



mgr inż. Magdalena Michalska-Ciekańska

Diagnostyka zmian skórnych z wykorzystaniem
głębokich sieci neuronowych

*Promotor:
prof. dr hab. inż. Oleksandra Hotra*

*Promotor pomocniczy:
dr hab. inż. Zbigniew Omiotek*

ROZPRAWA
DOKTORSKA

*Pragnę złożyć serdeczne podziękowania
prof. dr hab. inż. Aleksandrze Hotrze,
dr hab. inż. Zbigniewowi Omiotkowi
za cenne uwagi i opiekę promotorską
w trakcie pisania niniejszej rozprawy*

Rozprawy Doktorskie – Politechnika Lubelska

Dziedzina nauk inżyneryjno-technicznych
Dyscyplina naukowa automatyka, elektronika,
elektrotechnika i technologie kosmiczne

Politechnika Lubelska
ul. Nadbystrzycka 38
20-618 Lublin



POLITECHNIKA
LUBELSKA
WYDZIAŁ ELEKTROTECHNIKI
I INFORMATYKI

mgr inż. Magdalena Michalska-Ciekańska

Diagnostyka zmian skórnych z wykorzystaniem głębokich sieci neuronowych

Promotor:

prof. dr hab. inż. Aleksandra Hotra

Promotor pomocniczy:

dr hab. inż. Zbigniew Omiotek



POLITECHNIKA
LUBELSKA
LUBLIN UNIVERSITY
OF TECHNOLOGY

Lublin 2023

Spis treści

Streszczenie	7
Abstract	8
Wykaz ważniejszych symboli i akronimów	9
1. Wstęp	11
1.1. Uzasadnienie podjęcia tematyki pracy	11
1.2. Cel, teza i zakres rozprawy	15
2. Obraz kliniczny zmian skórnych	17
2.1. Przegląd popularnych zmian skórnych	18
2.2. Tradycyjne metody diagnostyczne	20
2.3. Diagnostyka z wykorzystaniem metod uczenia maszynowego	23
3. Uczenie głębokie	25
3.1. Uczenie głębokie w diagnostyce zmian skórnych	25
3.2. Sztuczne sieci neuronowe	27
3.3. Głębokie spłotowe sieci neuronowe	32
3.4. Trenowanie i diagnostyka modeli	35
3.5. Technika transfer learning	37
4. Materiał i metodyka badań	41
4.1. Materiał do badań	41
4.2. Wstępne przetwarzanie obrazów	43
4.3. Metodyka przeprowadzania badań	44
4.4. Założenia i ograniczenia przyjęte w badaniach	45
5. Sieci spłotowe wykorzystane w badaniach	47
5.1. VGG16 i VGG19	47
5.2. MobileNetV2	47
5.3. Xception	49
5.4. ResNet	49
5.5. InceptionResNetV2	51
5.6. InceptionV3	51
5.7. EfficientNetB7	53
5.8. DenseNet201	55
5.9. NASNetMobile	57
6. Metryki oceny jakości modeli	59
7. Trenowanie modeli klasyfikacyjnych	63
7.1. Klasyfikacja binarna (ZNAMIEJ – CZERNIAK)	66
7.2. Klasyfikacja binarna (ŁAGODNE – ZŁOŚLIWE)	69
7.3. Klasyfikacja wieloklasowa	72

8. Ewaluacja modeli	75
8.1. Testowanie modeli z wykorzystaniem zbioru ISIC.....	75
8.2. Testowanie modeli z wykorzystaniem zbioru MED-NODE	83
8.3. Testowanie modeli z wykorzystaniem zbioru PAD-UFES-20	85
8.4. Testowanie modeli z wykorzystaniem zbioru PH ²	90
8.5. Testowanie modeli z wykorzystaniem kombinacji zbiorów PAD-UFES-20 i PH ²	94
8.6. Porównanie wyników ewaluacji dla różnych zbiorów danych.....	96
9. Podsumowanie	99
9.1. Porównanie efektów pracy z wynikami innych autorów.....	100
9.2. Oryginalne wyniki uzyskane w pracy	106
9.3. Zastosowanie wyników badań.....	106
9.4. Plan dalszych prac	107
Literatura	109
Załącznik A. Listing kodu do segmentacji zmian skórnych	117
Załącznik B. Listing przykładowego kodu do trenowania modeli.....	120
Załącznik C. Listing przykładowego kodu do ewaluacji modeli.....	123

Streszczenie

Diagnostyka zmian skórnych z wykorzystaniem głębokich sieci neuronowych

Czerniak jest obecnie jedną z najbardziej niebezpiecznych chorób skóry. Oprócz niego na skórze pojawia się wiele innych chorób. Naukowcy rozwijają techniki wczesnego, nieinwazyjnego diagnozowania zmian skórnych na podstawie obrazów dermatoskopowych. W tym celu coraz częściej wykorzystywane są sieci neuronowe. Powstaje wiele narzędzi pozwalających na szybszą implementację sieci neuronowych, m.in. należy do nich biblioteka Keras. Standardowym badaniem diagnostycznym chorób skórnych jest dermatoskopia. Diagnoza jest potwierdzana badaniem histopatologicznym. W rozprawie przedstawiono badania polegające na analizie obrazów zmian skórnych przy użyciu konwolucyjnych sieci neuronowych. Materiał badawczy, w postaci wyników badań dermatoskopowych, pochodził z pięciu baz danych. Największy zbiór zawierał obrazy należące do 2400 pacjentów (1200 chorych i 1200 zdrowych). Badania wykazały, iż przy odpowiednim doborze wcześniej wytrenowanego modelu konwolucyjnej sieci neuronowej oraz jego odpowiedniej modyfikacji, możliwe jest osiągnięcie bardzo dobrych wyników klasyfikacji, przekraczających 90% dla wskaźników wrażliwości (TPR) oraz specyficzności (TNR). W rozprawie zaproponowano algorytm analizy obrazów dermatoskopowych zmian skórnych łagodnych: znamion melanocytowych, łagodnych rogowaceń barwnikowych, brodawki łojotokowej oraz złośliwych, należy do nich: czerniak złośliwy, rak podstawnokomórkowy, rak kolczystokomórkowy. Dla modelu sieci DenseNet201, dla klasyfikacji binarnej i wieloklasowej, uzyskano najlepsze wyniki wskaźników jakości klasyfikacji. W końcowej części pracy przedstawiono propozycję dalszych badań w kierunku opracowania systemu diagnostyki chorób skóry opartego o wyniki przedstawione w rozprawie.

Słowa kluczowe: *głębokie sieci neuronowe, uczenie głębokie, diagnostyka zmian skórnych, konwolucyjne sieci neuronowe, analiza obrazów, diagnostyka medyczna*

Abstract

Diagnosis of skin lesions using deep neural networks

Melanoma is currently one of the most dangerous skin diseases. In addition to it, many other diseases appear on the skin. Scientists are developing techniques for early, non-invasive diagnosis of skin lesions based on dermatoscopic images. For this purpose, neural networks are increasingly used. Many tools are being developed that allow for faster implementation of neural networks, including the Keras library. The standard diagnostic examination for skin diseases is dermatoscopy. The diagnosis is confirmed by histopathological examination. The dissertation presents research involving the analysis of images of skin lesions using convolutional neural networks. The research material, in the form of dermatoscopic test results, came from five databases. The largest collection contained images belonging to 2400 patients (1200 sick and 1200 healthy). Research has shown that with the appropriate selection of a previously trained model of convolutional neural network and its appropriate modification, it is possible to achieve very good classification results, exceeding 90% for sensitivity (TPR) and specificity (TNR) indicators. In the doctoral thesis, an algorithm for the dermatoscopic images analysis of benign skin lesions is proposed: nevus, benign pigmented keratoses, seborrheic keratoses and malignant ones, these include: malignant melanoma, basal cell carcinoma, squamous cell carcinoma. For the DenseNet201 network model, for binary and multiclass classification, the best results of classification quality indicators were obtained. In the final part of the work, a proposal is put forward for further research towards the development of a skin disease diagnostic system based on the results presented in the dissertation.

Keywords: *deep neural networks, deep learning, diagnostics of skin lesions, convolutional neural networks, image analysis, medical diagnostics*

Wykaz ważniejszych symboli i akronimów

Symbole

BRO	– brodawka łojotokowa
CZE	– czerniak złośliwy
DN	– model DenseNet201
EN	– model EfficientNetB7
F, FISHER	– metoda selekcji cech Fishera
I	– model InceptionV3
IRM	– model Xception
IRN	– model InceptionResNetV2
KOL	– rak kolczystokomórkowy
MN	– model MobileNetV2
NNM	– model NASNetMobile
POD	– rak podstawnokomórkowy
RN	– model ResNet50
ROG	– łagodne rogowacenie barwnikowe
V16	– model VGG16
V19	– model VGG19
X	– model Xception
ZNA	– znamię melanocytowe

Akronimy

ACC	– <i>Classification Accuracy</i> – dokładność klasyfikacji
AdaBoost	– <i>Adaptive Boosting</i>
AdaGrad	– <i>Adaptive Gradient Algorithm</i>
Adam	– <i>Adaptive Moment Estimation</i>
ASR	– <i>Age Standardised Rate</i> – wskaźnik standaryzowany pod względem wieku
AUC	– <i>Area Under ROC Curve</i> – pole pod krzywą ROC
CB	– <i>Convolution Base</i> – baza konwolucyjna
CNN	– <i>Convolution Neural Network</i> – splotowe sieci neuronow
DCNN	– <i>Deep Convolutional Neural Network</i> – głębokie konwolucyjne sieci neuronowe
DL	– <i>Deep Learning</i> – głębokie uczenie
ERR	– <i>Classification Error</i> – błąd klasyfikacji
FDR	– <i>False Discovery Rate</i> – wskaźnik fałszywego rozpoznania
FPR	– <i>False Positive Rate</i> – wskaźnik fałszywie dodatni
ISIC	– <i>International Skin Imaging Collaboration</i>
KNN	– <i>K-Nearest Neighbors</i>
LTU	– <i>Lineat Treshold Unit</i>
MLP	– <i>Multi – Layer Perceptron</i> – perceptron wielowarstwowy
NPV	– <i>Negative Predictive Value</i> – wartość predykcyjna ujemna

PPV	– <i>Positive Predictive Value</i> – wartość predycyjna dodatnia
ReLU	– <i>Rectified Linear Unit</i> – funkcja aktywacji warstwy sieci neuronowej
RMSProp	– <i>Root Mean Square Propagation</i>
RNN	– <i>Recurrent Neural Network</i> – rekurencyjne sieci neuronowe
ROC	– <i>Receiver operating Characteristic</i> – krzywa określająca jakość klasyfikatora
ROI	– <i>Regions of Interest</i> – obszary zainteresowania
SVM	– <i>Support Vector Machines</i> (z parametrem regulacji C)
TNR	– <i>True Negative Rate</i> – specyficzność
TPR	– <i>True Positive Rate</i> – wrażliwość, czułość

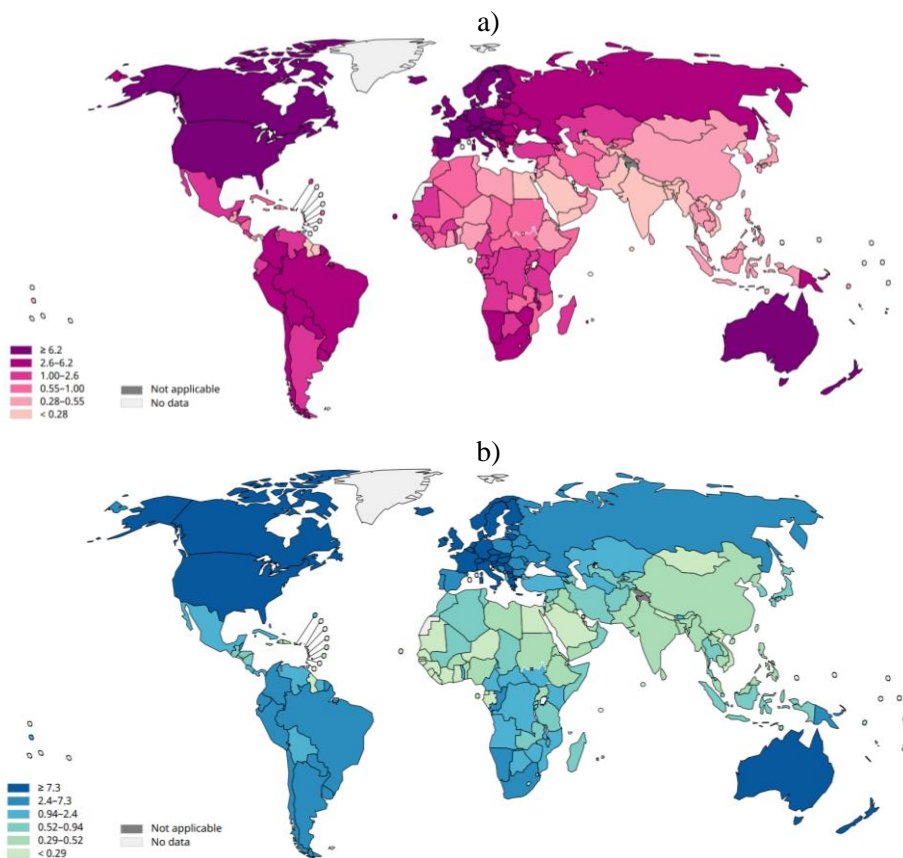
1. Wstęp

1.1. Uzasadnienie podjęcia tematyki pracy

Uczenie głębokie znajduje szerokie zastosowanie we wspomaganiu narzędzi diagnostycznych w dermatologii. Ponadto, jest skuteczne w rozróżnianiu czerniaka od łagodnych zmian skórnych na podstawie klinicznych obrazów dermatoskopowych. Sieci neuronowe są szeroko wykorzystywane w procesie klasyfikacji obrazów dermatoskopowych. Aby skrócić proces wstępnej diagnostyki, warto wykorzystać rozwijające się narzędzia diagnostyczne oparte o głębokie sieci neuronowe. Dzięki zastosowaniu specjalnych algorytmów i zaawansowanemu cyfrowemu przetwarzaniu otrzymanych wyników można uzyskać szybszą i bardziej skuteczną diagnozę. Natomiast rola lekarza specjalisty będzie się wówczas sprowadzała do jej kontroli i weryfikacji. Opisana w rozprawie metoda nie jest inwazyjna dla pacjenta. W niniejszej pracy autorka zdecydowała się skupić na jednostkach chorobowych takich jak, zmiany złośliwe skóry (rak kolczystokomórkowy, czerniak złośliwy, rak podstawnokomórkowy) oraz zmiany łagodne (znamię melanocytowe, łagodne rogowacenie barwnikowe, brodawka łojotokowa).

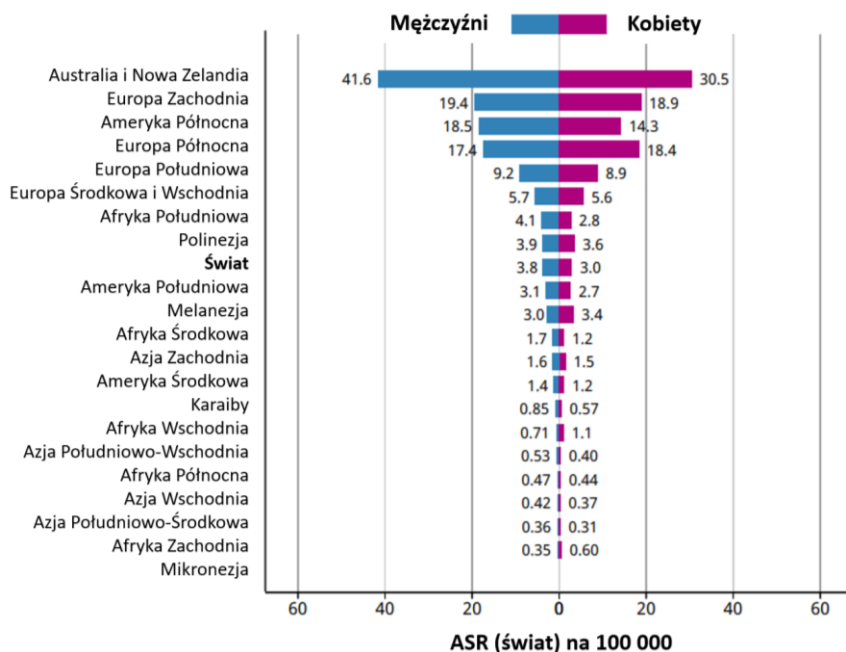
Czerniak skóry jest nowotworem o największej dynamice wzrostu, plasuje się na 17 miejscu pod względem liczby zachorowań na nowotwory na świecie według Światowej Organizacji Zdrowia (dane z 2020 r.). W roku 2020, liczba zachorowań na czerniaka na świecie sięgnęła 324 635, natomiast na czerniaka zmarło łącznie 57 043 osób [1]. Zgodnie z danymi Krajowego Rejestru Nowotworów (KRN) w 2016 r., w Polsce odnotowano 3 666 nowych przypadków zachorowań na czerniaka oraz 1 386 zgonów z jego powodu [2]. Na rysunku 1.1 przedstawiona została globalna zachorowalność na czerniaka dla kobiet i mężczyzn. Największa zachorowalność na czerniaka dla obu płci występuje w Australii, Nowej Zelandii, Ameryce Północnej, Południowej i Północnej Europie.

Na rysunku 1.2 przedstawiono zachorowalność na czerniaka skóry na świecie dla kobiet i mężczyzn w 2020 roku, wyrażoną wskaźnikiem ASR (*ang. Age – Standardised Rate*), w przeliczeniu na 100 000 osób. Według tych danych, zdecydowanie najczęściej na czerniaka chorują mężczyźni w Australii i Nowej Zelandii. W tym rejonie geograficznym, u większej liczby mężczyzn, niż kobiet diagnozowany był czerniak. Wysoka liczba zachorowań miała również miejsce w przypadku Zachodniej i Północnej Europy – drugie miejsce. Powyżej średniej światowej, która wynosi 3,8 dla mężczyzn, są również pozostałe regiony Europy, Południowa Afryka i Polinezja. Dane są określone jako ASR, wskaźnik standaryzowany pod względem wieku. Standaryzacja jest konieczna przy porównywaniu kilku populacji, wskaźnik ten jest również wyrażany na 100 000 mieszkańców. W regionach takich jak, Melanezja, południowo – wschodnia Azja, Mikronezja liczba zachorowań u kobiet była wyższa, niż u mężczyzn w 2020 roku.

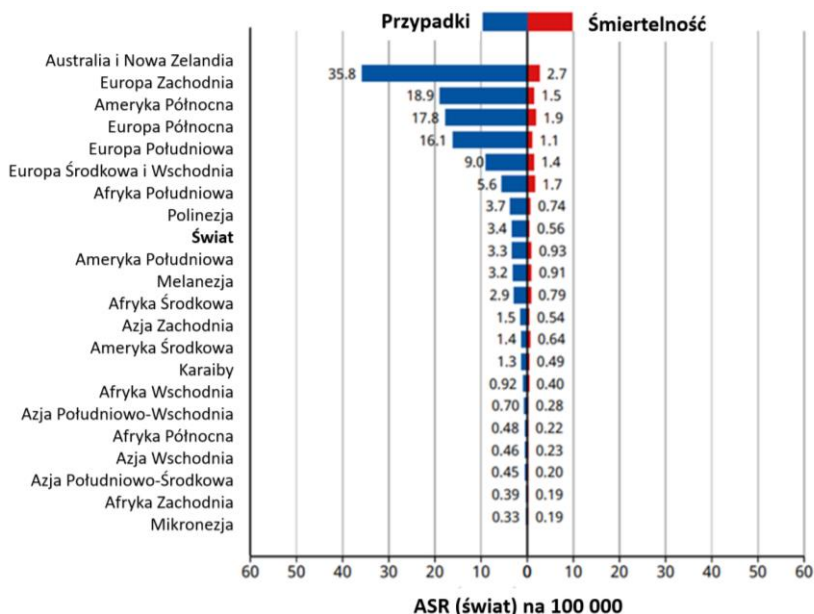


Rys. 1.1. Zachorowalność na czerniaka skóry na świecie w 2020 roku, w przeliczeniu na 100 000 mieszkańców dla: a) kobiet; b) mężczyzn [1]

Śmiertelność to liczba zgonów występujących w danym okresie, w określonej populacji. Może być ona wyrażona jako bezwzględna liczba zgonów rocznie lub jako wskaźnik na 100 000 mieszkańców rocznie, co zostało przedstawione na rysunku 1.3. Najwięcej osób umiera z powodu czerniaka w Australii i Nowej Zelandii. Kolejnym regionem jest Zachodnia Europa. Jednak występująca w tych rejonach świata duża różnica pomiędzy śmiertelnością, a średnią liczbą przypadków jest bardzo znaczna, w porównaniu do średniej światowej. Niepokojące są sytuacje, które mają miejsce w Południowej Afryce, Melanezji i Południowej Ameryce, gdzie współczynnik śmiertelności, w porównaniu do średniej liczby przypadków, jest wysoki.

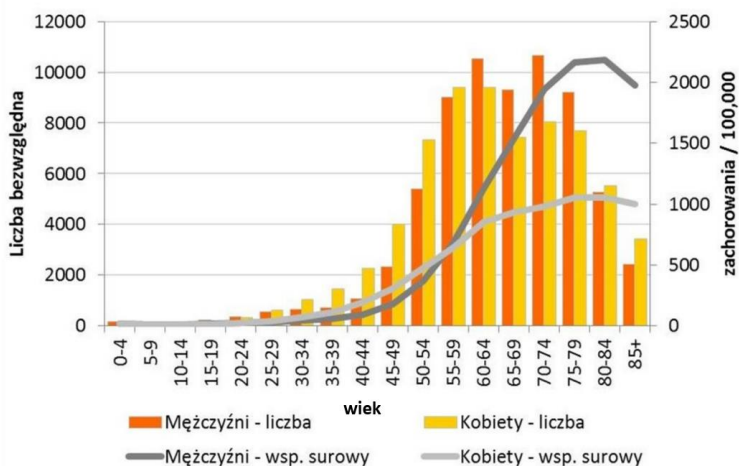


Rys. 1.2. Zachorowalność na czerniaka skóry na świecie dla kobiet i mężczyzn w przeliczeniu na 100 000 mieszkańców w 2020 roku [1]



Rys. 1.3. Śmiertelność w odniesieniu do średniej liczby zdiagnozowanych przypadków na czerniaka skóry na świecie w 2020 roku, w przeliczeniu na 100 000 mieszkańców [1]

Liczba zachorowań na czerniaka skóry wraz z wiekiem wzrasta u obu płci – współczynnik zachorowalności w ciągu trzydziestu lat niemal potroił się. Najwięcej zachorowań u obu płci notuje się między 50 a 64 rokiem życia. Poniżej 75 roku życia zachorowalność u kobiet jest wyższa, niż u mężczyzn. Rysunek 1.4 przedstawia liczbę zachorowań kobiet i mężczyzn w Polsce, w różnych grupach wiekowych.



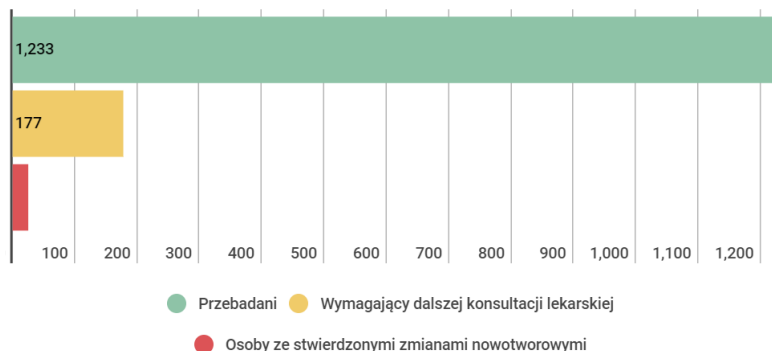
Rys. 1.4. Zachorowalność na czerniaka skóry w Polsce, w 2010 roku [3]

Kobiety i mężczyźni w każdej grupie wiekowej otrzymują coraz to więcej pozytywnych diagnoz zachorowalności na czerniaka skóry. Dla obu płci i we wszystkich grupach wiekowych od trzech dekad trwa wzrost zagrożenia czerniakiem skóry. W 2010 roku, w przedziale wiekowym 20–59 lat, zachorowalność na czerniaka wśród kobiet była wyższa, niż u mężczyzn.

Problem diagnostyki nowotworów skóry jest trudny, gdyż często bywa ignorowany przez pacjentów. Zbyt późno zgłaszają się oni do lekarza dermatologa w celu konsultacji zmiany skórnej. Pacjenci zgłaszają się do lekarza z już zaawansowanym stanem, wymagającym specjalistycznego leczenia onkologicznego. Niestety, część z tych przypadków jest trudna w leczeniu i w konsekwencji kończy się śmiercią pacjenta. Obserwowany jest stały wzrost zachorowalności i umieralności na czerniaka w Polsce. Odnotowano, iż liczba nowych przypadków na przestrzeni ostatnich 10 lat wzrosła ponad 70% [3, 4].

Wiele osób nie jest diagnozowanych pod kątem nowotworów skóry. Te nowotwory często są lekceważone przez pacjentów, którzy nie zgłaszają się do lekarza dermatologa ze zmianami skórnymi. Świadczą o tym wyniki badań przeprowadzonych przez NFZ w Polsce, opracowane po Dniu Otwartym poświęconym nowotworom skóry – rysunek 1.5. Skontrolowano ponad 1200 osób, z czego ponad 14% dostało zalecenie dalszej diagnostyki lekarskiej,

a u 2% osób lekarze już podczas pierwszej wizyty stwierdzili zmiany rakowe – czerniaki i inne nowotwory skóry.



Rys. 1.5. Wyniki badań dermatoskopowych poświęconych nowotworom skóry [5]

Wczesna diagnostyka to obszar zainteresowania naukowców zajmujących się, m.in., konwolucyjnymi sieciami neuronowymi. W ciągu ostatnich lat opracowano wiele prac dotyczących klasyfikacji obrazów dermatoskopowych. Zwiększenie zainteresowania procesami diagnostyki zmian skórnych przy użyciu sztucznej inteligencji spowodowało rozwój jej wykorzystania do diagnozy wielu chorób skóry. Powstały bardziej rozbudowane modele sieci pozwalające na klasyfikację kilku różnych chorób. Wielu autorów zajmuje się w ostatnim czasie problemem klasyfikacji zmian skórnych [6, 7, 8, 9, 10, 11, 12, 13, 14]. Tworzone są również publikacje zawierające przegląd dotychczas opracowanych metod klasyfikacji zmian skórnych na podstawie obrazów dermatoskopowych [15, 16].

Przeprowadzone przez autorkę badania dotyczące identyfikacji złośliwych zmian skórnych (rak kolczystokomórkowy, czerniak złośliwy, rak podstawnokomórkowy) oraz zmian łagodnych (znamię melanocytowe, łagodne rogowacenie barwnikowe, brodawka łojotokowa) wskazują na możliwość osiągnięcia wysokiej czułości klasyfikatora, na poziomie ponad 85%. Doniesienia literaturowe o sukcesach dotyczących zastosowania konwolucyjnych sieci neuronowych, skłoniły autorkę do rozpoczęcia badań z wykorzystaniem tych sieci do rozwiązania problemu diagnostyki zmian skórnych, czego wynikiem jest niniejsza rozprawa.

1.2. Cel, teza i zakres rozprawy

Celem niniejszej rozprawy jest wykazanie skuteczności głębokich sieci neuronowych, wytrenowanych na małym zbiorze obrazów dermatoskopowych, w rozpoznawaniu zmian skórnych.

Na podstawie przeprowadzonych badań, analiz i testów autorka postawiła następującą tezę:

„Zastosowanie głębokich sieci neuronowych oraz techniki *transfer learning* pozwoli zbudować na podstawie małego zbioru danych skuteczny model klasyfikacyjny do diagnostyki zmian skórnych”.

W kontekście uczenia od podstaw głębokich sieci neuronowych, liczebność obserwacji wykorzystana w badaniach była stosunkowo mała. W związku z tym, zastosowano technikę określaną często angielską nazwą *transfer learning*, polegającą na wykorzystaniu sieci, która została wcześniej wytrenowana na dużym zbiorze danych (zwykle trenowanie miało miejsce podczas wykonywania zadania klasyfikacji dużego zbioru obrazów). Takie podejście jest często stosowane w sytuacjach, gdy do dyspozycji są małe zbiory danych. Jeżeli zbiór danych użytych do wytrenowania modelu był wystarczająco duży i ogólny, przestrzenna hierarchia wyuczonych cech może skutecznie pełnić rolę ogólnego modelu do przetwarzania obrazów. Takie cechy mogą być pomocne przy rozwiązywaniu nowych problemów przetwarzania obrazów nawet wówczas, gdy problemy te dotyczą rozpoznawania innych klas, niż te oryginalne, użyte do wytrenowania modelu. W pracy udowodniono, iż wytrenowanie modelu na małym zbiorze danych (założono jego liczebność nie mniejszą, niż 400 obserwacji) pozwala na uzyskanie zadowalających wyników diagnostycznych.

Niniejsza rozprawa obejmuje część teoretyczną oraz część badawczą, przedstawiającą efekty przeprowadzonych eksperymentów, w wyniku których została udowodniona teza rozprawy. Na wstępie przedstawiono uzasadnienie podjęcia tematyki pracy oraz stan wiedzy w zakresie podobnych badań oraz wykorzystanych narzędzi. W części teoretycznej opisano przegląd popularnych w populacji zmian skórnych oraz opisano tradycyjne metody ich obrazowania. W dalszej części rozprawy opisano diagnostykę z wykorzystaniem metod uczenia maszynowego. Szczególną uwagę zwrócono na uczenie głębokie, w tym na sztuczne sieci neuronowe, głębokie splotowe sieci neuronowe oraz trenowanie i diagnostykę modeli. Kolejnym punktem jest opis zebranego materiału badawczego, zawierający charakterystykę grup pacjentów, parametry pozyskanych obrazów dermatoskopowych oraz sposób przygotowania materiału do badań i proces przetwarzania wstępnego obrazów. Opisano także metodykę badań oraz założenia i ograniczenia przyjęte w badaniach. W kolejnym rozdziale omówiono zastosowane w rozprawie splotowe sieci neuronowe oraz charakterystykę poszczególnych modeli. Część badawcza obejmuje opis wyników działania wykorzystanych modeli sieci konwolucyjnych oraz ich analizę poprzez porównanie podstawowych parametrów określających pracę sieci. Trenowanie modeli klasyfikacyjnych obejmuje klasyfikację binarną (ZNAMIĘ – CZERNIAK) i (ŁAGODNE – ZŁOŚLIWE) oraz klasyfikację wieloklasową. W ostatniej części rozprawy dokonano zestawienia oryginalnych osiągnięć oraz nakreślono kierunek dalszych badań. Całość została dopełniona o spis źródeł literaturowych oraz internetowych.

2. Obraz kliniczny zmian skórnych

Czerniak jest jednym z najszybciej rozwijających się na świecie nowotworów wśród osób o jasnej karnacji. Średni wzrost częstości występowania tej choroby u ludzi w ciągu ostatnich dekad wyniósł około 3%. Czerniak złośliwy jest jednym z najbardziej śmiertelnych nowotworów na świecie, ale jego wczesna diagnoza może prowadzić do skutecznego i ratującego życie leczenia.

Rozpoznanie chorób skóry w dermatologii można przeprowadzić na podstawie oględzin skóry, jest więc to proces w dużej mierze nieinwazyjny. Diagnoza lekarska wydawana jest na podstawie budowy anatomicznej, koloru i powierzchni zmiany. Jeśli badana zmiana wydaje się niebezpieczna podczas wstępnej oceny, zostaje zlecona jej biopsja, umożliwiając obrazowanie mikroskopowe. W ograniczonym zakresie można także poddać ocenie głębokość, wielkość zmiany skórnej oraz jej powierzchnię i cechy wewnętrzne. Wymaga to zastosowania obiektywnej i nieinwazyjnej metody oceny skóry, dlatego powstało wiele technik jej obrazowania.

Najczęściej stosowaną metodą oceny zmian skórnych jest dermatoskopia, która umożliwia ich wizualizację i ocenę. Trzy główne typy współczesnej dermatoskopii oparte są na wykorzystaniu światła spolaryzowanego, niespolaryzowanego, metody kontaktowej lub bezkontaktowej. Światło spolaryzowane pomaga uwidocznic głębokie struktury skóry, natomiast światło niespolaryzowane dostarcza informacji o strukturze powierzchni skóry. Na rysunku 2.1 przedstawiono współczesny dermatoskop. Nowoczesne dermatoskopy dają możliwość uzyskania kompletnych obrazów dzięki łączeniu tych dwóch trybów. Wysoka rozdzielczość zdjęć i odpowiednio duże powiększenie zmiany poprawia proces diagnostyki. Obraz otrzymywany jest dzięki silnemu źródłu światła i powiększającemu układowi optycznemu. Współczesne urządzenia posiadają także wyspecjalizowane oprogramowanie do obróbki zdjęć i wykonywania ich analiz. W gabinetach dermatologów coraz częściej pojawiają się wideodermatoskopy.

Przyczyny powstawania złośliwego czerniaka skóry nie są do końca poznane. Znaczny wpływ na ten proces ma promieniowanie UV, które pochodzi z promieni słońca lub solarium. UV prowadzi do mutacji w komórkach skóry. Często wykazywane są też predyspozycje genetyczne, dlatego lekarz podczas diagnostyki przeprowadza wywiad, czy w rodzinie chorego występowały zachorowania na czerniaka. Czerniaki często dotyczą ludzi o jasnej karnacji, z jasnymi oczami, włosami blond lub rudymi. Dotyczy to także ludzi posiadających piegi i skórę, która trudno się opala.



Rys. 2.1. Współczesny dermatoskop [17]

Skuteczne rozpoznawanie czerniaka nadal stanowi poważny problem, dlatego poszukiwane są nowe metody diagnostyczne. Obecnie, do oceny chorób skóry, wykorzystuje się: dermatoskopię, obrazowanie wielospektralne, ultrasonografię, optyczną tomografię koherencyjną, obrazowanie za pomocą rezonansu magnetycznego i konfokalną mikroskopię laserową. Są to techniki nieinwazyjne. Diagnostyka może być przeprowadzona bez pobierania próbek i bólu pacjenta. Ponadto, nie wymagają specjalnego przygotowania pacjenta do badania i dają wynik w krótkim okresie czasu.

2.1. Przegląd popularnych zmian skórnych

Znamiona barwnikowe, zwane również melanocytowymi, są rodzajem proliferacji melanocytów. Powstają już w życiu płodowym w skutek nieprawidłowego rozwoju elementów budujących skórę płodu. Znamiona dzieli się głównie na znamiona wrodzone i nabyte, natomiast te nabyte na łagodne i atypowe. W wieku dziecięcym i wczesnej młodości na skórze człowieka pojawiają się zmiany łagodne. W większości przypadków nie prowadzą one do rozwinięcia się nowotworu. Posiadają one regularną, symetryczną budowę, nie różnią się rozmieszczeniem melaniny w obrębie znamienia, z reguły mają kolor nieco ciemniejszy od naturalnego koloru skóry [18].

Inną, niebezpieczną grupę, stanowią znamiona dysplastyczne (atypowe), których barwa jest znacznie bardziej zróżnicowana, mają asymetryczny kształt, w większości przypadków przekraczający 5 mm. Brzegi tych zmian nie są regularne. Przyjmuje się je jako te, z których potencjalnie może rozwinąć się czerniak, dlatego wymagają one co kilka miesięcy obserwacji przez doświadczonego dermatologa, w celu diagnostyki. Zmiany skórne należy chronić przed słońcem (promieniowaniem UV). Dlatego należy stosować kremy z filtrem przeciwsłonecznym oraz nosić ubrania ochronne. Przyjmuje się, że jedno znamię na 30 000 zmian może ulec przekształceniu w postać nowotworową. Raki skóry mają pochodzenie nabłonkowe, należy do nich rak podstawnkomórkowy, kolczystokomórkowy i czerniak złośliwy. To właśnie czerniak jest najbardziej niebezpiecznym nowotworem skóry. Powstaje on na skutek szybkiego wzrostu komórek nazywanych melanocytami. Rak płaskonabłonkowy skóry jest drugą najczęstszą postacią raka skóry, charakteryzującą się nieprawidłowym, przyspieszonym wzrostem komórek płaskonabłonkowych. Kolejnym nowotworem jest rak podstawnkomórkowy, który najczęściej rozwija się na obszarach skóry wystawionych na działanie słońca. Na brązowej i ciemnej skórze często wygląda jak guzek, który jest brązowy, błyszczący i ma zwinięte brzegi. Najczęściej diagnozowane złośliwe formy zmian skórnych to raki skóry: płaskonabłonkowy (*squamous cell carcinoma*), czerniak (*melanoma*), podstawnkomórkowy (*basal cell carcinoma*). Wśród łagodnych zmian skórnych należy wymienić: rogowacenie łojotokowe (*seborrheic keratosis*), rogowacenie słoneczne (*actinic keratoses*), znamię (*nevus*).

Niezwykle istotne jest wykrycie nowotworu we wczesnym stadium. Im wcześniej zostanie wykryty czerniak, tym płycej będzie znajdował się w skórze. W tabeli 2.1 przedstawiono stopień rozwoju czerniaka wraz z jego krótką charakterystyką. W III i IV stopniu czerniak daje już przerzuty do innych tkanek i narządów, co jest niezwykle niebezpieczne. Wcześniejsza diagnoza czerniaka daje lepsze rokowania dla pacjenta. W Polsce ten nowotwór diagnozuje się każdego roku u ok. 3500 osób. Nawet wczesne stadium czerniaka, w momencie zgłoszenia się do dermatologa, nie oznacza sukcesu w diagnostyce, gdyż wizualna diagnostyka przez specjalistę osiąga skuteczność bliską 60%.

Na szczęście, wiele zmian skórnych ma łagodny przebieg. Należą do nich zmiany skórne, takie jak pieprzyk oraz przebarwienie skórne. Kolejnym niezłośliwym znamieniem jest rogowacenie słoneczne. To szorstka, łuszcząca się plama na skórze, która rozwija się po latach przebywania na słońcu. Jest przedinwazyjną postacią raka kolczystokomórkowego. Występowanie choroby jest ściśle związane z wiekiem pacjenta i wśród rasy kaukaskiej dotyczy 100% osobników powyżej 70. roku życia. Rogowacenie łojotokowe jest zwykle okrągłe lub owalne i ma kolor od jasnobrązowego do czarnego. Może rozwijać się pojedynczo lub w klastrach. Jest powszechnym nienowotworowym (łagodnym) znamieniem na skórze, a wraz z wiekiem człowieka pojawi się ich więcej na ciele.

