksela typu niecałkowitego, należy ją zaokrąglić do postaci całkowitej.

Można zauważyć, że proces konwersji obrazu z modelu barw *RGB* do *YUV* jest podobny do konwersji z modelu *CIEXYZ* do *CIELab*. Warstwa Y reprezentuje jasność piksela i jest wyliczana na podstawie wartości składowych kolorów R , G i B . Współczynniki 0.299, 0.587 i 0.114 odpowiadają za odpowiednie wagi, które uwzględniają różną percepcję

ludzkiego oka dla różnych kolorów i są wykorzystywane także w transformacji obrazów do postaci skali szarości. Składowe U i V reprezentują informacje o chrominancji, czyli o kolorach i są wyliczane jako różnice między składowymi B i Y oraz R i Y odpowiednio. Współczynniki 0.492 i 0.877 również uwzględniają różnice w percepcji ludzkiego oka dla kolorów niebieskiego i czerwonego.

C.2 Formuły zastosowane w wynikach

C.2.1 Odległość Minkowskiego

Odległość miejska

Odległość miejska (*Manhattan*, Równanie C.4) jest szczególnym nieeuklidesowym przypadkiem odległości Minkowskiego (Równanie 6.1) dla parametru $p = 1$. Nazwa odległości pochodzi od charakteru geometrii ulic dzielnicy Manhattan w Nowym Jorku, która odzwierciedla przestrzeń metryczną. Odległość miejska jest sumą wartości bezwzględnych różnic między współrzędnymi dwóch punktów, co stanowi analogię poruszania się po siatce ulic jedynie wzdłuż osi poziomej i pionowej. W odległości miejskiej różnice między punktami są traktowane jednakowo, dzięki czemu nie przyjmuje wartości spotęgowanych odstępami w dowolnych wymiarach.

$$d_{\text{Manhattan}}(x, y) = \sum_{i=1}^n |x_i - y_i| \quad (\text{C.4})$$

gdzie:

- $x = (x_1, x_2, \dots, x_n)$ i $y = (y_1, y_2, \dots, y_n)$ – współrzędne punktów w przestrzeni metrycznej,
- n - liczba wymiarów przestrzeni metrycznej,
- $|\cdot|$ - operator wartości bezwzględnej.

Odległość euklidesowa

Odległość euklidesowa (Równanie C.5) jest szczególnym przypadkiem odległości Minkowskiego (Równanie 6.1) dla parametru $p = 2$. Nazwa odległości pochodzi od przestrzeni euklidesowej, która stanowi przybliżenie przestrzeni fizycznej dla metryk makroskopowych. Odległość euklidesowa stanowi długość najkrótszej prostej linii łączącej dwa punkty znajdujące się w przestrzeni. Wartości odległości euklidesowej są podnoszone do kwadratu w celu przypisania większej wagi punktom, które są bardziej oddalone w dowolnym wymiarze.

$$d_{\text{Euclidean}}(x, y) = \left(\sum_{i=1}^n |x_i - y_i|^2 \right)^{\frac{1}{2}} = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad (\text{C.5})$$

gdzie:

- $x = (x_1, x_2, \dots, x_n)$ i $y = (y_1, y_2, \dots, y_n)$ – współrzędne punktów w przestrzeni metrycznej,
- n - liczba wymiarów przestrzeni metrycznej.

Odległość Czebyszewa

Odległość euklidesowa (Równanie C.6) jest szczególnym przypadkiem odległości Minkowskiego (szachowa, Równanie 6.1) dla parametru $p = \infty$. Nazwa odległości pochodzi od gry w szachy, gdzie w ten sposób wyraża się odcinek między polami szachownicy w ruchach, które musi wykonać figura króla. Odległość Czebyszewa znajduje zastosowanie w określaniu punktów jako różniące się gdy przyjmują inne wartości w jednym dowolnym wymiarze.

$$d_{\text{Chebyshev}}(x, y) = \lim_{p \rightarrow \infty} \left(\sum_{i=1}^n |x_i - y_i|^p \right)^{\frac{1}{p}} = \max_i |x_i - y_i| \quad (\text{C.6})$$

gdzie:

- $x = (x_1, x_2, \dots, x_n)$ i $y = (y_1, y_2, \dots, y_n)$ – współrzędne punktów w przestrzeni metrycznej,
- n - liczba wymiarów przestrzeni metrycznej,
- $|\cdot|$ - operator wartości bezwzględnej.

Dodatek D

Otwarte problemy

W niniejszym Dodatku skoncentrowano się na identyfikacji i opisie problemów, które pozostają otwarte w zakresie rozpoznawania zdjęć dna oka i analizy metod rozwiązujących powiązane kwestie. Przedstawiono również sugestie i wskazówki dla celów realizacji potencjalnych badań, które mogą przyczynić się do rozwiązania prezentowanych problemów. Celem ułatwienia zrozumienia istoty, problemy podzielono na trzy kategorie: problemy związane z modelowaniem decyzyjnym i z lokalizacją obiektów oraz problemy dotyczące analizy i wydajności metod.

D.1 Modelowanie decyzyjne i lokalizacja obiektów

Jednym z przeznaczeń lokalizacji środków cech anatomicznych dna oka jest wsparcie decyzji w procesie oceny atrybutów anatomicznych zdjęć dna oka. Otwarte kwestie dotyczące modelowania decyzyjnego w klasyfikacji anatomicznej zdjęć dna oka można przedstawić następująco:

- zastosowanie spłotowych sieci neuronowych w klasyfikacji zdjęcia względem centracji oraz strony oka,
- zastosowanie dowolnego klasyfikatora w rozpoznaniu centracji i strony oka na podstawie próbki zawierającej współrzędne środka dołeczka i

środką tarczy nerwu wzrokowego,

- zastosowanie klasyfikatora zespołowego w rozpoznaniu centracji i strony oka na podstawie zarówno zdjęcia dna oka, jak i próbki zawierającej współrzędne środka dołączka i środka tarczy nerwu wzrokowego,
- zastosowanie narzędzi z rodziny *AutoML* [BSPS22] w celu w rozpoznaniu centracji i strony oka na podstawie zdjęcia dna oka oraz próbki zawierającej współrzędne środka dołączka i środka tarczy nerwu wzrokowego,
- lokalizacja środków cech anatomicznych na zdjęciu dna oka za pomocą narzędzi *AutoML*,
- porównanie efektywności wyżej wymienionych podejść między sobą lub z metodami *MSiLR* i *MSLW*.

Z uwagi na brak dostępnego zbioru zdjęć dna oka, na potrzeby testów, zawierającego opisy pod kątem atrybutów anatomicznych i jakościowych, przygotowano publiczne repozytorium pod nazwą *Połączony Zbiór Zdjęć (PZZ)*, które jest dostępne w serwisie *GitHub* [PZZ23]. W repozytorium znajdują się opisy zawierające informacje o centracji, obecności artefaktów i stronie oka dla zdjęć dna oka zaczerpniętych z trzech publicznych baz.

D.2 Analiza i wydajność metod

Jak wspomniano w Rozdziale 6, metoda *MSiLR* charakteryzuje się krótszym czasem wyznaczania lokalizacji oraz niższym zużyciem pamięci operacyjnej, w porównaniu z metodą *MSLW*, co przekłada na jej niższą efektywność wraz z pokryciem. Jednak nie wszystkie techniki wykorzystywane w metodach *MSiLR* oraz *MSLW* mają określoną złożoność obliczeniową, co uniemożliwia dokładne określenie całkowitej złożoności obydwu metod. Określenie złożoności każdej z technik wykorzystywanych w obydwu metodach zaproponowanych w rozprawie może umożliwić wyczerpujące porównanie ich modelu obliczeń.