Tabela 2.1. Stopnie zaawansowania klinicznego czerniaka [5, 19, 20, 21]

Stopień	Opis charakterystycznych cech
0	Postać nieprzekraczająca naskórka, nienaciekająca
I	Brak zajęcia węzłów chłonnych, brak przerzutów, guz z owrzodzeniem nie grubszym, niż 1mm lub bez owrzodzenia poniżej 2mm
II	Brak zajęcia węzłów chłonnych, brak przerzutów, wyróżnia się 3 podstopnie – ich cechy dominujące to grubość zmiany pierwotnej
III	Przerzuty do regionalnych węzłów chłonnych
IV	Przerzuty w odległych narządach

2.2. Tradycyjne metody diagnostyczne

Głównym narzędziem diagnostycznym w dermatologii jest dermatoskopia. Dzięki dermatoskopowi wykonywane są zdjęcia zmian skórnych o wysokiej rozdzielczości. Współczesne dermatoskopy wyposażone są w coraz to bardziej zaawansowany układ optyczny, specjalistyczne oprogramowanie i algorytmy przetwarzania obrazu. Powiększają one niewielkich rozmiarów obszar skóry ze zmianą, eliminują odbicia i rozproszenia światła od powierzchni skóry, pomagają lekarzowi wizualnie odróżnić złośliwe zmiany skórne od niezłośliwych. Dermatoskopia jest nieinwazyjną techniką obrazowania zapewniająca ukazanie wielu cech zmian skórnych. Obrazy skóry są poddawane analizie przez dermatologa. Jednak, gdy jest on niedoświadczony, nowotwór może nie zostać zdiagnozowany. Przyjmuje się, że dermatolodzy są w stanie ze skutecznością rzędu 75 – 85% prawidłowo zdiagnozować zmianę skórą na podstawie badania dermatoskopem [22]. Dalsza diagnostyka zmiany skórnej to jej biopsja i badanie histopatologiczne.

Przyjęto wiele metod poprawy skuteczności badania dermatoskopowego, pozwalających na ocenę przez dermatologa zmiany skórnej pod kątem wykrycia czerniaka skóry. W procesie diagnostyki, lekarze posługują się wybraną przez nich metodą (skalą). O zakwalifikowaniu zmiany jako złośliwej decyduje wartość wybrana ze skali. W zależności od wybranej cechy (np. barwa, tekstura, nieregularność zmiany) obliczana jest finalna wartość. We współczesnej diagnostyce chorób skóry sieci neuronowe znajdują szerokie zastosowanie. Powoli wypierają one klasyczne metody i analizę wzorca. Metoda analizy wzorca służy do oceny barwnikowych wykwitów skórnych. Naturalny kolor skóry (różny dla każdego człowieka) często stanowi punkt odniesienia w procesie analizowania kolorów zmiany. Wykwit barwnikowy zwykle składa się z więcej, niż jednego koloru, co może stanowić utrudnienie w rozróżnieniu barw przez ludzkie oczy.

Do pozostałych, stworzonych przez lekarzy skal, dzięki którym diagnozowane są zmiany skóry, należą: skala ABCD (czasem rozszerzana do postaci ABCDE), analiza tekstury [23], algorytm CASH, skala Huntera, skala Clarka i Breslowa, skala TNM, lista 3 punktów, lista 7 punktów, metoda


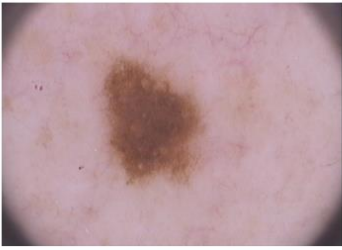


Menzies, Chaos – Clue. Skala ABCD(E) jest stosowana przez lekarzy w przypadku diagnozowania czerniaka. Na jej podstawie możliwe jest obliczenie wskaźnika TDS określającego, czy badana zmiana ma cechy czerniaka:

$$TDC = 1,3 \times A + 0,1 \times B + 0,5 \times C + 0,5 \times D, \quad (1)$$

gdzie: *A* to asymetryczny kształt względem każdej osi; *B* to brzegi nierówne i postrzępione; *C* – kolor różnorodny, *D* – średnica > 5 mm lub dynamika zmian morfologicznych w guzie; w rozszerzonej wersji – *E* oznacza uwypuklenie się zmiany ponad otaczający naskórek.

Rysunek 2.2 przedstawia założenia metody ABCD wykorzystywanej w diagnostyce zmian skórnych obrazów dermatoskopowych. W ich ocenie zastosowano wzór TDS i określone przedziały diagnostyczne. Przyznane punkty określają, czy zmiana jest łagodna, czy zagrożona wystąpieniem czerniaka.

W procesie klasycznej diagnostyki, wykorzystywane są również metody oparte na TDS, skali Glasgow, a także wiele innych. Skala Glasgow uznawana jest za zestaw objawów klinicznych czerniaka rozwiniętego miejscowo. W dermatoskopii, duże znaczenie ma obecność określonych kolorów i ich liczba. Skala Huntera daje wynik w zakresie od zera do trzynastu punktów. Objawy kliniczne sugerujące podejrzenie czerniaka są często pogrupowane w dwa systemy: skalę ABCD i siedmiopunktową skalę Glasgow. Chaos – Clue to prosta metoda szybkiej oceny podejrzenia zmian barwnikowych skóry za pomocą dermatoskopu. Algorytm CASH charakteryzuje porównywalną czułość przy nieco większej swoistości. Do wskaźników oceny skuteczności lekarzy zalicza się swoistość oraz czułość [24, 25]. Skale Clarka [25] oraz Breslowa [26] określają stopnie naciekania zmian w głąb skóry. W skali Breslowa istnieją cztery stopnie naciekania. Poszczególne stopnie odpowiadają głębokości naciekania wyrażonej w milimetrach [27]. Im wyższy stopień, tym rokowania są gorsze. W badaniu dermatoskopowym, poza analizą wzorca zmian barwnikowych, stosuje się algorytmy diagnostyki dermatoskopowej (reguła 7-punktowa, 3-punktowa) i zasadę Menzies.

Reguła	Obraz dermoskopowy	Cecha	Opis	Punkcja
A		Asymetria (kształt, kolor, struktury)	względem żadnej, jednej lub obu osi	0–2
B		Odgraniczenie	ostre lub nieostre w segmentach	0–8
C		Kolor	różne (biały, czerwony, jasno- i ciemnobrązowy, czarny, szaro-niebieski)	1–6
D		Struktury różnicujące	siatka, zatarcia struktury, kropki, ciała barwnikowe, wypustki promieniste lub pseudopodia	1–5

Całkowita punkcja podsumowująca poszczególne cechy (TDS):

A = asymetria (asymmetry): 0–2×1,3

B = brzegi (border): 0–8×0,1

C = barwy (colours): 1–6×0,5

D = struktury różnicujące (different): 1–5×0,5

Interpretacja wyniku TDS (Total Dermoscopy Score)

1,0–4,75: zmiana łagodna

4,8–5,45: zmiana podejrzana, wymaga kontroli dermoskopowej co 3 miesiące lub wycięć chirurgicznie w całości

5,46–8,9: zmiana wysoce podejrzana w kierunku czerniaka

Rys. 2.2. Przykładowa skala ABCD zdiagnozowanych zmian skórnych [18]

Analizowane cechy zależą od dostępnych metod diagnostyki, lecz nie wszystkie cechy obrazu są zauważalne przez człowieka. Metody fraktalne są narzędziem do pomiaru nieregularności występujących w ramach zmiany skórnej. Wymiar pudełkowy opisuje liczbowo struktury geometryczne, przydatny jest w klasyfikacji zmiany i przypisywaniu mu wybranych cech. Idea możliwego związku między rakiem a fraktalami została zasugerowana w wielu pracach.

2.3. Diagnostyka z wykorzystaniem metod uczenia maszynowego

W ciągu ostatniej dekady, dużą popularność zyskały metody automatycznej diagnostyki zmian skórnych na podstawie obrazów dermatoskopowych [28]. Powstanie metod automatycznych zawdzięczamy rozwojowi uczenia maszynowego, którego celem jest budowanie systemu zdolnego do automatycznego doskonalenia się i nabywania nowych umiejętności na podstawie dostarczonych informacji. Biorąc pod uwagę nadzór człowieka w procesie trenowania, uczenie maszynowe może być podzielone na uczenie nadzorowane, nienadzorowane oraz uczenie przez wzmacnianie. W uczeniu nadzorowanym, zbiór z danymi uczącymi zawiera rozwiązania danego problemu.

Jeśli złośliwa odmiana zmiany skórnej jest zdiagnozowana we wczesnym stadium, szanse na wyleczenie są większe. Opracowano dotychczas wiele technik wykrywania granic zmian na obrazach dermatoskopowych. Aby proces ten przebiegł prawidłowo, należy zwrócić uwagę na wiele aspektów. Jednym z nich jest wstępne przygotowanie obrazów dermatoskopowych. W celu – poddania obrazów procesowi segmentacji, należy je dobrze do tego przygotować. Niezbędne są techniki wstępnego przetwarzania obrazu [29]. Wyświetlane obrazy często zawierają artefakty, takie jak naczynia krwionośne, włosy, linijki z podziałką oraz oznaczenia naniesione przez lekarzy. Opracowano narzędzia usuwające automatycznie artefakty z obrazów dermatoskopowych, są to: filtry krawędziowe, algorytm *DullRazor* i inne. Artefakty na obrazach przeszkadzają w procesie identyfikacji zmiany skórnej. Wiele dostępnych baz dermatologicznych zawiera obrazy z artefaktami: znacznikami medycznymi, naniesioną skalą, ciemnymi włosami, pęcherzykami powietrza. Takie obrazy mają słabą jakość (brak w nich kontrastu lub są prześwietlone) lub obraz nie obejmuje całego obrysu zmiany.

Kolejnym etapem jest przygotowanie odpowiedniej bazy obrazów treningowych i testowych dla danej klasy. Obrazy są wcześniej prawidłowo zdiagnozowane przez lekarzy, w oparciu o biopsję. Przed etapem segmentacji, wyznacza się obszary zainteresowania (ROI – ang. *regions of interest*), które obejmują zmiany skórne. Do monitorowania granic zmian skórnych wykorzystywana jest komputerowa analiza obrazu. Segmentacja obrazu dzieli obrazy na regiony lub kategorie odpowiadające obiektom lub częściom obiektów. Powstało wiele prac porównujących efekty różnych rodzajów segmentacji. Obecnie, wykorzystuje się szereg automatycznych algorytmów

segmentacji krawędziowej, aktywne kontury, K-średnich, metodę Otsu, metodę rozmytych c-średnich, metodę "center split", metodę "split and merge" oraz progowanie adaptacyjne. Analizowana jest także zmienność koloru, cechy kolorystyczne czy teksturalne. Zwraca się szczególną uwagę na statystyki składowych koloru oraz macierz gradientu obrazu i przejść tonalnych.

Analiza i klasyfikacja obrazu może być przydatnym narzędziem do diagnozowania czerniaka. Stosowane są do tego celu algorytmy oparte o SVM [30, 31], regresję liniową, drzewa decyzyjne, klasyfikator KNN [32], AdaBoost i ich modyfikacje. Algorytmy te są testowane na wielu zestawach różnych obrazów dermatoskopowych, co pozwala na większą skuteczność w ocenie nowotworu. Naukowcy porównują czułość, specyficzność oraz inne miary jakości różnych klasyfikatorów, tak aby stworzyć najskuteczniejsze narzędzie diagnostyczne.

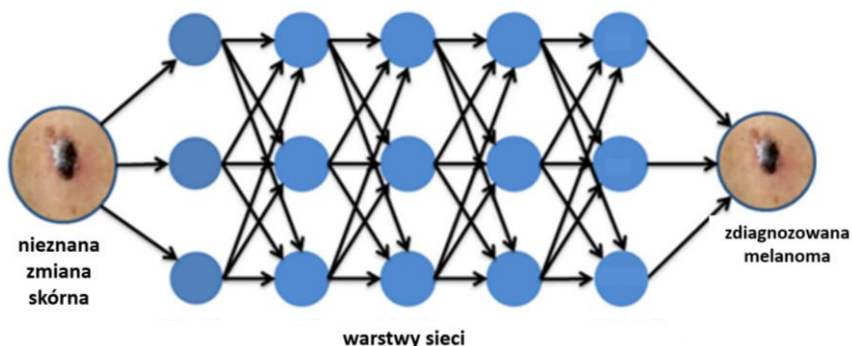
3. Uczenie głębokie

3.1. Uczenie głębokie w diagnostyce zmian skórnych

Techniki głębokiego uczenia (ang. *deep learning*, *DL*) [33] oraz inne techniki uczenia maszynowego [34] pomogły zwiększyć efektywność analizy obrazu. Wykorzystanie metod opartych o DL było możliwe dzięki zaimplementowaniu odpowiednich algorytmów. Zakładają one stworzenie modelu składającego się z sieci utworzonej z wielu warstw. Dzięki nim, dostarczone do modelu dane wejściowe, w postaci obrazów, podlegają wcześniej określonym procesom, które prowadzą do otrzymania danych wynikowych w postaci diagnozy. Uczenie głębokie jest obszarem uczenia maszynowego, w którym kolejne warstwy stworzonego modelu uczą się z każdym krokiem lepszych reprezentacji na podstawie dostarczonych informacji [35].

Diagnoza określa, które z obrazów należą do osób chorych, a które do zdrowych. Głębokie uczenie pozwoliło na zdiagnozowanie wielu chorób, dzięki zastosowaniu procesu segmentacji i klasyfikacji [36]. Statystyki wskazują na największe zainteresowanie wykorzystaniem sztucznych sieci neuronowych w procesie segmentacji [37]. Często, w tym celu wykorzystywane są CNN, a w ostatnich latach także DCNN, które stosowane są do segmentacji regionu zainteresowania, treningu sieci oraz do klasyfikacji obrazów dermatoskopowych [38, 39, 40]. Przegląd dostępnych metod wykorzystujących sieci opracowano, m.in., w [29, 41]. Proces głębokiego uczenia znalazł zastosowanie w wykrywaniu patologicznych zmian w strukturach anatomicznych, segmentacji tkanek, klasyfikacji obrazów medycznych oraz komputerowo wspomaganej wczesnej diagnostyce różnego rodzaju chorób [42, 43, 44].

Głębokie sieci neuronowe składają się z wielu warstw, każda z nich identyfikuje bardziej złożone elementy obrazu wejściowego. Na rysunku 3.1 przedstawiono przykład struktury, która analizuje obrazy dermatoskopowe. Wejściowy obraz zmiany skórnej może być klasyfikowany, jako zmiana łagodna lub złośliwa. Warstwy sieci są ze sobą w określony sposób połączone, a także są im nadane odpowiednie wagi. Każdy węzeł (niebieskie koło) w danej warstwie jest połączony z każdym węzłem w kolejnej warstwie. Sieć spłotowa w rzeczywistości jest znacznie bardziej złożoną strukturą, niż przedstawiona na rysunku 3.1. Liczba warstw w tym modelu sieci wynosi od kilkudziesięciu do kilkuset, co znacznie odróżnia je od modeli uczenia tradycyjnego. Kolejne warstwy sieci przekształcają zbiór danych wejściowych na efekt końcowy – zbiór danych wyjściowych. Szereg następujących po sobie operacji jest parametryzowany przez wagi każdej z warstw modelu. Istnieje wiele różnych metod uczenia sieci, m.in., algorytm wstecznej propagacji błędów.



Rys. 3.1. Schemat budowy sieci głębokiej [43]

Wśród architektur głębokich sieci neuronowych wymienia się perceptrony wielowarstwowe, ograniczone maszyny Boltzmanna, głębokie sieci przekonań, splotowe sieci neuronowe, rekurencyjne sieci neuronowe. W ostatnich latach najlepsze efekty w obszarze rozpoznawania obrazów osiąga się dzięki konwolucyjnym sieciom neuronowym [45]. Miały one szanse na rozwój dzięki nowym technikom skutecznego uczenia głębokich sieci i zwiększeniu wydajności obliczeniowej.

Uczenie głębokie jest także skuteczne w rozróżnianiu czerniaka od łagodnych zmian skórnych na podstawie klinicznych obrazów dermatoskopowych [39, 46]. Dynamiczny rozwój technik głębokiego uczenia umożliwia ich zastosowanie w nowych obszarach, do których należy, m.in., identyfikacja zmian skórnych [47, 48, 37]. Przykładem mogą być wyniki badań zaprezentowane w pozycji [44], gdzie otrzymane rezultaty były porównywalne z wynikami otrzymanymi dzięki tradycyjnym badaniom dermatologicznym.

Realizowane są również projekty międzynarodowe, których celem jest zebranie różnorodnej bazy danych, a następnie znalezienie skutecznej metody klasyfikacji zmian skórnych. Takie projekty wzbudziły duże zainteresowanie wśród naukowców. W większości prac liczba obrazów treningowych dochodzi do kilkuset, a nawet kilku tysięcy. Obrazy te były już wcześniej zdiagnozowane przez lekarzy specjalistów. Ich diagnozy zostały potwierdzone badaniami histopatologicznymi. Zebranie tak dużej bazy danych nie jest łatwe, wymaga współpracy naukowców i dermatologów.

W ramach badań prezentowanych w literaturze, najczęściej przeprowadzana jest klasyfikacja wieloklasowa oraz binarna. Klasyfikacja binarna obejmuje tylko dwie klasy: czerniaka kontra łagodne zmiany lub niełośliwe zmiany skórne bez pigmentu [49]. Rozbudowane klasyfikacje wieloklasowe pozwalają na ocenę wielu schorzeń skóry. Głównie diagnozowaną chorobą skórną, powodującą powstawanie zmian skórnych jest czerniak skóry. Diagnozowany jest on najczęściej wraz z łagodnymi zmianami skórnymi [49, 50, 51]. W klasyfikacji binarnej, sieci neuronowe trenowane są na dwóch zestawach

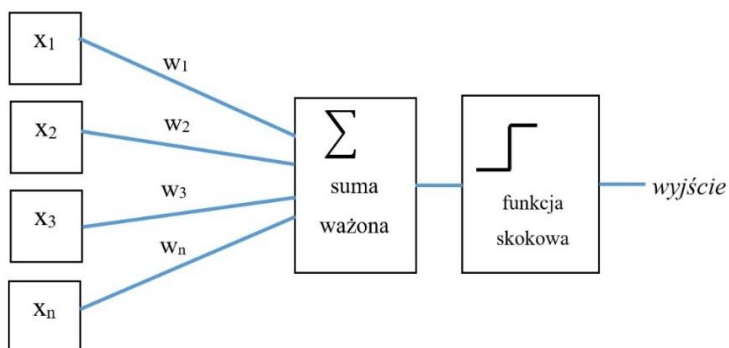
danych, pierwszy zawiera obrazy należące do osób zdrowych, a drugi do pacjentów z rozpoznaniem choroby.

Zwiększenie zainteresowania procesami diagnostyki zmian skórnych przy użyciu sztucznej inteligencji spowodowało rozwój wielu narzędzi. Powstały bardziej rozbudowane modele sieci, pozwalające na klasyfikację wielu różnych schorzeń. Klasyfikacja wieloklasowa pomaga przede wszystkim w diagnostyce włókniaka skóry, czerniaka, zmian naczyniowych, łagodnego rogowacenia, [52, 53, 54, 55].

Liczba prac naukowych dotyczących zastosowania konwolucyjnych sieci neuronowych oraz głębokiego uczenia z wykorzystaniem obrazów medycznych wciąż wzrasta. Opublikowane w 2021 i 2022 roku artykuły dotyczące zastosowania sieci konwolucyjnych przekonują, że możliwe jest osiągnięcie wysokiej efektywności klasyfikatorów. Dostępne metody głębokiego uczenia umożliwiły otrzymanie bardzo dobrych wyników w diagnostyce wielu chorób dzięki nowym algorytmom. Przedmiotem prezentowanej rozprawy jest analiza obrazów dermatoskopowych przeprowadzona za pomocą różnych modeli konwolucyjnych sieci neuronowych.

3.2. Sztuczne sieci neuronowe

Zaproponowany w 1957 roku perceptron jest najprostszą z architektur sztucznych sieci neuronowych. Składa się on wyłącznie z warstw wejścia i wyjścia. Mogą się one uczyć wyłącznie wzorców, które można rozdzielić liniowo. Perceptron to jednowarstwowa jednostka liniowa, która nazywana jest neuronem. Dostarczaniem danych wejściowych zajmują się neurony wejściowe. W perceptronie występuje również neuron obciążeniowy. Na rysunku 3.2. przedstawiony jest perceptron. Należy go oczywiście wytrenować, podając dane na wejścia $x_1, x_2, x_3, \dots, x_n$ oraz modyfikując wagi wejść $w_1, w_2, w_3, \dots, w_n$ i połączeń między warstwami neuronów. Istotne są również, obliczane przez perceptron, suma ważona i funkcja skokowa. Czynności te mają dać na wyjściu wynik *wyjście* z pożądanymi wartościami.



Rys. 3.2. Perceptron złożony z jednego neuronu

Na podstawie różnych danych wejściowych neuronu określamy wartość aktywacji według wzoru:

$$a(x) = \sum_i w_i x_i, \quad (2)$$

gdzie: x_i – wartość wejścia neuronu, w_i – wartość połączenia pomiędzy neuronem i a wyjściem. Perceptrony mają wiele wspólnego z algorytmami regresji logistycznej i są ograniczone przez klasyfikatory liniowe. Następnie, po opisanej sumie, realizowana jest funkcja skokowa:

$$h_w(x) = \text{skok}(a(x)), \quad (3)$$

Najczęściej wykorzystywanymi narzędziami do realizacji funkcji skokowych są funkcje Heaviside'a i signum. Warto również wspomnieć o regule Heba. Najczęściej odmiana reguły Heba jest używana do uczenia perceptronów. Nadaje ona wagę połączeniu pomiędzy dwoma neuronami, która rośnie, gdy mają ten sam stan na wyjściu. Zwracany błędny wynik dla każdego neuronu wyjściowego powoduje zwiększanie się wag połączeń ze wszystkimi wejściami. Otrzymane wyniki powodują pojawienie się prognozy prawidłowej. Zasada polegająca na uwzględnianiu w uczeniu popełnianego błędu sieci przedstawia się następująco:

$$w_{i,j}^{(\text{następny krok})} = w_{i,j} + \eta(y_j - y_{jj})x_i, \quad (4)$$

gdzie: $w_{i,j}$ – waga połączenia pomiędzy i -tym neuronem wejściowym i j -tym neuronem wyjściowym; x_i – i -ta wartość wejściowa próbki uczącej; y_{jj} – wynik j -ego neuronu wyjściowego dla próbki uczącej; y_j – docelowy wynik j -tego neuronu wyjściowego dla przykładu uczącego; η – współczynnik uczenia.

Znacznie więcej możliwości daje jednak perceptron wielowarstwowy (MLP). W jego skład wchodzi jedna warstwa wejściowa, co najmniej jedna warstwa ukryta (LTU) oraz warstwa wyjściowa (LTU). Każda w warstw, poza wyjściową zawiera neuron obciążeniowy i jest połączona z warstwą następną. Głęboką siecią neuronową nazywamy sieć zawierającą w swojej strukturze co najmniej dwie warstwy ukryte. Uczenie perceptronu wielowarstwowego jest możliwe dzięki algorytmowi propagacji wstecznej. W ostatnim czasie najbardziej popularne stały się funkcje logistycznego (sigmoidalnego) tangensa hiperbolicznego oraz funkcja ReLU.

Wielowarstwowe perceptrony wykorzystywane są w klasyfikacji binarnej, jak i wieloklasowej. W klasyfikacji binarnej, każdy z przypadków rozpatrywany jest jako zdrowy lub chory, gdzie każde wyjście odpowiada właśnie jednej z dwóch klas. Funkcją aktywacji w warstwie wyjściowej jest wówczas funkcja

logistyczna. W tym przypadku, obliczane jest prawdopodobieństwo \hat{p} przynależności próbki x do klasy pozytywnej:

$$\hat{p} = h_{\theta}(x) = \sigma(\theta^T \cdot x), \quad (5)$$

gdzie: θ^T – transponowany wektor parametrów modelu, zawierający wagę obciążenia θ_0 oraz wagi cech $\theta_1, \theta_2, \dots, \theta_n$; h_{θ} – funkcja hipotezy wykorzystująca parametry θ modelu.

Klasyfikacja wieloklasowa zakłada przynależność analizowanego przypadku do trzech lub większej liczby klas, które wzajemnie się wykluczają. Pojedyncze funkcje aktywacji są zastąpione w tym przypadku funkcją *Softmax*. Dla danej próbki x obliczony zostaje wynik funkcji $s_k(x)$ dla każdej z klas k :

$$s_k(x) = (\theta^k)^T \cdot x, \quad (6)$$

gdzie θ^k jest własnym, wyspecjalizowanym wektorem parametrów klasy k . Wynik każdej klasy dla próbki x , można oszacować poprzez obliczenie prawdopodobieństwa \hat{p}_k przynależności próbki do klasy k :

$$\hat{p}_k = \sigma(s(x))_k = \frac{e^{s_k(x)}}{\sum_{j=1}^K e^{s_j(x)}}, \quad (7)$$

gdzie: K – liczba klas; $s(x)$ – wektor zawierający wyniki klasy dla próbki x ; $\sigma(s(x))_k$ – szacowane prawdopodobieństwo przynależności próbki x do klasy k przy znanych wynikach wszystkich klas.

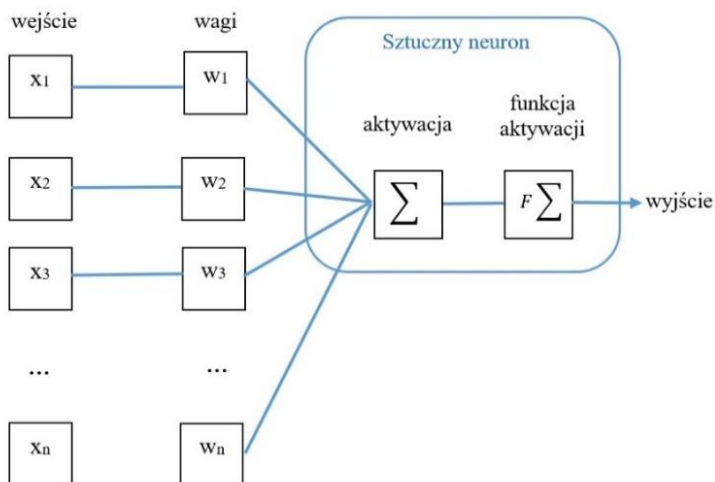
Wynik każdego neuronu odpowiada oszacowanemu prawdopodobieństwu przynależności do danej klasy. Następnie, obliczona zostaje prognoza klasy o najwyższym wyniku:

$$\hat{y} = \arg_k \max \sigma(s(x))_k = \arg_k \max s_k(x) = \arg_k \max ((\theta^{(k)})^T \cdot x), \quad (8)$$

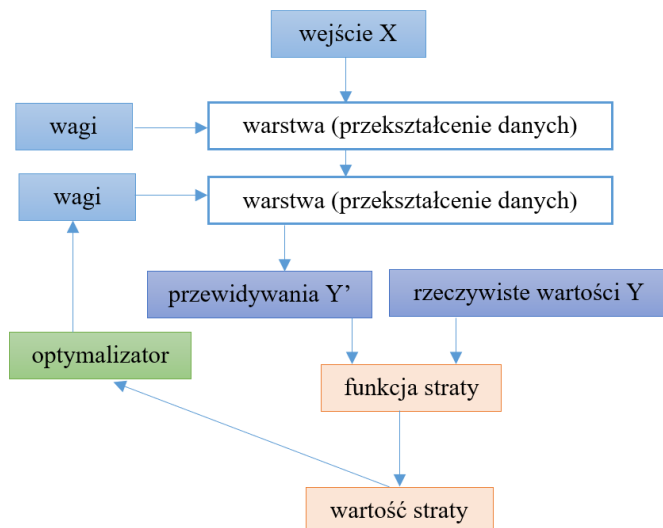
gdzie: *argmax* – operator zwracający wartość zmiennej maksymalizującej funkcję. W równaniu (8) zwraca on wartość parametru k maksymalizującą szacowane prawdopodobieństwo $\sigma(s(x))_k$. Uczenie jest procesem przeprowadzanym z wykorzystaniem wielu różnych technik, a jego głównym celem jest znalezienie optymalnych wag dla poszczególnych warstw neuronów.

Architektura sieci neuronowej definiuje sposób połączenia neuronów, liczbę neuronów w każdej warstwie oraz liczbę warstw. Połączone ze sobą warstwy tworzące sieć przypisują przewidywane wartości wyjściowe do danych wejściowych. Warstwa jest podstawową jednostką budującą sieć. Przyjmuje ona dane wejściowe z tensorów i generuje dane wyjściowe w postaci tensorów. Warstwy są określane przez wagi $w_1 \dots w_n$, których wartości można ustalić. Do

sieci wprowadzane są dane wejściowe $x_1 \dots x_n$ i odpowiadające im docelowe etykiety. Budowa sztucznego neuronu została przedstawiona na rysunku 3.3. Sztuczny neuron odbiera sygnał wejściowy (aktywację), generuje wewnętrzny stan neuronu poprzez funkcję aktywacji. W kolejnym kroku, wygenerowane dane są przekazywane dalej do kolejnych neuronów. Ich działanie jest kontrolowane za pomocą wag.



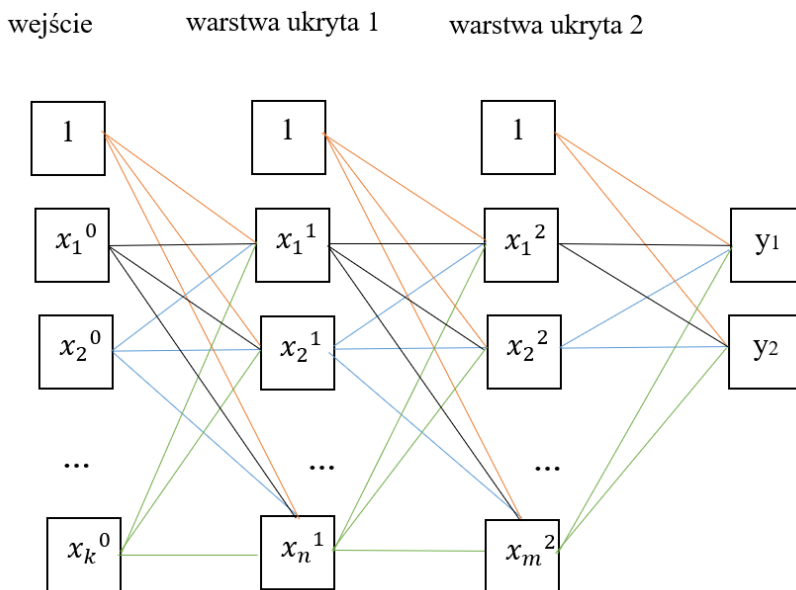
Rys. 3.3. Schemat budowy sztucznego neuronu



Rys. 3.4. Zależności między siecią, jej warstwami, funkcją straty i optymalizatorem

W funkcjonowaniu sieci neuronowych istotna jest funkcja straty, która definiuje sygnał zwrotny używany w procesie uczenia. To ona porównuje wyniki przewidywań sieci z docelowymi etykietami i na ich podstawie oblicza wartość straty. Optymalizator korzysta z wartości funkcji straty podczas modyfikowania wag sieci [56], określając następny przebieg trenowania. Rysunek 3.4 przedstawia zależności między siecią, jej warstwami, funkcją straty i optymalizatorem.

Stałym elementem sieci neuronowej jest funkcja straty, która jest wartością minimalizowaną w procesie trenowania. Trenowanie sieci to proces, który poszukuje zestawu odpowiednich wartości do obsługi procesu przetwarzania tensora. Funkcja straty jest modyfikowana przy użyciu optymalizatora, który implementuje jeden z wariantów algorytmu spadku wzdłuż gradientu [57]. Rysunek 3.5 przedstawia schemat trójwarstwowej sieci neuronowej z dwiema warstwami ukrytymi. Z lewej strony znajdują się neurony tworzące wejście, dalej znajdują się wagi. Neurony po lewej stronie reprezentują *wejście*, środkowa kolumna przedstawia wagi dla każdego połączenia warstw ukrytych. Neurony wyjściowe reprezentują dwie klasy: y_1 i y_2 . Przedstawiona architektura ma k neuronów wejściowych. Pierwsza warstwa ukryta ma n ukrytych neuronów, natomiast druga zawiera m ukrytych neuronów. W rzeczywistej sieci może występować znacznie więcej warstw ukrytych.



Rys. 3.5. Schemat trójwarstwowej sieci neuronowej z dwiema warstwami ukrytymi

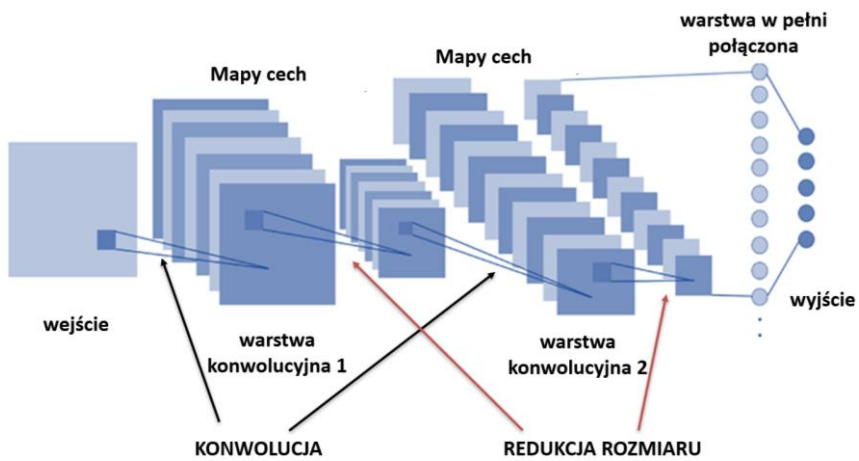
W działaniu sieci neuronowej bardzo istotnym elementem jest funkcja aktywacji, która generuje sygnał wyjściowy neuronu. Sygnał ten jest następnie

propagowany do neuronów w kolejnej warstwie sieci. Funkcja aktywacji przybiera różne postacie, m.in.: funkcji tożsamości, funkcji logistycznej, funkcji progowej aktywacji, czy funkcji tangensa hiperbolicznego. W odpowiedniku neuronu biologicznego występuje połączenie funkcji tożsamości z funkcją progową. Takie połączenie nazywane jest funkcją ReLU [57]. W zależności od problemu, który jest do rozwiązania wybierana jest odpowiednia funkcja aktywacji. Również, do różnych zadań dedykowane są różne architektury głębokiego uczenia [46]. Jako zaawansowane topologie sieci głębokiego uczenia należy wymienić, m.in.: sieci o dwóch rozgałęzieniach, sieci typu multihead, czy bloki inercji [37].

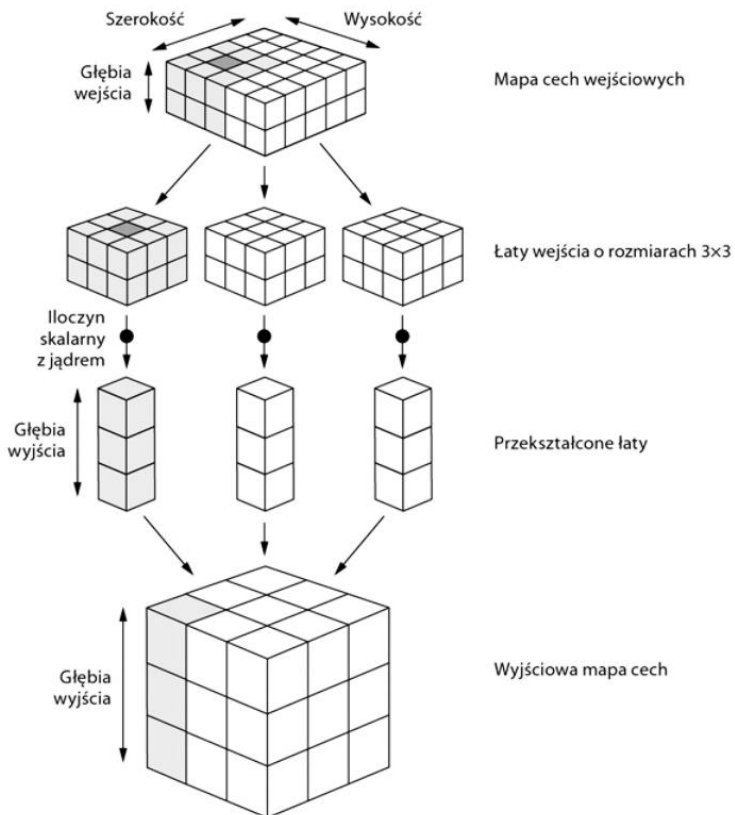
3.3. Głębokie splotowe sieci neuronowe

Rysunek 3.6 przedstawia typową architekturę sieci splotowej, która jest wykorzystywana do klasyfikacji obrazów. Podstawą takiej sieci jest baza splotowa, która składa się z warstwy wejściowej i kilku warstw splotowych połączonych ze sobą warstwami łączącymi (*pooling*). Neurony danej warstwy połączone są z neuronami kolejnej warstwy poprzez określoną funkcję aktywacji (np. *ReLU*) [46]. Neurony z pierwszej warstwy splotowej łączone są z pikselami znajdującymi się w polach recepcyjnych obrazu wejściowego. Natomiast neurony z drugiej warstwy splotowej łączą się z niewielkim obszarem pola recepcyjnego pierwszej warstwy. W polach recepcyjnych, np. o wymiarach 3×3 lub 5×5 , znajdują się lokalne wzorce, które odnajduje sieć [58]. Dane znajdujące się w polu recepcyjnym wcześniejszej warstwy poddawane są operacji konwolucji i tworzą następną warstwę. W każdej kolejnej warstwie sieć koncentruje się na większej liczbie szczegółów charakteryzujących daną cechę. Dzięki tej operacji powstają mapy cech, których rozmiar jest redukowany (mniejsza liczba parametrów) za pomocą warstw łączących (*pooling*). Zadaniem warstw konwolucyjnych jest znalezienie zbioru cech charakterystycznych obrazu wejściowego, umożliwiającego jego klasyfikację przy użyciu zbudowanego modelu. Model z pewnym prawdopodobieństwem określa przynależność obrazu wejściowego do określonej klasy [58].

Na rysunku 3.7 przedstawiono schemat operacji konwolucji. Zakłada ona przesuwanie okien (pól recepcyjnych) o określonych wymiarach (np. 3×3 lub 5×5) po mapie cech. Każde z okien wybiera pewną łąkę i zawarte w niej cechy. Łąka ulega przekształceniu dzięki iloczynowi tensorowemu z wyuczoną macierzą wag (filtrem, jądrem konwolucji) [60]. Tensor utworzony z wykorzystaniem tego samego filtru tworzy mapę cech, którą można opisać poprzez wysokość i szerokość. Pewna liczba map cech tworzy warstwę konwolucyjną. Z kolei, filtr to tensor trójwymiarowy (wolumin), który można opisać poprzez: szerokość, wysokość i głębokość.