W kontekście obecnego kryzysu energetycznego, w obliczu którego stoi świat, jedną z istotnych potrzeb stanowi precyzyjne wyznaczenie

konsumpcji energii metod obliczeniowych, co dotyczy również metod *MSiLR* oraz *MSLW*. Energetyczna efektywność odgrywa istotną rolę w dzisiejszej erze technologicznej, gdzie złożone i wyrafinowane algorytmy wymagają znacznych zasobów obliczeniowych i energetycznych. Dostarczenie wnikliwych wyników w tym zakresie może mieć istotne znaczenie dla optymalnego wyboru i wdrażania tych rozwiązań, co może dalej prowadzić do bardziej zrównoważonych i wydajnych rozwiązań naukowych, technologicznych, a także do lepiej zoptymalizowanego przetwarzania informacji pod kątem oszczędności energii.

W obliczu rosnącego zapotrzebowania na wydajne i szybkie sposoby przetwarzania danych, zauważono również potrzebę ponownej zrównoleglonej na GPU implementacji algorytmów wykorzystywanych w metodzie *MSiLR*. Zauważono, że istniejące implementacje dostępne w bibliotece *OpenCV* wykazały pewne zyski wydajności w stosunku do domyślnych sekwencyjnych wersji wykonywanych na CPU, ale nie były one w pełni w stanie wykorzystać potencjału zrównoleglenia ich na GPU. Wprowadzenie ponownej implementacji tych algorytmów na GPU ma na celu optymalizację operacji na poziomie równoległym, wykorzystując możliwości równoległego przetwarzania danych, które oferuje GPU i potencjał wynikający z charakteru tych algorytmów. Biorąc pod uwagę ponowną zrównolegloną implementację technik przetwarzania obrazów, warto także wziąć pod uwagę brak równoległych implementacji algorytmów *Convex hull* oraz *Spaghetti labeling*, co jest możliwe do wykonania i włączenia ich do struktur biblioteki *OpenCV*.

Bibliografia

- [AAM14] Anzar Alam, Mohd. Abdullah, and Ravi Shankar Mishra. Colour contrast enhancement method by scaling the DC coefficients in CIE-LAB colour space. *International Journal of Computer Applications*, 97(22):1–6, July 2014.
- [AB99] Martin Anthony and Peter L. Bartlett. *Neural Network Learning*. Cambridge University Press, November 1999.
- [AB09] Sanjeev Arora and Boaz Barak. *Computational Complexity: A Modern Approach*. Cambridge University Press, USA, 1st edition, 2009.
- [ABG20] Stefano Allegretti, Federico Bolelli, and Costantino Grana. Optimized block-based algorithms to label connected components on gpus. *IEEE Transactions on Parallel and Distributed Systems*, 31(2):423–438, 2020.
- [Ace23] Acer ConceptD CN516-72G - specyfikacja sprzętowa. <https://store.acer.com/pl-pl/conceptd-5-laptop-cn516-72g-czarny>, 2023 (dostęp: 30 czerwca 2023).
- [ADFM19] Silke Aumann, Sabine Donner, Jörg Fischer, and Frank Müller. Optical coherence tomography (OCT): Principle and technical realization. In *High Resolution Imaging in Microscopy and Ophthalmology*, pages 59–85. Springer International Publishing, 2019.
- [AFB16] Muhammad Abdullah, Muhammad Moazam Fraz, and Sarah A. Barman. Localization and segmentation of optic disc in retinal images using circular hough transform and grow cut algorithm. *PeerJ*, 4:e2003, May 2016.

- [ALB⁺18] Michael D. Abràmoff, Philip T. Lavin, Michele Birch, Nilay Shah, and James C. Folk. Pivotal trial of an autonomous AI-based diagnostic system for detection of diabetic retinopathy in primary care offices. *npj Digital Medicine*, 1(1), August 2018.
- [ALE⁺16] Michael David Abràmoff, Yiyue Lou, Ali Erginay, Warren Clarida, Ryan Amelon, James C. Folk, and Meindert Niemeijer. Improved Automated Detection of Diabetic Retinopathy on a Publicly Available Dataset Through Integration of Deep Learning. *Investigative Ophthalmology & Visual Science*, 57(13):5200–5206, 10 2016.
- [ALT22] Ridvy Avyodri, Samuel Lukas, and Hendra Tjahyadi. Optical character recognition (ocr) for text recognition and its post-processing method: A literature review. In *2022 1st International Conference on Technology Innovation and Its Applications (ICTIIA)*, pages 1–6, 2022.
- [AMAZ17] Saad Albawi, Tareq Abed Mohammed, and Saad Al-Zawi. Understanding of a convolutional neural network. In *2017 International Conference on Engineering and Technology (ICET)*, pages 1–6, 2017.
- [AMD23a] CPU AMD Ryzen 9 3950X - specyfikacja. <https://www.amd.com/en/support/cpu/amd-ryzen-processors/amd-ryzen-9-desktop-processors/amd-ryzen-9-3950x>, 2023 (dostęp: 02 lipca 2023).
- [AMD23b] GPU NVIDIA Tesla T4 - specyfikacja. <https://www.nvidia.com/en-us/data-center/tesla-t4/>, 2023 (dostęp: 02 lipca 2023).
- [AMD23c] Vulkan Graphics API – AMD. <https://www.amd.com/en/technologies/vulkan>, 2023 (dostęp: 21 lipca 2023).
- [AME⁺22] Elie Abitbol, Alexandra Miere, Jean-Baptiste Excoffier, Carl-Joe Mehanna, Francesca Amoroso, Samuel Kerr, Matthieu Ortala, and Eric H Souied. Deep learning-based classification of retinal vascular diseases using ultra-widefield colour fundus photographs. *BMJ Open Ophthalmology*, 7(1):e000924, February 2022.
- [APHB07] Marco Aiello, Ian Pratt-Hartmann, and Johan Van Benthem, editors. *Handbook of Spatial Logics*. Springer Netherlands, 2007.

- [App23a] Metal Overview – Apple Developer. <https://developer.apple.com/metal/>, 2023 (dostęp: 21 lipca 2023).
- [App23b] Apple Silicon – Apple Developer Documentation. <https://developer.apple.com/documentation/apple-silicon>, 2023 (dostęp: 21 lipca 2023).
- [Apt21] Rajendra S. Apte. Age-related macular degeneration. *New England Journal of Medicine*, 385(6):539–547, August 2021.
- [Arc23] Arche Linux - dystrybucja systemu operacyjnego Linux. <https://archlinux.org>, 2023 (dostęp: 02 lipca 2023).
- [ARS23] Performance Trends in AI. <https://srconstantin.wordpress.com/2017/01/28/performance-trends-in-ai/>, 2023 (dostęp: 28 lipca 2023).
- [AWZ⁺20] Cheolhong An, Yiqian Wang, Junkang Zhang, Dirk-Uwe G. Bartsch, and William R. Freeman. Fovea localization neural network for multimodal retinal imaging. In Michael E. Zelinski, Tarek M. Taha, Jonathan Howe, Abdul A. Awwal, and Khan M. Iftekharuddin, editors, *Applications of Machine Learning 2020*. SPIE, August 2020.
- [AZH⁺21] Laith Alzubaidi, Jinglan Zhang, Amjad J. Humaidi, Ayad Al-Dujaili, Ye Duan, Omran Al-Shamma, J. Santamaría, Mohammed A. Fadhel, Muthana Al-Amidie, and Laith Farhan. Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. *Journal of Big Data*, 8(1), March 2021.
- [BABG20] Federico Bolelli, Stefano Allegretti, Lorenzo Baraldi, and Costantino Grana. Spaghetti labeling: Directed acyclic graphs for block-based connected components labeling. *IEEE Transactions on Image Processing*, 29:1999–2012, 2020.
- [Bac06] Paul Gustav Heinrich Bachmann. *Die Analytische Zahlentheorie / Dargestellt Von Paul Bachmann*. University of Michigan Library, Ann Arbor, MI, September 2006.
- [BAV13] Mawardi Bahri, Ryuichi Ashino, and Rémi Vaillancourt. Convolution theorems for quaternion fourier transform: Properties and applications. *Abstract and Applied Analysis*, 2013:1–10, 2013.

- [BBL⁺23] Bernd Bischl, Martin Binder, Michel Lang, Tobias Pielok, Jakob Richter, Stefan Coors, Janek Thomas, Theresa Ullmann, Marc Becker, Anne-Laure Boulesteix, Difan Deng, and Marius Lindauer. Hyperparameter optimization: Foundations, algorithms, best practices, and open challenges. *WIREs Data Mining and Knowledge Discovery*, 13(2), January 2023.
- [BCBG18] Federico Bolelli, Michele Cancilla, Lorenzo Baraldi, and Costantino Grana. Toward reliable experiments on the performance of connected components labeling algorithms. *Journal of Real-Time Image Processing*, 17(2):229–244, March 2018.
- [BDMS⁺22] Valérie Biousse, Helen V Danesh-Meyer, Amit M Saindane, Cédric Lamirel, and Nancy J Newman. Imaging of the optic nerve: technological advances and future prospects. *The Lancet Neurology*, 21(12):1135–1150, December 2022.
- [BH89] Eric B. Baum and David Haussler. What size net gives valid generalization? *Neural Computation*, 1(1):151–160, March 1989.
- [BM03] Peter L Bartlett and Wolfgang Maass. Vapnik-chervonenkis dimension of neural nets. *The handbook of brain theory and neural networks*, pages 1188–1192, 2003.
- [BMK⁺18] Saikat Basu, Supratik Mukhopadhyay, Manohar Karki, Robert DiBiano, Sangram Ganguly, Ramakrishna Nemani, and Shreekanth Gayaka. Deep neural networks for texture classification—a theoretical analysis. *Neural Networks*, 97:173–182, January 2018.
- [BMM98] Peter L. Bartlett, Vitaly Maiorov, and Ron Meir. Almost linear vc-dimension bounds for piecewise polynomial networks. *Neural Computation*, 10(8):2159–2173, 1998.
- [BMS⁺19] Muhammad Naseer Bajwa, Muhammad Imran Malik, Shoaib Ahmed Siddiqui, Andreas Dengel, Faisal Shafait, Wolfgang Neumeier, and Sheraz Ahmed. Two-stage framework for optic disc localization and glaucoma classification in retinal fundus images using deep learning. *BMC Medical Informatics and Decision Making*, 19(1), July 2019.
- [Bra00] G. Bradski. The OpenCV Library. *Dr. Dobb’s Journal of Software Tools*, 2000.

- [Bre96] Leo Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, August 1996.
- [Bre99] Leo Breiman. Pasting small votes for classification in large databases and on-line. *Machine Learning*, 36(1/2):85–103, 1999.
- [Bre01] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.
- [BSPS22] Maroua Bahri, Flavia Salutarì, Andrian Putina, and Mauro Sozio. AutoML: state of the art with a focus on anomaly detection, challenges, and research directions. *International Journal of Data Science and Analytics*, 14(2):113–126, February 2022.
- [BZW⁺23] Yaru Bai, Xinpeng Zhang, Congcong Wang, Haojie Gu, Meng Zhao, and Fan Shi. Microaneurysms detection in retinal fundus images based on shape constraint with region-context features. *Biomedical Signal Processing and Control*, 85:104903, August 2023.
- [CCP⁺21] Giovanni Capobianco, Carmine Cerrone, Andrea Di Placido, Daniel Durand, Luigi Pavone, Davide Donato Russo, and Fabio Sebastiano. Image convolution: a linear programming approach for filters design. *Soft Computing*, 25(14):8941–8956, April 2021.
- [Che19] Li-Pang Chen. Mehryar mohri, afshin rostamizadeh, and ameer talwalkar: Foundations of machine learning, second edition. *Statistical Papers*, 60(5):1793–1795, July 2019.
- [Coo71] Stephen A. Cook. The complexity of theorem-proving procedures. In *Proceedings of the Third Annual ACM Symposium on Theory of Computing*, STOC '71, page 151–158, New York, NY, USA, 1971. Association for Computing Machinery.
- [CS14] Scott Chacon and Ben Straub. *Pro git*. Apress, 2014.
- [Cud23a] NVIDIA Cuda Toolkit - zestaw bibliotek do wsparcia obliczeń na GPU zgodnych z technologią CUDA. <https://developer.nvidia.com/cuda-toolkit>, 2023 (dostęp: 02 lipca 2023).
- [CuD23b] NVIDIA Cuda Deep Neural Network (CuDNN) - biblioteka do obsługi sieci neuronowych na procesorach graficznych zgodnych

- z technologią Cuda. <https://developer.nvidia.com/cudnn>, 2023 (dostęp: 02 lipca 2023).
- [CWA⁺23] Lauren J. Coan, Bryan M. Williams, Venkatesh Krishna Adithya, Swati Upadhyaya, Ala Alkafri, Silvester Czanner, Rengaraj Venkatesh, Colin E. Willoughby, Srinivasan Kavitha, and Gabriela Czanner. Automatic detection of glaucoma via fundus imaging and artificial intelligence: A review. *Survey of Ophthalmology*, 68(1):17–41, January 2023.
- [DCP⁺16] Benjamin Michael Davis, Laura Crawley, Milena Pahlitzsch, Fatimah Javaid, and Maria Francesca Cordeiro. Glaucoma: the retina and beyond. *Acta Neuropathologica*, 132(6):807–826, August 2016.
- [Din21] Phu-Hung Dinh. A novel approach based on grasshopper optimization algorithm for medical image fusion. *Expert Systems with Applications*, 171:114576, June 2021.
- [DK23] Barış Dinç and Yasin Kaya. A novel hybrid optic disc detection and fovea localization method integrating region-based convnet and mathematical approach. *Wireless Personal Communications*, 129(4):2727–2748, March 2023.
- [DPK⁺23] Cameron Duic, Kristina Pfau, Tiarnan D.L. Keenan, Henry Wiley, Alisa Thavikulwat, Emily Y. Chew, and Catherine Cukras. Hyperreflective foci in age-related macular degeneration are associated with disease severity and functional impairment. *Ophthalmology Retina*, 7(4):307–317, April 2023.
- [DWZ22] Ran Duan, Hongxun Wu, and Renfei Zhou. Faster matrix multiplication via asymmetric hashing. <https://arxiv.org/abs/2210.10173>, 2022.
- [DYC⁺19] Xibin Dong, Zhiwen Yu, Wenming Cao, Yifan Shi, and Qianli Ma. A survey on ensemble learning. *Frontiers of Computer Science*, 14(2):241–258, August 2019.
- [EHKV89] Andrzej Ehrenfeucht, David Haussler, Michael Kearns, and Leslie Valiant. A general lower bound on the number of examples needed for learning. *Information and Computation*, 82(3):247–261, September 1989.