Rys. 3.6. Typowa architektura sieci konwolucyjnej [59]



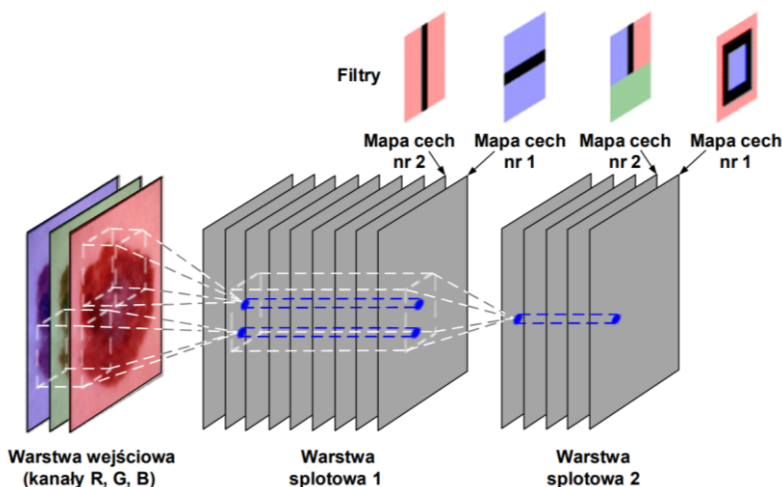
Rys. 3.7. Schemat działania konwolucji [56]

Operacja splotu realizuje filtrację liniową. Po tej operacji, intensywność piksela obrazu na wyjściu staje się liniową kombinacją pikseli otaczających go w obrazie wejściowym:

$$G(i, j) = H(i, j) ** f(i, j), \quad (9)$$

gdzie: $G(i, j)$ – obraz wyjściowy, $H(i, j)$ – obraz źródłowy, $f(i, j)$ – funkcja maski (filtru). Operator $**$ oznacza dwuwymiarowy splot (konwolucję).

Na rysunku 3.8 przedstawiona została rzeczywista budowa sieci splotowej. Warstwa splotowa zbudowana jest z wielu map cech, a każda mapa korzysta z tego samego filtra. Zbiór map cech tworzy trójwymiarową strukturę warstwy konwolucyjnej. Obecność różnych map cech pozwala wykrywać wiele charakterystycznych cech znajdujących się w różnych miejscach na obrazie wejściowym. Wynika to z faktu, że każda mapa korzysta z innego filtra, wyczulonego na specyficzne cechy obrazu. Rolą mapy cech jest odnalezienie na obrazie wejściowym wszystkich elementów najbardziej podobnych do filtra. Wykorzystanie dużej liczby map zapewnia wykrycie wielu charakterystycznych cech. Proces uczenia sieci splotowej zapewnia dobieranie właściwych wag.



Rys. 3.8. Rzeczywista budowa sieci splotowej [46]

Podczas treningu sieci stosowane są różne techniki wspierające jakość tego procesu. Jedną z nich jest ograniczenie pojemności sieci. Mniejsza pojemność oznacza mniejszą liczbę parametrów modelu, a to oznacza mniejsze prawdopodobieństwo przetrenowania sieci. Inne sposoby eliminowania nadmiernego dopasowania to regularyzacja L1, L2 oraz Max-norm [61]. Często, w ramach przeciwdziałania przetrenowaniu, stosowane jest także porzucanie (ang. *dropout*).

Podczas klasyfikacji binarnej, w ostatniej warstwie sieci, jako funkcja aktywacji, stosowana jest funkcja logistyczna (*sigmoid*). W tym przypadku,

zadaniem neuronu warstwy wyjściowej jest oszacowanie prawdopodobieństwa przynależności próbki do klasy pozytywnej. Podczas klasyfikacji wieloklasowej, ostatnia warstwa zawiera liczbę neuronów równą liczbie wszystkich klas obserwacji. W tym przypadku, jako funkcja aktywacji stosowana jest funkcja *softmax*, która szacuje prawdopodobieństwo przynależności próbki wejściowej do każdej z klas. Do najczęściej wykorzystywanych metod optymalizacji należą: optymalizacja momentum, AdaGrad, RMSProp, algorytm Adam (ang. *adaptive moment estimation*, szacowanie adaptacyjnego momentu) [61]. Wybór odpowiedniej metody optymalizacji jest jednym z czynników wpływających na jakość zbudowanego modelu.

3.4. Trenowanie i diagnostyka modeli

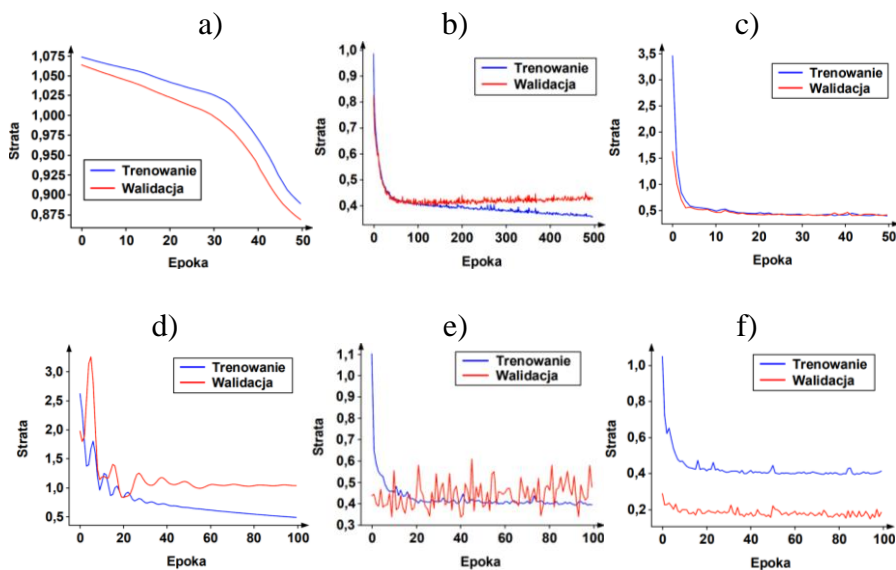
Do walki z nadmiernym dopasowaniem stosowane są metody nazywane metodami regulacji. Należą do nich, m.in.: zgromadzenie większego zbioru uczącego, redukcja pojemności modelu i wczesne zatrzymanie. Inne metody polegają na dodaniu do funkcji straty kosztu związanego z dużymi wartościami wag [61]. Wyróżniane są tutaj:

- regularyzacja L1 – dodany koszt jest proporcjonalny do bezwzględnej wartości współczynników wag;
- regularyzacja L2 – dodany koszt jest proporcjonalny do kwadratu wartości współczynników;
- regularyzacja max-norm – dla każdego z neuronów, wagi połączeń wejściowych w są ograniczone tak, że $\|w\|_2 \leq r$, gdzie r to hiperparametr metody max-norm, a $\|\cdot\|_2$ jest członem regularyzacji L2.

Do metod regularyzacji należy także porzucanie [62, 63], które zakłada losowe wybranie pewnej liczby cech wyjściowych warstwy podczas uczenia i zastąpieniu tych wartości zerami. Kolejną techniką jest dogenerowanie danych. Celem jest tutaj otrzymanie większej liczby elementów należących do zbioru uczącego w wyniku losowych przekształceń obrazów wejściowych. Dogenerowanie danych zwraca obrazy, które wyglądają wiarygodnie i nie powtarzają się.

Całemu procesowi uczenia sieci towarzyszy diagnostyka modelu na podstawie krzywych uczenia i walidacji. Na podstawie przebiegu wspomnianych krzywych można stwierdzić, czy w danym modelu mamy do czynienia z niedotrenowaniem, przetrenowaniem czy dopasowaniem. Kiedy model nie jest w stanie uzyskać niskiej wartości błędu dla zbioru treningowego, mówimy o niedotrenowaniu (rysunek 3.9a). Model nie jest wówczas w stanie nauczyć się zespołu cech znajdujących się w zbiorze treningowym lub nie ma określonej pojemności w stosunku do złożoności danych. Dodatkowo może dojść do opadania krzywej strat trenowania. Jeśli wartości strat spadają pod koniec wykresu, należy dalej trenować model i poprawić jego wydajność. Odwrotną sytuacją jest przetrenowanie sieci (rysunek 3.9b), kiedy to model jest za bardzo dopasowany do danych uczących, szumu statystycznego i fluktuacji losowych. Wzrasta wówczas błąd generalizacji bazujący na wydajności dla zbioru

walidacyjnego. Model ma wówczas większą pojemność, niż wymaga tego rozwiązywany problem. W takim przypadku, krzywa straty trenowania wykazuje tendencję opadającą, podczas gdy krzywa straty walidacji spada, a następnie zaczyna wzrastać. Stanem, do którego się dąży jest dopasowanie (rysunek 3.9c). Następuje ono, gdy krzywe straty trenowania i walidacji wykazują tendencję opadającą, następnie ich przebieg jest ustabilizowany, a na końcu występuje mała różnica pomiędzy krzywymi.



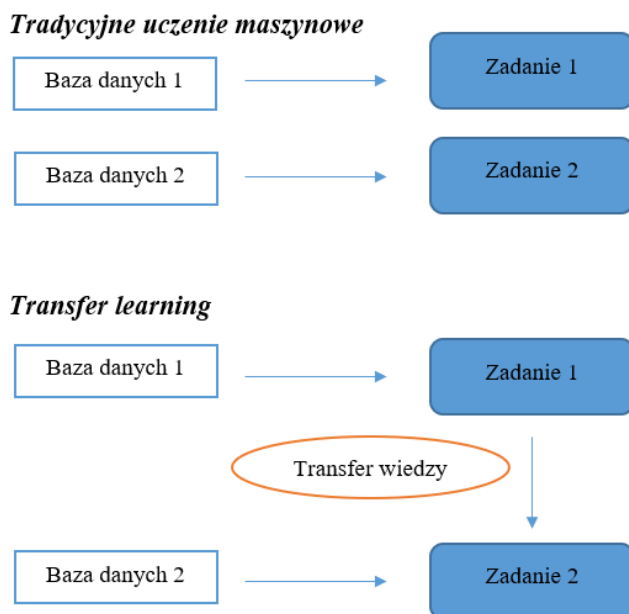
Rys. 3.9. Przykładowe wyniki procesu trenowania sieci: a) niedotrenowanie (model wymaga dalszego trenowania); b) przetrenowanie; c) dopasowanie modelu; d) zbiór treningowy jest niereprezentatywny (zbiór treningowy jest zbyt mały, w porównaniu ze zbiorem walidacyjnym); e) zbiór walidacyjny jest niereprezentatywny (zbiór walidacyjny jest zbyt mały, w porównaniu ze zbiorem treningowym); f) zbiór walidacyjny jest niereprezentatywny (zbiór walidacyjny jest łatwiejszy do predykcji, niż zbiór treningowy) [46]

W procesie trenowania sieci może wystąpić również sytuacja niereprezentatywnych zbiorów danych. Niereprezentatywność dotyczy zbioru treningowego, gdy ten nie dostarcza do modelu odpowiedniej liczby danych o potrzebnych cechach (krzywe wykazują tendencję opadającą z dużym odstępem między nimi). Natomiast, gdy niereprezentatywny jest zbiór walidacyjny, krzywa straty walidacji oscyluje wokół krzywej straty trenowania (rysunek 3.9e). Zdarzają się również sytuacje, gdy zbiór walidacyjny jest łatwiejszy do predykcji, niż zbiór treningowy (rysunek 3.9f).

3.5. Technika *transfer learning*

Modele oparte o głębokie sieci neuronowe mogą osiągnąć wysoką dokładność, jeżeli podczas uczenia korzystają z dostatecznie dużej ilości danych (tysiące obserwacji). Jednak zgromadzenie takich danych w wyniku obrazowania medycznego nie zawsze jest możliwe. Najczęściej, bazy obrazów dermatoskopowych nie są bardzo liczne, dlatego liczebność zgromadzonych obrazów dla zbioru treningowego może okazać się niewystarczająca do osiągnięcia zadowalających wyników. W procesie uczenia głębokich sieci neuronowych bardzo często wykorzystywana jest technika określana jako *transfer learning* [64]. Technika ta pozwala uczyć się nowych danych na podstawie wcześniej zdobytej już wiedzy, która może zostać wykorzystana do rozwiązywania innych, podobnych problemów.

Podczas tradycyjnego uczenia maszynowego, każde nowe zadanie jest rozwiązywane typowo przez jeden model bazujący na przewidzianym dla niego zbiorze danych. Jeśli rozwiązać należy dwa problemy klasyfikacji (korzystające z różnych zbiorów danych), należy użyć dwóch różnych modeli. Natomiast korzystając z techniki *transfer learning* po przeszkoleniu klasyfikatora z pierwszego zadania, można skorzystać ze zdobytej już wiedzy i zbudować klasyfikator do rozwiązania drugiego zadania [65]. Na rysunku 3.10 przedstawiono porównanie tradycyjnego uczenia maszynowego i techniki *transfer learning*.



Rys. 3.10. Porównanie tradycyjnego uczenia maszynowego i transfer learning

Przy stosowaniu techniki *transfer learning* odłączamy od modelu źródłowego oryginalny klasyfikator znajdujący się na jego końcu. Pozostała część modelu, odpowiedzialna za ekstrakcję cech, nosi nazwę bazy konwolucyjnej. Następnie, na koniec bazy konwolucyjnej dołączamy własny klasyfikator. Taka struktura jest trenowana w dwóch etapach. W etapie pierwszym (ekstrakcja cech) cała baza konwolucyjna jest zamrożona, a trenowaniu podlega tylko nowy klasyfikator dołączony na jej koniec. Etap drugi (dostrajanie) polega na tym, że odmrażamy pewną liczbę końcowych warstw bazy konwolucyjnej i trenujemy je razem z własnym klasyfikatorem.

To jak dużo warstw bazy konwolucyjnej trenujemy, a jak dużo zamrażamy, zależy od wielkości nowego zbioru treningowego oraz jego podobieństwa do zbioru, na którym model został wstępnie wytrenowany. W związku z tym, *transfer learning* zakłada cztery warianty (rysunek 3.11).

W I ćwiartce na rysunku 3.11 mamy do czynienia z dużym zbiorem nowych danych treningowych, ale istotnie różniących się od danych użytych do wstępnego wytrenowania modelu. Z uwagi na posiadanie dużego zestawu danych, nowy model może być trenowany od podstaw (wszystkie warstwy podlegają trenowaniu).

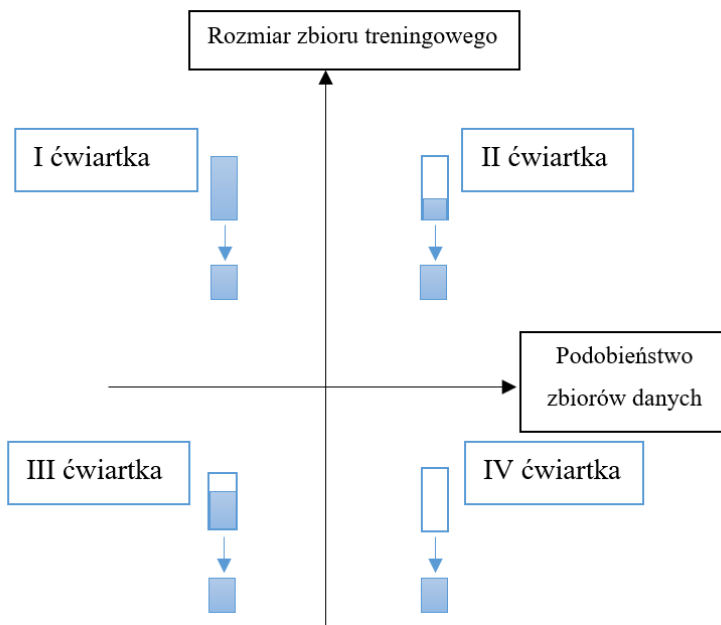
W II ćwiartce z rysunku 3.11, podobieństwo elementów nowego zbioru treningowego do elementów zbioru użytego do wstępnego wytrenowania jest większe, niż w I ćwiartce. W praktyce oznacza to, że zadaniem nowego modelu jest rozpoznawanie identycznych (lub bardzo podobnych) obrazów, niż te, które zostały wykorzystane do wytrenowania modelu źródłowego. W tym przypadku, ponownie trenuje się tylko kilka końcowych warstw bazy konwolucyjnej, tworzących najczęściej ostatni blok. Natomiast, wcześniejsze warstwy zostają zamrożone, to znaczy nie są poddane procesowi trenowania. Pamiętajmy, że początkowe warstwy bazy konwolucyjnej odpowiadają za ekstrakcję ogólnych cech obiektów, podczas gdy dalsze wyłaniają ich specyficzne cechy.

W przypadku, gdy występuje mała liczebność nowego zbioru treningowego oraz jego małe podobieństwo do zbioru użytego do wstępnego wytrenowania, procesowi uczenia poddawana jest większa liczba końcowych warstw bazy konwolucyjnej (np. 2 ostatnie bloki). Innymi słowy, procesowi ponownego trenowania należy podać większą liczbę warstw, niż w przypadku ćwiartki II z rysunku 3.11. Właśnie z taką sytuacją mamy do czynienia w niniejszej pracy.

Ostatnia, IV ćwiartka opisuje sytuację, w której liczba elementów nowego zbioru treningowego jest mała, ale stopień jego podobieństwa do zbioru użytego do wstępnego wytrenowania jest duży. W tym przypadku, cała baza konwolucyjna może być zamrożona, a budowa modelu może być ograniczona tylko do wytrenowania własnego klasyfikatora.

Podsumowując, technika *transfer learning* pozwala uzyskać następujące korzyści:

- użycie wstępnie wytrenowanego modelu pozwala najczęściej uzyskać lepsze wyniki, w porównaniu z trenowaniem modelu od początku;
- skrócenie czasu potrzebnego na budowę modelu;
- możliwość wykorzystania małych baz danych.



Legenda:

- warstwy zamrożone
- warstwy podlegające trenowaniu

Im większe wypełnienie prostokąta, tym więcej warstw podlega trenowaniu

Duży prostokąt – wstępnie wytrenowany model (baza konwolucyjna)

Mały prostokąt – własny klasyfikator

Rys. 3.11. Rodzaje techniki *transfer learning* zależne od wielkości nowego zbioru treningowego oraz jego podobieństwa do zbioru, na którym model został wstępnie wytrenowany

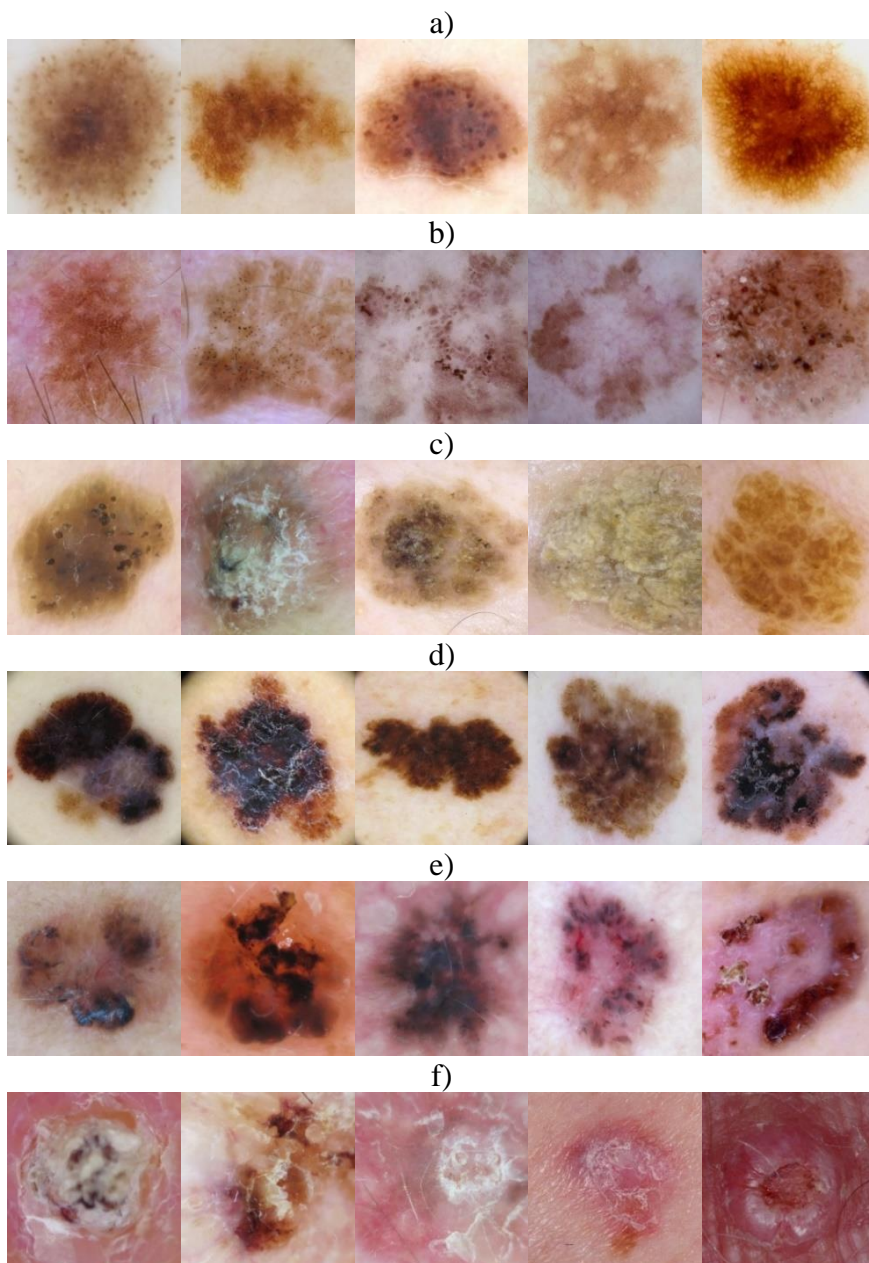
4. Materiał i metodyka badań

4.1. Materiał do badań

Dostępnych jest wiele baz danych zawierających obrazy dermatoskopowe. Wśród nich, najbardziej rozbudowaną i największą jest baza ISIC [66]. Zawiera ona ponad 20 000 obrazów wcześniej zdiagnozowanych zmian skórnych. Znaną bazą jest także PH² [67]. Z kolei, baza MED-NODE [68] zawiera 100 obrazów czerniaka i 70 zmian skórnych. Na rysunku 4.1 przedstawiono przykładowe zmiany pochodzące z bazy danych obrazowych ISIC. W ramach pracy zbudowano modele klasyfikacji binarnej i wieloklasowej z wykorzystaniem sześciu klas. Trzy z nich obejmowały obrazy łagodnych zmian skórnych: znamię melanocytowe, łagodne rogowacenie barwnikowe, brodawka łojotokowa. Kolejne trzy dotyczyły zmian zdiagnozowanych jako złośliwe, należały do nich: czerniak złośliwy, rak podstawnokomórkowy, rak kolczystokomórkowy. W tabeli 4.1 zamieszczono podział zmian na sześć klas z uwzględnieniem nazw chorób również w języku angielskim. Skrótowa nazwa klasy obserwacji będzie używana dalszej części pracy.

Tabela 4.1. Rodzaje zmian skórnych ze zbioru ISIC wykorzystane w badaniach

Typ zmiany	Nazwa zmiany	Klasa obserwacji
Łagodna	Znamię melanocytowe (<i>Nevus</i>)	ZNA
	Łagodne rogowacenie barwnikowe (<i>Pigmented benign keratosis</i>)	ROG
	Brodawka łojotokowa (<i>Seborrheic keratosis</i>)	BRO
Złośliwa	Czerniak złośliwy (<i>Malignant melanoma</i>)	CZE
	Rak podstawnokomórkowy (<i>Basal cell carcinoma</i>)	POD
	Rak kolczystokomórkowy (<i>Squamous cell carcinoma</i>)	KOL

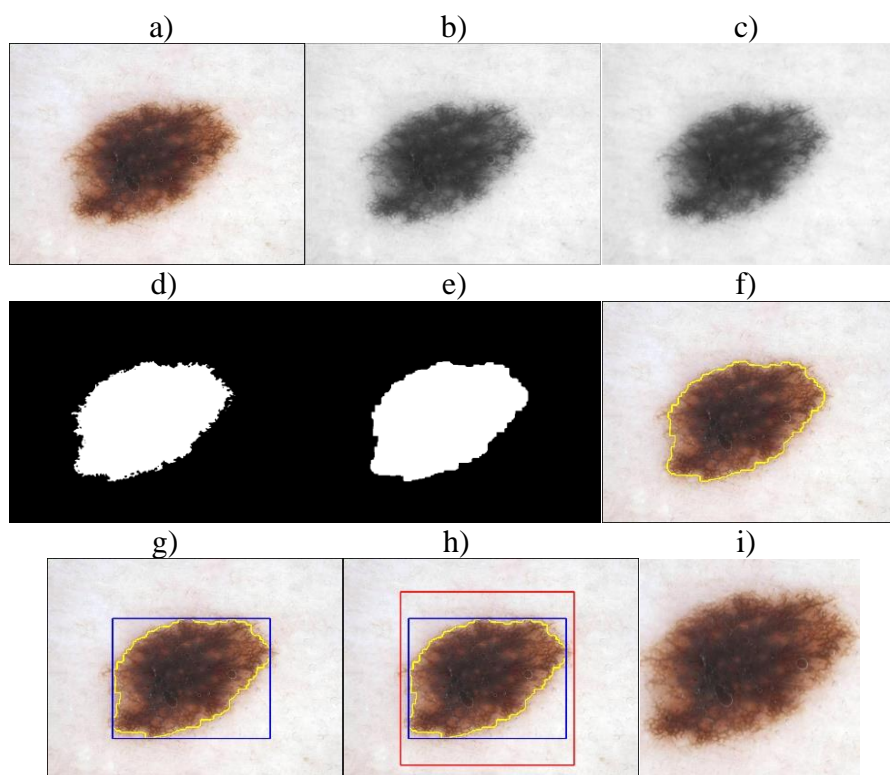


Rys. 4.1. Przykładowe zmiany skórne ze zbioru ISIC wykorzystane w badaniach: a) znamię melanocytowe; b) łagodne rogowacenie barwnikowe; c) brodawka łojotokowa; d) czerniak złośliwy; e) rak podstawnkomórkowy; f) rak kolczystokomórkowy

4.2. Wstępne przetwarzanie obrazów

Obrazy zmian skórnych mogą mieć różną jakość i zawierać wiele artefaktów, dlatego segmentacja zmiany nie jest łatwym procesem. Najczęściej, popularne metody segmentacji (np. progowanie) stają się niewystarczające. Zanim przeprowadzono segmentację, wszystkie obrazy zostały przeskalowane w taki sposób, że największy wymiar (szerokość lub wysokość) miała nie więcej, niż 1024 piksele. Niżej scharakteryzowano główne etapy wstępnego przetwarzania obrazów, natomiast pośrednie wyniki przetwarzania przedstawiono na rysunku 4.2. Załącznik A zawiera listing kodu w języku Python, który zastosowano do realizacji opisanego niżej procesu segmentacji.

1. Wczytanie obrazu źródłowego z 24-bitową głębią kolorów (rysunek 4.2a).
2. Konwersja obrazu do 8-bitowej skali szarości (rysunek 4.2b).
3. Filtracja medianowa z maską o rozmiarze 5x5 pikseli (rysunek 4.2c).
4. Progowanie obrazu z wykorzystaniem metody Otsu (rysunek 4.2d).
5. Operacje morfologiczne (zamknięcie, otwarcie) z wykorzystaniem ośmiu iteracji oraz elementu strukturalnego o rozmiarze 3x3 piksele (rysunek 4.2e).
6. Wykrycie wszystkich konturów na obrazie.
7. Znalezienie konturu o największej powierzchni. W tym przypadku, wykorzystano obserwację polegającą na tym, że statystycznie, spośród wszystkich konturów występujących na obrazie, ten należący do obiektu zainteresowania ma największą powierzchnię.
8. Znalezienie prostokąta ograniczającego kontur największego obiektu na obrazie.
9. Znalezienie kwadratu zawierającego prostokąt ograniczający kontur największego obiektu na obrazie. Podczas wyznaczania wielkości kwadratu zastosowano warunek polegający na tym, że długość boku kwadratu ma być o 5% większa od największego rozmiaru prostokąta ograniczającego kontur zmiany skórnej. Celem tego warunku było pozostawienie na ROI pewnego marginesu skóry wokół zmiany.
10. Skopiowanie z obrazu źródłowego ROI wyznaczonego przez znaleziony kwadrat pod warunkiem, że kwadrat całkowicie zawierał się w obrazie źródłowym. Jeśli warunek ten nie był spełniony, kopiowanie ROI nie było wykonywane.
11. Przeskalowanie ROI do rozmiaru 224x224 piksele.



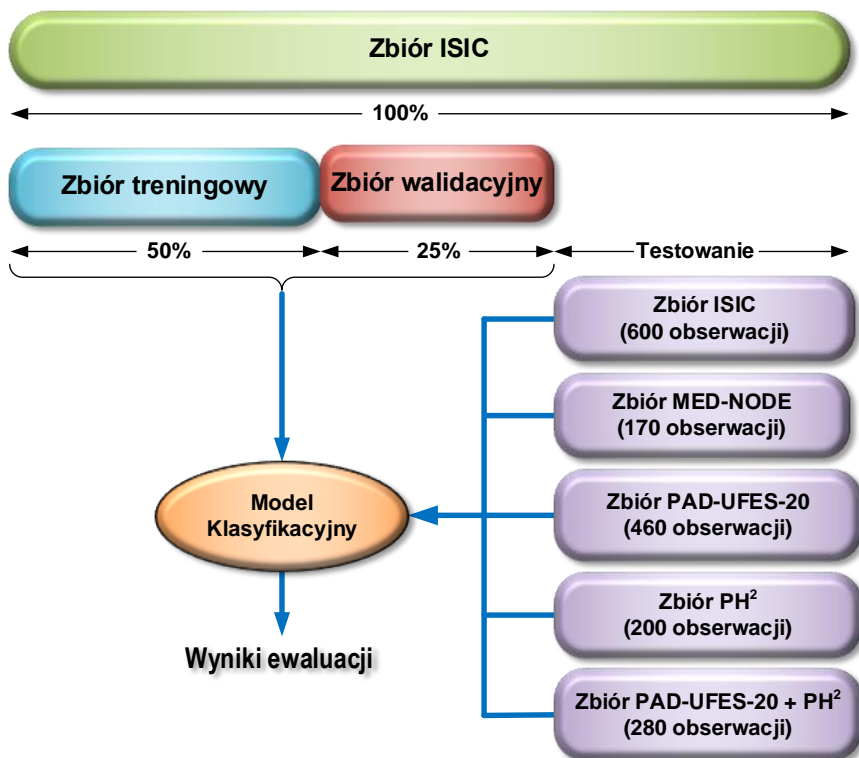
Rys. 4.2. Przykładowe wyniki najważniejszych etapów wstępnego przetwarzania obrazu: a) obraz źródłowy (1024x768 pikseli); b) obraz po konwersji do 8-bitowej skali szarości; c) obraz po filtracji medianowej z maską o rozmiarze 5x5 pikseli; d) obraz po progowaniu obrazu z wykorzystaniem metody Otsu; e) obraz po operacjach morfologicznych (zamknięcie, otwarcie) z wykorzystaniem ośmiu iteracji oraz elementu strukturalnego o rozmiarze 3x3 piksele; f) obraz z zaznaczonym konturem obiektu o największej powierzchni (żółta linia); g) obraz z prostokątem ograniczającym kontur największego obiektu na obrazie (niebieska linia); h) obraz z kwadratem zawierającym prostokąt ograniczający kontur największego obiektu na obrazie (czerwona linia); i) obszar zainteresowania skopiowany z obrazu źródłowego i przeskalowany do rozmiaru 224x224 piksele

4.3. Metodyka przeprowadzania badań

Zakres wykonanych badań w pracy przedstawiono na rysunku 4.3 Diagnostyka zmian skórnych poprowadzona została w trzech kierunkach:

- odróżnienie czerniaków złośliwych skóry od niezłośliwych znamion melanocytowych;
- odróżnienie niezłośliwych zmian skórnych od zmian złośliwych;
- rozróżnienie pomiędzy sobą wybranych zmian skórnych.

Modele klasyfikacyjne zbudowano na podstawie zbioru ISIC, który przygotowano tak, że 50% pełnego zbioru stanowiło część treningową, 25% walidacyjną oraz 25% testową. Obserwacje przypisywane były do poszczególnych podzbiorów w sposób losowy. Testowanie modeli klasyfikacyjnych przeprowadzono z wykorzystaniem pięciu zbiorów danych: ISIC [66], MED-NODE [68], PAD-UFES-20 [69], PH² [67], PAD-UFES-20 + PH² [67, 69].



Rys. 4.3. Zakres badań

4.4. Założenia i ograniczenia przyjęte w badaniach

Podczas badań przyjęto następujące założenia i ograniczenia:

1. Pod pojęciem „mały” zbiór danych rozumiany jest zbiór liczący nie mniej, niż 400 obserwacji. W związku z tym, do budowy modeli wykorzystano te klasy zmian skórnych, których liczebność w zbiorze ISIC była równa co najmniej 400. Wcześniejsze doświadczenie autorki oraz wyniki innych badań świadczą, iż przyjęta minimalna liczebność obserwacji jest wystarczająca do efektywnego wytrenowania modeli sieci konwolucyjnych z wykorzystaniem techniki *transfer learning*.

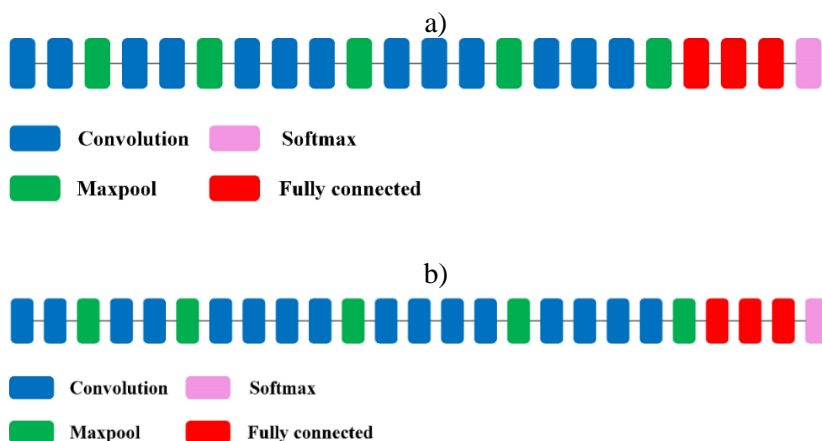
2. W badaniach zastosowano obrazy bez artefaktów, takich jak: włosy, pęcherzyki, linijki i znaczki naniesione przez dermatologa. Opracowano dotychczas wiele skutecznych algorytmów służących do usuwania artefaktów z obrazów medycznych, np. *DullRazor*, a projektowanie nowych nie wchodziło w zakres pracy.
3. Jako obszar zainteresowania przyjęto kwadrat o wielkości 224×224 piksele, przedstawiający całą zmianę wraz z wąskim marginesem skóry wokół niej. Kształt obrazu wynikał z wymagań zastosowanych modeli, a jego rozmiar należał do standardowych rozmiarów obrazów wejściowych, które mogą być przetwarzane przez zastosowane sieci.
4. Podczas badań wykorzystano bibliotekę Keras ze względu na jej dużą popularność, obszerną dokumentację, łatwość implementacji z wykorzystaniem języka Python oraz obszerny zbiór różnych architektur sieci konwolucyjnych do przetwarzania obrazów.
5. Z biblioteki Keras wybrano do badań 10 sieci konwolucyjnych, które różniły się między sobą architekturą, głębokością topologiczną oraz liczbą parametrów.
6. Strojenie modeli klasyfikacyjnych ograniczono do określenia optymalnej liczby epok trenowania na podstawie wykresów krzywych uczenia i walidacji. Zakres badań nie obejmował poszukiwania optymalnej liczby warstw splotowych podlegających dostrajaniu – zawsze odpowiadała ona dwóm ostatnim blokom bazy splotowej.

5. Sieci spłotowe wykorzystane w badaniach

W rozdziale 5 omówiono architektury konwolucyjnych sieci neuronowych, które wykorzystano w badaniach przedstawionych w rozprawie. Wiele terminów związanych z architekturą sieci konwolucyjnych nie ma swoich odpowiedników w języku polskim, dla części z nich zastosowano oryginalną nomenklaturę w języku angielskim.

5.1. VGG16 i VGG19

Sieć VGG charakteryzuje się najmniejszą głębokością topologiczną z małym filtrem spłotowym 3×3 . Wśród sieci VGG największy sukces odniosły dwie z nich – VGG16 i VGG19. VGG16 (rysunek 5.1a) składa się z 13 warstw spłotowych i trzech warstw w pełni połączonych. Natomiast VGG19 (rysunek 5.1b) ma 16 warstw spłotowych i trzy w pełni połączone warstwy. Obie sieci używają niewielkich filtrów spłotowych o wymiarze 3×3 . Zwiększa to głębokość topologiczną sieci i przyczynia się do efektywniejszego procesu uczenia [6].



Znaczenie terminów:

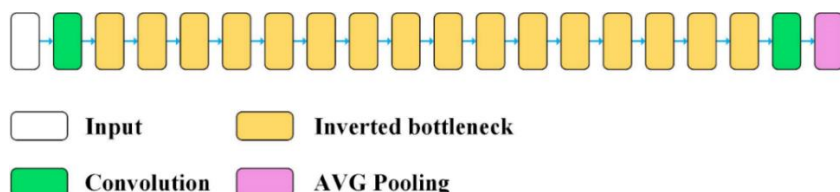
convolution – warstwa konwolucyjna; *fully connected* – warstwy w pełni połączone; *softmax* – funkcja aktywacji; *maxpool* – maksymalizująca warstwa łącząca.

Rys. 5.1. Schemat budowy modeli sieci: a) VGG16 oraz b) VGG19 [70]

5.2. MobileNetV2

Sieć MobileNetV2 składa się z 53 warstw spłotowych i jednej warstwy łączącej, wykorzystuje ona wydajne operacje konwolucyjne. Warstwy w pełni połączone są w tej sieci zastępowane uśredniającą warstwą łączącą.