- [Eni23] Eniac History. <https://www.seas.upenn.edu/about/history-heritage/eniac/>, 2023 (dostęp: 28 lipca 2023).
- [Erd08] Peter Erdi. *Complexity Explained*. Springer Berlin Heidelberg, 2008.
- [FAG⁺04] Donald S. Fong, Lloyd Aiello, Thomas W. Gardner, George L. King, George Blankenship, Jerry D. Cavallerano, Fredrick L. Ferris, and Ronald Klein and. Retinopathy in diabetes. *Diabetes Care*, 27(suppl_1):s84–s87, January 2004.
- [FLA23] Flake8 – dokumentacja. <https://flake8.pycqa.org/en/latest/>, 2023 (dostęp: 02 sierpnia 2023).
- [Fri01] Jerome H. Friedman. Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, 29(5), October 2001.
- [FS95] Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. In *Lecture Notes in Computer Science*, pages 23–37. Springer Berlin Heidelberg, 1995.
- [Fun23] Fundus Photo Labeller - narzędzie do etykietowania zdjęć dna oka. <https://github.com/betacord/FundusPhotoLabeller>, 2023 (dostęp: 27 czerwca 2023).
- [FWS⁺19] H. Fu, Boyang Wang, Jianbing Shen, Shanshan Cui, Yanwu Xu, Jiang Liu, and Ling Shao. Evaluation of retinal image quality assessment networks in different color-spaces. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 2019.
- [FWWL20] Chao Fang, Xiao-Peng Wang, Qing-Sheng Wang, and Jin-Cheng Liang. Construction algorithm for adaptive morphological structuring elements based on the neighborhood gray difference changing vector field and relative density. *IEEE Access*, 8:54928–54939, 2020.
- [FZL⁺22] Yinghua Fu, Ge Zhang, Jiang Li, Dongyan Pan, Yongxiong Wang, and Dawei Zhang. Fovea localization by blood vessel vector in abnormal fundus images. *Pattern Recognition*, 129:108711, September 2022.

- [GBK⁺22] Andrzej Grzybowski, Piotr Brona, Tomasz Krzywicki, Magdalena Gaca-Wysocka, Arleta Berlińska, and Anna Święch. Variability of grading DR screening images among non-trained retina specialists. *Journal of Clinical Medicine*, 11(11):3125, May 2022.
- [GPU23] GPU Audio is the world's first company to unlock your GPU to power your music and audio production workflows. <https://www.gpu.audio>, 2023 (dostęp: 21 lipca 2023).
- [Grz20] Andrzej Grzybowski, editor. *Current Concepts in Ophthalmology*. Springer International Publishing, 2020.
- [Grz21] Andrzej Grzybowski, editor. *Artificial Intelligence in Ophthalmology*. Springer International Publishing, 2021.
- [GSNR23] Andrzej Grzybowski, Panisa Singhanetr, Onnisa Nanegrungsunk, and Paisan Ruamviboonsuk. Artificial intelligence for diabetic retinopathy screening using color retinal photographs: From development to deployment. *Ophthalmology and Therapy*, 12(3):1419–1437, March 2023.
- [GTD13] Thierry Géraud, Hugues Talbot, and Marc Van Droogenbroeck. Algorithms for mathematical morphology. In *Mathematical Morphology*, pages 323–353. John Wiley & Sons, Inc., February 2013.
- [GYA⁺69] Fatih Cakir Gundogan, Umit Yolcu, Fahrettin Akay, Abdullah Ilhan, Gokhan Ozge, and Salih Uzun. Diabetic macular edema: A short review. *Pakistan Journal of Medical Sciences*, 32(2), December 1969.
- [HAE⁺21] Md. Kamrul Hasan, Md. Ashraful Alam, Md. Toufick E Elahi, Shidhartho Roy, and Robert Martí. DRNet: Segmentation and localization of optic disc and fovea from diabetic retinopathy image. *Artificial Intelligence in Medicine*, 111:102001, January 2021.
- [HC15] Lifeng He and Yuyan Chao. A very fast algorithm for simultaneously performing connected-component labeling and euler number computing. *IEEE Transactions on Image Processing*, 24(9):2725–2735, September 2015.

- [HHM16] Somaia Awad Hassan, A.M. Hemeida, and Mountasser M.M. Mahmoud. Performance evaluation of matrix-matrix multiplications using intel's advanced vector extensions (AVX). *Microprocessors and Microsystems*, 47:369–374, November 2016.
- [HRF23] Zbiór danych HRF. <https://www5.cs.fau.de/research/data/fundus-images/>, 2023 (dostęp: 25 czerwca 2023).
- [HRG⁺17] Lifeng He, Xiwei Ren, Qihang Gao, Xiao Zhao, Bin Yao, and Yuyan Chao. The connected-component labeling problem: A review of state-of-the-art algorithms. *Pattern Recognition*, 70:25–43, October 2017.
- [HSZ87] Robert M. Haralick, Stanley R. Sternberg, and Xinhua Zhuang. Image analysis using mathematical morphology. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-9(4):532–550, 1987.
- [Hug15] Christopher J. Hughes. Single-instruction multiple-data execution. In *Single-Instruction Multiple-Data Execution*, 2015.
- [Hum77] Robert Hummel. Image enhancement by histogram transformation. *Computer Graphics and Image Processing*, 6(2):184–195, April 1977.
- [IC13] Yonny S. Izmantoko and Heung-Kook Choi. Time complexity measurement on CUDA-based GPU parallel architecture of morphology operation. *Journal of Korea Multimedia Society*, 16(4):444–452, April 2013.
- [IDR23a] Baza zdjęć IDRID23. <https://idrid.grand-challenge.org>, 2023 (dostęp: 11 czerwca 2023).
- [IDR23b] Lista zwycięskich rozwiązań wyzwania IDRID. <https://idrid.grand-challenge.org/Leaderboard/>, 2023 (dostęp: 25 czerwca 2023).
- [IHKM12] Noor Ibraheem, Mokhtar Hasan, Rafiqul Zaman Khan, and Pramod Mishra. Understanding color models: A review. *ARPJN Journal of Science and Technology*, 2, 01 2012.
- [IK87] J. Illingworth and J. Kittler. The adaptive hough transform. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-9(5):690–698, September 1987.

- [Ima23] ImageNet – Large Scale Visual Recognition Challenge. <https://www.image-net.org/challenges/LSVRC/>, 2023 (dostęp: 28 lipca 2023).
- [Int23] Intel Core i7-11800H - specyfikacja. <https://www.intel.pl/content/www/pl/pl/products/sku/213803/intel-core-i711800h-processor-24m-cache-up-to-4-60-ghz/specifications.html>, 2023 (dostęp: 30 czerwca 2023).
- [IZW⁺23] Wai Cheng Iao, Weixing Zhang, Xun Wang, Yuxuan Wu, Duoru Lin, and Haotian Lin. Deep learning algorithms for screening and diagnosis of systemic diseases based on ophthalmic manifestations: A systematic review. *Diagnostics*, 13(5):900, February 2023.
- [JB15] Asha Merin Jose and Arun A. Balakrishnan. A novel method for glaucoma detection using optic disc and cup segmentation in digital retinal fundus images. In *2015 International Conference on Circuits, Power and Computing Technologies [ICCPCT-2015]*, pages 1–5, 2015.
- [JCQ23] Glenn Jocher, Ayush Chaurasia, and Jing Qiu. YOLO by Ultralytics. <https://github.com/ultralytics/ultralytics>, January 2023.
- [JKK11] Minwoo Jang, Kukhyun Kim, and Kanghee Kim. The performance analysis of ARM NEON technology for mobile platforms. In *Proceedings of the 2011 ACM Symposium on Research in Applied Computation*. ACM, November 2011.
- [JPS⁺20] Yukang Jiang, Jianying Pan, Yanhe Shen, Jin Zhu, Jiamin Huang, Huirui Xie, Xueqin Wang, and Yan Luo. A fast and effective method of macula automatic detection for retina images. <https://arxiv.org/abs/2010.03122>, 2020.
- [KAG23] Przeznaczeniem platformy kaggle jest udostępnianie zbiorów danych oraz rywalizacja konkursowa między zespołami badaczy danych z całego świata. <https://kaggle.com>, 2023 (dostęp: 12 czerwca 2023).
- [KB14] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. <https://arxiv.org/abs/1412.6980>, 2014.

- [KC22] Connor Kenyon and Collin Capano. Apple silicon performance in scientific computing. <https://arxiv.org/abs/2211.00720>, 2022.
- [Ker10] Douglas A Kerr. The cie xyz and xyy color spaces. *Colorimetry*, 1(1):1–16, 2010.
- [KKKS22] Diksha Khurana, Aditya Koli, Kiran Khatter, and Sukhdev Singh. Natural language processing: state of the art, current trends and challenges. *Multimedia Tools and Applications*, 82(3):3713–3744, July 2022.
- [KLW⁺76] D.J. Ketcham, R.W. Lowe, J.W. Weber, Hughes Aircraft Co Culver City Calif Display Systems, and Human Factors Dept. *Image Enhancement Techniques for Cockpit Displays*. Defense Technical Information Center, 1976.
- [KM97] Marek Karpinski and Angus Macintyre. Polynomial bounds for VC dimension of sigmoidal and general pfaffian neural networks. *Journal of Computer and System Sciences*, 54(1):169–176, February 1997.
- [KNB⁺20] Yong Dae Kim, Kyoung Jin Noh, Seong Jun Byun, Soochahn Lee, Tackeun Kim, Leonard Sunwoo, Kyong Joon Lee, Si-Hyuck Kang, Kyu Hyung Park, and Sang Jun Park. Effects of hypertension, diabetes, and smoking on age and sex prediction from retinal fundus images. *Scientific Reports*, 10(1), March 2020.
- [KR06] Brian W Kernighan and Dennis M Ritchie. *C Programming Language*. Perason, 2 edition, 2006.
- [Krz21] Tomasz Krzywicki. Selected image analysis methods for ophthalmology. In *Artificial Intelligence in Ophthalmology*, pages 77–86. Springer International Publishing, 2021.
- [Krz22] Tomasz Krzywicki. *Systemy uczące się: sztuczne sieci neuronowe i uczenie głębokie*. Wydawnictwo Uniwersytetu Warmińsko-Mazurskiego w Olsztynie, Olsztyn, 2022.
- [KS21] Isoon Kanjanasurat and Nitjaree Satayarak. Optic disk and fovea localization by using the direction of blood vessels and morphology operation. In *2021 11th International Conference on Biomedical Engineering and Technology*. ACM, March 2021.

- [KSH12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012.
- [KVB88] Nick Kanopoulos, Nagesh Vasanthavada, and Robert L Baker. Design of an image edge detection filter using the sobel operator. *IEEE Journal of solid-state circuits*, 23(2):358–367, 1988.
- [KW52] J. Kiefer and J. Wolfowitz. Stochastic estimation of the maximum of a regression function. *The Annals of Mathematical Statistics*, 23(3):462–466, September 1952.
- [KYC23] Yong Woo Kim, Young In Yun, and Hyuk Jin Choi. Screening fundus photography predicts and reveals risk factors for glaucoma conversion in eyes with large optic disc cupping. *Scientific Reports*, 13(1), January 2023.
- [LAE⁺16] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. SSD: Single shot MultiBox detector. In *Computer Vision – ECCV 2016*, pages 21–37. Springer International Publishing, 2016.
- [LBD⁺89] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1(4):541–551, 1989.
- [LDF⁺20] Bao Chau K. Ly, Ethan B. Dyer, Jessica L. Feig, Anna L. Chien, and Sandra Del Bino. Research techniques made simple: Cutaneous colorimetry: A reliable technique for objective skin color measurement. *Journal of Investigative Dermatology*, 140(1):3–12.e1, January 2020.
- [Len23] Alessandro Lenci. Understanding natural language understanding systems. a critical analysis. <https://arxiv.org/abs/2303.04229>, 2023.
- [LMB⁺14] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft

- COCO: common objects in context. *CoRR*, abs/1405.0312, 2014.
- [LSD18] Xuechen Li, Linlin Shen, and Jiang Duan. Optic disc and fovea detection using multi-stage region-based convolutional neural network. In *Proceedings of the 2nd International Symposium on Image Computing and Digital Medicine*, ISICDM 2018, page 7–11, New York, NY, USA, 2018. Association for Computing Machinery.
- [LTX⁺22] Jahanzaib Latif, Shanshan Tu, Chuangbai Xiao, Sadaqat Ur Rehman, Azhar Imran, and Yousaf Latif. ODGNet: a deep learning model for automated optic disc localization and glaucoma classification using fundus images. *SN Applied Sciences*, 4(4), March 2022.
- [Luo15] Ming Ronnier Luo. CIELAB. In *Encyclopedia of Color Science and Technology*, pages 1–7. Springer Berlin Heidelberg, 2015.
- [Mem23] Biblioteka Memory Profiler. https://github.com/pythonprofilers/memory_profiler, 2023 (dostęp: 30 czerwca 2023).
- [Mer87] M. Merritt. Spatial and temporal analysis of weather radar reflectivity images. In *ICASSP '87. IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 12, pages 606–609, 1987.
- [MM18] Partha Maji and Robert Mullins. On the reduction of computational complexity of deep convolutional neural networks. *Entropy*, 20(4):305, April 2018.
- [MMRS06] John McCarthy, Marvin L. Minsky, Nathaniel Rochester, and Claude E. Shannon. A proposal for the dartmouth summer research project on artificial intelligence, august 31, 1955. *AI Magazine*, 27(4):12, Dec. 2006.
- [MP43] Warren S. McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*, 5(4):115–133, December 1943.
- [MPM15] Sanket Mehta, Arpita Patel, and Jagrat Mehta. Ccd or cmos image sensor for photography. In *2015 International Conference*