Stosunkowo mała liczba parametrów pozwala na uzyskanie mniej rozbudowanych i szybciej uczących się modeli. Ponadto, filtry o różnych rozmiarach są używane na tej samej warstwie, zapewniając bardziej szczegółowe informacje i wyodrębniając wzorce o różnych rozmiarach. W celu zmniejszenia złożoności obliczeniowej i liczby parametrów, stosowana jest warstwa splotowa 1×1 . Zalety tej sieci ujawniają się podczas trenowania od podstaw małych modeli na ograniczonym zbiorze danych. Schemat architektury tej sieci został przedstawiony na rysunku 5.2.



Znaczenie terminów:

input – wejście; *convolution* – warstwa konwolucyjna; *inverted bottleneck* – odwrócone wąskie gardło; *AVG Pooling* – uśredniająca warstwa łącząca.

Rys. 5.2. Uproszczonego schematu budowy modelu MobileNetV2 [70]

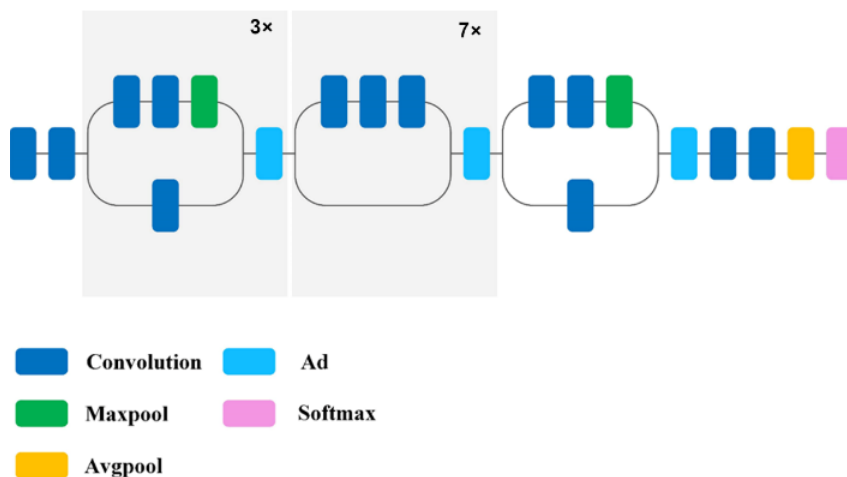
Input	Operator	t	c	n	s
$224^2 \times 3$	conv2d	-	32	1	2
$112^2 \times 32$	bottleneck	1	16	1	1
$112^2 \times 16$	bottleneck	6	24	2	2
$56^2 \times 24$	bottleneck	6	32	3	2
$28^2 \times 32$	bottleneck	6	64	4	2
$14^2 \times 64$	bottleneck	6	96	3	1
$14^2 \times 96$	bottleneck	6	160	3	2
$7^2 \times 160$	bottleneck	6	320	1	1
$7^2 \times 320$	conv2d 1x1	-	1280	1	1
$7^2 \times 1280$	avgpool 7x7	-	-	1	-
$1 \times 1 \times 1280$	conv2d 1x1	-	k	-	-

Rys. 5.3. Szczegółowa architektura modelu sieci MobileNetV2 [71]

Warto przyjrzeć się szczegółowemu modelowi architektury omawianej sieci, który został przedstawiony na rysunku 5.3. W poszczególnych wierszach zawarta jest informacja o kolejności powtarzających się n – razy warstw i liczbie kanałów wyjściowych – c . Każda z pierwszych warstw występująca w danej sekwencji posiada krok – s , natomiast pozostałe w sekwencji mają krok równy 1. Należy zwrócić uwagę, że wszystkie warstwy splotowe mają jądra konwolucji 3×3 .

5.3. Xception

Sieć Xception zbudowana jest w oparciu o głęboki stos 36 warstw splotowych z liniowymi połączeniami szczałkowymi (rysunek 5.4). Łącznie, model ma 71 warstw. Istnieją dwie ważne warstwy splotowe w tej architekturze. Pierwszą z nich jest głęboka warstwa splotowa. W niej znajduje się splot przestrzenny, niezależny w każdym kanale danych wejściowych. Następnie występuje warstwa splotowa z jądrem 1×1 , która mapuje kanały wyjściowe na nową przestrzeń kanałów, za pomocą splotu głębokiego [72]. Omawiany model tworzą dwa bloki przedstawione na rysunku 5.4, przy czym pierwszy blok występuje 3 razy, a drugi 7 razy.



Znaczenie terminów:

convolution – warstwa konwolucyjna; *add* – warstwa sumująca; *avgpool* – uśredniająca warstwa łącząca; *softmax* – funkcja aktywacji; *maxpool* – maksymalizująca warstwa łącząca.

Rys. 5.4. Uproszczony schemat budowy modelu sieci Xception [70]

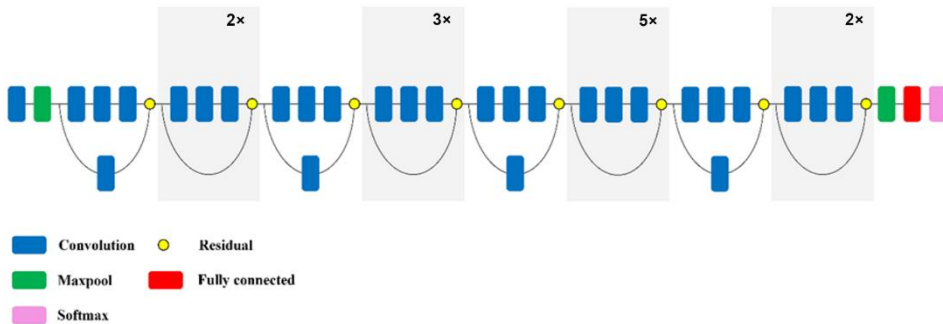
5.4. ResNet

Na rysunku 5.5 przedstawiono szczegółową architekturę modeli należących do rodziny sieci ResNet. Modele te mogą mieć 18, 34, 50, a nawet 101 lub 152 warstwy. Wśród tych modeli, najczęściej wykorzystywany jest model ResNet50. Zbudowany jest on z 48 warstw splotowych, warstwy *max-pooling* i warstwy *AVGpooling*.

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2.x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$
conv3.x	28×28	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 8$
conv4.x	14×14	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 23$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 36$
conv5.x	7×7	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

Rys. 5.5. Szczegółowa architektura modeli należących do rodziny sieci ResNet [73]

Model ResNet50 tworzy pierwsza warstwa o parametrach 7×7, z krokiem (*stride*) równym 2. Następna z warstw to *max pool* 3×3, krok = 2. Następnie występują jądra konwolucji 1×1, 3×3, 1×1, każda z tych warstw powtarzana jest trzykrotnie, dając łącznie 9 warstw. Następnie występują 3 bloki *conv3.x* (12 warstw), *conv4.x* (18 warstw), *conv5.x* (9 warstw), szczegóły odnośnie liczby ich jąder są zamieszczone na rysunku 5.5. Ostatnia z warstw modelu realizuje operację *AVGpooling* oraz implementuje funkcję aktywacji *softmax*. Model ResNet50 posiada zatem 50 warstw. Uproszczony schemat sieci ResNet przedstawiono na rysunku 5.6.



Znaczenie terminów:

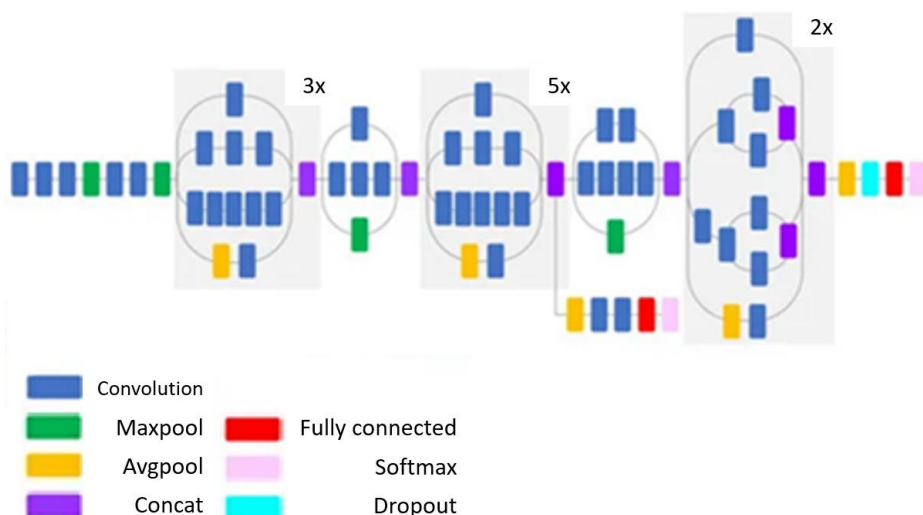
convolution – warstwa konwolucyjna; *residual* – warstwa rezydualna; *fully connected* – warstwy w pełni połączone; *softmax* – funkcja aktywacji; *maxpool* – maksymalizująca warstwa łącząca.

Rys. 5.6. Uproszczony schemat budowy modelu sieci ResNet [70]

Type	Kernel size/stride	Input size
Convolution	$3 \times 3/2$	$299 \times 299 \times 3$
Convolution	$3 \times 3/1$	$149 \times 149 \times 32$
Convolution	$3 \times 3/1$	$147 \times 147 \times 32$
Pooling	$3 \times 3/2$	$147 \times 147 \times 64$
Convolution	$3 \times 3/1$	$73 \times 73 \times 64$
Convolution	$3 \times 3/2$	$71 \times 71 \times 80$
Convolution	$3 \times 3/1$	$35 \times 35 \times 192$
Inception module	Three modules	$35 \times 35 \times 288$
Inception module	Five modules	$17 \times 17 \times 768$
Inception module	Two modules	$8 \times 8 \times 1,280$
Pooling	8×8	$8 \times 8 \times 2,048$
Linear	Logits	$1 \times 1 \times 2,048$
Softmax	Output	$1 \times 1 \times 1,000$

Rys. 5.8. Szczegółowy schemat budowy modelu sieci InceptionV3 [79]

Na rysunku 5.9 widoczny jest uproszczony schemat budowy sieci InceptionV3. Na tym schemacie dodatkowo zostały ujęte warstwa porzucania i warstwa konkatencji.



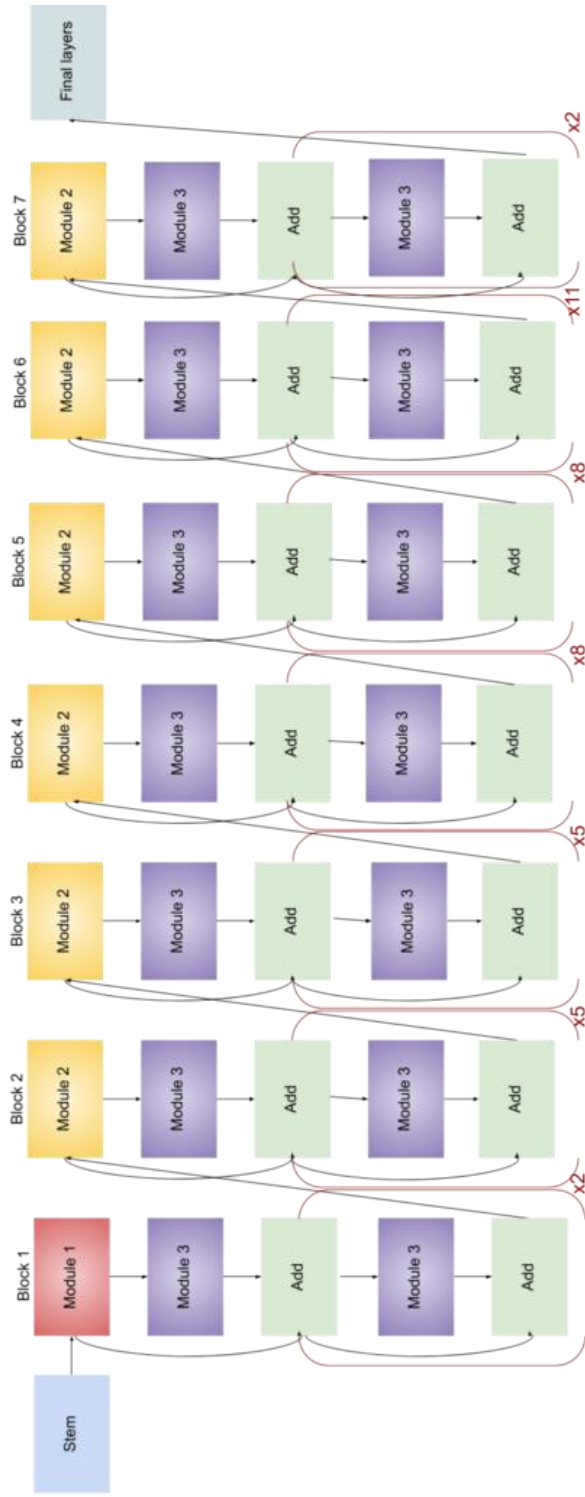
Znaczenie terminów:

convolution – warstwa konwolucyjna; *maxpool* – maksymalizująca warstwa łącząca; *avgpool* – uśredniająca warstwa łącząca; *fully connected* – warstwy w pełni połączone; *softmax* – funkcja aktywacji; *concat* – warstwa konkatencji; *dropout* – warstwa porzucania.

Rys. 5.9. Uproszczony schemat budowy modelu sieci InceptionV3 [70]

5.7. EfficientNetB7

Modele z rodziny EfficientNet charakteryzują się wysoką wydajnością [80]. Sieci tego typu, zależnie od wersji, zbudowane są z pewnej liczby bloków, z których każdy składa się z modułów. W pracy zastosowano najbardziej zaawansowany model należący tej grupy, a mianowicie model B7. Jest on najwydajniejszy z całej rodziny i składa się z 7 bloków tworzących 813 warstw (rysunek 5.10). Więcej informacji o tym modelu sieci zawarto w pracach [81,82].



Znaczenie terminów:

Stem – moduł wejściowy; *Module 1* – moduł przekazujący dane do wszystkich pozostałych modułów; *Module 2* – moduł przekazujący dane do wszystkich pozostałych modułów z wyjątkiem modułu 1; *Module 3* – moduł realizujący tzw. połączenie skrótowe (omijające) dla pozostałych modułów; *Add* – moduł sumujący; *Final layers* – warstwy wyjściowe.

Rys. 5.10. Uproszczone schemat budowy modelu sieci EfficientNetB7 [80]

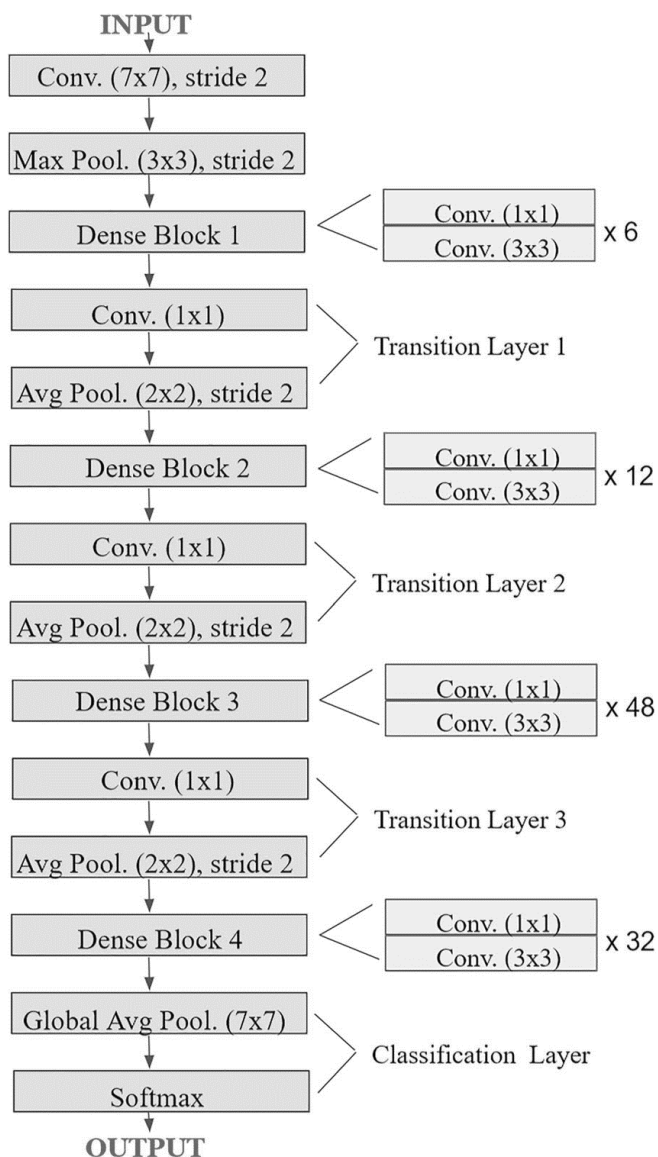
5.8. DenseNet201

Rodzina sieci DenseNet ma również rozbudowaną architekturę. Mogą ją tworzyć 121, 169, 201, a nawet 264 warstwy. Rysunek 5.11 przedstawia szczegółową strukturę rodziny modeli sieci DenseNet.

Layers	Output Size	DenseNet-121	DenseNet-169	DenseNet-201	DenseNet-264
Convolution	112 × 112	7 × 7 conv, stride 2			
Pooling	56 × 56	3 × 3 max pool, stride 2			
Dense Block (1)	56 × 56	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$
Transition Layer (1)	56 × 56 28 × 28	1 × 1 conv 2 × 2 average pool, stride 2			
Dense Block (2)	28 × 28	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$
Transition Layer (2)	28 × 28 14 × 14	1 × 1 conv 2 × 2 average pool, stride 2			
Dense Block (3)	14 × 14	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 24$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 48$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 64$
Transition Layer (3)	14 × 14 7 × 7	1 × 1 conv 2 × 2 average pool, stride 2			
Dense Block (4)	7 × 7	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 16$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 48$
Classification Layer	1 × 1	7 × 7 global average pool 1000D fully-connected, softmax			

Rys. 5.11. Szczegółowa architektura rodziny modeli sieci DenseNet [83]

Model DenseNet201 składa się z warstwy splotowej z 64 filtrami o wymiarach 7×7 i kroku równym 2. Kolejną warstwą jest warstwa *max-pooling* 3×3 (krok=2). Następnie występuje pierwszy z gęstych (w pełni połączonych) bloków, tzw. *Dense Block(1)* 1×1 i 3×3, powtarzający się 6 razy, co daje łącznie 12 warstw modelu. Następnie występuje warstwa konwolucyjna 1x1 i łącząca 2×2 z krokiem = 2. Kolejne 24 warstwy modelu tworzą drugi gęsty blok – *Dense Block(2)*, a po nim znajduje się warstwa przejściowa – *Transition Layer(2)*. Potem występuje kolejny *Dense Blok(3)*, zawierający 96 warstw, trzecia warstwa przejściowa i czwarty – ostatni już gęsty blok zawierający 64 warstwy. Na końcu znajduje się warstwa klasyfikacyjna (*Classification Layer*) zawierająca warstwę *global AVG pooling* oraz warstwę w pełni połączoną z funkcją *softmax*. DenseNet201 ma więc 201 warstw splotowych [12]. Na rysunku 5.12 zamieszczono przykładowy, uproszczony schemat sieci DenseNet201. Architektura tej sieci wspiera eliminację problemu znikającego gradientu wynikającego z dużej złożoności sieci.



Znaczenie terminów:

conv –warstwa konwolucyjna; *avg pool* – uśredniająca warstwa łącząca; *dense block* – blok warstw gęsto połączonych; *transition layer* – warstwa przejściowa; *classification layer* – warstwa klasyfikacji; *softmax* – funkcja aktywacji; *maxpool* – maksymalizująca warstwa łącząca.

Rys. 5.12. Uproszczony schemat budowy modelu sieci DenseNet201 [42]

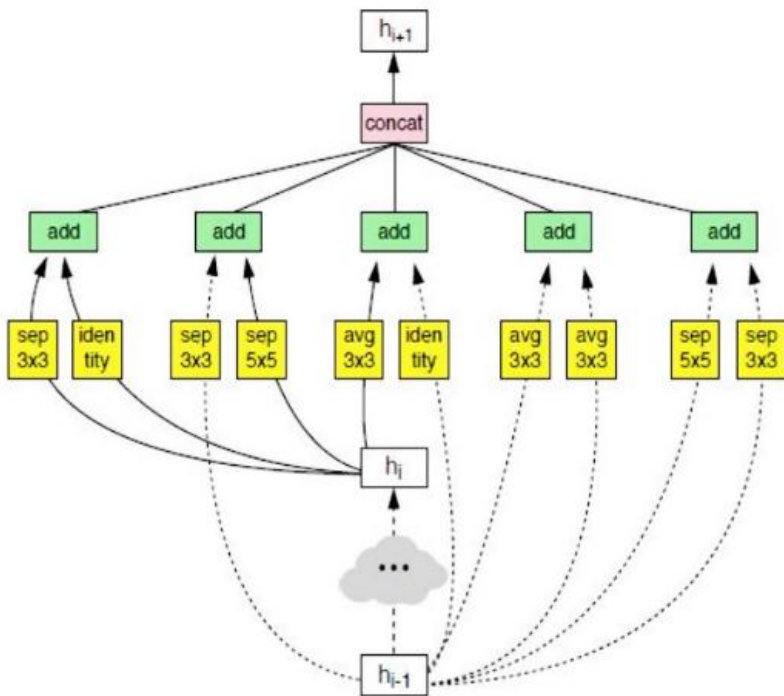
Najczęściej stosowane są architektury polegające na połączeniu wszystkich warstw, w celu zapewnienia maksymalnego przepływu informacji między nimi.

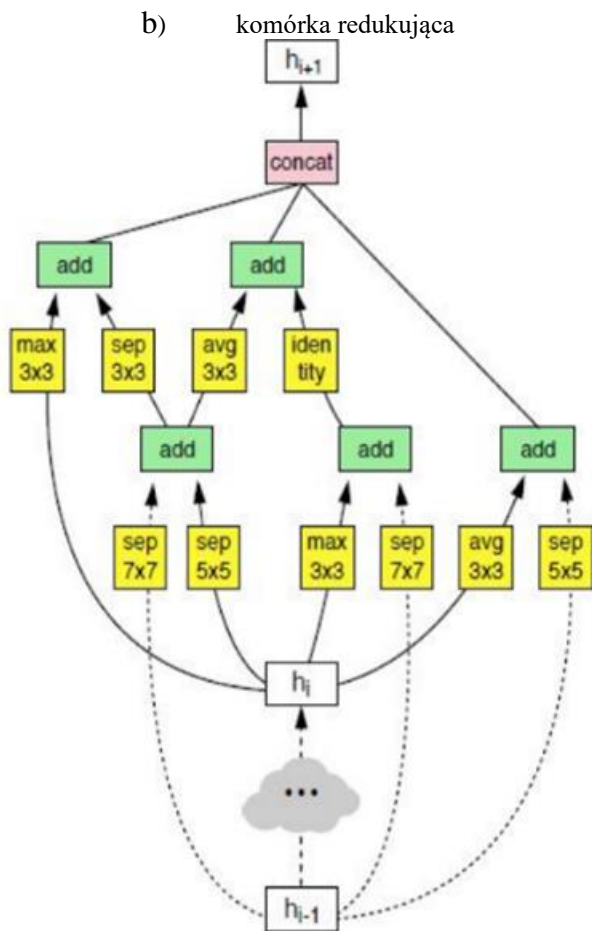
W tej konfiguracji każda warstwa pobiera dane wejściowe ze wszystkich poprzednich warstw i przekazuje własne mapy obiektów do wszystkich kolejnych warstw. Mapy obiektów są połączone na każdej warstwie, aby przekazać informacje z poprzednich warstw do kolejnych. Taka architektura sieci eliminuje konieczność uczenia się zbędnych informacji, a tym samym liczba parametrów jest znacznie zmniejszona. Jest to również sposób na przekazanie informacji dzięki połączeniu wszystkich warstw. Sieć działa bardzo dobrze w przypadku małego zestawu danych treningowych. Sieć DenseNet201 radzi sobie również bardzo dobrze z problemem nadmiernego dopasowania [16, 58].

5.9. NASNetMobile

Architektura sieci NASNetMobile stworzona została przez zespół Google Brain [84]. NASNet (*Neural Architecture Search Net*) składa się z dwóch typów komórek. Pierwszym typem jest komórka normalna, która odpowiedzialna jest za zwracanie na wyjściu mapy cech o takim samym rozmiarze, co na wejściu. Drugim typem jest komórka redukująca, która na wyjściu zwraca zmniejszoną mapę cech. W takiej mapie, wysokość i szerokość jest zmniejszona o pewien współczynnik. Na rysunku 5.13 przedstawiono schemat budowy sieci NASNetMobile wraz z podziałem na komórki normalne (a) i redukujące (b). Więcej informacji o tym modelu sieci zawarto w pracach [84,85].

a) komórka normalna





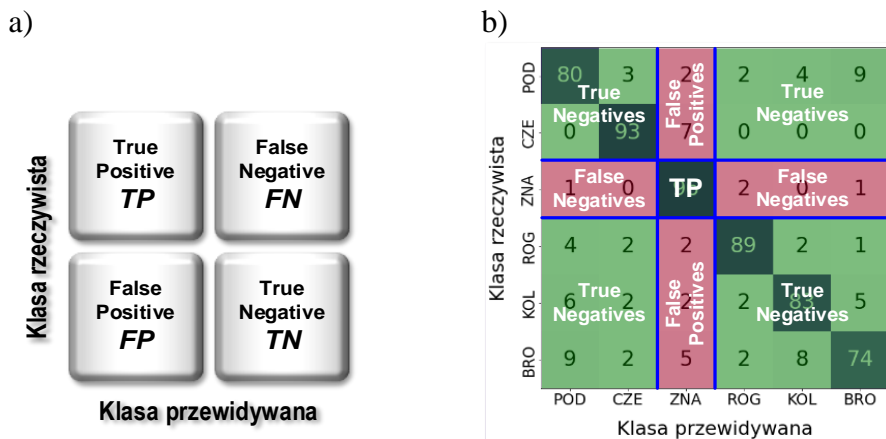
Znaczenie terminów:

add – warstwa sumująca; *concat* – warstwa konkatencji; *avg*– uśredniająca warstwa łącząca; *max* – maksymalizująca warstwa łącząca; *identity* – warstwa tożsamościowa (argument wejściowy zwraca na wyjście).

Rys. 5.13. Architektura sieci NASNet dla 5 bloków: a) bloki normalne; b) bloki redukujące [84]

6. Metryki oceny jakości modeli

Do oceny skuteczności działania klasyfikatorów wykorzystuje się różne miary jakości klasyfikacji. Związane z tym pojęcia omówiono, m.in., w [86, 87, 88]. Podstawą do obliczenia wspomnianych miar jest macierz pomyłek, która podaje wyniki przewidywań obserwacji należących do poszczególnych klas (rysunek 6.1).



Rys. 6.1. Interpretacja macierzy pomyłek: a) klasyfikacja binarna; b) klasyfikacja wieloklasowa. Znaczenie symboli: TP (prawdziwie pozytywne) – liczba przypadków pozytywnych zdiagnozowanych, jako pozytywne (chore); FN (fałszywie negatywne) – liczba przypadków pozytywnych zdiagnozowanych jako negatywne (zdrowe); TN (prawdziwie negatywne) – liczba przypadków negatywnych zdiagnozowanych, jako negatywne; FP (fałszywie pozytywne) – liczba przypadków negatywnych zdiagnozowanych, jako pozytywne

W dalszej części scharakteryzowano wybrane miary jakości klasyfikacji. Czułość (wrażliwość, *ang. sensitivity, True Positive Rate, TPR*), zwana także pełnością (*ang. Recall, REC*), określana jest, jako stosunek wyników prawdziwie dodatnich (TP) do sumy prawdziwie dodatnich (TP) i fałszywie ujemnych (FN). Czułość równa 100% oznacza, że wszystkie zmiany charakterystyczne dla stanu choroby zostały rozpoznane. Czułość zdefiniowana jest wzorem (10):

$$TPR = REC = \frac{TP}{TP + FN} \quad (10)$$

Swoistość (specyficzność, *ang. specificity, True Negative Rate, TNR*) to stosunek wyników prawdziwie ujemnych (TN) do sumy prawdziwie ujemnych i fałszywie dodatnich (FP). Wynik dla testu swoistości równy 100% oznaczałoby, że wszyscy ludzie zdrowi w wykonanym teście diagnostycznym zostaliby zdiagnozowali, jako pacjenci zdrowi. Swoistość wyrażona jest wzorem (11):

$$TNR = \frac{TN}{TN + FP} = 1 - \frac{FP}{TN + FP}. \quad (11)$$

Wskaźnik fałszywie dodatni (*ang. False Positive Rate, FPR*) to szacowane prawdopodobieństwo błędu polegającego na przewidzeniu choroby zdrowego pacjenta:

$$FPR = 1 - \frac{FP}{TN + FP}. \quad (12)$$

Dokładność (*ang. Accuracy, ACC*) to prawdopodobieństwo klasyfikacji przypadków zależących do kategorii zdrowych i chorych:

$$ACC = \frac{TP + TN}{TP + TN + FP + FN}. \quad (13)$$

Krzywa ROC (*ang. Receiver Operating Characteristic Curve*) pokazana na rysunku 6.2 powstaje w oparciu o dane czułości i specyficzności. Służy ona porównaniu skuteczności różnego rodzaju klasyfikatorów [89, 90]. Obszar pod krzywą opisuje skuteczność klasyfikatora, która może przyjmować wartości w zakresie 0 – 1 [86]. Im bliżej krzywa znajduje się lewego górnego rogu wykresu, tym większe jest pole pod wykresem i tym wyższa jest skuteczność klasyfikatora. Błąd klasyfikacji (*ang. Classification Error, ERR*) wyrażony jest zależnością:

$$ERR = \frac{FP + FN}{TP + TN + FP + FN} = 1 - ACC. \quad (14)$$

Wskaźnik fałszywie dodatni (*ang. False Positive Rate, FPR*) to prawdopodobieństwo błędu przyporządkowania pacjenta zdrowego do grupy chorych pacjentów:

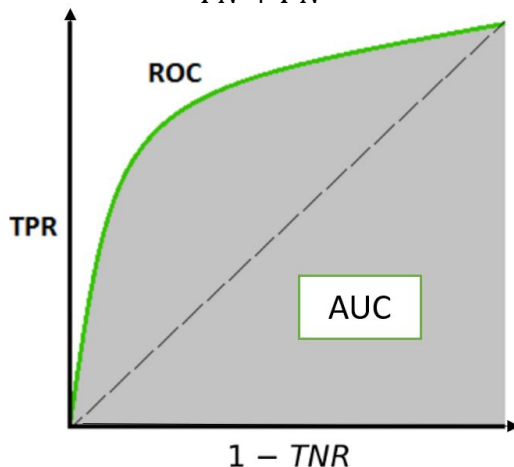
$$FPR = \frac{FP}{FP + TN}. \quad (15)$$

Wartość predykcyjna dodatnia (*ang. Positive Predictive Value, PPV*) to prawdopodobieństwo klasyfikacji przypadków osób chorych poprawnie zdiagnozowanych jako chore:

$$PPV = \frac{TP}{TP + FP}. \quad (16)$$

Wartość predykcyjna ujemna (*ang. Negative Predictive Value, NPV*) określa prawdopodobieństwo klasyfikacji przypadków osób zdrowych poprawnie przyporządkowanych do grupy osób zdrowych:

$$NPV = \frac{TN}{TN + FN} \quad (17)$$



Rys. 6.2. Krzywa ROC

Wskaźnik fałszywego rozpoznania (*ang. False Discovery Rate, FDR*) to prawdopodobieństwo tego, że przypadki fałszywie zdiagnozowane zostały zakwalifikowane jako chore. Wskaźnik określany jest zależnością:

$$FDR = \frac{FP}{FP + TP} \quad (18)$$

Precyzja (*ang. Precision, PRE*) to wskaźnik określający jak duża część wybranych obserwacji jest trafna, co pokazuje wzór:

$$PRE = \frac{TP}{TP + FP} \quad (19)$$

Wskaźnik F1 (*ang. F1 Score, F1*) jest średnią harmoniczną precyzji i pełności. Jego wysoka wartość świadczy zarówno o dużych wartościach precyzji i pewności. Wskaźnik określany jest zależnością:

$$F1 = 2 \cdot \frac{PRE \cdot REC}{PRE + REC} \quad (20)$$

7. Trenowanie modeli klasyfikacyjnych

W niniejszym rozdziale przedstawiono wyniki trenowania modeli klasyfikacyjnych z wykorzystaniem dziesięciu różnych architektur konwolucyjnych sieci neuronowych dostępnych w pakiecie Keras: MobileNetV2 [90, 71], DenseNet201 [58], NASNetMobile [85], Xception [72], VGG16, VGG19 [59], ResNet50 [74, 91], InceptionV3 [92], InceptionResNetV2 [77], EfficientNetB7. Wszystkie te modele zostały wytrenowane na zbiorze ImageNet zawierającym ok. 1,4 miliona obrazów podzielonych na 1 000 klas. W badaniach wykorzystano modele charakteryzujące się różną głębokością topologiczną (tabela 7.1).

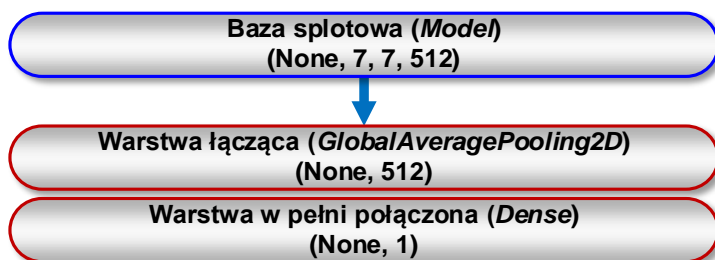
Tabela 7.1. Modele zastosowane w badaniach, które zostały wcześniej wytrenowane na zbiorze ImageNet

Nr	Model	Liczba warstw bazy konwolucyjnej	Dostrajane warstwy bazy konwolucyjnej
1	VGG16	19	12÷19
2	VGG19	22	13÷22
3	Xception	132	117÷132
4	MobileNetV2	154	136÷154
5	ResNet50	175	155÷175
6	InceptionV3	311	250÷311
7	DenseNet201	595	142÷595
8	NASNetMobile	769	346÷769
9	InceptionResNetV2	780	631÷780
10	EfficientNetB7	806	553÷806

Do budowy modeli wykorzystany był komputer z systemem Windows 10 64-bity, procesorem Intel Core i5-3470 3.20 GHz i pamięcią RAM 32 GB. Zastosowano platformę TensorFlow 1.13.1, bibliotekę Keras 2.2.4 [89] oraz język programowania Python 3.7.3. Obliczenia były wykonywane z wykorzystaniem GPU (NVIDIA GeForce GTX 1060 3 GB), platformy CUDA 9.0 oraz biblioteki cuDNN 7.0.

Modele składały się z dwóch części. Pierwszą były bazy splotowe, a drugą własne klasyfikatory położone na końcu sieci. Wykorzystano bazy splotowe, ponieważ wyuczone przez nie reprezentacje prezentują ogólne koncepcje, które są odpowiednie do rozwiązywania różnych problemów związanych z przetwarzaniem obrazu. Zamiast klasyfikatorów oryginalnych, zastosowano własne, zbudowane z warstwy łączącej oraz warstwy gęsto połączonej. Reprezentacje, których wyuczyły się własne klasyfikatory były specyficzne dla badanego zestawu klas, na których modele trenowano. Rysunek 7.1 przedstawia ogólny schemat implementacji sieci głębokiej, która składa się z bazy splotowej

oraz nowego klasyfikatora binarnego dodanego na jej koniec. W przypadku klasyfikatora wieloklasowego, liczba neuronów na wyjściu warstwy gęsto połączonej jest równa liczbie klas, czyli 6.



Rys. 7.1. Ogólny schemat implementacji sieci głębokiej, składającej się bazy splotowej (niebieska linia) oraz własnego klasyfikatora (czerwona linia). Znaczenie poszczególnych klas biblioteki Keras jest następujące: *GlobalAveragePooling2D* – warstwa łącząca (zwraca wartości średnie każdego z kanałów); *Dense* – warstwa w pełni połączona. Mapa cech ma kształt (*próbki, wysokość, szerokość, kanały*). Pierwszy wymiar, równy *None*, oznacza dowolną liczbę próbek

Podczas budowy modeli, zastosowano architekturę opartą na sekwencyjnym stosie warstw, zaimplementowanych w postaci klas biblioteki Keras. Po bazie splotowej, występuje pierwsza warstwa własnego klasyfikatora (*GlobalAveragePooling2D*). Realizuje ona operację łączenia danych dwuwymiarowych na podstawie wartości średniej i zawiera 512 neuronów na wyjściu. Następnie, dodano warstwę w pełni połączoną, która interpretuje cechy wyłonione przez bazę splotową i na tej podstawie prognozuje przynależność bieżącej obserwacji do określonej klasy.

Trenowanie modelu realizowane było w dwóch fazach:

1. **Ekstrakcja cech** – baza splotowa została zamrożona, a dodana warstwa w pełni połączona (tworząca nowy klasyfikator) została losowo zainicjowana i wytrenowana przez 50 epok, z wykorzystaniem techniki dogenerowania danych. Zamrażanie warstw ma na celu uniemożliwienie aktualizacji ich wag podczas uczenia. Dzięki temu, reprezentacje wyuczone wcześniej przez bazę splotową nie są modyfikowane podczas treningu nowych warstw.
2. **Dostrajanie** – 2 górne bloki bazy splotowej zostały odmrożone i wytrenowane przez 50 epok razem z nową warstwą w pełni połączoną (także z wykorzystaniem dogenerowania danych). Na końcu, cała baza splotowa została odmrożona.

Podczas trenowania modeli zastosowano technikę regularyzacji zwaną dogenerowaniem danych (ang. *data augmentation*). Jej celem było wygenerowanie większej liczby elementów zbioru uczącego na drodze losowych przekształceń, zwracających obrazy wyglądające wiarygodnie. Wykonywane były następujące operacje: obrót obrazu, przekształcenia pionowe i poziome,

przycinanie, przybliżanie i oddalanie fragmentów oraz odbicie połowy obrazu w płaszczyźnie poziomej i pionowej. W efekcie tych przekształceń, trenowane modele nigdy nie przetwarzały dwukrotnie tego samego obrazu.

Tabela 7.2. Ustawienia zastosowane podczas budowy modeli

Zastosowanie	Parametr	Wartość
trenowanie	liczba epok	50 (ekstrakcja cech), 50 (dostrajanie)
	wielkość wsadu	8
model	funkcja straty	binarna entropia krzyżowa, kategoryczna entropia krzyżowa
	optymalizator	Adam
	metryka	dokładność klasyfikacji
optymalizator	szybkość uczenia	10^{-3} (ekstrakcja cech), 10^{-5} (dostrajanie)
	stała β_1	0,9
	stała β_2	0,999
	stała ε	10^{-7}
dogenerowanie danych	zakres obrotu	40°
	zakres zmiany szerokości	0,2
	zakres zmiany wysokości	0,2
	zakres przycinania	0,2
	zakres powiększenia	0,2
	przerzucanie w poziomie	True

Jako funkcję straty, zastosowano binarną entropię krzyżową w przypadku klasyfikacji binarnej lub kategoryczną entropię krzyżową dla klasyfikacji wieloklasowej. Jako algorytmu optymalizacji, użyto optymalizatora Adam z wartością współczynnika uczenia równą 10^{-3} dla fazy ekstrakcji cech oraz 10^{-5} dla fazy dostrajania. Niska wartość tego współczynnika podczas dostrajania wynikała z tego, że modyfikacje reprezentacji dostrajanych warstw bazy plotowej powinny być minimalne. Zbyt duże zmiany tych wartości mogłyby zaszkodzić reprezentacji danych. Jako miarę skuteczności procesu uczenia, zastosowano dokładność klasyfikacji. W warstwie wyjściowej klasyfikatora zastosowano funkcję aktywacji *sigmoid* oraz 1 neuron dla klasyfikatora binarnego lub funkcję aktywacji *softmax* oraz 6 neuronów w przypadku klasyfikatora wieloklasowego. Wytrenowane modele zapisywane były na dysku w postaci plików w formacie HDF5. W tabeli 7.2 zamieszczono bardziej szczegółowe informacje na temat ustawień zastosowanych podczas budowy

modeli. Załącznik B zawiera listing przykładowego kodu w języku Python, który zastosowano do trenowania modeli.

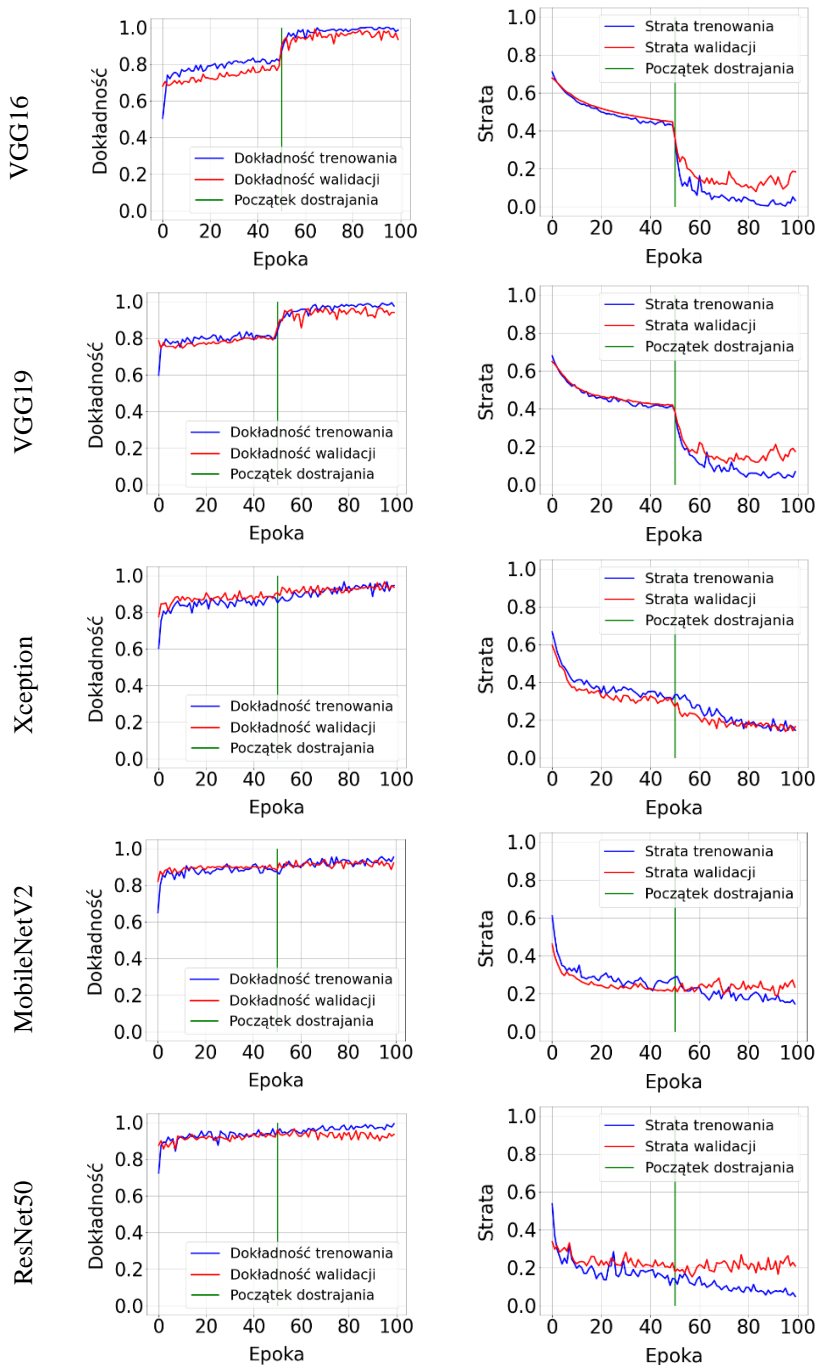
7.1. Klasyfikacja binarna (ZNAMIE – CZERNIAK)

Do klasyfikacji binarnej wybrano łącznie 800 obrazów dermatoskopowych łagodnych zmian skórnych (znamion) oraz czerniaków. Obrazy zostały podzielone na zbiór treningowy, walidacyjny i testowy. Zbiór treningowy stanowił 50% zbioru pełnego, co daje 400 obrazów dermatoskopowych. Zbiór testowy i walidacyjny liczyły po 25% zbioru pełnego (łącznie 400 obrazów). Dokładne liczebności podzbiorów zostały przedstawione w tabeli 7.3. Klasyfikację binarną wykonano dla dziesięciu sieci: VGG16, VGG19, Xception, MobileNetV2, ResNet50, InceptionV3, InceptionResNetV2, EfficientNetB7, DenseNet201, NASNetMobile. Uczenie sieci przeprowadzono w dwóch etapach opisanych w rozdziale 8. W pierwszej fazie (ekstrakcja cech) trenowano nowe warstwy sieciowe (nowy klasyfikator), a cała baza konwolucyjna została zamrożona. W drugiej fazie (dostrajanie) pewna liczba końcowych warstw bazy konwolucyjnej została odmrożona i trenowana wraz z nowym klasyfikatorem.

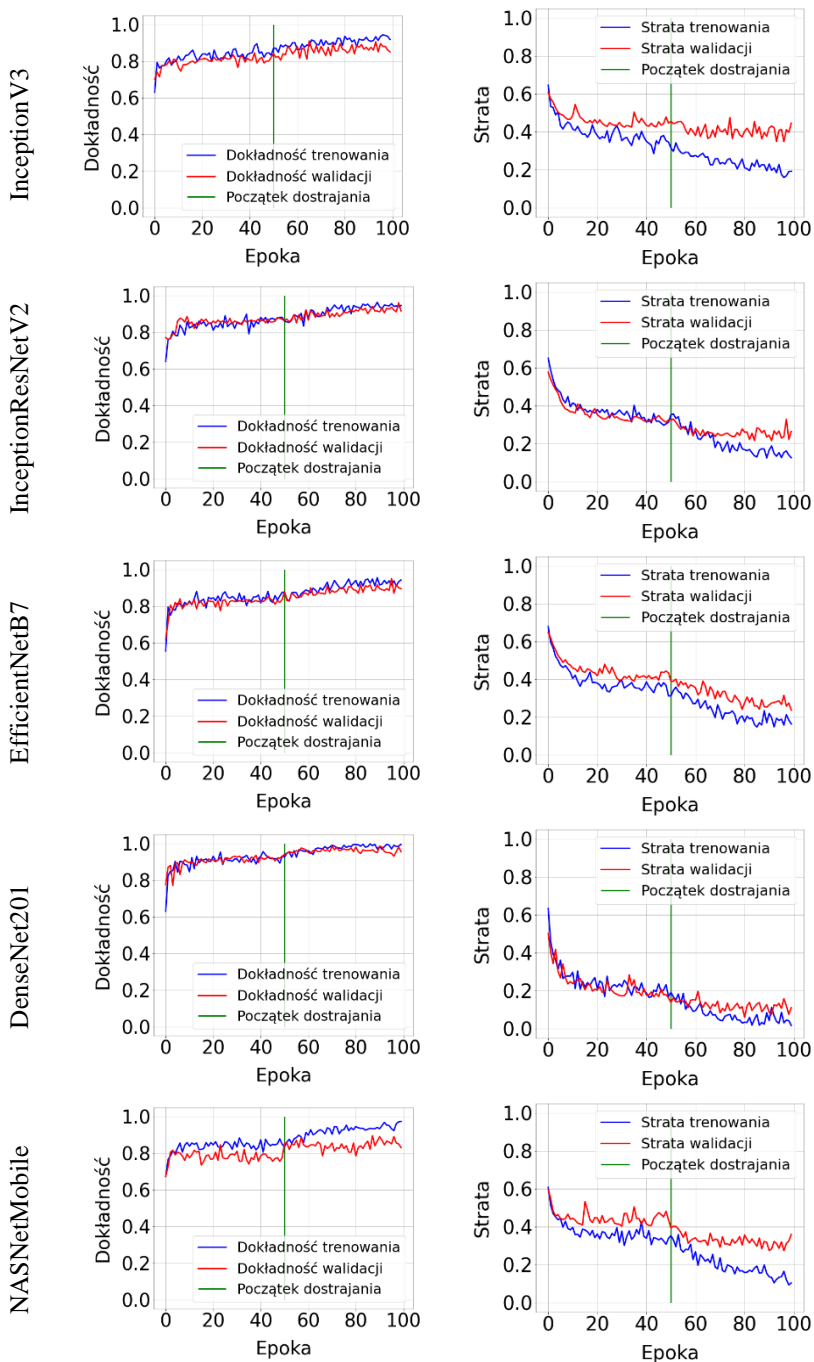
Na rysunkach 7.2a i 7.2b. przedstawiono wykresy dokładności oraz straty trenowania i walidacji wyżej wymienionych modeli sieci. Na ich podstawie widać, że model InceptionV3 charakteryzuje się najslabszym dopasowaniem do danych w porównaniu do pozostałych modeli. Taki wniosek można wysnuć na podstawie największej różnicy pomiędzy wartością straty trenowania a walidacji. Możliwą przyczyną takiego stanu jest zbyt mała pojemność modelu. Dla pozostałych sieci osiągnięto lepsze wyniki ewaluacji, dzięki lepszemu dopasowaniu. Najlepsze dopasowanie miało miejsce w przypadku wykresu strat dla modelu InceptionResNetV2.

Tabela 7.3. Liczebność podzbiorów podczas klasyfikacji binarnej z wykorzystaniem obrazów znamion i czerniaków

Zbiór danych	Liczba obserwacji	Część zbioru pełnego	ZNAMIE	CZERNIAK
Pełny	800	1,00	400	400
Treningowy	400	0,50	200	200
Walidacyjny	200	0,25	100	100
Testowy	200	0,25	100	100



Rys. 7.2a. Wykresy dokładności oraz straty trenowania i walidacji modeli: VGG16, VGG19, Xception, MobileNetV2, ResNet50



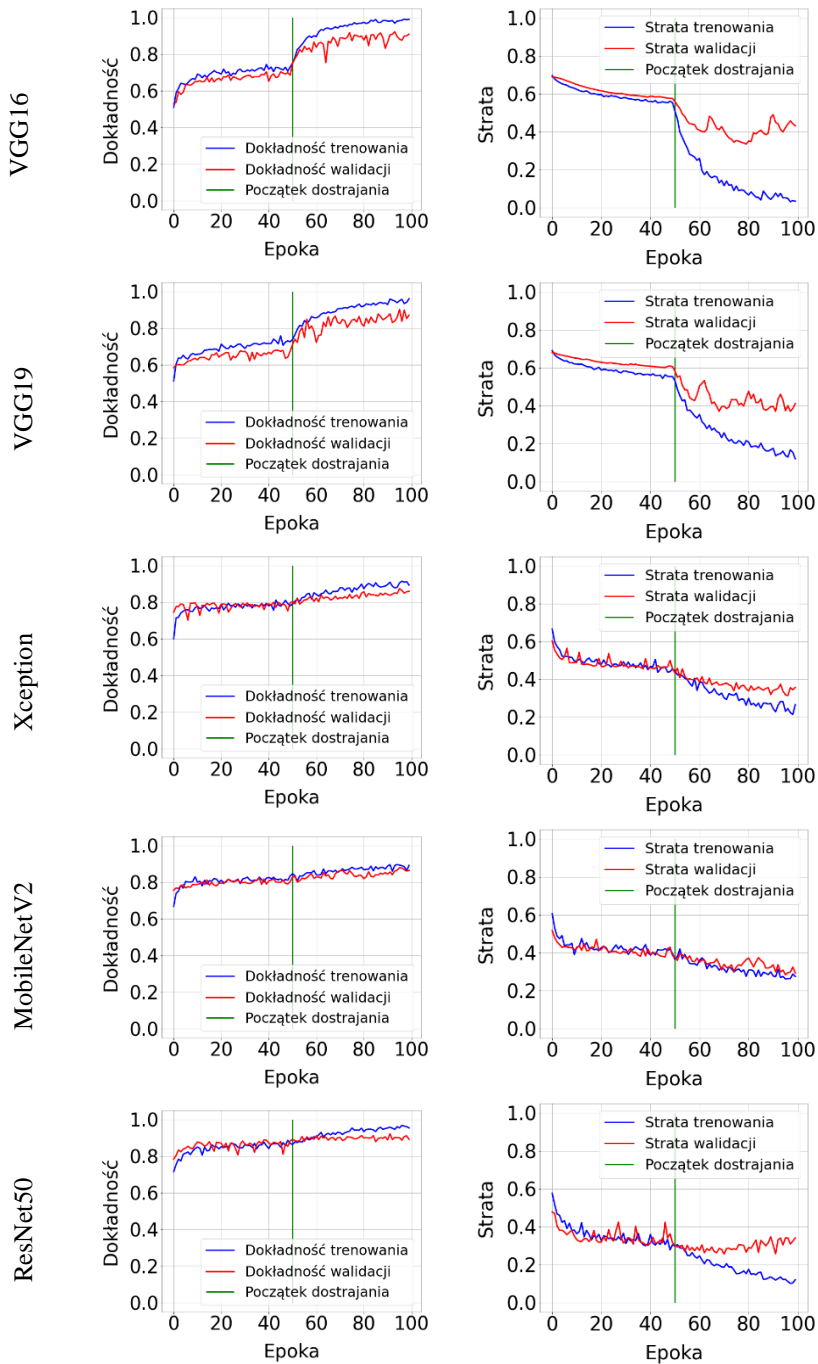
Rys. 7.2b. Wykresy dokładności oraz straty trenowania i walidacji modeli: InceptionV3, InceptionResNetV2, EfficientNetB7, DenseNet201, NASNetMobile

7.2. Klasyfikacja binarna (ŁAGODNE – ZŁOŚLIWE)

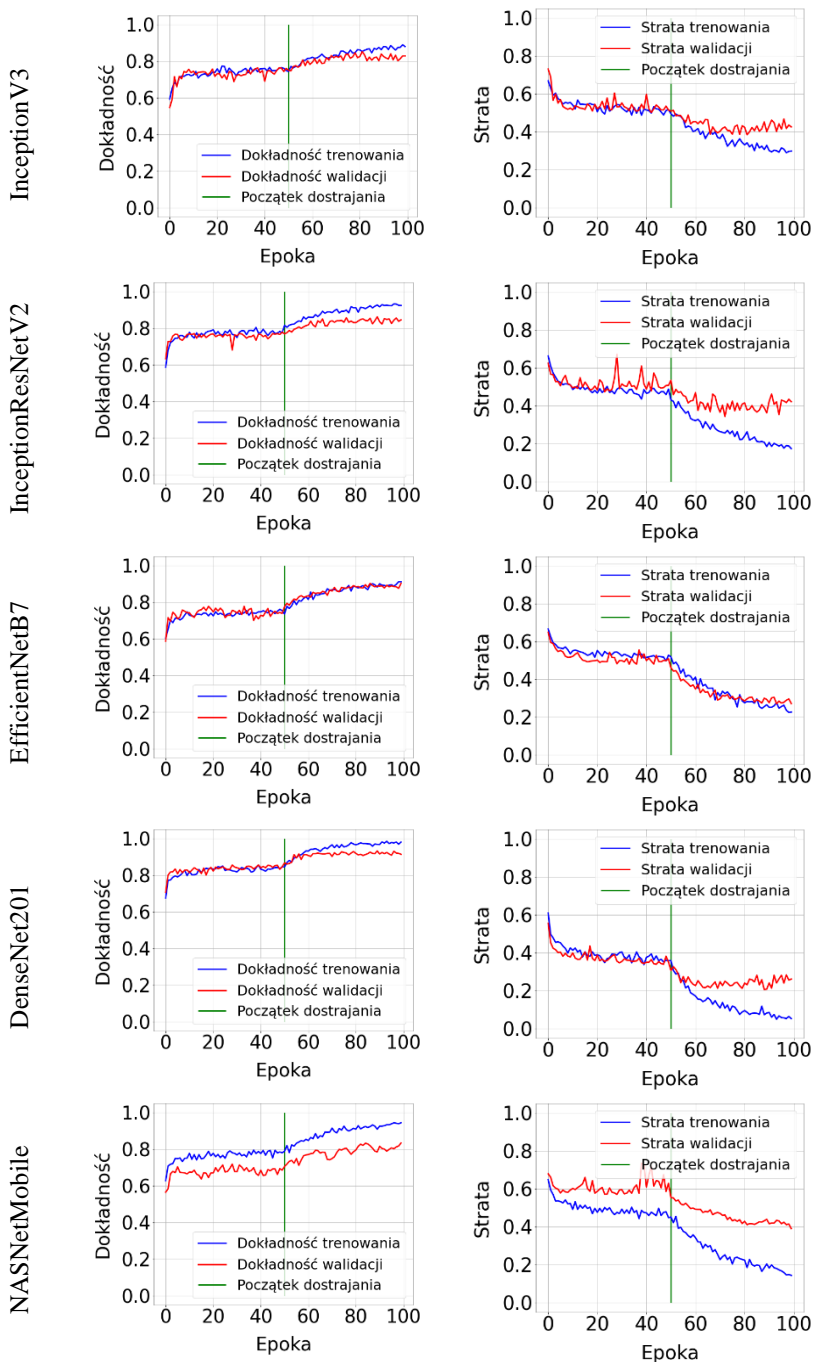
Podczas badań opisanych w niniejszym rozdziale, pełny zbiór danych zawierał 2400 obserwacji, wśród których 1200 reprezentowało zmiany łagodne i 1200 zmiany złośliwe. Obu kategoriom obrazów przypisano etykiety, odpowiednio „zdrowy” oraz „chory”. Następnie, pełny zbiór danych podzielono w sposób losowy na podzbiory: uczący, walidacyjny i testowy, przy czym w każdym z tych podzbiorów liczebność obserwacji należących do poszczególnych klas była jednakowa. Zbiór treningowy stanowił 50% zbioru pełnego (600 obserwacji), walidacyjny 25% (300 obserwacji) i testowy także 25% (300 obserwacji). Dokładne liczebności podzbiorów zostały przedstawione w tabeli 7.4. Klasyfikację binarną wykonano również dla dziesięciu sieci: VGG16, VGG19, Xception, MobileNetV2, ResNet50, InceptionV3, InceptionResNetV2, EfficientNetB7, DenseNet201, NASNetMobile. Na rysunkach 7.3a i 7.3b zostały przedstawione wykresy dokładności oraz straty trenowania i walidacji wyżej wymienionych modeli sieci. Największe niedopasowanie modelu występuje w przypadku sieci NASNetMobile. Natomiast najmniejszy odstęp pomiędzy krzywymi straty trenowania i walidacji jest dla sieci MobileNetV2 i InceptionResNetV2.

Tabela 7.4. Liczebność podzbiorów podczas klasyfikacji binarnej z wykorzystaniem obrazów zmian łagodnych i złośliwych

Zbiór danych	Liczba obserwacji	Część zbioru pełnego	ŁAGODNE	ZŁOŚLIWE
Pełny	2400	1,00	1200	1200
Treningowy	1200	0,50	600	600
Walidacyjny	600	0,25	300	300
Testowy	600	0,25	300	300



Rys. 7.3a. Wykresy dokładności oraz straty trenowania i walidacji modeli: VGG16, VGG19, Xception, MobileNetV2, ResNet50



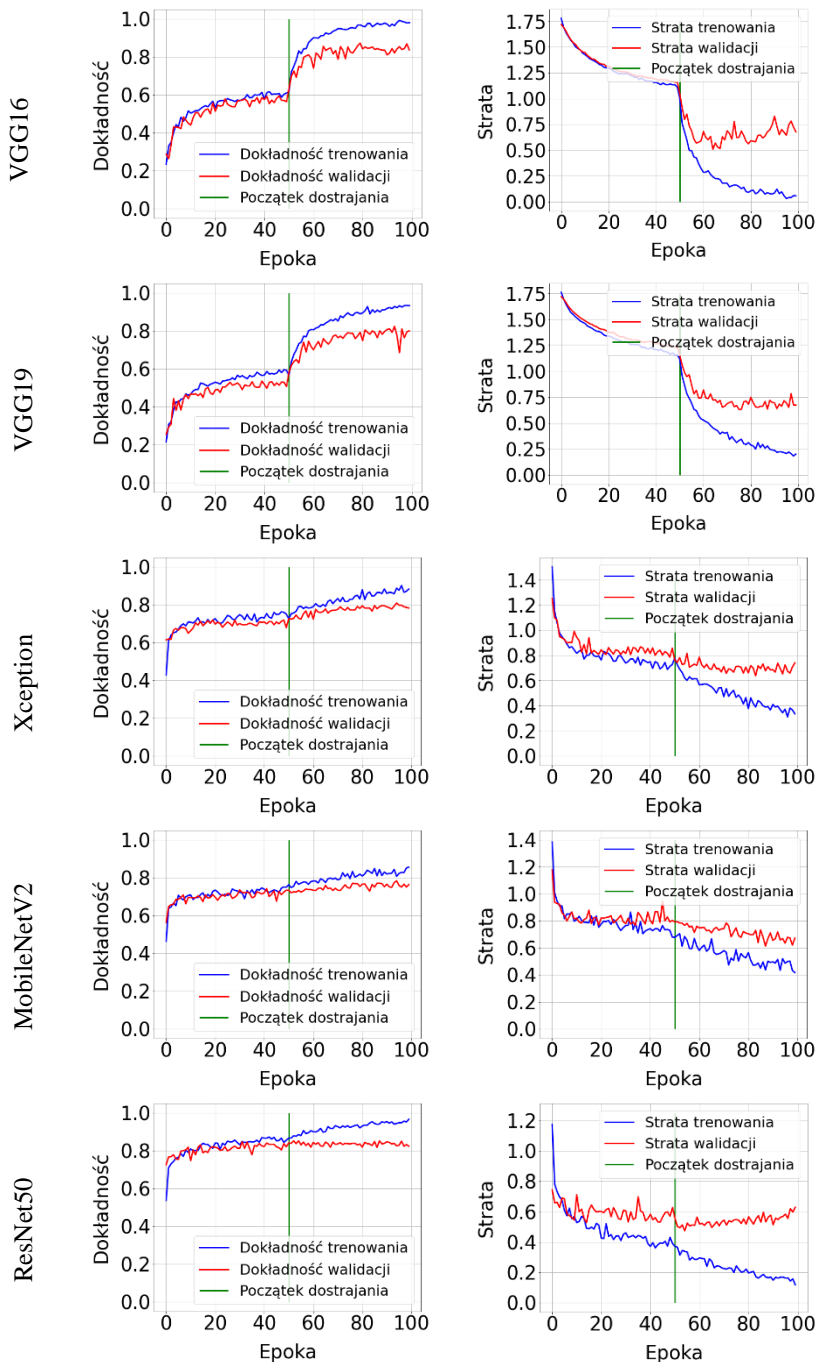
Rys. 7.3b. Wykresy dokładności oraz straty trenowania i walidacji modeli: InceptionV3, InceptionResNetV2, EfficientNetB7, DenseNet201, NASNetMobile

7.3. Klasyfikacja wieloklasowa

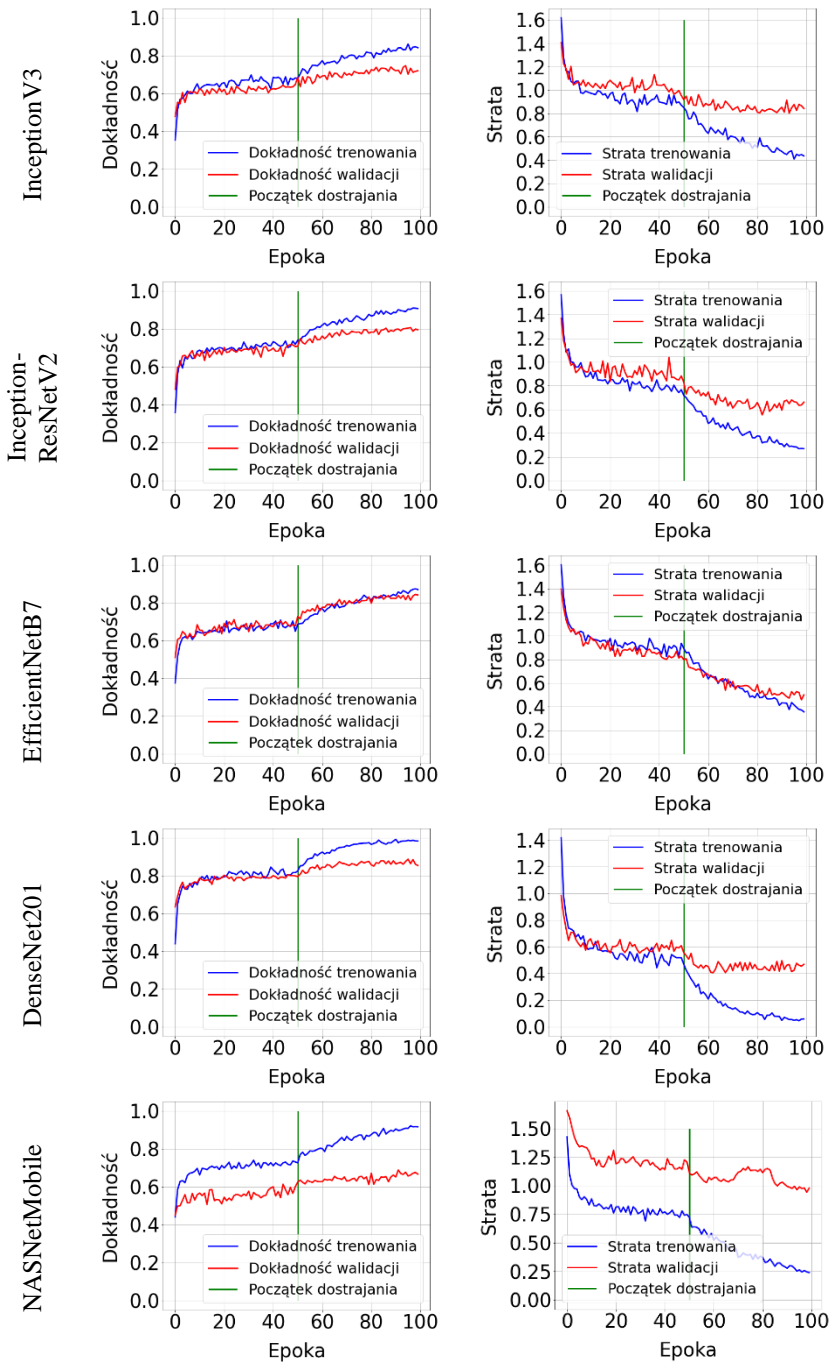
W celu przeprowadzenia klasyfikacji wieloklasowej wybrano łącznie 2400 obrazów dermatoskopowych. Dane zostały podzielone na sześć klas reprezentujących różne zmiany skórne. Do diagnozowanych zmian złośliwych należały: rak kolczystokomórkowy (KOL), czerniak złośliwy (CZE), rak podstawnokomórkowy (POD). Klasy reprezentujące zmiany skórne łagodne to: znamię melanocytowe (ZNA), łagodne rogowacenie barwnikowe (ROG), brodawka łojotokowa (BRO). Obrazy zostały podzielone na zbiór treningowy, walidacyjny i testowy w stosunku odpowiednio 50%, 25%, 25% wszystkich użytych obserwacji. Dokładne liczebności podzbiorów zostały przedstawione w tabeli 7.5. Modele klasyfikacji wieloklasowej zbudowano dla dziesięciu sieci: VGG16, VGG19, Xception, MobileNetV2, ResNet50, InceptionV3, InceptionResNetV2, EfficientNetB7, DenseNet201, NASNetMobile. Na rysunkach 7.4a i 7.4b zostały przedstawione wykresy dokładności oraz straty trenowania i walidacji wyżej wymienionych modeli sieci. Najstabilniej dopasowanymi modelami okazały się ResNet50, VGG19 i NASNetMobile, natomiast oznaki przetrenowania wykazał model VGG16 (od pewnego momentu strata walidacji zaczęła narastać) Z kolei, najlepsze dopasowanie wystąpiło w przypadku sieci EfficientNetB7.

Tabela 7.5. Liczebność podzbiorów podczas klasyfikacji wieloklasowej z wykorzystaniem sześciu rodzajów zmian skórnych

Zbiór danych	Liczba obserwacji	Część zbioru pełnego	POD	CZE	ZNA	ROG	KOL	BRO
Pełny	2400	1,00	400	400	400	400	400	400
Treningowy	1200	0,50	200	200	200	200	200	200
Walidacyjny	600	0,25	100	100	100	100	100	100
Testowy	600	0,25	100	100	100	100	100	100



Rys. 7.4a. Wykresy dokładności oraz straty trenowania i walidacji modeli: VGG16, VGG19, Xception, MobileNetV2, ResNet50

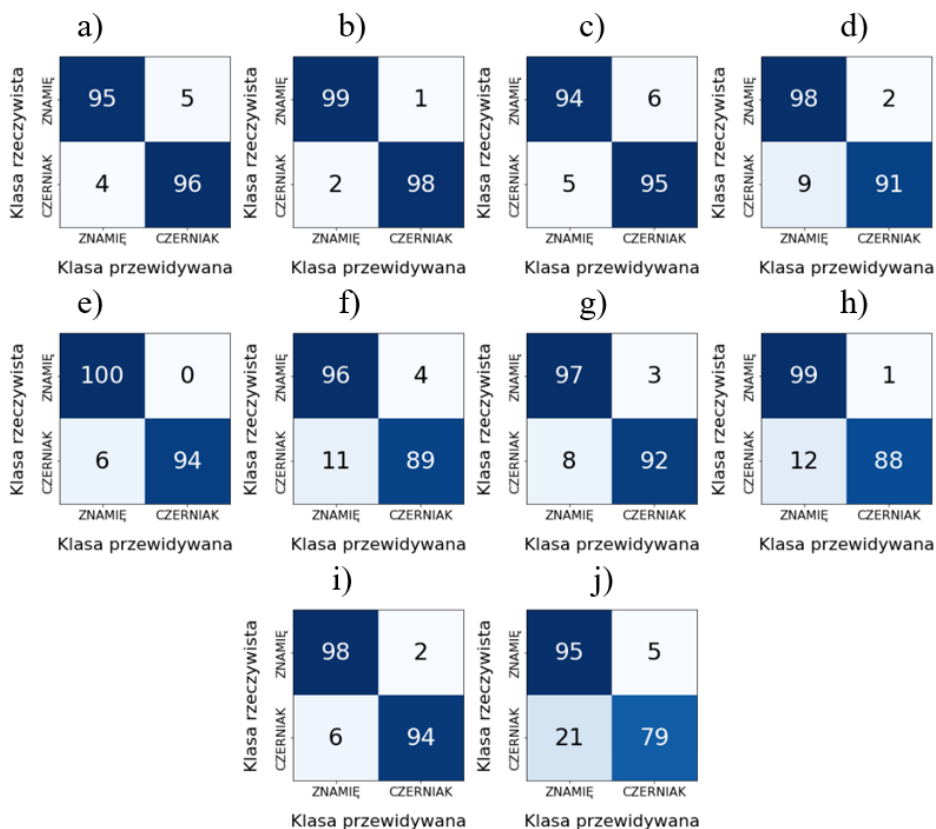


Rys. 7.4b. Wykresy dokładności oraz straty trenowania i walidacji modeli: InceptionV3, InceptionResNetV2, EfficientNetB7, DenseNet201, NASNetMobile

8. Ewaluacja modeli

8.1. Testowanie modeli z wykorzystaniem zbioru ISIC

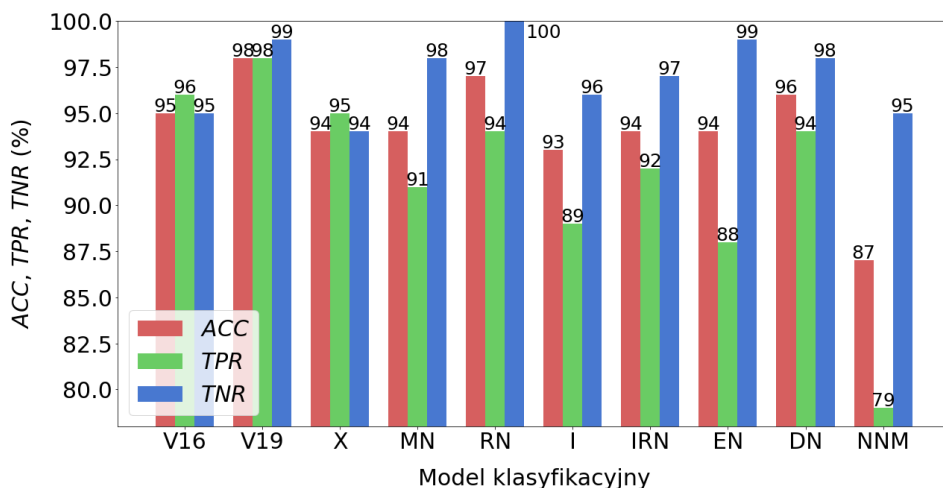
W pierwszym etapie procesu ewaluacji przeprowadzono testowanie zbudowanych modeli z wykorzystaniem zbioru ISIC. Między innymi, zbudowano macierze pomyłek, które pokazują rozkład poprawnych i niepoprawnych przypadków klasyfikacji (rysunek 8.1). Wartości w nich zawarte odpowiadają kolejno parametrom TN, FP, FN oraz TP. Na ich podstawie możliwe było wyznaczenie pozostałych parametrów określających jakość pracy klasyfikatorów. Niskie wartości wyżej wymienionych parametrów zaznaczono kolorem jaśniejszym.



Rys. 8.1. Macierze pomyłek modeli klasyfikacji binarnej (ZNAMIE – CZERNAK):
a) VGG16; b) VGG19; c) Xception; d) MobileNetV2; e) ResNet50;
f) InceptionV3; g) InceptionResNetV2; h) EfficientNetB7; i) DenseNet201;
j) NASNetMobile

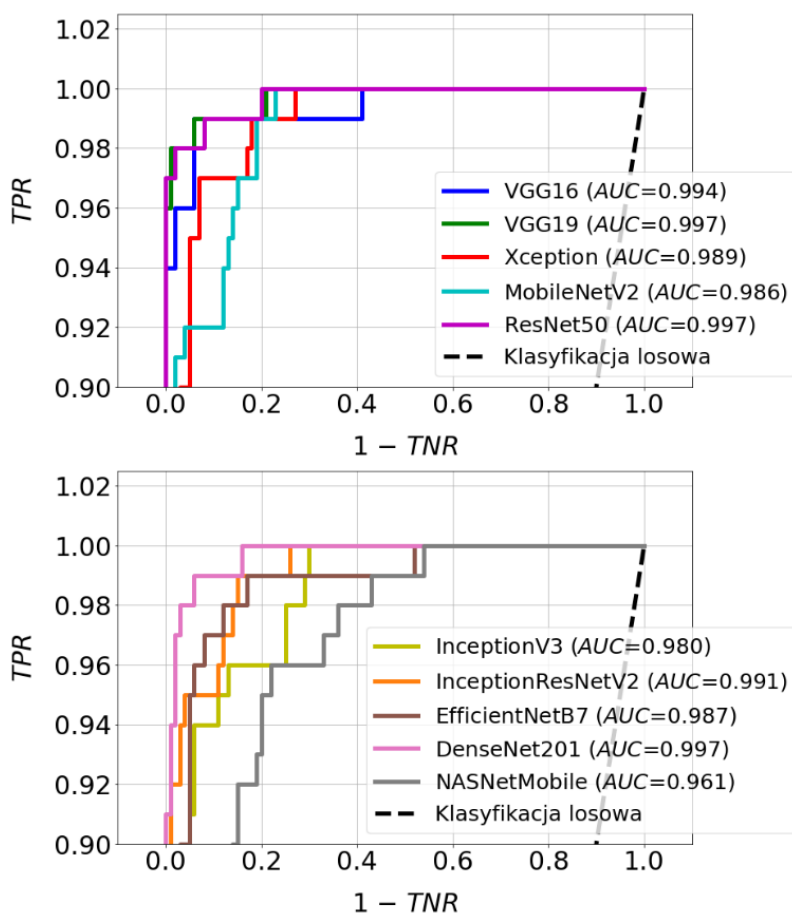
Załącznik C zawiera listing przykładowego kodu w języku Python, który zastosowano do ewaluacji modeli. Model VGG19 pomylił się tylko raz rozpoznając zamię jako czerniaka i tylko 2 razy pomylił się diagnozując czerniaka, jako zmianę. Wśród modeli sieci o niskiej liczbie pomyłek należy wymienić także: VGG16, Xception i DenseNet201. Oczywiście, podobne liczby pomyłek świadczą o porównywalnej skuteczności modeli.

W ramach ewaluacji modeli, obliczono dokładność (ACC) klasyfikacji binarnej (ZNAMIE – CZERNIAK), czułość (TPR), swoistość (TNR) i pole pod krzywą ROC (wskaźnik AUC) (rysunek 8.2. oraz rysunek 8.3.). Najwyższą skuteczność 98% uzyskano dla modelu sieci VGG19. Wysoką dokładność osiągnęły także modele: ResNet50 (ACC = 97%), DenseNet201 (ACC = 96%). Zdecydowanie najniższą wartość dokładności uzyskano dla modelu NASNetMobile (ACC = 87%), co jest wynikiem najslabszego dopasowania tego modelu do danych treningowych. Należy zwrócić uwagę na wysoką dokładność diagnostyczną modelu VGG19 (TPR = 98%) oraz bardzo dobrą skuteczność oferowaną przez model ResNet50 (TNR = 100%), VGG19 i EfficientNetB (TNR = 99%), MobileNetV2 i DenseNet201 (TNR = 98%) (rysunek 8.2).