- on Communications and Signal Processing (ICCSPP)*, pages 0291–0294, 2015.
- [MS02] G. Matheron and Jean Paul Frédéric Serra. The birth of mathematical morphology. In *Proceedings of VIth International Symposium on Mathematical Morphology*, page 1–16, 2002.
- [MSI23] Implementacja metody MSiLR. <https://github.com/betacord/MSiLR>, 2023 (dostęp: 02 sierpnia 2023).
- [MSL23] Implementacja metody MSLW. <https://github.com/betacord/MSLW>, 2023 (dostęp: 02 sierpnia 2023).
- [Mun96] David C. Munson. A note on lena. *IEEE Transactions on Image Processing*, 5(1):3–3, 1996.
- [MYP23] mypy – dokumentacja. <https://mypy.readthedocs.io/en/stable/>, 2023 (dostęp: 02 sierpnia 2023).
- [NAB⁺23] Syntia Nusanti, Lazuardiah Anandi, BrigittaMarcia Budihardja, Erika Angraini, and RonaAli Badjrai. The use of artificial intelligence in detecting papilledema from fundus photographs. *Taiwan Journal of Ophthalmology*, 13(2):184, 2023.
- [Nov62] A. B. Novikoff. On convergence proofs on perceptrons. In *Proceedings of the Symposium on the Mathematical Theory of Automata*, volume 12, pages 615–622, New York, NY, USA, 1962. Polytechnic Institute of Brooklyn.
- [NPR⁺09] Sandrina Nunes, Isabel Pires, Andreia Rosa, Lilianne Duarte, Rui Bernardes, and José Cunha-Vaz. Microaneurysm turnover is a biomarker for diabetic retinopathy progression to clinically significant macular edema: Findings for type 2 diabetics with nonproliferative retinopathy. *Ophthalmologica*, 223(5):292–297, 2009.
- [NS56] A. Newell and H. Simon. The logic theory machine—a complex information processing system. *IRE Transactions on Information Theory*, 2(3):61–79, 1956.
- [NVI23] Technologia NVIDIA Cuda. <https://developer.nvidia.com/cuda-zone>, 2023 (dostęp: 30 czerwca 2023).

- [ORS23] Ophthalmic repository sample generator - narzędzie do generowania prób zawartości baz zdjęć dna oka oraz atrybutów technicznych. <https://github.com/betacord/OphthalmicRepositorySampleGenerator>, 2023 (dostęp: 12 czerwca 2023).
- [Ots79] Nobuyuki Otsu. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1):62–66, 1979.
- [Pap93] Christos H Papadimitriou. *Computational Complexity*. Pearson, Upper Saddle River, NJ, November 1993.
- [pdt20] The pandas development team. pandas-dev/pandas: Pandas. <https://doi.org/10.5281/zenodo.3509134>, February 2020.
- [PGM⁺19] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.
- [PHL⁺16] Nishtha Panwar, Philemon Huang, Jiaying Lee, Pearse A. Keane, Tjin Swee Chuan, Ashutosh Richhariya, Stephen Teoh, Tock Han Lim, and Rupesh Agrawal. Fundus photography in the 21st century—a review of recent technological advances and their implications for worldwide healthcare. *Telemedicine and e-Health*, 22(3):198–208, March 2016.
- [PIP23] Python package index - pypi. <https://pypi.org/>, 2023 (dostęp: 02 sierpnia 2023).
- [PKKJ14] Michal Podpora, Grzegorz Paweł Korbaś, and Aleksandra Kawala-Janik. YUV vs RGB—choosing a color space for human-machine interaction. In *Annals of Computer Science and Information Systems*. PTI, September 2014.
- [PMMB22] Biagio Peccerillo, Mirco Mannino, Andrea Mondelli, and Sandro Bartolini. A survey on hardware accelerators: Taxonomy, trends,

- challenges, and perspectives. *Journal of Systems Architecture*, 129:102561, August 2022.
- [Pol23] Lech Polkowski. Hit-or-miss topology - Encyclopedia of Mathematics — encyclopediaofmath.org. https://encyclopediaofmath.org/index.php?title=Hit-or-miss_topology&oldid=51364, 2021 (dostęp: 10 lipca 2023).
- [PPK⁺18] Prasanna Porwal, Samiksha Pachade, Ravi Kamble, Manesh Kokare, Girish Deshmukh, Vivek Sahasrabuddhe, and Fabrice Meriaudeau. Indian Diabetic Retinopathy Image Dataset (IDRID): A Database for Diabetic Retinopathy Screening Research. *Data*, 3(3):25, July 2018.
- [PPK⁺20] Prasanna Porwal, Samiksha Pachade, Manesh Kokare, Girish Deshmukh, Jaemin Son, Woong Bae, Lihong Liu, Jianzong Wang, Xinhui Liu, Liangxin Gao, TianBo Wu, Jing Xiao, Fengyan Wang, Baocai Yin, Yunzhi Wang, Gopichandh Danala, Linsheng He, Yoon Ho Choi, Yeong Chan Lee, Sang-Hyuk Jung, Zhongyu Li, Xiaodan Sui, Junyan Wu, Xiaolong Li, Ting Zhou, Janos Toth, Agnes Baran, Avinash Kori, Sai Saketh Chennamsetty, Mohammed Safwan, Varghese Alex, Xingzheng Lyu, Li Cheng, Qin hao Chu, Pengcheng Li, Xin Ji, Sanyuan Zhang, Yaxin Shen, Ling Dai, Oindrila Saha, Rachana Sathish, Tânia Melo, Teresa Araújo, Balazs Harangi, Bin Sheng, Ruogu Fang, Debdoot Sheet, Andras Hajdu, Yuanjie Zheng, Ana Maria Mendonça, Shaoting Zhang, Aurélio Campilho, Bin Zheng, Dinggang Shen, Luca Giancardo, Gwenolé Quéllec, and Fabrice Mériaudeau. IDRiD: Diabetic retinopathy – segmentation and grading challenge. *Medical Image Analysis*, 59:101561, January 2020.
- [PTC⁺20] Chun-Cheng Peng, Cheng-Jung Tsai, Ting-Yi Chang, Jen-Yuan Yeh, Hsun Dai, and Min-Hsiu Tsai. A fast and noise tolerable binarization method for automatic license plate recognition in the open environment in taiwan. *Symmetry*, 12(8):1374, August 2020.
- [Pyt23] Strona internetowa języka Python. <https://www.python.org>, 2023 (dostęp: 26 czerwca 2023).
- [PZZ23] Połączony Zbiór Zdjęć - dostęp online. <https://github.com/betacord/pzz>, 2023 (dostęp: 16 czerwca 2023).

- [Qui86] J. R. Quinlan. Induction of decision trees. *Machine Learning*, 1(1):81–106, March 1986.
- [RDGF16] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 779–788, 2016.
- [Rez04] Ali M. Reza. Realization of the contrast limited adaptive histogram equalization (CLAHE) for real-time image enhancement. *The Journal of VLSI Signal Processing-Systems for Signal, Image, and Video Technology*, 38(1):35–44, August 2004.
- [RH00] Raúl Rojas and Ulf Hashagen, editors. *The First Computers: History and Architectures*. MIT Press, Cambridge, MA, USA, 2000.
- [RHW86] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, October 1986.
- [RKHD23] Dillon Reis, Jordan Kupec, Jacqueline Hong, and Ahmad Daoudi. Real-time flying object detection with yolov8. <https://arxiv.org/abs/2305.09972>, 2023.
- [RM51] Herbert Robbins and Sutton Monro. A stochastic approximation method. *The Annals of Mathematical Statistics*, 22(3):400–407, September 1951.
- [Roj96] Raúl Rojas. *Neural Networks*. Springer Berlin Heidelberg, 1996.
- [Roj97] R. Rojas. Konrad zuse's legacy: the architecture of the z1 and z3. *IEEE Annals of the History of Computing*, 19(2):5–16, 1997.
- [Ros58] F. Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6):386–408, 1958.
- [RP66] Azriel Rosenfeld and John L. Pfaltz. Sequential operations in digital picture processing. *Journal of the ACM*, 13(4):471–494, October 1966.