Rys. 8.2. Wyniki testowania modeli klasyfikacji binarnej (ZNAMIE – CZERNIAK)

Krzywe ROC przedstawione na rysunku 8.3 potwierdzają wysoką skuteczność Modeli VGG19, ResNet50 i DenseNet201. Parametr AUC przyjmuje dla tych sieci wartości 0,997. Niższe wartości parametru AUC mają kolejno modele VGG16, InceptionResNetV2, Xception, EfficientNetB7. Natomiast zdecydowanie najgorsza skuteczność została osiągnięta dla NASNetMobile (AUC = 0,961).

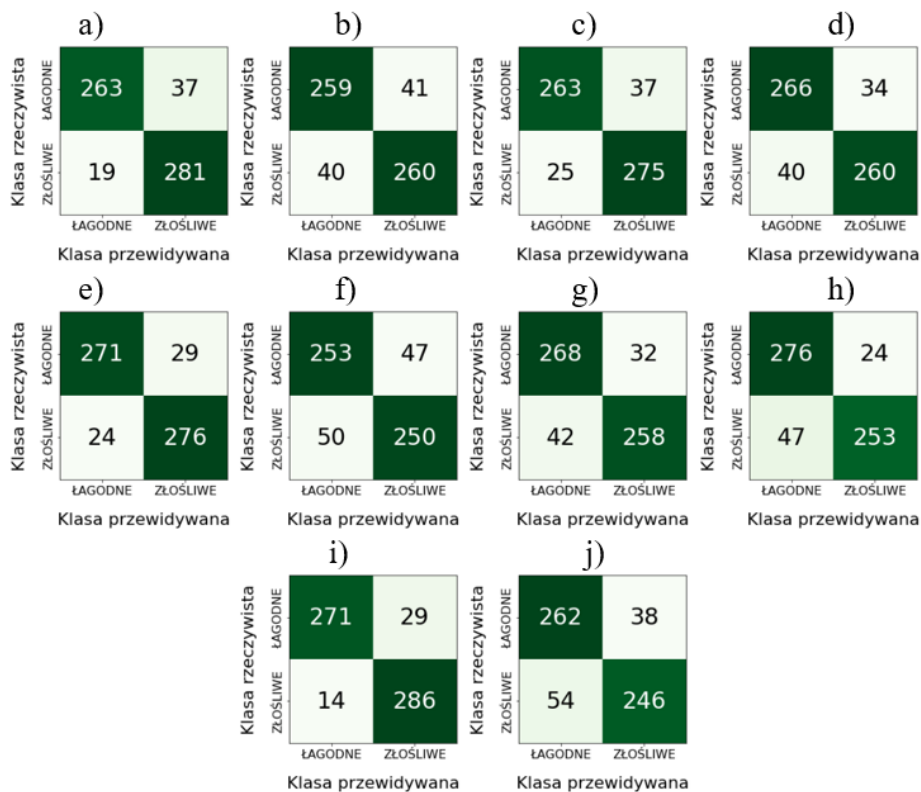


Rys. 8.3. Krzywe ROC wraz z parametrami AUC modeli klasyfikacji binarnej (ZNAMIE – CZERNIAK)

W kolejnym etapie ewaluacji z wykorzystaniem zbioru ISIC przeprowadzono testowanie klasyfikatorów binarnych służących do rozróżniania zmian łagodnych i złośliwych.

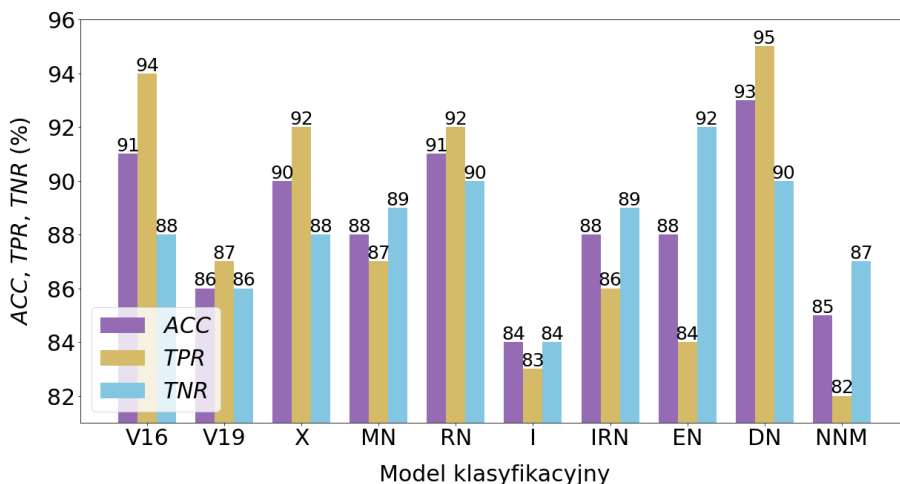
Na rysunku 8.4 przedstawiono macierze pomyłek poszczególnych modeli. Wartości w nich zawarte odpowiadają kolejno parametrom TN, FP, FN oraz TP. Niższym wartościom wyżej wymienionych parametrów odpowiadają jaśniejsze kolory. Na podstawie niskiej wartości liczby pomyłek klasyfikator określa się jako skuteczny. Model DenseNet201 pomylił się tylko 29 razy rozpoznając znamie łagodne jako złośliwe i tylko 14 razy pomylił się diagnozując znamie złośliwe, jako łagodne. Wśród modeli sieci o niskiej liczbie pomyłek należy wymienić także: VGG16, Xception i ResNet50. Najwięcej pomyłek w klasyfikacji obrazów testowych popełnił model InceptionV3, ponieważ aż 47 łagodnych zmian zakwalifikował, jako złośliwe i 50 zmian złośliwych, jako

łagodne. Podobne liczby pomyłek świadczą o porównywalnej skuteczności modeli.

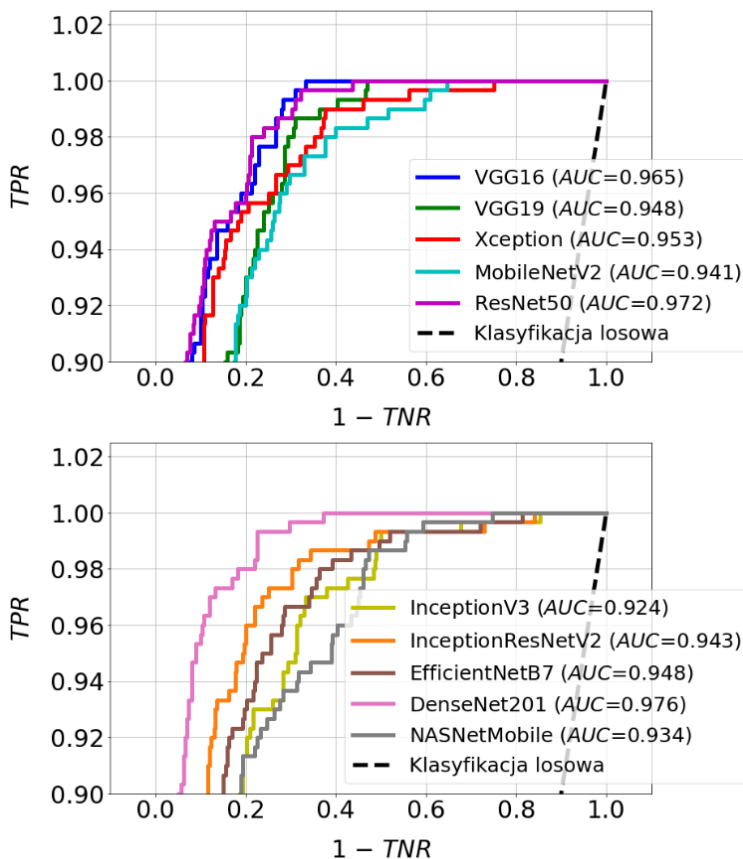


Rys. 8.4. Macierze pomyłek modeli klasyfikacji binarnej (ŁAGODNE – ZŁOŚLIWE):
a) VGG16; b) VGG19; c) Xception; d) MobileNetV2; e) ResNet50;
f) InceptionV3; g) InceptionResNetV2; h) EfficientNetB7; i) DenseNet201;
j) NASNetMobile)

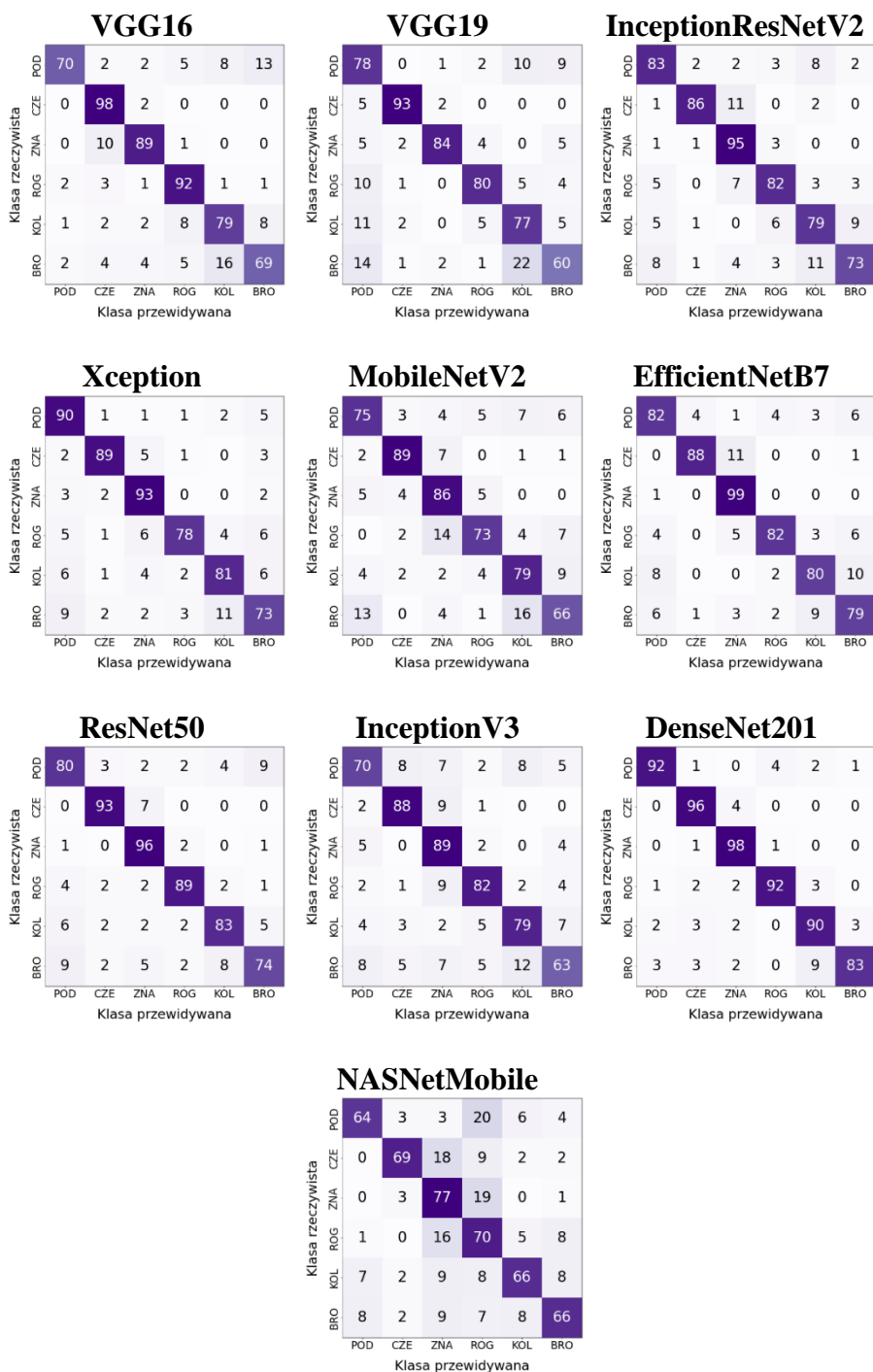
Obliczono także dokładność (ACC) klasyfikacji binarnej (ŁAGODNE – ZŁOŚLIWE), czułość (TPR), swoistość (TNR) i pole pod krzywą ROC (wskaźnik AUC) (rysunku 8.5. oraz rysunku 8.6.). Zdecydowanie najskuteczniejsze okazały się modele: DenseNet201 (ACC = 93%), VGG16 i ResNet50 (ACC = 91%) oraz Xception (ACC = 90%). Najwyższą czułość, równą 95% uzyskano dla modelu sieci DenseNet201 oraz 94% dla modelu VGG16. Nieco niższą czułość osiągnęły modele Xception i ResNet50 (TPR = 92%). Z kolei, najniższą wartość TPR (82%) ma model NASNetMobile.



Rys. 8.5. Wyniki testowania modeli klasyfikacji binarnej (ŁAGODNE – ZŁOŚLIWE)



Rys. 8.6. Krzywe ROC wraz z parametrami AUC modeli klasyfikacji binarnej (ŁAGODNE – ZŁOŚLIWE)



Rys. 8.7. Macierze pomyłek modeli klasyfikacji wieloklasowej

Krzywe ROC wraz z parametrami AUC modeli klasyfikacji binarnej (ŁAGODNE – ZŁOŚLIWE) przedstawiono na rysunku 8.6. Wartości parametrów AUC były dla wymienionych wyżej modeli wyższe i wynosiły – 0,976 dla DenseNet201, 0,972 dla ResNet50 i 0,965 dla modelu VGG16. Najniższą wartość wskaźnika AUC uzyskały modele: NASNetMobile (AUC = 0,934) i InceptionV3 (ACU = 0,924).

Wyniki testowania modeli klasyfikacji wieloklasowej zaprezentowano na rysunku 8.7. W tym przypadku, wykorzystano sześć klas obejmujących obrazy zmian skórnych złośliwych – rak kolczystokomórkowy (KOL), czerniak złośliwy (CZE), rak podstawnokomórkowy (POD) i łagodnych – znamię melanocytowe (ZNA), łagodne rogowacenie barwnikowe (ROG) i brodawka łojotokowa (BRO). Najmniej pomyłek w klasyfikacji obrazów testowych popełniła sieć DenseNet201, uzyskała ona aż 551 prawidłowo zdiagnozowanych przypadków. Sieć EfficientNetB7 prawidłowo zdiagnozowała 510 zdjęć. Natomiast najwięcej pomyłek popełniła sieć NASNetMobile – prawidłowo sklasyfikowała tylko 412 zdjęć.

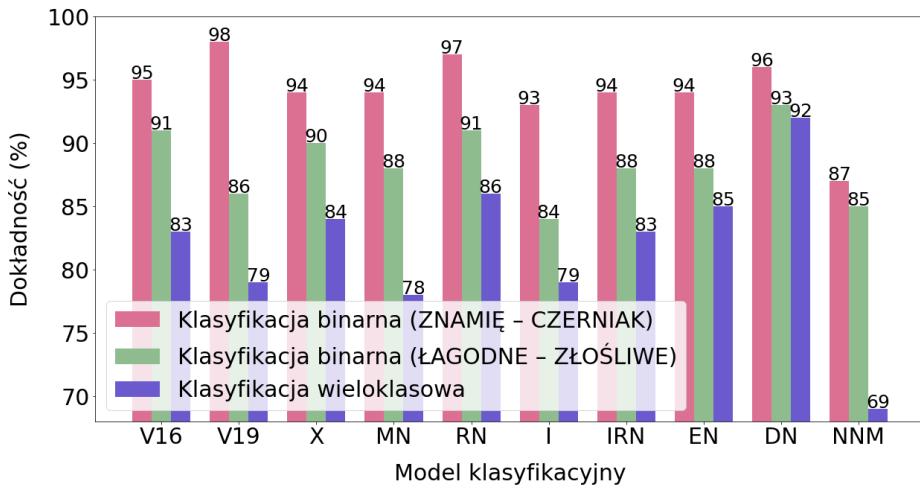
W ramach ewaluacji obliczono dokładność klasyfikacji wieloklasowej, precyzję, pewność, wskaźnik F1 i stratę (tabela 8.1). Skuteczność zbudowanych modeli okazała się zróżnicowana. Najwyższą dokładność, równą 92% uzyskano dla modelu sieci DenseNet201, natomiast 86% uzyskano dla modelu ResNet50. Najwyższą wartość pełności, równą 0,99 uzyskała sieć EfficientNetB7 dla klasyfikacji znamienia (ZNA). Najwyższa wartość precyzji, równa 0,95 została uzyskana dla sieci EfficientNetB7 i Inception-ResNetV2 (czerniak), DenseNet201 (łagodne rogowacenie łojotokowe i brodawka łojotokowa). Wskaźnik F1 najwyższą wartość (0,94) uzyskał dla sieci DenseNet201, przy diagnostyce znamienia.

Tabela 8.1. Wyniki testowania modeli klasyfikacji wieloklasowej

Model	Zmiana	Precyzja	Pełność	Wskaźnik F1	Dokładność	Strata
VGG16	POD	0,93	0,70	0,80	0,83	0,65
	CZE	0,82	0,98	0,89		
	ZNA	0,89	0,89	0,89		
	ROG	0,83	0,92	0,87		
	KOL	0,76	0,79	0,77		
	BRO	0,76	0,69	0,72		
VGG19	POD	0,63	0,78	0,70	0,79	0,68
	CZE	0,94	0,93	0,93		
	ZNA	0,94	0,84	0,89		
	ROG	0,87	0,80	0,83		
	KOL	0,68	0,77	0,72		
	BRO	0,72	0,60	0,66		

Model	Zmiana	Precyzja	Pełność	Wskaźnik F1	Dokładność	Strata
Xception	POD	0,78	0,90	0,84	0,84	0,52
	CZE	0,93	0,89	0,91		
	ZNA	0,84	0,93	0,88		
	ROG	0,92	0,78	0,84		
	KOL	0,83	0,81	0,82		
MobileNetV2	BRO	0,77	0,73	0,75	0,78	0,67
	POD	0,76	0,75	0,75		
	CZE	0,89	0,89	0,89		
	ZNA	0,74	0,86	0,79		
	ROG	0,83	0,73	0,78		
ResNet50	KOL	0,74	0,79	0,76	0,86	0,43
	BRO	0,74	0,66	0,70		
	POD	0,80	0,80	0,80		
	CZE	0,91	0,93	0,92		
	ZNA	0,84	0,96	0,90		
InceptionV3	ROG	0,92	0,89	0,90	0,79	0,64
	KOL	0,86	0,83	0,84		
	BRO	0,82	0,74	0,78		
	POD	0,77	0,70	0,73		
	CZE	0,84	0,88	0,86		
Inception-ResNetV2	ZNA	0,72	0,89	0,80	0,83	0,59
	ROG	0,85	0,82	0,83		
	KOL	0,78	0,79	0,79		
	BRO	0,76	0,63	0,69		
	POD	0,81	0,83	0,82		
EfficientNetB7	CZE	0,95	0,86	0,90	0,85	0,48
	ZNA	0,80	0,95	0,87		
	ROG	0,85	0,82	0,83		
	KOL	0,77	0,79	0,78		
	BRO	0,84	0,73	0,78		
DenseNet201	POD	0,81	0,82	0,82	0,92	0,33
	CZE	0,95	0,88	0,91		
	ZNA	0,83	0,99	0,90		
	ROG	0,91	0,82	0,86		
	KOL	0,84	0,80	0,82		
NASNetMobile	BRO	0,77	0,79	0,78	0,69	0,85
	POD	0,94	0,92	0,93		
	CZE	0,91	0,96	0,93		
	ZNA	0,91	0,98	0,94		
	ROG	0,95	0,92	0,93		
NASNetMobile	KOL	0,87	0,90	0,88	0,69	0,85
	BRO	0,95	0,83	0,89		
	POD	0,80	0,64	0,71		
	CZE	0,87	0,69	0,77		
	ZNA	0,58	0,77	0,66		
NASNetMobile	ROG	0,53	0,70	0,60	0,69	0,85
	KOL	0,76	0,66	0,71		
NASNetMobile	BRO	0,74	0,66	0,70	0,69	0,85
	POD	0,80	0,64	0,71		

Na rysunku 8.8 przedstawiono wyniki testowania modeli dla różnych rodzajów klasyfikacji. Największą dokładnością odznacza się klasyfikacja binarna (ZNAMIĘ – CZERNIAK). Siecią, która najlepiej poradziła sobie z tym rodzajem klasyfikacji była VGG19 (98%) oraz ResNetV2 (97%). Na drugim miejscu znajduje się klasyfikacja binarna (ŁAGODNE – ZŁOŚLIWE). Wysoką dokładnością, na poziomie 93%, wykazała się w tym przypadku sieć DenseNet201 oraz ResNetV2 i VGG16 (po 91%). Natomiast najniższą dokładnością charakteryzuje się klasyfikacja wieloklasowa. Najskuteczniejsza okazała się w tym przypadku sieć DenseNet201 (92%).

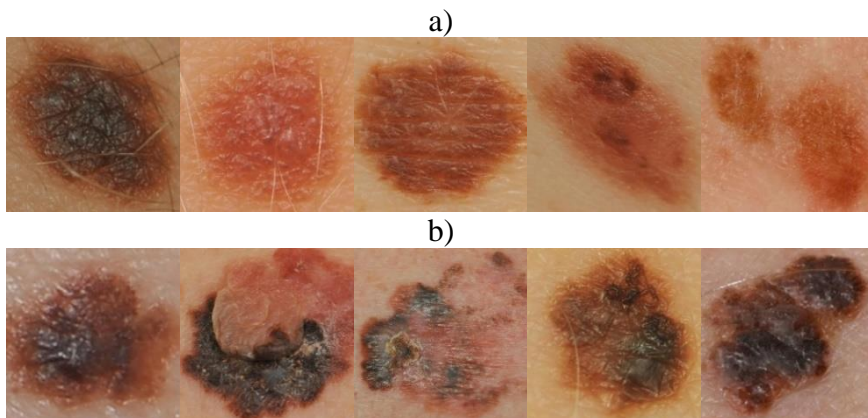


Rys. 8.8. Wyniki testowania modeli dla różnych rodzajów klasyfikacji

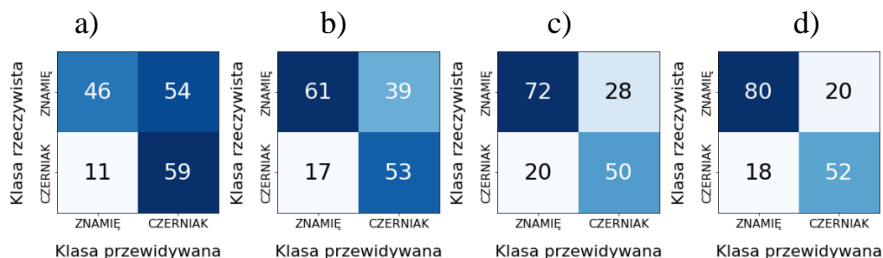
8.2. Testowanie modeli z wykorzystaniem zbioru MED-NODE

Cztery modele klasyfikacji binarnej typu (ZNAMIĘ – CZERNIAK), które okazały się najskuteczniejsze podczas testowania z wykorzystaniem zbioru ISIC, wybrano do dalszej ewaluacji za pomocą zbioru MED-NODE [68]. Na rysunku 8.9. przedstawiono przykłady obrazów z bazy znamion skórnych i czerniaków. Do testowania wykorzystano 100 obrazów znamion i 70 obrazów czerniaka.

Zbudowano macierze pomyłek dla czterech testowanych modeli (rysunek 8.10). Najmniej pomyłek zostało zarejestrowanych dla modelu DenseNet201, a najwięcej dla VGG16.

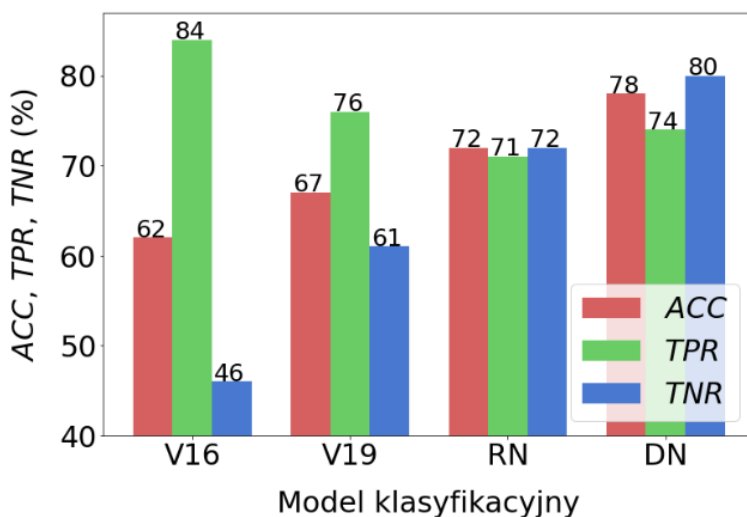


Rys. 8.9. Przykładowe obrazy zmian skórnych ze zbioru MED-NODE: a) znamię; b) czerniak

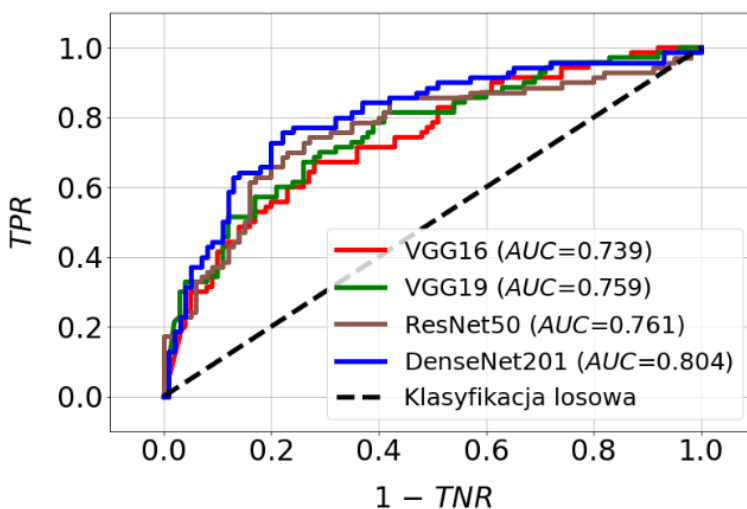


Rys. 8.10. Macierze pomylek modeli klasyfikacji binarnej (ZNAMIE – CZERNIAK): a) VGG16; b) VGG19; c) ResNet50; d) DenseNet201

Wyniki ewaluacji zaprezentowano na rysunku 8.11. Najskuteczniejsze okazały się modele DenseNet201 i ResNet50, dla których wartości parametrów ACC osiągnęły poziom odpowiednio 78% i 72%. Najmniej skuteczny okazał się model VGG16 (ACC = 62%). Zdecydowanie najdokładniejszy okazał się model DenseNet201, dla którego ACC=78%, TPR=74%, TNR=80%. Krzywe ROC przedstawione na rysunku 8.12 potwierdzają wysoką skuteczność modelu DenseNet201 (AUC = 0,804). Dla pozostałych modeli obszar pod krzywą ROC jest mniejszy i zmienia się w zakresie $0,739 \pm 0,761$.



Rys. 8.11. Wyniki testowania modeli klasyfikacji binarnej ZNAMIEŃ – CZERNAK

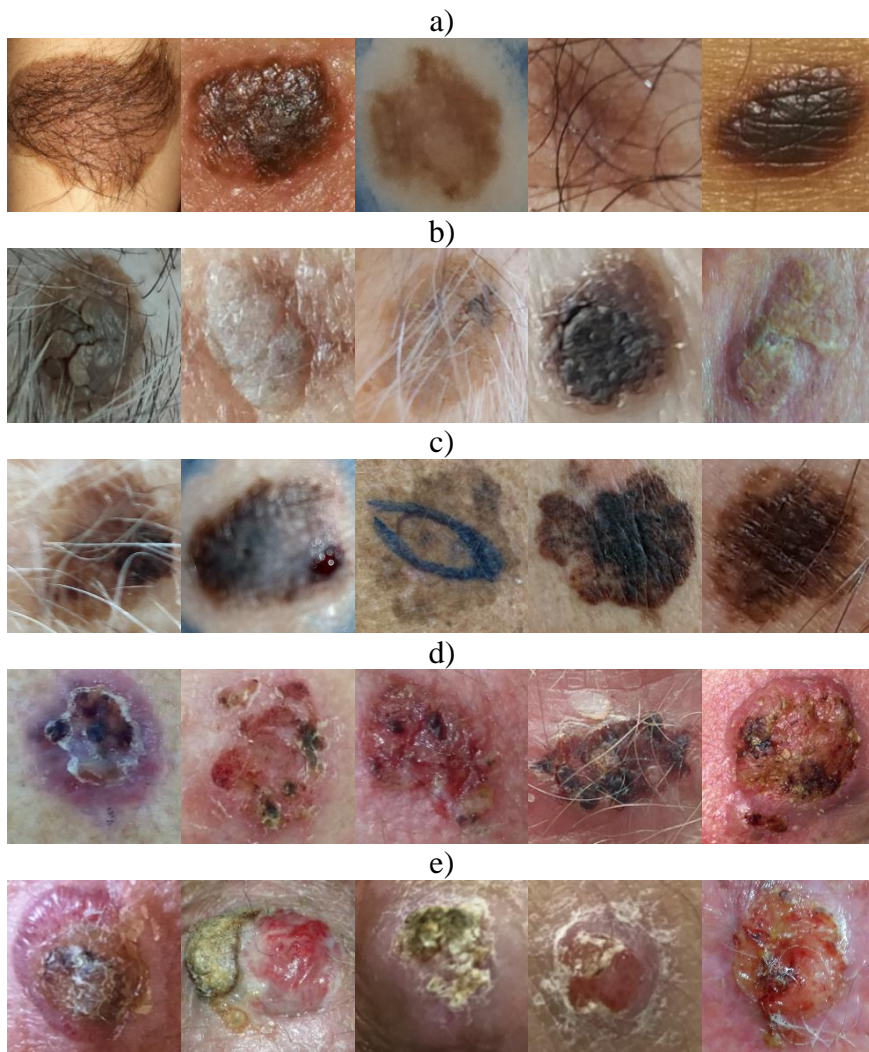


Rys. 8.12. Krzywe ROC wraz z parametrami AUC modeli klasyfikacji binarnej ZNAMIEŃ – CZERNAK

8.3. Testowanie modeli z wykorzystaniem zbioru PAD-UFES-20

W kolejnym etapie ewaluacji wykorzystano obrazy ze zbioru PAD-UFES-20 [69]. Na rysunku 8.13. przedstawiono przykłady obrazów znajdujących się we wspomnianej bazie. Na potrzeby ewaluacji, z bazy wykorzystano 130 obrazów

dla klasyfikacji binarnej (ZNAMIĘ – CZERNIAK). Cały zbiór dla tego wariantu testowania zawierał 80 obrazów znamion oraz 50 obrazów czerniaków.

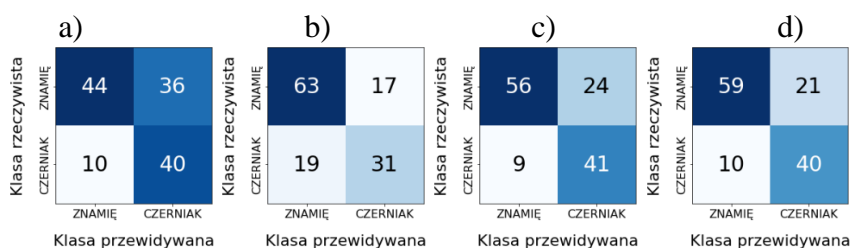


Rys. 8.13. Przykładowe obrazy zmian skórnych ze zbioru PAD-UFES-20: a) znamię; b) brodawka łojotokowa; c) czerniak; d) rak podstawokomórkowy; e) rak kolczystokomórkowy

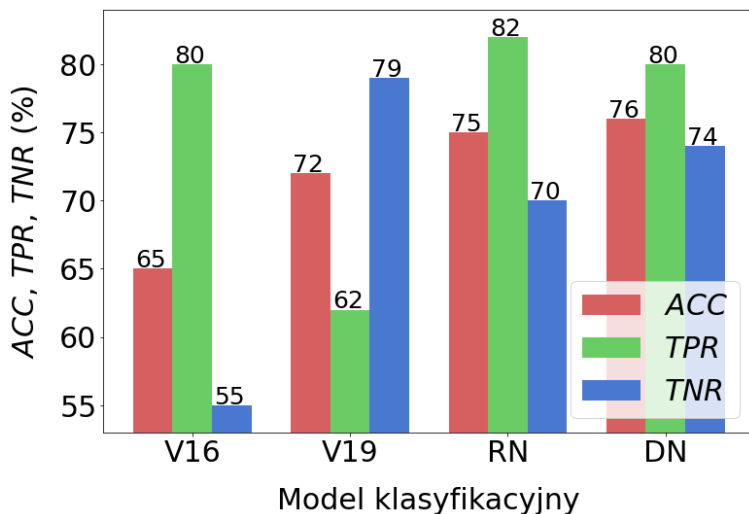
Podczas testowania, wykorzystano te same modele klasyfikacji binarnej (ZNAMIĘ – CZERNIAK), co dla zbioru MED-NODE. Rysunek 8.14 przedstawia macierze pomyłek uzyskane w wyniku ewaluacji modeli. Najmniejszą łączną liczbę pomyłek obserwujemy dla modelu DenseNet201 – tylko 31 pomyłek (10 czerniaków zaklasyfikowano jako znamiona i 21 znamion zakwalifikowano jako czerniaki). Z kolei, model ResNet50 pomylił się

w klasyfikacji nieco mniej, niż VGG19. Natomiast, najwięcej pomyłek zostało popełnionych przez model VGG16.

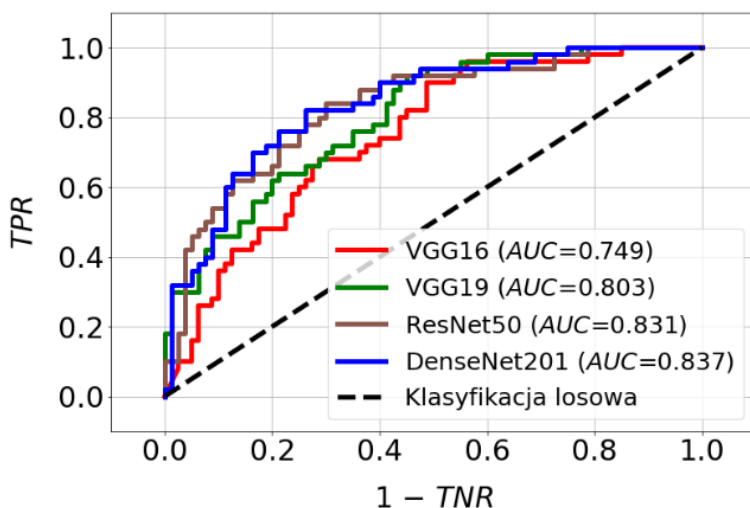
Podczas ewaluacji obliczono dokładność klasyfikacji (ACC), czułość (TPR), swoistość (TNR) i pole pod krzywą ROC (wskaźnik AUC). Wartości tych miar jakości zaprezentowano na rysunku 8.15. oraz rysunku 8.16. Najwyższą dokładność (ACC = 76%) uzyskano dla modelu DenseNet201. Wysoką dokładność osiągnęły także modele: ResNet50 (ACC = 75%) i VGG19 (ACC = 72%). Zdecydowanie najniższą wartość dokładności uzyskano dla modelu VGG16 (ACC = 65%). Należy zwrócić uwagę na wysoką czułość modelu ResNet50 (TPR = 82%) oraz bardzo dobrą specyficzność osiągniętą przez model VGG19 (TNR = 79%). Model VGG16 ma najniższą wartość specyficzności (TNR = 55%), natomiast najniższą wartość czułości uzyskał model VGG19 (TPR = 62%) (rysunku 8.15).



Rys. 8.14. Macierze pomyłek modeli klasyfikacji binarnej (ZNAMIE – CZERNIAK): a) VGG16; b) VGG19; c) ResNet50; d) DenseNet201



Rys. 8.15. Wyniki testowania modeli klasyfikacji binarnej (ZNAMIE – CZERNIAK)



Rys. 8.16. Krzywe ROC wraz z parametrami AUC modeli klasyfikacji binarnej (ZNAMIĘ – CZERNIAK)

Krzywe ROC testowanych modeli, przedstawione na rysunku 8.16, potwierdzają wcześniejszą ocenę. Wartości parametru AUC były dla dwóch wymienionych wyżej modeli wyższe i wynosiły – 0,837 dla DenseNet201 i 0,831 dla modelu ResNet50. Najniższe wartości wskaźnika AUC uzyskały modele VGG19 (AUC = 0,803) i VGG16 (AUC = 0,749).

Zbiór PAD-UFES-20 wykorzystano także w kolejnym etapie ewaluacji, przy czym tym razem przetestowano modele klasyfikacji binarnej typu (ŁAGODNE – ZŁOŚLIWE). Modele zbudowano przy użyciu tych samych architektur sieci, co te wykorzystane we wcześniejszym eksperymencie, to znaczy: VGG16, VGG19, ResNet50 i DenceNet201. Niżej podano liczebność obserwacji należących do zbioru testowego.

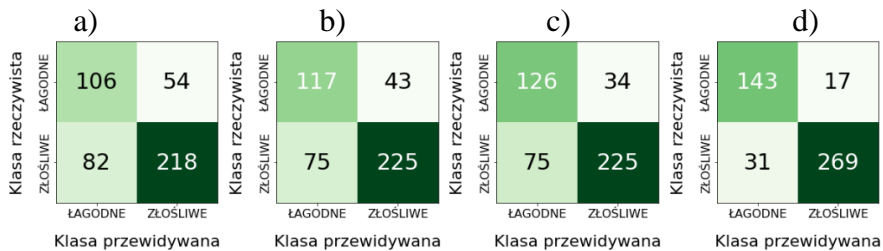
Zmiany łagodne:

- znamię – 80;
- brodawka łojotokowa – 80.

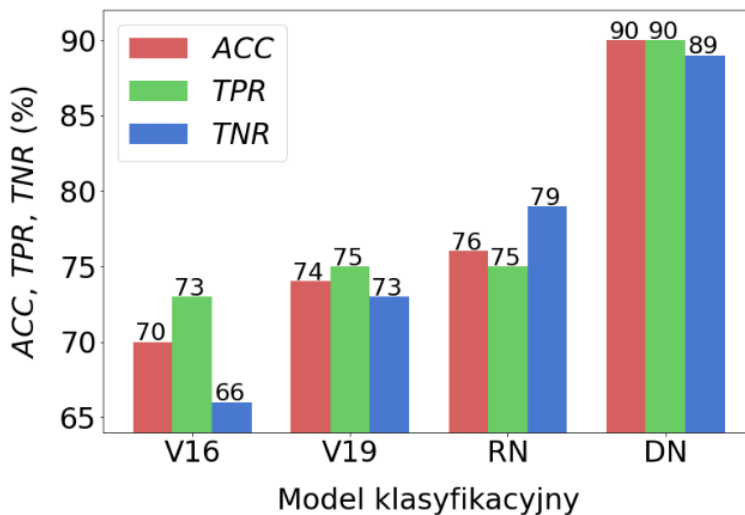
Zmiany złośliwe:

- rak koleczystokomórkowy – 100;
- rak podstawnokomórkowy – 200.

Na rysunku 8.17 przedstawiono macierze pomyłek. Najmniejszą łączną liczbę błędów obserwujemy dla modelu DenseNet201 – tylko 48 pomyłek (31 zmian złośliwych zaklasyfikowano jako zmiany łagodne i 17 zmian łagodnych zakwalifikowano jako złośliwe). Dla modelu ResNet50 zarejestrowano nieco mniejszą liczbę pomyłek, niż dla modelu VGG19. Natomiast, model VGG16 popełnił najwięcej pomyłek – aż 136 obrazów zostało błędnie zdiagnozowanych.

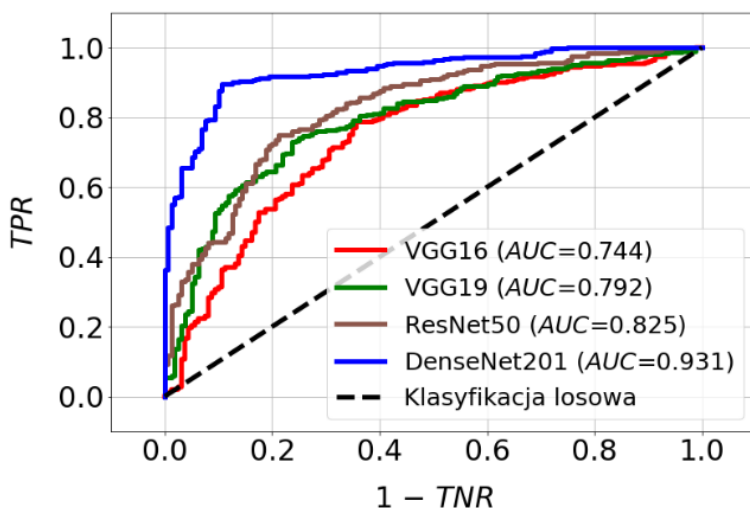


Rys. 8.17. Macierze pomyłek modeli klasyfikacji binarnej (ŁAGODNE – ZŁOŚLIWE): a) VGG16; b) VGG19; c) ResNet50; d) DenseNet201)



Rys. 8.18. Wyniki testowania modeli klasyfikacji binarnej (ŁAGODNE – ZŁOŚLIWE))

Wyniki ewaluacji przedstawiono na rysunku 8.18. Zdecydowanie największe wartości miar jakości osiągnął model DenseNet201 (ACC = 90%, TPR = 90%, TNR = 89%). Mniej skuteczne okazały się modele ResNet50 i VGG19 – wartości parametrów ACC były równe w ich przypadku odpowiednio 76% i 74%. Z kolei, najmniej skuteczny okazał się model VGG16 (ACC = 70%). Krzywe ROC przedstawione na rysunku 8.19 potwierdzają wysoką skuteczność modelu DenseNet201 (AUC = 0,931). Dla pozostałych modeli obszar pod krzywą ROC jest zdecydowanie mniejszy i zmienia się w zakresie 0,744÷0.825.



Rys. 8.19. Krzywe ROC wraz z parametrami AUC modeli klasyfikacji binarnej (ŁAGODNE – ZŁOŚLIWE)

8.4. Testowanie modeli z wykorzystaniem zbioru PH²

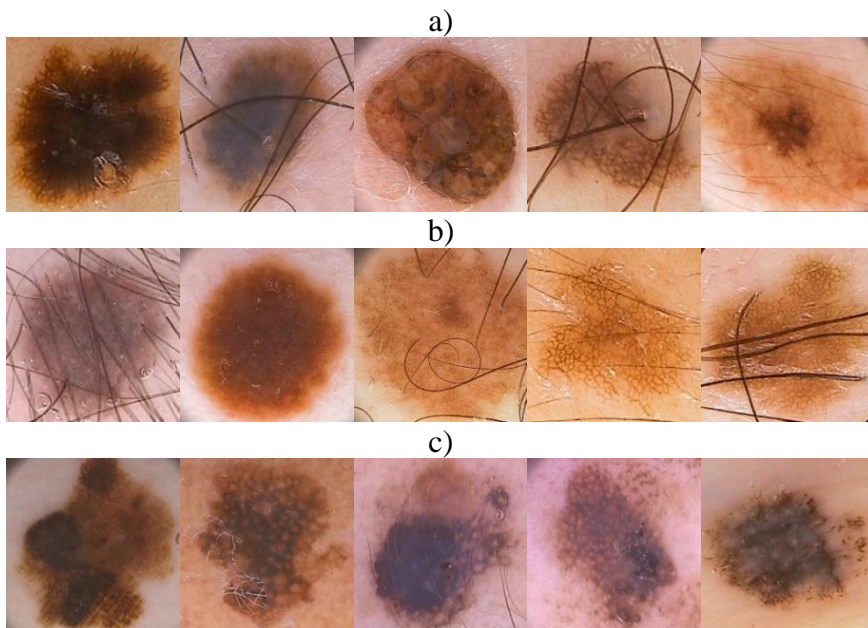
W kolejnym etapie ewaluacji, wykonano testowanie modeli VGG16, VGG19, ResNet50 i DenseNet201 z wykorzystaniem zbioru obrazów zawartych w bazie PH² [67]. Wspomniane modele wykorzystano do dwóch rodzajów klasyfikacji binarnej – (ZNAMIEŃ NORMALNE – CZERNIAK) oraz (ZNAMIEŃ NIETYPOWE – CZERNIAK). Na rysunku 8.20 znajdują się przykłady obrazów zmian skórnych pochodzących ze wspomnianej bazy PH².

Zastosowano następującą liczebność obserwacji zbioru testowego:

- znamień normalne – 80;
- znamień nietypowe – 80;
- czerniak – 40.

Rysunek 8.21 przedstawia macierze pomyłek dla klasyfikacji (ZNAMIEŃ NORMALNE – CZERNIAK). Najmniejszą łączną liczbę pomyłek obserwujemy dla modelu DenseNet201 – tylko 31 błędów (7 czerniaków błędnie zaklasyfikowano jako znamiona normalne i 24 znamion normalnych błędnie zdiagnozowano jako czerniaki). Modele ResNet50 i VGG19 pomyliły się w klasyfikacji tylko 34 razy. Natomiast, model VGG16 popełnił najwięcej błędów – 36 obrazów zostało błędnie zdiagnozowanych.

Zbudowano także macierze pomyłek dla klasyfikacji (ZNAMIEŃ NIETYPOWE – CZERNIAK) (rysunek 8.22). W tym przypadku, najmniej pomyłek (tylko 15) zostało zarejestrowanych dla modeli DenseNet201 i VGG19, a najwięcej (19) dla modelu ResNet50.



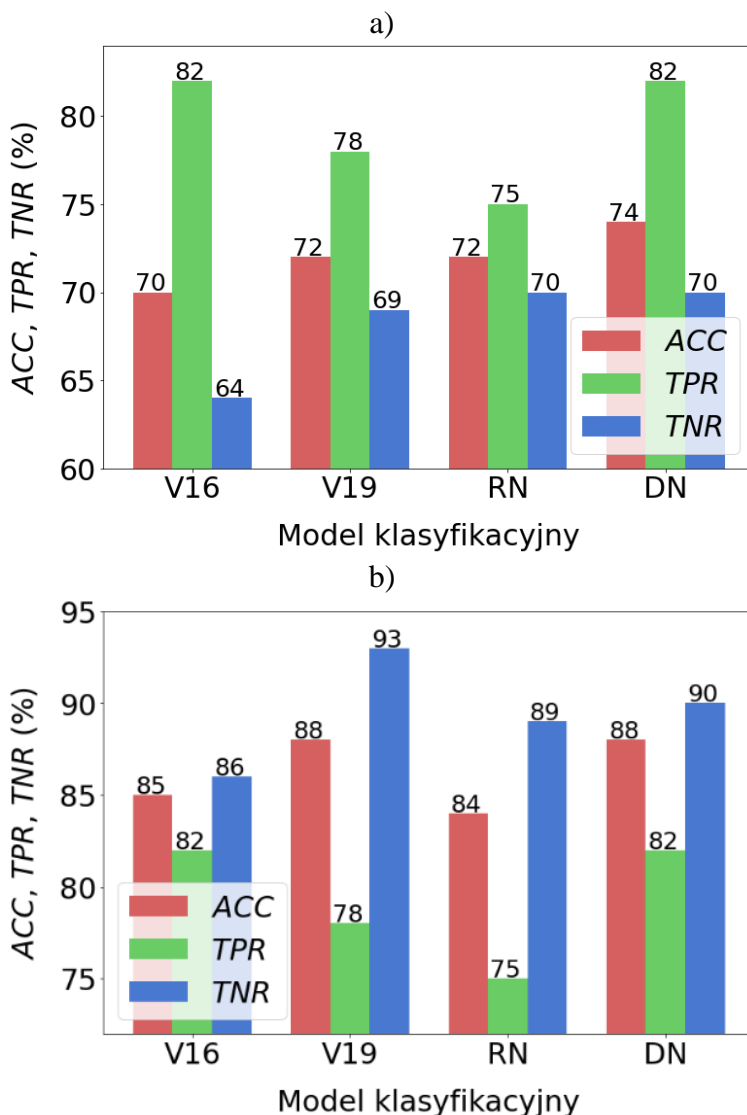
Rys. 8.20. Przykładowe obrazy zmian skórnych ze zbioru PH²: a) znamię normalne; b) znamię nietypowe; c) czerniak

	a)	b)	c)	d)				
Klasa rzeczywista	ZNAMIE NORMALNE	51	29	56	24			
	CZERNIAK	7	33	10	30			
	ZNAMIE NORMALNE	55	25	56	24			
	CZERNIAK	9	31	10	30			
	Klasa przewidywana		Klasa przewidywana		Klasa przewidywana		Klasa przewidywana	

Rys. 8.21. Macierze pomyłek modeli klasyfikacji binarnej (ZNAMIE NORMALNE – CZERNIAK): a) VGG16; b) VGG19; c) ResNet50; d) DenseNet201

	a)	b)	c)	d)				
Klasa rzeczywista	ZNAMIE NIETYPOWE	69	11	71	9			
	CZERNIAK	7	33	10	30			
	ZNAMIE NIETYPOWE	74	6	71	9			
	CZERNIAK	9	31	10	30			
	Klasa przewidywana		Klasa przewidywana		Klasa przewidywana		Klasa przewidywana	

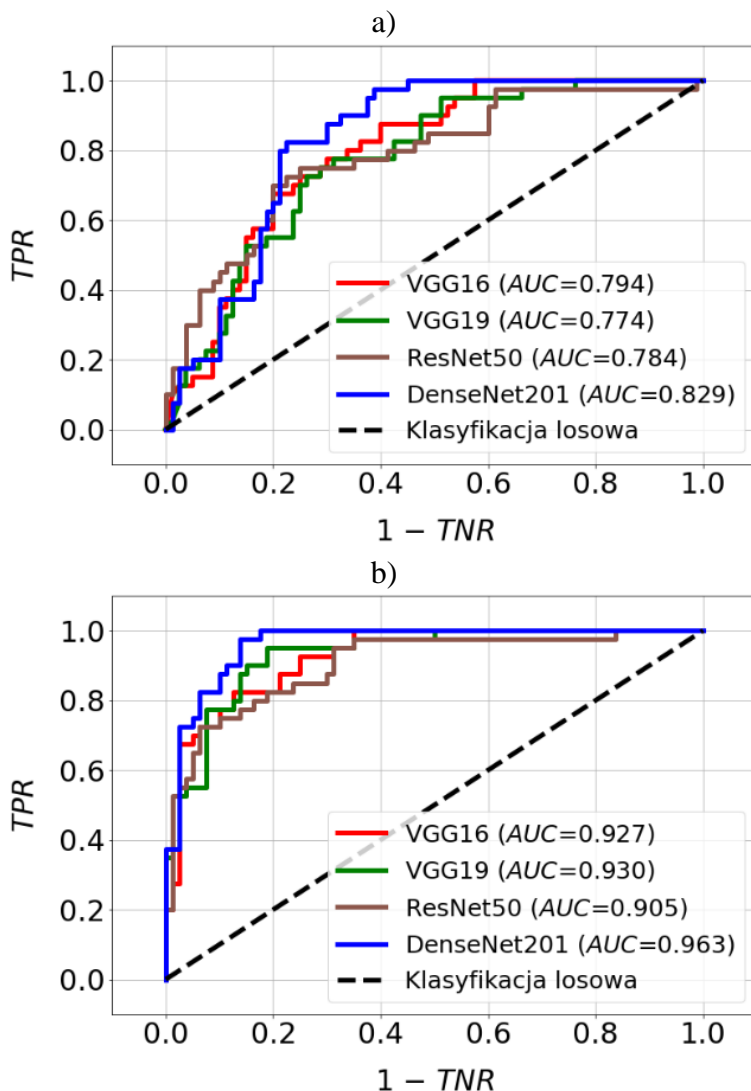
Rys. 8.22. Macierze pomyłek modeli klasyfikacji binarnej (ZNAMIE NIETYPOWE – CZERNIAK): a) VGG16; b) VGG19; c) ResNet50; d) DenseNet201



Rys. 8.23. Wyniki testowania modeli klasyfikacji binarnej: a) ZNAMIEŃ NORMALNE – CZERNIAK; b) ZNAMIEŃ NIETYPOWE – CZERNIAK

Wyniki testowania modeli klasyfikacji binarnej (ZNAMIEŃ NORMALNE – CZERNIAK) zaprezentowano na rysunku 8.23a). Najskuteczniejszy okazał się model DenseNet201 (ACC = 74%). Pozostałe modele, tj. ResNet50 i VGG19 uzyskały wartość parametru ACC na poziomie 72%. Najmniej skuteczny okazał się model VGG16 (ACC = 70%). Na uwagę zasługuje fakt, iż modele VGG16 i DenseNet201 osiągnęły najwyższą wartość czułości (TPR = 82%). Ponadto, dla wszystkich modeli wskaźnik TPR osiągał wyższą wartość niż TNR. Na rysunku 8.23b przedstawiono wyniki testowania modeli dla klasyfikacji binarnej

(ZNAMIĘ NIETYPOWE – CZERNIĄK). W tym przypadku, najwyższą dokładność (ACC = 88%) otrzymano dla modeli DenseNet201 i VGG19. Z kolei, najniższą dokładność uzyskał model ResNet50 (ACC = 84%). W tym eksperymencie, odwrotnie niż w poprzednim, wskaźnik TPR był dla każdego modelu niższy, niż TNR. Należy zauważyć, iż wskaźnik TNR osiągnął największą wartość, równą 93%, dla modelu VGG19.



Rys. 8.24. Krzywe ROC wraz z parametrami AUC modeli klasyfikacji binarnej:
a) ZNAMIE NORMALNE – CZERNIĄK; b) ZNAMIE NIETYPOWE – CZERNIĄK

Krzywe ROC przedstawione zostały na rysunku 8.24. Należy zauważyć, iż obszar pod krzywą ROC jest znacznie mniejszy dla modeli do klasyfikacji ZNAMIEŃ NORMALNE – CZERNIAK, niż dla modeli do klasyfikacji ZNAMIEŃ NIETYPOWE – CZERNIAK. Wysoką skuteczność modelu DenseNet201 (klasyfikacja ZNAMIEŃ NORMALNE – CZERNIAK) potwierdza wartość AUC, równa 0,829. Dla pozostałych modeli, parametr AUC jest mniejszy i zmienia się w zakresie $0,774 \div 0,794$. Zdecydowanie wyższe wartości parametru AUC otrzymano dla modeli użytych podczas klasyfikacji binarnej ZNAMIEŃ NIETYPOWE – CZERNIAK (rysunek 8.24b). Najwyższą wartość tego parametru ($AUC = 0,963$) otrzymano dla modelu DenseNet201. Dla pozostałych modeli, obszar pod krzywą ROC jest mniejszy i zmienia się w zakresie $0,905 \div 0,930$.

8.5. Testowanie modeli z wykorzystaniem kombinacji zbiorów PAD-UFES-20 i PH²

W ostatnim etapie walidacji wykorzystano obrazy należące do zbiorów PAD-UFES-20 oraz PH². Zbiór testowy składał się z pewnej liczby obrazów należących do obu wymienionych baz. Liczebność tego zbioru była następująca:
Zbiór PH²:

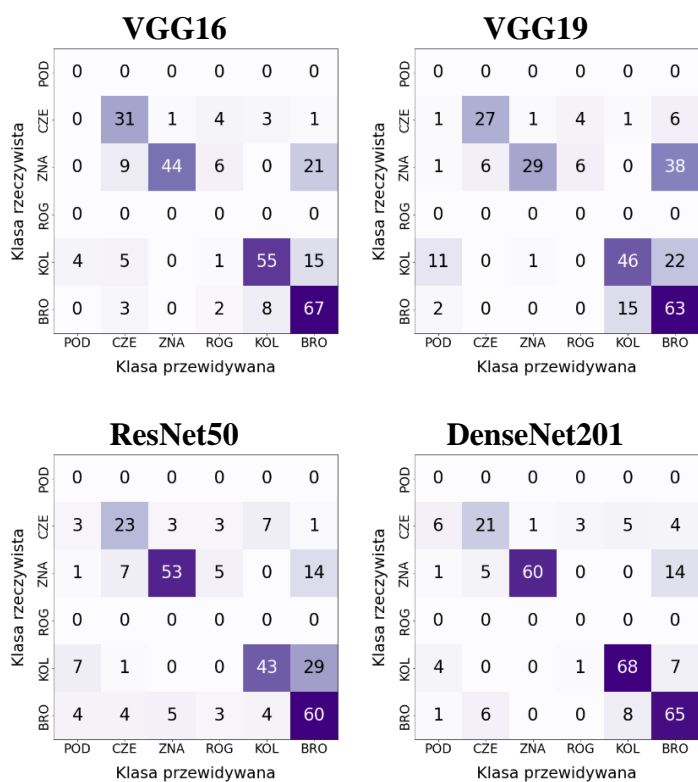
- czerniak – 40;
- znamień nietypowe – 80.

Zbiór PAD-UFES-20:

- rak kolczystokomórkowy – 80;
- brodawka łojotokowa – 80.

W ramach ewaluacji przeprowadzono testowanie modeli VGG16, VGG19, ResNet50 i DenseNet201 dla klasyfikacji wieloklasowej. Wyniki tego eksperymentu, w postaci macierzy pomyłek, przedstawiono na rysunku 8.25. Należy w tym miejscu wyjaśnić, że w macierzy pomyłek widnieje sześć klas, ponieważ modele zbudowano na podstawie zbioru ISIC, w którym zawarte były obrazy należące do wszystkich sześciu klas. Obejmowały one zmiany złośliwe – rak kolczystokomórkowy (KOL), czerniak złośliwy (CZE), rak podstawnokomórkowy (POD) oraz zmiany łagodne – znamień nietypowe (ZNA), łagodne rogowacenie barwnikowe (ROG), brodawka łojotokowa (BRO). Na potrzeby niniejszego eksperymentu wykorzystano cztery klasy – KOL, CZE, ZNA oraz BRO, ponieważ połączone zbiory PAD-UFES-20 i PH² zawierały obrazy należące do tych klas.

Największą liczbę prawidłowo zdiagnozowanych przypadków (214) zarejestrowano dla modelu DenseNet201. Z kolei, model VGG16 prawidłowo zdiagnozował 197 obrazów, a w przypadku modelu ResNet50 była to liczba 179. Najmniej prawidłowych klasyfikacji miało miejsce dla modelu VGG19, liczba ta wynosiła 165.



Rys. 8.25. Macierze pomyłek modeli klasyfikacji wieloklasowej

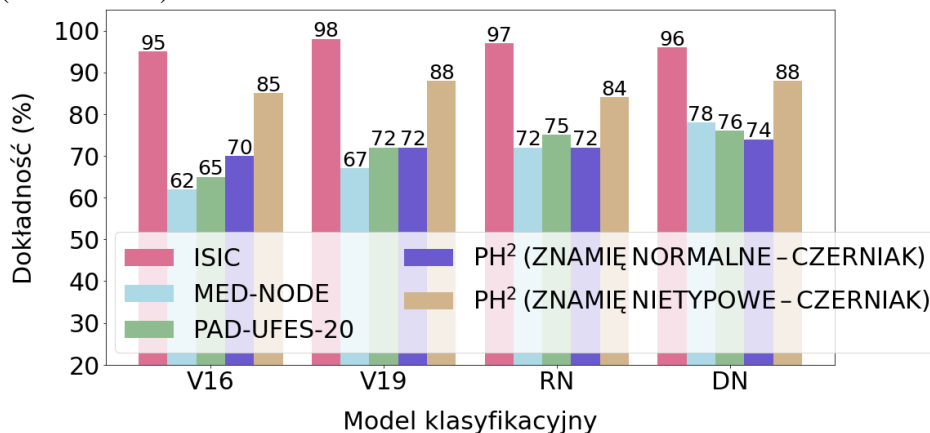
Tabela 8.2. Wyniki testowania modeli klasyfikacji wieloklasowej

Model	Zmiana	Precyzja	Pełność	Wskaźnik F1	Dokładność	Strata
VGG16	CZE	0,65	0,78	0,70	0,70	1,32
	ZNA	0,98	0,55	0,70		
	KOL	0,83	0,69	0,75		
	BRO	0,64	0,84	0,73		
VGG19	CZE	0,82	0,68	0,74	0,59	1,77
	ZNA	0,94	0,36	0,52		
	KOL	0,74	0,57	0,65		
	BRO	0,49	0,79	0,60		
ResNet50	CZE	0,66	0,57	0,61	0,64	1,22
	ZNA	0,87	0,66	0,75		
	KOL	0,80	0,54	0,64		
	BRO	0,58	0,75	0,65		
DenseNet201	CZE	0,66	0,53	0,58	0,76	0,95
	ZNA	0,98	0,75	0,85		
	KOL	0,84	0,85	0,84		
	BRO	0,72	0,81	0,76		

W wyniku testowania modeli obliczono dokładność klasyfikacji wieloklasowej, precyzję, pełność, wskaźnik F1 oraz stratę (tabela 8.2). Najwyższą dokładność uzyskano dla modeli DenseNet201 ($ACC = 76\%$), oraz VGG16 ($ACC = 70\%$). Najwyższą wartość pełności (0,85) uzyskała sieć DenseNet201 dla klasyfikacji raka kolczastokomórkowego (KOL). Najwyższa wartość precyzji to 0,98, uzyskano ją dla modeli VGG16 i DenseNet201 przy klasyfikacji znamienia (ZNA). Wskaźnik F1 osiągnął najwyższą wartość (0,85) dla modelu DenseNet201 przy diagnostyce znamienia.

8.6. Porównanie wyników ewaluacji dla różnych zbiorów danych

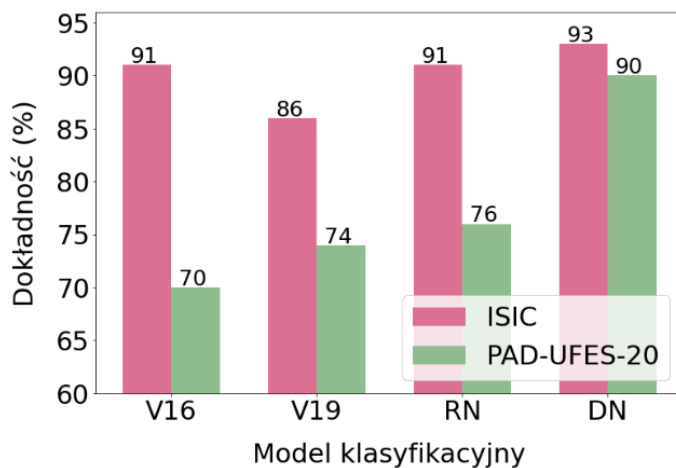
Na rysunku 8.26 porównano wyniki testowania modeli klasyfikacji binarnej (ZNAMIEŃ – CZERNIAK) dla różnych zbiorów danych. Zdecydowanie największą dokładność uzyskano dla modeli testowanych na danych pochodzących z bazy ISIC. Osiągnięto w tym przypadku dokładność równą co najmniej 95%. Najlepszy wynik uzyskał model VGG19 ($ACC = 98\%$). Jednak, uśredniając dokładność dla wszystkich pięciu zbiorów danych, najskuteczniejszy okazał się model DenseNet201 ($ACC_{sr} = 82,4\%$). Najgorsze wyniki uzyskano dla modelu VGG16 oraz danych pochodzących z bazy MED-NODE ($ACC = 62\%$).



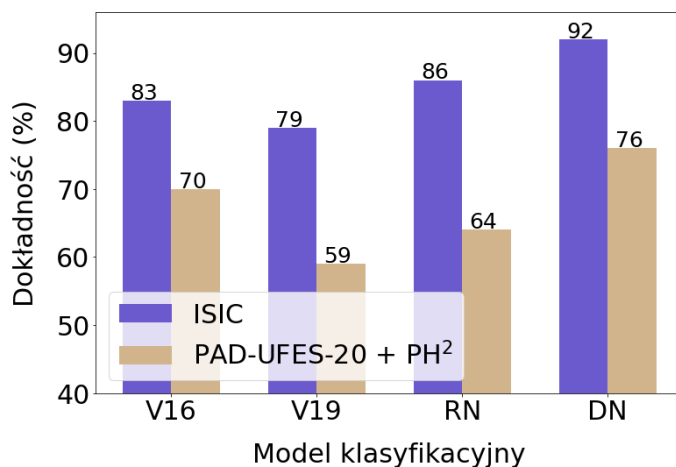
Rys. 8.26. Wyniki testowania modeli klasyfikacji binarnej (ZNAMIEŃ – CZERNIAK) dla różnych zbiorów danych

Na rysunku 8.27 przedstawiono wyniki testowania modeli klasyfikacji binarnej (ŁAGODNE – ZŁOŚLIWE) dla dwóch zbiorów danych – ISIC oraz PAD-UFES-20. Zdecydowanie największą dokładność, równą 93%, uzyskano dla modelu DenseNet201 oraz danych pochodzących z bazy ISIC. Również dla tego modelu wyniki testowania na podstawie zbioru PAD-UFES-20 były bardzo wysokie ($ACC = 90\%$). Klasyfikacja przy użyciu modelu VGG16 oraz danych pochodzących z bazy PAD-UFES-20 dała najniższą dokładność, równą 70%.

Dokładności testowania modeli dla danych z bazy ISIC znajdowały się w przedziale 86÷93%, a dla bazy PAD-UFES-20 był to zakres 70÷90%.



Rys. 8.27. Wyniki testowania modeli klasyfikacji binarnej (ŁAGODNE – ZŁOŚLIWE) dla różnych zbiorów danych



Rys. 8.28. Wyniki testowania modeli klasyfikacji wieloklasowej dla różnych zbiorów danych

Dwa zbiory danych, a mianowicie ISIC oraz PAD-UFES-20+PH² wykorzystano także do ewaluacji modeli klasyfikacji wieloklasowej. Uzyskane wyniki, w postaci ogólnej dokładności klasyfikacji, porównano na rysunku 8.28. Zdecydowanie największą dokładność, równą 92%, uzyskano dla modelu DenseNet201 oraz danych pochodzących z bazy ISIC. Również dla tego modelu wynik testowania na podstawie zbioru PAD-UFES-20+PH² był najlepszy

i wynosił 76%. Klasyfikacja przy użyciu modelu VGG19 oraz danych pochodzących ze zbioru PAD-UFES-20+PH² dała najniższą dokładność, równą 59%. Dokładność testowania modeli klasyfikacji wieloklasowej dla danych ISIC znajdowały się w przedziale 83÷92%, a dla zbioru PAD-UFES-20+PH² był to zakres 59÷76%.

9. Podsumowanie

W wyniku przeprowadzonych badań zbudowano szereg klasyfikatorów, które z punktu widzenia rozpoznawanych obrazów można podzielić na trzy grupy. Proces ewaluacji pozwolił wybrać z każdej grupy model najskuteczniejszy w kontekście dokładności rozpoznawania określonych zmian skórnych. W przypadku klasyfikacji binarnej typu ZNAMIE – CZERNIAK, najlepszy okazał się model VGG19 (ACC = 98%) testowany na podstawie zbioru ISIC. W przypadku danych, które nie były wykorzystywane w procesie uczenia, tj. PH², najskuteczniejsze okazały się modele VGG19 oraz DenseNet201, których dokładność wyniosła 88%. Jak widać, nastąpiło tutaj 10-procentowe pogorszenie dokładności, jednak pomimo tego, skuteczność tych modeli należy wciąż ocenić jako bardzo wysoką.

Drugi kierunek badań polegał na budowie modelu binarnego do klasyfikacji typu ŁAGODNE – ZŁOŚLIWE. Dla tego wariantu, najlepszy okazał się model DenseNet201. W przypadku obrazów pochodzących ze zbioru ISIC, dokładność testowania tego modelu wyniosła 93%. Nieznacznie niższą skuteczność (ACC = 90%) uzyskano także dla modelu DenseNet201 oraz obrazów należących do zbioru, który nie brał udziału w procesie uczenia, tj. PAD-UFES-20.

Trzecia grupa modeli zbudowanych w ramach badań realizowała klasyfikację wieloklasową takich zmian skórnych, jak: rak kolczystokomórkowy, czerniak złośliwy, rak podstawnokomórkowy, znamie melanocytowe, łagodne rogowacenie barwnikowe, brodawka łojotokowa. W tym przypadku, najlepszy model, który był testowany na podstawie obrazów ze zbioru ISIC (DenseNet201), osiągnął dokładność równą 92%. Znaczne pogorszenie dokładności testowania zaobserwowano dla zbioru, który nie uczestniczył w procesie uczenia, tj dla PAD-UFES-20 + PH². Najlepszy model, którym był w tym przypadku także DenseNet201, osiągnął dokładność na poziomie 76%.

Przytoczone wyniki świadczą o występowaniu pewnych różnic w wynikach ewaluacji pomiędzy zbiorem ISIC, a zbiorami, które nie uczestniczyły w procesie uczenia. Skuteczność modeli dla zbioru ISIC była w każdym przypadku wyższa, niż dla pozostałych zbiorów. Jak zaznaczono w początkowej części pracy, do budowy modeli wykorzystano zbiór ISIC, ponieważ jest to największy ogólnodostępny zbiór obrazów dermatoskopowych. Zawiera on wiele zdiagnozowanych przypadków, co pozwoliło wybrać na potrzeby niniejszych badań dostatecznie dużą liczbę obrazów reprezentujących sześć różnych zmian skórnych. Należy zaznaczyć, że wybrane obrazy nie zawierały artefaktów w postaci włosów, znaczników dermatologicznych, pęcherzyków powietrza, czy zrogowaciałego naskórka, dlatego stanowiły bardzo dobry materiał dla zbioru uczącego. W przypadku innych zbiorów danych, czyli tych, które wykorzystano tylko do testowania modeli, zrezygnowano ze wstępnej selekcji obrazów z uwagi na ich znacznie mniejszą liczebność, w porównaniu ze zbiorem ISIC. W związku z tym, obrazy te zawierały szereg artefaktów, które nie występowały w zbiorze uczącym. Oczywiście, skutkowało to niższą

dokładnością testowania, w porównaniu z obrazami należącymi do zbioru ISIC, na którym modele zostały wytrenowane.

Zaletą przeprowadzonych badań jest to, iż zbudowane modele zostały poddane ewaluacji z wykorzystaniem ogólnodostępnych zbiorów danych, takich jak ISIC, PH2 MED-NODE, PAD-UFES-20. Dzięki temu, istnieje możliwość porównania uzyskanych wyników z wynikami innych autorów. Szersze porównanie wyników otrzymanych przez autorkę z innymi dostępnymi pracami przedstawiono w podrozdziale 9.1.

Przeprowadzone badania wykazały, że odpowiednio wytrenowane głębokie konwolucyjne sieci neuronowe mogą być bazą do budowy skutecznej metody diagnostycznej służącej do rozpoznawania różnych zmian skórnych na podstawie obrazów dermatoskopowych. Wykazano także, iż zastosowanie techniki *transfer learning* pozwala zbudować na podstawie małego zbioru danych skuteczny model klasyfikacyjny do diagnostyki zmian skórnych. **Tym samym, teza rozprawy została udowodniona.**

9.1. Porównanie efektów pracy z wynikami innych autorów

W tabelach zamieszczonych w niniejszym podrozdziale przedstawiono najlepsze wyniki określające jakość klasyfikacji zmian skórnych. Wyniki te porównano uwzględniając rodzaj wykonywanej przez autorów klasyfikacji: binarna (tabela 9.1) lub wieloklasowa (tabela 9.2). Przedstawione wyniki uzyskano na podstawie tych samych lub podobnych danych, co te użyte przez autorkę. Pogrubioną czcionką zaznaczono wysokie wartości poszczególnych wskaźników jakości. W tabelach wstawiono symbol ”-,, gdy dana wartość nie została podana. W tabelach 9.1 i 9.2 podane zostały najwyższe otrzymane wartości wskaźników jakości dla danej sieci.

W tabeli 9.1 zostały porównane otrzymane przez autorkę wyniki klasyfikacji binarnej z wynikami innych autorów [7, 93, 94, 95, 96, 97, 98, 99, 100]. Zarówno w przypadku własnych, jak i innych zacytowanych badań, korzystano z podobnych ogólnodostępnych baz danych. Najczęściej były to bazy: ISIC, MED-NODE, PH². W większości przypadków, klasyfikowano zmiany łagodne, czerniaki skóry, zmiany skórne nie będące czerniakami czy znamiona złośliwe. W zestawieniu uwzględniono następujące wskaźniki: AUC, ACC, TPR, TNR. Na tym tle, wyniki uzyskane przez autorkę wydają się obiecujące.

Jedną z najwyższych wartości ACC spośród wyników uzyskanych dla klasyfikacji binarnej uzyskał model sieci VGG19, który został zaproponowany w niniejszej pracy (ACC=98%). Model ten został wytrenowany i przetestowany na zbiorze danych obrazowych ISIC i realizował klasyfikację zmiany skórnej do kategorii ZNAMIEŃ lub CZERNIAK. Dla tego rodzaju klasyfikacji, najwyższą dokładność (98,2%) uzyskano w [101] dla sieci MobileNetV2. Zbiór uczący i treningowy stanowiła tam baza ISIC 2020, z której wyłoniono 2 klasy: MELANOMA oraz BENIGN.

Najwyższą wartość wskaźnika TPR (98,6%) osiągnięto w [93] podczas klasyfikacji binarnej typu MELANOMA – NON-MELANOMA. Autorzy

trenowali i testowali model InceptionV3 na zbiorze MED-NODE. Aby dokonać klasyfikacji typu ZNAMIEŃ – CZERNIAK, autorka trenowała, m.in., sieć DenseNet201 na zbiorze ISIC i testowała ją na MED-NODE. Wartość wskaźnika TPR wyniosła wówczas 74%.

Najlepszą wartość wskaźnika AUC (0,997) autorka otrzymała również podczas klasyfikacji typu ZNAMIEŃ – CZERNIAK dla zbioru ISIC oraz sieci VGG19. Wysoki wynik AUC, równy 0,986 otrzymano także w [93] dla sieci InceptionV3 i bazy danych MED-NODE, podczas klasyfikacji MELANOMA – NON-MELANOMA. W niniejszej pracy, podczas testowania sieci DenseNet201 na bazie MED-NODE, uzyskano wartość AUC równą 0,804. Różnice w przytoczonych wyżej wynikach wynikają z faktu, iż trenowanie sieci odbywało się na innym zestawie danych, niż jej testowanie.

W tabeli 9.2 przedstawiono porównanie indeksów jakości klasyfikacji wieloklasowej (ACC, pełność, precyzja, wskaźnik F1) z wynikami innych autorów. Wyniki własne, uzyskane dla 4 klas, porównano z innymi wynikami osiągniętymi dla 3 klas obrazów. W tym przypadku, autorka uzyskała najlepsze efekty dla modelu DenseNet201, jednak wyniki te okazały się niższe od zaprezentowanych w pracach [7, 93, 95, 102]. Powodem był fakt, iż uczenie i testowanie modeli odbywało się na obrazach należących do różnych zbiorów danych. Natomiast wyniki innych autorów uzyskane były w efekcie uczenia i testowania modeli na tej samej bazie obrazów. Należy jednak zauważyć, że uzyskane przez autorkę wartości ACC=76% i pełności=80% nie odbiegają znacznie od wartości uzyskanych w pracy [7], gdzie uzyskano ACC=81,57%, pełność=75,33%. W cytowanej pracy, wskaźnik F1 uzyskał taką samą wartość (75,75%), co w niniejszych badaniach.

W tabeli 9.2 porównano także wyniki autorki uzyskane podczas klasyfikacji zmian skórnych na 6 klas z wynikami innych autorów, gdzie dokonano klasyfikacji z wykorzystaniem 7 klas [6, 7, 13, 93, 102, 103, 104, 105]. W przypadku innych autorów, w większości prac, zbiory danych uczących i treningowych pochodziły z tej samej bazy obrazów dermatoskopowych. Należy w tym miejscu podkreślić, iż autorka, podczas procesu klasyfikacji 6 różnych klas, uzyskała dla modelu DenseNet201 (zbiór danych ISIC) bardzo wysoki wynik dokładności (ACC=92%), w porównaniu z wynikami innych autorów zaprezentowanymi w tabeli 9.2. Nieco wyższą dokładność (ACC=93,2%) uzyskano dla sieci InceptionResnetV2 w pracach [13] i [103], gdzie wykorzystano zbiór HAM10000. Z kolei, mniejszą wartość ACC otrzymano w [93], [7] i [80] podczas klasyfikacji z wykorzystaniem 7 klas dla modelu ResNet50 testowanego na zbiorze ISIC-18. Należy także zauważyć, że precyzja (92,17%) uzyskana przez autorkę dla sieci DenseNet201 okazała się największa. Nieco niższy wynik (91,43%) uzyskano w pracy [93] dla sieci ResNet50. Ponadto, autorka uzyskała wartość wskaźnika F1 (91,67%), która okazała się także najwyższa spośród wyników uzyskanych dla 7 klas obserwacji.