- [RTM19] Aditya Raj, Anil Kumar Tiwari, and Maria G. Martini. Fundus image quality assessment: survey, challenges, and future scope. *IET Image Processing*, 13(8):1211–1224, May 2019.
- [rtx23] NVIDIA RTX 3060 - specyfikacja. <https://www.nvidia.com/pl-pl/geforce/graphics-cards/30-series/rtx-3060-3060ti/>, 2023 (dostęp: 30 czerwca 2023).
- [Sau72] N Sauer. On the density of families of sets. *Journal of Combinatorial Theory, Series A*, 13(1):145–147, July 1972.
- [Sch90] Robert E. Schapire. The strength of weak learnability. *Machine Learning*, 5(2):197–227, June 1990.
- [Sch22] Juergen Schmidhuber. Annotated history of modern ai and deep learning. <https://arxiv.org/abs/2212.11279>, 2022.
- [Ser83] Jean Serra. *Image Analysis and Mathematical Morphology*. Academic Press, Inc., USA, 1983.
- [SF68] Irwin Sobel and G. M. Feldman. An isotropic 3×3 image gradient operator. In *Pattern Classification and Scene Analysis*, pages 271–272, 1968.
- [SGS10] John E. Stone, David Gohara, and Guochun Shi. Opencl: A parallel programming standard for heterogeneous computing systems. *Computing in Science & Engineering*, 12(3):66–73, 2010.
- [Sha48] C. E. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27(3):379–423, 1948.
- [Sha88] Claude E. Shannon. Programming a computer for playing chess. In *Computer Chess Compendium*, pages 2–13. Springer New York, 1988.
- [ShCYE17] Vivienne Sze, Yu hsin Chen, Tien-Ju Yang, and Joel S. Emer. Efficient processing of deep neural networks: A tutorial and survey. *Proceedings of the IEEE*, 105:2295–2329, 2017.
- [She89] Spencer E. Sherman. The history of the ophthalmoscope. In *History of Ophthalmology*, pages 221–228. Springer Netherlands, 1989.

- [SK23] Yukihiisa Suzuki and Motohiro Kiyosawa. Relationship between diabetic nephropathy and development of diabetic macular edema in addition to diabetic retinopathy. *Biomedicines*, 11(5):1502, May 2023.
- [Skl82] Jack Sklansky. Finding the convex hull of a simple polygon. *Pattern Recognition Letters*, 1(2):79–83, December 1982.
- [Smi79] Alvy Ray Smith. Tint fill. *ACM SIGGRAPH Computer Graphics*, 13(2):276–283, August 1979.
- [SMKA19] G. Susrutha, K. Mallikarjun, M. Ajay Kumar, and M. Ashok. Analysis on fft and dwt transformations in image processing. In *2019 International Conference on Emerging Trends in Science and Engineering (ICESE)*, volume 1, pages 1–4, 2019.
- [Soi04] Pierre Soille. *Morphological Image Analysis*. Springer Berlin Heidelberg, 2004.
- [Soy20] Derya Soydaner. A comparison of optimization algorithms for deep learning. *International Journal of Pattern Recognition and Artificial Intelligence*, 34(13):2052013, April 2020.
- [SSA⁺23] Mohammad Hasan Shahriari, Hamideh Sabbaghi, Farkhondeh Asadi, Azamosadat Hosseini, and Zahra Khorrami. Artificial intelligence in screening, diagnosis, and classification of diabetic macular edema: A systematic review. *Survey of Ophthalmology*, 68(1):42–53, January 2023.
- [Sta07] Gregory Van Stavern. Optic disc edema. *Seminars in Neurology*, 27(3):233–243, July 2007.
- [Sto74] M. Stone. Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society. Series B (Methodological)*, 36(2):111–147, 1974.
- [Str69] Volker Strassen. Gaussian elimination is not optimal. *Numerische Mathematik*, 13(4):354–356, August 1969.
- [Str13] Bjarne Stroustrup. *The C++ Programming Language*. Addison-Wesley Professional, 4th edition, 2013.
- [TCE23] Juan Terven and Diana Cordova-Esparza. A comprehensive review of yolo: From yolov1 and beyond. <https://arxiv.org/abs/2304.00501>, 2023.

- [TIO23] Tiobe index. <https://www.tiobe.com/tiobe-index/>, 2023 (dostup: 30 lipca 2023).
- [Top19] Eric Topol. *Deep Medicine: How Artificial Intelligence Can Make Healthcare Human Again*. Basic Books, Inc., USA, 1st edition, 2019.
- [TSB⁺19] Huishi Toh, Alexander Smolentsev, Rachel V. Bozadjian, Patrick W. Keeley, Madison D. Lockwood, Ryan Sadjadi, Dennis O. Clegg, Barbara A. Blodi, Peter J. Coffey, Benjamin E. Reese, and James A. Thomson. Vascular changes in diabetic retinopathy—a longitudinal study in the Nile rat. *Laboratory Investigation*, 99(10):1547–1560, October 2019.
- [TTH22] Buket Toptaş, Murat Toptaş, and Davut Hanbay. Detection of optic disc localization from retinal fundus image using optimized color space. *Journal of Digital Imaging*, 35(2):302–319, January 2022.
- [Tur50a] A. M. Turing. Computing machinery and intelligence. *Mind*, 59(236):433–460, 1950.
- [Tur50b] A. M. Turing. Computing machinery and intelligence. *Mind*, 59(236):433–460, 1950.
- [TWS⁺23] Omer Trivizki, Liang Wang, Yingying Shi, David Rabinovitch, Prashanth Iyer, Giovanni Gregori, William Feuer, and Philip J. Rosenfeld. Symmetry of macular fundus features in age-related macular degeneration. *Ophthalmology Retina*, March 2023.
- [Val84] L. G. Valiant. A theory of the learnable. *Commun. ACM*, 27(11):1134–1142, nov 1984.
- [Vap00] Vladimir N. Vapnik. *The Nature of Statistical Learning Theory*. Springer New York, 2000.
- [Vaz01] Vijay V Vazirani. *Approximation algorithms*, volume 1. Springer, 2001.
- [VC15] V. N. Vapnik and A. Ya. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. In *Measures of Complexity*, pages 11–30. Springer International Publishing, 2015.

- [VEN23] Python virtual environments: A primer. <https://realpython.com/python-virtual-environments-a-primer/>, 2023 (dostęp: 02 sierpnia 2023).
- [VGO⁺20] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020.
- [VK04] Vladimir Vezhnevets and Vadim Konushin. "growcut interactive multi-label nd image segmentation by cellular automata. *Graphicon*, 1, 11 2004.
- [vRWC01] Guido van Rossum, Barry Warsaw, and Nick Coghlan. Style guide for Python code. PEP 8, 2001.
- [Wan22] Han Wang. A bibliographic study and quantitative analysis of age-related macular degeneration and fundus images. *Ann. Ophthalmol. Vis. Sci.*, 5:1–8, 2022.
- [Wei95] Mark Allen Weiss. *Data structures and algorithm analysis*. Benjamin-Cummings Publishing Co., Inc., 1995.
- [Win23] Windows 11 - specyfikacja. <https://www.microsoft.com/pl-pl/windows/windows-11>, 2023 (dostęp: 30 czerwca 2023).
- [WKR11] S. Wolf, B. Kirchhof, and M. Reim. *Ocular Fundus: From Findings to Diagnosis*. Thieme, 2011.
- [WLCL22] Han Wang, Kelvin Kam Lung Chong, and Zefeng Li. Applications of ai to age-related macular degeneration: a case study and a brief review. In *2022 International Conference on Computer Engineering and Artificial Intelligence (ICCEAI)*, pages 586–589, 2022.

- [WM10] Wes McKinney. Data Structures for Statistical Computing in Python. In Stéfan van der Walt and Jarrod Millman, editors, *Proceedings of the 9th Python in Science Conference*, pages 56 – 61, 2010.
- [WTW⁺22] Tao Wei, Yonghong Tian, Yaowei Wang, Yun Liang, and Chang Wen Chen. Optimized separable convolution: Yet another efficient convolution operator. *AI Open*, 3:162–171, 2022.
- [WWX⁺19] Jun Wu, Jiapei Wang, Jie Xu, Yiting Wang, Kaiwei Wang, Zongjiang Shang, Dayong Ding, Xirong Li, Gang Yang, Xuemin Jin, Yanting Wang, Fangfang Dai, and Jianping Fan. Fovea localization in fundus photographs by faster r-CNN with physiological prior. In *Ophthalmic Medical Image Analysis*, pages 156–164. Springer International Publishing, 2019.
- [yC91] Santiago Ramón y Cajal. New ideas on the structure of the nervous system in man and vertebrates. *Journal of Cognitive Neuroscience*, 3(3):300–301, 1991.
- [YJD22] Keyuan Yu, Kun Jin, and Xiangyang Deng. Review of deep reinforcement learning. In *2022 IEEE 5th Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC)*, volume 5, pages 41–48, 2022.
- [YOL23] Parametry treningu sieci YOLOv8 rekomendowane przez autorów pierwszej publicznej implementacji. <https://docs.ultralytics.com/modes/train/#arguments>, 2023 (dostęp: 02 lipca 2023).
- [YXL22] Fuchao Yu, Xianchao Xiu, and Yunhui Li. A survey on deep transfer learning and beyond. *Mathematics*, 10(19):3619, October 2022.

Skorowidz

- AI, 37–39, 41
- analiza komponentów połączonych
 - Spaghetti labeling, 70, 121, 132
- aparatura medyczna
 - oftalmoskop, 40
- baza
 - BFO21, 102, 140
 - IDRID, 103, 143, 159
- biblioteka
 - Flake8, 33
 - mypy, 33
 - OpenCV, 32, 148, 160
 - pandas, 33
 - PyTorch, 32
 - SciPy, 33
 - Ultralytics, 32, 33, 148, 160
- błąd
 - SBE, 164
- cechy anatomiczne siatkówki, 29,
 - 100, 105, 106
 - dołeczek, 30, 41, 100, 113
 - lokalizacja, 103
 - naczynia krwionośne, 40
 - plamka żółta, 29, 40, 41, 103, 113
 - tarcza nerwu wzrokowego, 29, 40, 41, 100, 103, 126
 - tętnice, 41
 - współrzędne środka, 105
 - żyły, 41
- dane pacjentów
 - anonimizacja, 101
- Fundacja Okulistyka 21, 102
- fundus kamera
 - Kowa VX-10 alpha, 105
 - Topcon camera Nw-400, 102
- jednostka chorobowa, 100
 - metaboliczna
 - cukrzyca, 28
 - okulistyczna
 - AMD, 28, 29, 42
 - DME, 28, 29, 42, 102
 - DR, 28, 102
 - jaskra, 28, 29, 42
 - ODE, 28, 29, 42
 - systemowa, 29
- klasyfikacja binarna, 85
- lokalizacja cech anatomicznych
 - pokrycie, 159
- lokalizacja cech anatomicznych siatkówki, 29, 30
 - dokładność, 30, 31
 - dołeczek, 42

- pokrycie, 30, 31
- tarcza nerwu wzrokowego, 42
- metoda
 - MSiLR, 161
 - MSiRL, 111
 - MSLW, 139, 161
- model barw, 51
 - chrominacja, 54, 129
 - CIELab, 53, 115, 126, 144, 192
 - CIEXYZ, 192
 - RGB, 40, 51, 115, 126, 144, 192
 - YUV, 54, 126, 194
- morfologia matematyczna, 72
 - Black-hat, 74, 120
 - domknięcie, 74, 130
 - dylacja, 72, 121
 - erozja, 72, 121, 130
 - otwarcie, 74, 130
 - Top-hat, 74, 120
- obiekt
 - kontur
 - Convex hull, 68, 130
- obraz
 - adnotacja, 100
 - binaryzacja, 60
 - cyfrowy, 50
 - filtr
 - gaussowski, 58, 116
 - Sobela, 59
 - filtrowanie, 56
 - histogram
 - AHE, 66
 - CLAHE, 66, 115, 144
 - wyrównywanie, 63, 65, 66
 - kolorowy, 51
 - przetwarzanie, 110
 - rozpoznawanie, 46
 - segmentacja, 45, 103, 115
 - maska, 105
 - splot, 56, 57, 116
 - transformata, 56
 - obrazowanie medyczne, 100
 - fundoskopia, 40, 100
 - fundus kamera, 40
 - odległość
 - Czebyszewa, 196
 - Euklidesa, 196
 - Manhattan, 195
 - Minkowskiego, 195
 - ORSG, 101
 - perceptron, 81, 82
 - uczenie, 83
 - wielowarstwowy, 81
 - pole widzenia, 28, 102
 - powłoka wypukła konturu, 68
 - procesor
 - CPU, 148, 161
 - GPU, 96, 148, 161
 - progowanie
 - Otsu, 62, 121, 130
 - proste, 60
 - przetwarzanie równoległe, 97
 - Python, 32, 98, 148, 160
 - siatkówka, 40, 41, 100
 - sieci neuronowe, 41, 79, 109
 - sieć neuronowa
 - architektura
 - YOLO, 94
 - YOLOv8, 144, 162
 - uczenie, 83
 - algorytm propagacji wstecznej błędu, 83
 - uczenie transferowe, 89, 147
 - ekstrakcja cech, 90
 - fine-tuning, 90
 - system
 - IDx-DR, 103

- sztuczny neuron, 79
- uczenie maszynowe, 81
 - koncept, 81
 - nadzorowane, 83
 - optymalizacja gradientowa, 83
 - próbka, 82
- uczenie zespołowe
 - bagging, 92
 - boosting, 91
 - pasting, 94, 139
- wymiar
 - VC, 86
- zdjęcie dna oka, 100, 106
 - artefakty optyczne, 101, 105, 106
 - atrybuty
 - anatomiczne, 30, 101
 - epidemiologiczne, 101
 - jakościowe, 30, 101
 - techniczne, 101
- centracja, 42
 - brak, 105
 - dołeczek, 102, 105
 - tarcza nerwu wzrokowego, 102, 105
- jakość, 42, 100, 106
- strona, 42
- zmiany
 - chorobowe, 100
 - naczyniowe, 28
 - mikroaneuryzmat, 28
- złożoność
 - czasowa, 47
 - obliczeniowa, 47
 - pamięciowa, 49