Tabela 9.1. Porównanie wyników testowania modeli klasyfikacji binarnej

Lp.	Autor	Baza danych uczących/ Wyodrębnione klasy/ Baza danych testujących/ Rodzaj sieci	ACC %	TPR %	TNR %	AUC
1	Własne wyniki	ISIC/2 klasy (ZNAMIEŃ – CZERNIAK)/ ISIC/ VGG19	98	98	99	0,997
2	Własne wyniki	ISIC/2 klasy (ŁAGODNE – ZŁOŚLIWE)/ ISIC/ DenseNet201	93	95	90	0,976
3	Własne wyniki	ISIC/2 klasy (ZNAMIEŃ – CZERNIAK)/ MED- NODE/ DenseNet201	78	74	80	0,804
4	Własne wyniki	ISIC/2 klasy (ZNAMIEŃ – CZERNIAK)/ PAD-UFES-20/ DenseNet201	76	80	74	0,837
5	Własne wyniki	ISIC/2 klasy (ZNAMIEŃ ŁAGODNE – ZNAMIEŃ ZŁOŚLIWE)/ PAD-UFES-20/ DenseNet201	90	90	89	0,931
6	Własne wyniki	ISIC/2 klasy (ZNAMIEŃ NORMALNE – CZERNIAK)/ PH ² / DenseNet201	74	82	70	0,829
7	Własne wyniki	ISIC/2 klasy (ZNAMIEŃ NIETYPOWE – CZERNIAK)/ PH ² / VGG19	88	78	93	0,930
		DenseNet201	88	82	90	0,963
8	[7]	ISIC 2016/ 2 klasy (ZNAMIEŃ ŁAGODNE – CZERNIAK)/ ISIC 2016/ InceptionResNetV2	81,79	81,8	71,4	0,766
9	[93]	MED-NODE/2 klasy (MELANOMA – NON-MELANOMA)/ MED-NODE/ InceptionV3	97,1	98,6	96	0,986
10	[93]	PH ² klasy (MELANOMA – NON-MELANOMA)/ PH ² / InceptionV3	97,2	98,5	96	0,983
11	[94]	DermIs, Med-Node and PH ² / 2klasy (BENIGN – MELANOMA)/ DermIs, Med-Node and PH ² / ResNet-50 i DenseNet-201	95,2	92,8	96,7	0,985

Lp.	Autor	Baza danych uczących/ Wyodrębnione klasy/ Baza danych testujących/ Rodzaj sieci	ACC %	TPR %	TNR %	AUC
12	[95]	ISIC and PH ² / 2 klasy (BENIGN – MELANOMA) /ISIC 2016/ VGG16	63,5	60,1	74,1	-
13	[96]	ISIC-Archive, ISBI 2016, MED-Node, PH ² / 2 klasy (BENIGN – MELIGNAT)/ MobileNetV2 dla testowania na ISIC-Archive	85	85	85	-
		Mobile NetV2 dla testowania na ISBI 2016	83	36	95	-
		Mobile NetV2 dla testowania na MED-Node	75	76	73	-
		Mobile NetV2 dla testowania na PH ² /	72	33	92	-
14	[97]	ISIC/ 2 klasy (BENIGN – MELIGNAT)/ ISIC/ VGG16	81,33	78,66	-	-
15	[99]	ISIC/ 2 klasy (BENIGN NEVI – MELANOMA)/ ISIC/ ResNet50	-	82,3	77,9	-
16	[98]	ISIC 2019/ 2 klasy (MELANOMA – NON-MELANOMA)/ ISIC 2019/ ResNet50V2	66	86	-	-
17	[100]	ISIC 2017/2 klasy (BENIGN NEVI – MELANOMA)/ ISIC 2017/ ResNet152	87,2	82,0	92,5	0.904
18	[101]	ISIC 2020/2 klasy (BENIGN DISEASES – MELANOMA)/ ISIC 2020/ MobileNetV2	98,2	98,1	-	-

Tabela 9.2. Porównanie wyników testowania modeli klasyfikacji wieloklasowej

Lp.	Autor	Baza danych uczących/Wyodrębnione klasy/ Baza danych testujących/ Rodzaj sieci/	ACC %	Pełność %	Precyzja %	Wskaźnik F1 %
1	Własne wyniki	ISIC/ 4 klasy (rak koleczystokomórkowy, czerniak złośliwy, znamię melanocytowe, brodawka łojotokowa)/ PAD-UFES-20 i PH ² / DenseNet201	76	73,5	80	75,75
2	[93]	PH ² / 3 klasy (nietypowe, znamię melanocytowe, czerniak złośliwy)/ PH ² / ResNet50	99,8	95,64	96,33	-
3	[102]	HAM10000/ 3 klasy (normalne, rak, czerniak złośliwy)/ HAM10000/ Xception	100	-	97,07	95,53
3	[7]	ISIC-17/ 3 klasy (znamię łagodne, rogowacenie łojotokowe, czerniak złośliwy)/ ISIC-17/ ResNet50	81,57	75,33	-	75,75
5	[95]	PH ² / ISIC/ 3 klasy (rogowacenie słoneczne, powszechnie zmiany, czerniak złośliwy)/ PH ² / VGG16	95,7	89,2	-	-
6	Własne wyniki	ISIC/ 6 klas (rak koleczystokomórkowy, czerniak złośliwy, rak podstawkomórkowy, znamię melanocytowe, łagodne rogowacenie barwnikowe, brodawka łojotokowa /ISIC/ DenseNet201	92	91,83	92,17	91,67
7	[80]	ISIC-18/ 7 klas (czerniak złośliwy, znamię melanocytowe, rak podstawkomórkowy, rogowacenie słoneczne, łagodne rogowacenie barwnikowe, dermatofibroma, zmiany naczyniowe)/ ISIC-18/ ResNet50	87,1	77	78,6	-
8	[7]	ISIC-18/ 7 klas (znamię łagodne, rogowacenie łojotokowe, rak podstawkomórkowy, rogowacenie słoneczne, dermatofibroma, zmiany naczyniowe, czerniak złośliwy)/ ISIC-18/ ResNet50	89,28	81,00	-	81,28

Lp.	Autor	Baza danych uczących/Wyodrębnione klasy/ Baza danych testujących/ Rodzaj sieci/	ACC %	Pełność %	Precyzja %	Wskaźnik F1 %
9	[83]	ISIC-18/ 7 klas (rogowacenie słoneczne, rak podstawnocomórkowy, łagodne rogowacenie barwnikowe, Dermatofibroma, znamię melanocytoowe, czerniak złośliwy, zmiany naczyniowe)/ ISIC-18/ ResNet50	90	89,37	91,43	-
10	[6]	HAM10000/7 klas (rogowacenie słoneczne, rak podstawnocomórkowy, łagodne rogowacenie barwnikowe, dermatofibroma, znamię melanocytoowe, czerniak złośliwy, zmiany naczyniowe)/ HAM10000/ MobileNet	83,1	83	89	83
11	[13]	HAM10000/ 7 klas (znamię melanocytoowe, czerniak złośliwy, łagodne rogowacenie barwnikowe, rak podstawnocomórkowy, rogowacenie słoneczne, zmiany naczyniowe, dermatofibroma)/ HAM10000/ InceptionResNetV2	93,2	88	87	88
12	[103]	HAM10000/ 7 klas (znamię melanocytoowe, czerniak złośliwy Melanoma, łagodne rogowacenie barwnikowe, rogowacenie słoneczne, zmiany naczyniowe, dermatofibroma, rak podstawnocomórkowy)/ HAM10000/ InceptionResnetV2	93,2	88	87	-
13	[13]	HAM10000/ 7 klas (rogowacenie słoneczne, rak podstawnocomórkowy, łagodne rogowacenie barwnikowe, dermatofibroma, znamię melanocytoowe, czerniak złośliwy, zmiany naczyniowe)/ HAM10000/ MobileNet	83,1	83	89	-
14	[104]	HAM10000/ 7 klas (rogowacenie słoneczne, rak podstawnocomórkowy, łagodne rogowacenie barwnikowe, dermatofibroma, znamię melanocytoowe, czerniak złośliwy, zmiany naczyniowe /HAM10000/ EfficientNet B4	87,91	88	88	87
15	[105]	HAM10000/ 7 klas (rogowacenie słoneczne, rak podstawnocomórkowy, łagodne rogowacenie barwnikowe, dermatofibroma, znamię melanocytoowe, czerniak złośliwy, zmiany naczyniowe)/ HAM10000/ Xception	89,66	-	88,76	89,02

9.2. Oryginalne wyniki uzyskane w pracy

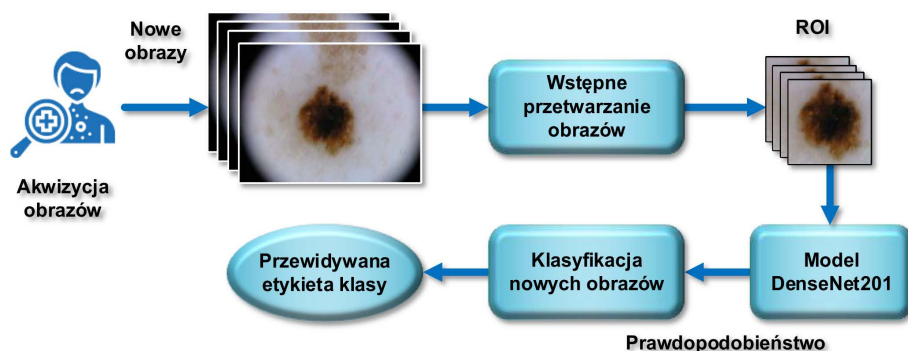
W niniejszej rozprawie przedstawiono zagadnienia dotyczące klasyfikacji znamion skórnych na podstawie obrazów dermatoskopowych z wykorzystaniem techniki *transfer learning*. Do budowy modeli wykorzystano różne architektury głębokich sieci konwolucyjnych dostępne w pakiecie Keras. Za najważniejsze oryginalne osiągnięcia pracy uważa się:

- Zgromadzenie materiału do badań pochodzącego z ogólnodostępnych repozytoriów obrazów dermatoskopowych: ISIC, PH², MED-NODE, PAD-UFES-20. Obrazy zostały dobrane pod kątem spełnienia określonych kryteriów danej choroby skórnej, tak aby cechy uwzględnione podczas klasyfikacji pozwoliły na uzyskanie jak najbardziej wiarygodnego wyniku.
- Opracowanie i implementacja w języku Python algorytmu służącego do segmentacji zmian skórnych. Zastosowanie algorytmu pozwoliło zautomatyzować etap ekstrakcji obszarów zainteresowania z obrazów źródłowych.
- Opracowanie i implementacja w języku Python skryptów służących do trenowania i ewaluacji modeli klasyfikacyjnych.
- Zbudowanie 30 modeli klasyfikacyjnych służących do rozpoznawania zmian skórnych z wykorzystaniem 10 różnych architektur głębokich konwolucyjnych sieci neuronowych. Uzyskane wyniki pozwoliły ocenić skuteczność poszczególnych architektur w rozwiązywaniu postawionych zadań klasyfikacji.
- Przeprowadzenie ewaluacji modeli klasyfikacyjnych pozwalającej wyłonić modele najskuteczniejsze dla poszczególnych wariantów klasyfikacji: klasyfikacja binarna typu ZNAMIEŃ – CZERNIAK, klasyfikacja binarna typu ŁAGODNE – ZŁOŚLIWE, klasyfikacja wieloklasowa.
- Opracowanie ogólnej struktury systemu do klasyfikacji zmian skórnych opartej na wynikach badań. W systemie zaproponowano wykorzystanie wariantów modelu DenseNet201, zależnych od rodzaju przeprowadzanej klasyfikacji.

9.3. Zastosowanie wyników badań

Porównanie uzyskanych efektów z wynikami innych autorów pozwala ocenić je jako bardzo dobre. Przypomnijmy, iż najskuteczniejszy model, tj. DenseNet201 uzyskał dokładność testowania równą 96% dla klasyfikacji ZNAMIEŃ – CZERNIAK, 93% w przypadku klasyfikacji ŁAGODNE – ZŁOŚLIWE oraz 92% dla klasyfikacji wieloklasowej. Taka sytuacja stwarza możliwości wykorzystania tego modelu do budowy systemu komputerowego do automatycznej diagnostyki zmian skórnych. Na rysunku 9.1 przedstawiono ogólną koncepcję systemu do klasyfikacji nowych obrazów, opartą na wynikach badań. Nowe obrazy dermatoskopowe są wstępnie przetwarzane, w wyniku czego zostaje z nich wyodrębniony obszar zainteresowania (ROI). Po ekstrakcji

ROI, nowe obserwacje przekazywane są na wejście modelu. Klasyfikacja odbywa się za pomocą modelu DenseNet201. Należy w tym miejscu dodać, iż w pracy zbudowano 3 warianty tego modelu, każdy dla innego typu klasyfikacji. W systemie automatycznej klasyfikacji zastosowany będzie model odpowiadający aktualnie wybranemu typowi klasyfikacji, np. klasyfikacji ZNAMIEŃ – CZERNIAK lub klasyfikacji wieloklasowej. Zadaniem modelu jest analiza cech obrazu przekazanych na jego wejście i obliczenie prawdopodobieństwa przynależności obserwacji do jednej z klas. Na tej podstawie, nowej obserwacji zostaje przypisana odpowiednia etykieta klasy. W przypadku klasyfikacji binarnej, może to być „CZERNIAK” lub „ZNAMIEŃ”, a w przypadku klasyfikacji wieloklasowej, byłyby to nazwa zmiany skórnej.



Rys. 9.1. Ogólna struktura systemu do klasyfikacji zmian skórnych oparta na wynikach badań

9.4. Plan dalszych prac

Dalsze prace związane z kontynuacją badań przedstawionych w rozprawie będą obejmowały próbę budowy kompletnego systemu komputerowego służącego do diagnostyki zmian skórnych na podstawie obrazów dermatoskopowych. Wdrożenie takiego systemu będzie poprzedzone szeregiem działań, które scharakteryzowano poniżej.

W celu uzyskania bardziej wiarygodnych wyników ewaluacji, proces testowania należy przeprowadzić z wykorzystaniem większej liczby obrazów. Jest to trudne zadanie, ponieważ nie ma zbyt wiele ogólnodostępnych baz obrazów dermatoskopowych spełniających wymagane kryteria w zakresie liczebności oraz jakości obrazów.

Wyniki ewaluacji pokazały pewne różnice w dokładności modeli testowanych na obrazach pochodzących z innych baz, niż ta na której owe modele były trenowane (baza ISIC). W związku z tym, podczas przyszłych prac, warto uwzględnić w zbiorach treningowych szerszą pulę ogólnodostępnych baz

danych. Należy się w tym przypadku liczyć z pewnym spadkiem dokładności modeli, lecz wyższa, niż obecnie, skuteczność klasyfikatorów dla szerszej populacji obrazów warta jest poniesienia takiego kosztu.

W niniejszych badaniach wykorzystano maksymalnie 6 rodzajów zmian skórnych, z uwagi na ograniczoną wielkość dostępnych repozytoriów obrazów. Zgromadzenie w przyszłości odpowiednio licznego zbioru danych obrazowych pozwoliłoby zwiększyć liczbę rozpoznawanych zmian w procesie klasyfikacji wieloklasowej. Ponadto, pojawiłyby się możliwości w zakresie budowy modeli do rozpoznawania stopnia zaawansowania stanu patologicznego wybranej jednostki chorobowej.

Podczas przygotowania próbek do procesu uczenia i ewaluacji modeli istotne jest odpowiednie określenie obszaru zainteresowania. W celu zapewnienia szybkiego działania systemu diagnostycznego, proces segmentacji obszaru zainteresowania powinien w sposób automatyczny odbywać się podczas wykonywania zdjęcia zmiany skórnej za pomocą dermatoskopu. Taka możliwość znacznie skróciłaby cały proces przygotowywania próbek. W związku z tym, w toku przyszłych prac nastąpi opracowanie dedykowanej aplikacji komputerowej zintegrowanej z widetermatoskopem, w której zaimplementowane będą algorytmy automatycznej segmentacji obszaru zainteresowania oraz predykcji nowych obserwacji. Zwiększy to szanse opracowanej metody na jej wprowadzenie do szerszego zastosowania w gabinetach dermatologicznych.

Podsumowując, wyniki przedstawione w rozprawie uważam za perspektywiczne i atrakcyjne pod kątem możliwości aplikacyjnych. Moim zdaniem, mogą się one w istotnym zakresie przyczynić rozwiązania problemu diagnostyki zmian skórnych, która w ostatnich latach nabiera coraz większego znaczenia.

Literatura

- [1] Raport Światowej Organizacji Zdrowia odnośnie czerniaka skóry w 2020 roku, (dostęp 30.03.2022). <https://gco.iarc.fr/today/data/factsheets/cancers/16-Melanoma-of-skin-fact-sheet.pdf>
- [2] Gershenwald, J., Scolyer, R., Hess, K., Sondak, V., Long, G., Ross, M., Lazar, A., Faries, M., Kirkwood, J., McArthur, G., Haydu, L., Eggermont, A., Flaherty, K., Balch, C., Thompson, J., For members of the American Joint Committee on Cancer Melanoma Expert Panel and the International Melanoma Database and Discovery Platform. Melanoma staging: Evidence-based changes in the American Joint Committee on Cancer eighth edition cancer staging manual. *CA Cancer J Clin.* 2017, 67(6), 472–492.
- [3] Wojciechowska, U., Didkowska, J., Zachorowania i zgony na nowotwory złośliwe w Polsce. Krajowy Rejestr Nowotworów. *Narodowy Instytut Onkologii im. Marii Skłodowskiej-Curie – Państwowy Instytut Badawczy.* (dostęp 10.02.2022), <http://onkologia.org.pl/raporty/>
- [4] Zakład Epidemiologii i Prewencji Nowotworów Centrum Onkologii – Instytut w Warszawie. Krajowy Rejestr Nowotworów (KRN). (dostęp 02.08.2019), <http://onkologia.org.pl/>
- [5] Materiały Narodowego Funduszu Zdrowia – aktualności. (dostęp 10.02.2022), <https://www.nfz.gov.pl/aktualnosci/aktualnosci-oddzialow/czerniak-pod-lupa-az-1200-badan-w-jeden-dzien,367.html>
- [6] Iqbal, I., Younus, M., Walayat, K., Ullah, K., Kakar, M., Ma, J., Automated multi-class classification of skin lesions through deep convolutional neural network with dermoscopic images. *Computerized Medical Imaging and Graphics* 2021, 88, 101843.
- [7] Al-Masni, M.A., Kim, D.H., Kim, T.S., Multiple skin lesions diagnostics via integrated deep convolutional networks for segmentation and classification. *Computer Methods and Programs in Biomedicine* 2020, 190, 105351.
- [8] Srinivasu, P.N., SivaSai, J.G., Ijaz, M.F., Bhoi, A.K., Kim, W., Kang, J.J., Classification of skin disease using deep learning neural networks with MobileNet V2 and LSTM. *Sensors* 2021, 21, 2852.
- [9] Cassidy, B., Kendrick, C., Brodzicki, A., Jaworek-Korjakowska, J., Yap, M., Analysis of the ISIC image datasets: usage, benchmarks and recommendations. *Medical Image Analysis* 2022, 75, 102305.
- [10] Wang, Y., Cai, J., Louie, D., Wang, J., Lee, T., Incorporating clinical knowledge with constrained classifier chain into a multimodal deep network for melanoma detection. *Computers in Biology and Medicine* 2021, 137, 104812.
- [11] Saeed, J., Zeebaree, S., Skin lesion classification based on deep convolutional neural networks architectures, *JASTT* 2021, 2(01), 41–51.
- [12] Raza, R., Zulfiqar, F., Tariq, S., Anwar, G.B., Sargano, A., Habib, Z., Melanoma classification from dermoscopy images using ensemble of convolutional neural networks. *Mathematics* 2022, 10(1), 26.
- [13] Chaturvedi, S., Gupta, K., Prasad, P., Skin Lesion Analyser: An Efficient Seven-Way Multi-class Skin Cancer Classification Using MobileNet. *Advances in Intelligent Systems and Computing* 1141, Springer, Singapore, 2020.

- [14] Villa-Pulgarin, J., Ruales-Torres, A., Arias-Garzón, D., Bravo-Ortiz, M., Arteaga-Arteaga, H., Mora-Rubio, A., Alzate-Grisales, J., Mercado-Ruiz, E., Hassaballah, M., Orozco-Arias, S., Cardona-Morales, O., Tabares-Soto, R., Optimized Convolutional Neural Network Models for Skin Lesion Classification. *Computers. Materials & Continua Tech Science Press* 2022, CMC 70(2).
- [15] Kareem, O.S., Abdulazee, A.M., Zeebaree ,D.Q., Skin lesions classification using deep learning techniques: Review. *Asian Journal of Research in Computer Science* 2021, 9(1), 1–22, AJRCOS.68652.
- [16] Adegun, A., Viriri, S., Deep learning techniques for skin lesion analysis and melanoma cancer detection: a survey of state of the art. *Artif Intell Rev* 2021, 54, 811–841.
- [17] FotoFinder dermatoskop. (dostęp 22.03.2023), <https://www.fotofinder.de/en/>
- [18] Kamińska-Winciorek, G., Śpiewak, R., Podstawy dermoskopii zmian melanocytowych dla początkujących. *Postepy Hig Med Dosw (online)* 2011, 65, 501–508.
- [19] Rutkowski, P., Wysocki, P.J, Nasierowska-Guttmejer, A., i in., Cutaneous melanomas. *Oncol Clin Pract* 2019, 15.
- [20] American Joint Committee on Cancer. *Melanoma of the Skin Staging*. 7th Edition. (dostęp 26.08.2019), <https://cancerstaging.org/references-tools/quickreferences/Documents/MelanomaSmall.pdf>.
- [21] AOTMiT - Wniosek o objęcie refundacją leku Keytruda (pembrolizumab) we wskazaniu: Leczenie czerniaka skóry lub błon śluzowych (ICD-10C43) (leczenie adjuwantowe). Analiza weryfikacyjna. (dostęp 26.08.2019), http://bipold.aotm.gov.pl/assets/files/zlecenia_mz/2019/121/AWA/121_AWA_O T.4331.32.2019_Keytruda_adjuwant_czerniak_25.07_BIP.pdf [.]
- [22] Haenssle, H., Fink, C., Toberer, F., Winkler, J., Stolz, W., Deinlein, T., i in., Man against machine reloaded: performance of a market-approved convolutional neural network in classifying a broad spectrum of skin lesions in comparison with 96 dermatologists working under less artificial conditions. *Ann Oncol* 2020, 31, 137–143.
- [23] Stolz, W., Riemann, A., Cognetta, A., Pillet, L., Abmayr, W., Holzel, D., Bilek, P., Nachbar, F., Landthaler, M., ABCD rule of dermatoscopy – a new practical method for early recognition of malignant-melanoma. *European Journal of Dermatology* 1994, 4(7), 521–527.
- [24] Damilola, A., Okuboyejo, O., Automating skin disease diagnosis using image classifications. *Proceedings of the world congress on engineering and computer science II*, San Francisco, USA, 2013.
- [25] Henning, J., Dusza, S., Wang, S., The CASH (color, architecture, symmetry, and homogeneity) algorithm for dermoscopy. *Archives of Dermatology* 2007, 56, 45–52.
- [26] Emery, J.D, Hunter, J., Hall, P.N., Accuracy of siascopy for pigmented skin lesions encountered in primary care: development and validation of a new diagnostic algorithm. *BMC Dermatology* 2010, 10, 1–9.
- [27] Clark, W.H, From, L., Bernardino, E.A., Histogenesis and biologic behavior of primary human malignant melanomas of the skin. *Cancer Research* 1969, 29, 705–726.

- [28] Codella, N., Nguyen, B., Pankanti, S., Gutman, D, Helba, B., Halpern, A., Smith, J.R., Deep learning ensembles for melanoma recognition in dermoscopy images. *IBM Journal of Research and Development* 2017, 61(4/5), 173.
- [29] Codella, N., Cai, J., Abedini, M., Garnavi, R., Halpern, A., Smith, J. R., Deep learning, sparse coding, and SVM for melanoma recognition in dermoscopy images. *Machine Learning in Medical Imaging*, Springer 2015. 118–126.
- [30] Ho, L., Fully automated GrowCut-based segmentation of melanoma in dermoscopic images. *Journal of Young Investigators* 2019, 36(2), 1117.
- [31] Barata, C., Figueiredo, M.A., Celebi, M.E., Marques, J.S., Local features applied to dermoscopy images: Bag-of-features versus sparse coding. *Iberian Conference on Pattern Recognition and Image Analysis*, Springer 2017, 528–536.
- [32] Shahid, S.M., Khan, S., Dermoscopy images classification based on color, texture and shape features using SVM. W: The 3rd International Conference on Next Generation Computing (ICNGC2017b), Kaohsiung, Taiwan, 21–24.12.2017.
- [33] Jaworek-Korjakowska, J., A deep learning approach to vascular structure segmentation in dermoscopy colour images. *BioMed Research International* 2018, 5049390.
- [34] Jaworek-Korjakowska, J., Computer-aided diagnosis of micro-malignant melanoma lesions applying support vector machines. *BioMed Research International* 2016, 4381972.
- [35] Dzierżak, R., *Zastosowanie deep learningu w analizie obrazów medycznych*. Prace doktorantów Wydziału Elektrotechniki i Informatyki Politechniki Lubelskiej, wybrane zagadnienia z zakresu elektrotechniki i inżynierii biomedycznej, Politechnika Lubelska, Lublin, 2018, 59–69.
- [36] Jaworek-Korjakowska, J., Kleczek, P., Automatic classification of specific melanocytic lesions using artificial intelligence. *BioMed Research International* 2016, 8934242.
- [37] Litjens, G., Kooi T., Bejnordi, B.E., Setio, A.A.A., Ciompi, F., Ghafoorian, M., van der Laak, J.A., Ginneken, B., Sánchez, C. I., A survey on deep learning in medical image analysis. *Medical Image Analysis* 2017, 42, 60–88.
- [38] Lopez, A., Giro-i-Nieto, X., Burdick, J., Marques, O., Skin lesion classification from dermatoscopic images using deep learning techniques. W: *2017 13th IASTED international conference on biomedical engineering (BioMed)*, IEEE 2017, 49–54.
- [39] Zhang, J., Xie, Y., Wu, Q., Xia, Y., Skin lesion classification in dermoscopy images using synergic deep learning. *Medical Image Computing and Computer Assisted Intervention–MICCAI 2018: 21st International Conference*, Granada, Spain, 16-20.09.2018, Springer International Publishing, Proceedings Part II 11, 12–20.
- [40] Ge, Z., Demyanov, S., Chakravorty, R., Bowling, A., Garnavi, R., Skin disease recognition using deep saliency features and multimodal learning of dermoscopy and clinical images. Descoteaux, M., Maier-Hein L., Franz A., Jannin, P., Collins D., Duchesne S. (eds.) *Springer* 2017, Cham LNCS 10435, 250–258.
- [41] Haenssle, H., Fink, C., Schneiderbauer, R., Toberer, .F, Buhl, T., Blum, A., Managaint machine: diagnostic performance of a deep learning convolutional neural network for dermoscopic melanoma recognition in comparison to 58 dermatologists. *Ann Oncol* 2018, 29, 1836–1842.

- [42] Vijay, Ch., Aayush, J., Gianchandani, N., Dilbag, S., Manjit, K., Classification of the COVID-19 infected patients using DenseNet201 based deep transfer learning. *Journal of biomolecular Structure & Dynamics* 2020, 39.
- [43] Young, A., Xiong, M., Pfau, J., Keiser, M., Wei, M., Artificial intelligence in dermatology: A Primer. *Journal of Investigative Dermatology* 2020, 140, 1504–1512.
- [44] Murphree, D., Puri, P., Shamim, H., Bezalel, S., Drage, L., Wang, M., Pittelkow, M., Carter, R., Davis, M., Bridges, A., Mangold, A., Yiannias, J., Tollefson, M., Lehman, J., Meves, A., Otley, C., Sokumbi, O., Hall, M., Comfere, N., Deep learning for dermatologists: Part I. *J AM ACAD DERMATOL* 2020, 1–9.
- [45] Jaworek-Korjakowska, J., i in., Melanoma thickness prediction based on convolutional neural network with VGG-19 model transfer learning. W: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2019, 2748–2756.
- [46] Omiotek Z., *Wybrane problem modelowania predykcyjnego w diagnostyce technicznej i medycznej*. Politechnika Lubelska, Lublin 2021.
- [47] Jaworek-Korjakowska, J., Kleczek, P., Region adjacency graph approach for acral melanocytic lesion segmentation. *Applied Sciences* 2018, 8, 1430.
- [48] Dinggang, S., Guorong, W., Heung-Il, S., Deep learning in medical image analysis. *The Annual Review of Biomedical Engineering* 2017, 9, 221–248.
- [49] Yu, L., Chen, H., Dou, Q., Qin, J., Heng, P.A., Automated melanoma recognition in dermoscopy images via very deep residual networks. *IEEE Trans. Med. Imaging* 2017, 36(4), 994–1004.
- [50] Esteva, A., Dermatologist-level classification of skin cancer with deep neural networks. *Nat. Res.* 2017, 542(7639), 115–118.
- [51] Codella N., Nguyen Q., Pankanti s., Gutman D., Helba B., Halpern A., Smith J., Deep learning ensembles for melanoma recognition in dermoscopy images. *arXiv:1610.04662*, 2016.
- [52] Attia, M., Hossny, M., Nahavandi, S., Yazdabadi, A., Skin melanoma segmentation using recurrent and convolutional neural networks. W: *Biomedical Imaging (ISBI 2017)*, 2017 IEEE 14th International Symposium, IEEE 2017, 292–296.
- [53] Nijeweme-d’Hollosy, W., van Velsen, L., Poel, M., Groothuis-Oudshoorn C., Soer R., Hermens H., Evaluation of three machine learning models for self-referral decision support on low back pain in primary care. *Int. J. Med. Inform.* 2018, 110, 31–41.
- [54] Nida, N., Irtaza, A., Javed, A., Yousaf, M., Mahmood, M., Melanoma lesion detection and segmentation using deep region based convolutional neural network and fuzzy C-means clustering. *International Journal of Medical Informatics* 2019, 124, 37–48.
- [55] Ge, Y., Li, B., Zhao, Y., Guan, E., Yan, W., Melanoma segmentation and classification in clinical images using deep learning. *Proceedings of the 2018 10th International Conference on Machine Learning and Computing* 2018, 252–256.
- [56] Chollet F., *Deep Learning. Praca z językiem Python i biblioteką Keras*. Helion, Gliwice 2019.
- [57] Zocca V., Spcagna G., Slater D., Roelants P., *Deep learning, Uczenie głębokie z językiem Python, Sztuczna inteligencja i sieci neuronowe*. Helion, Gliwice 2018.

- [58] Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K., Densely connected convolutional networks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, USA, 21–26.07.2017, 4700–4708.
- [59] Simonyan, K., Zisserman, A., Very deep convolutional networks for large-scale image recognition. W: *International Conference on Learning Representations (ICLR 2015)*, San Diego, USA, 2015.
- [60] Albelwi, S., Mahmood, A., A Framework for designing the architectures of deep convolutional neural networks. *Entropy* 2017, 19(6), 242.
- [61] Rumelhart, D.E., Hinton, G.E., Williams, R.J., Learning representations by back-propagating errors. *Nature* 1986, 323.
- [62] Hinton, G.E., Srivastava, N., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.R., Improving neural networks by preventing co-adaptation of feature detectors. *arXiv:1207.0580v1 [cs.NE]* 2012.
- [63] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R., Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research* 2014, 15, 1929–1958.
- [64] Lu, S., Lu, Z., Zhang, Y.D., Pathological brain detection based on AlexNet and transfer learning. *Journal of computational science* 2019, 30, 41–47.
- [65] Kassem, M. A, Hosny, K. M, Fouad, M. M., Skin lesions classification into eight classes for ISIC 2019 using deep convolutional neural network and transfer learning. *IEEE Access* 2020, 8, 114822–114832.
- [66] ISIC Archive – baza danych obrazowych. (dostęp: 23.03.2022), <https://www.isic-archive.com/#!/topWithHeader/onlyHeaderTop/gallery>.
- [67] PH² – baza danych obrazowych. (dostęp: 23.03.2022), <https://www.fc.up.pt/addi/ph2%20database.html>.
- [68] MED-NODE – baza danych obrazowych. (dostęp: 23.03.2022), http://www.cs.rug.nl/~imaging/databases/melanoma_naevi/.
- [69] PAD-UFES-20 – baza danych obrazowych. (dostęp: 23.03.2022), <https://data.mendeley.com/datasets/zr7vgbcyr2/1>.
- [70] Mahdianpari, M., Salehi, B., Rezaee, M., Mohammadimanesh F., Zhang Y., Very deep convolutional neural networks for complex land cover mapping using multispectral remote sensing imagery. *Remote Sens.* 2018, 10(7).
- [71] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Liang, C., MobileNetV2: Inverted residuals and linear bottlenecks. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* 2018, 4510–4520.
- [72] Chollet, F., Xception: Deep learning with depthwise separable convolutions. *arXiv:1610.02357*, 2016.
- [73] Architektura rodziny modeli ResNet50. (dostęp: 8.11.2022), <https://iq.opengenus.org/resnet50-architecture/>.
- [74] He, K., Zhang, X., Ren, S., Sun, J., Deep residual learning for image recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, USA, 27-30.06.2016, 770-778.
- [75] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V. Rabinovich, A., *Going deeper with convolutions. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, USA, 7-12.06.2015.
- [76] Bhatia, Y., Bajpayee ,A., Raghuvanshi, D. Mittal H., Image captioning using google’s inception-resnet-v2 and recurrent neural network. W: *2019 Twelfth Int. Conf. on Contemporary Computing*, Noida, India, 2019, 1–6.

- [77] Szegedy, C., Ioffe, S., Vanhoucke, V., Alemi, A.A., Inception-V4, Inception-ResNet and the impact of residual connections on learning. *AAAI* 2017, 4278–4284.
- [78] Shahin, A. H., Kamal, A., Elattar, M. A., Deep ensemble learning for skin lesion classification from dermoscopic images. W: *2018 9th Cairo International Biomedical Engineering Conference (CIBEC)*, Cairo, Egypt, 2018, 150–153.
- [79] Feng, C., Zhang, H., Wang, S., Li, Y., Wang, H. Yan, F., Structural damage detection using deep convolutional neural network and transfer learning. *KSCE Journal of Civil Engineering* 2019, 23, 4493–4502.
- [80] Tan, M., Le, Q., EfficientNet: Rethinking model scaling for convolutional neural networks. *Proceedings of the 36th International Conference on Machine Learning*, PMLR 2019, 97, 6105–6114.
- [81] Baheti, B, Innani, S., Gajre, S., Talbar, S., *Eff-UNet: A Novel architecture for semantic segmentation. Unstructured Environment.* 2022, 1473–1481.
- [82] Akyel, C., Arıç, N., LinkNet-B7: Noise removal and lesion segmentation in images of skin cancer. *Mathematics* 2020, 10, 736.
- [83] Architektura rodziny modeli DenseNet. (dostęp: 8.11.2022), <https://pytorch.org/assets/images/densenet2.png>.
- [84] Addagarla, S., Chakravarthi, G., Anitha, P., Real time multi-scale facial mask detection and classification using deep transfer learning techniques. *International Journal of Advanced Trends in Computer Science and Engineering* 2010, 9, 4402–4408.
- [85] Zoph, B., Vasudevan, V., Shlens, J., Le, Q.V., Learning transferable architectures for scalable image recognition. *IEEE conference on computer vision and pattern recognition* 2018, 8697–8710.
- [86] Omiotek Z., *Automatyczna klasyfikacja obrazów USG tarczycy, rozprawa doktorska.* Politechnika Lubelska, Lublin 2014.
- [87] Harańczyk G., *Krzywe ROC, czyli ocena jakości klasyfikatora i poszukiwanie optymalnego punktu odcięcia.* StatSoft Polska, Kraków 2010.
- [88] Harańczyk G., Stępień M., *Ilustrowana sztuka podejmowania decyzji.* Matematyka, Społeczeństwo, Nauczanie 2008, 41, 12–15.
- [89] Keras. Keras library. (dostęp 15.03. 2022), <https://keras.io/>
- [90] Howard, A., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M. Hartwig, A., MobileNets: Efficient convolutional neural networks for mobile vision applications. *arXiv:1704.04861v1*, 2017.
- [91] He, K., Zhang, X., Ren, S., Sun, J., Identity mappings in deep residual networks. *arXiv:1603.05027v3*, 2016.
- [92] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna Z. Rethinking the Inception architecture for computer vision. *arXiv:1512.00567v3*, 2015.
- [93] Jeyakumar, J., Jude, A., Henry, A., Hemanth, J., Comparative analysis of melanoma classification using deep learning techniques on dermoscopy images. *Electronics* 2022, 11, 2918.
- [94] Safdar, K., Akbar, S., Shoukat, A., A majority voting based ensemble approach of deep learning classifiers for automated melanoma detection. *2021 International Conference on Innovative Computing (ICIC) IEEE* 2021, 1–6.
- [95] Singh, L, Janghel, R.R, Sahu, S.P., A deep learning-based transfer learning framework for the early detection and classification of dermoscopic images of melanoma. *Biomed Pharmacol J* 2021, 14(3).

- [96] Indraswari, R., Rokhana, R., Herulambang, W., Melanoma image classification based on MobileNetV2 network. *Procedia Computer Science* 2022, 197, 198–207.
- [97] Romero Lopez, A., Giro-i-Nieto, X., Burdick, J., Marque,s O., Skin lesion classification from dermoscopic images using deep learning techniques. W: *2017 13th IASTED International Conference on Biomedical Engineering (BioMed)* 2017, 49–54, Innsbruck, Austria.
- [98] Aljohani, K., Turki, T., Automatic classification of melanoma skin cancer with deep convolutional neural networks. *AI 3* 2022, 512–525.
- [99] Brinker, T.J., Hekler, A., Enk, A.H., Berking, C., Haferkamp, S., Hauschild, A., Weichenthal, M., Klode, J., Schadendorf, D., Holland-Letz, T., i in., Deep neural networks are superior to dermatologists in melanoma image classification. *Eur. J. Cancer* 2019, 119, 11–17.
- [100] Acosta, M.F.J., Tovar, L.Y.C., Garcia-Zapirain, M.B., Percybrooks, W.S., Melanoma diagnosis using deep learning techniques on dermoscopic images. *BMC Med. Imaging* 2021, 21, 6.
- [101] Rashid, J., Ishfaq, M., Ali, G., Saeed, M.R., Hussain, M, Alkhalifah, T., Alturise, F., Samnd, N., Skin cancer disease detection using transfer learning technique. *Appl. Sci.* 2022, 12, 5714.
- [102] Lu, X, Firoozeh Abolhasani Zadeh, Y.A., Deep learning-based classification for melanoma detection using XceptionNet. *Journal of Healthcare Engineering* 2022, 2196096.
- [103] Chaturvedi, S., Tembhurne, J., Diwan, T., A multi-class skin cancer classification using deep convolutional neural networks. *Multimedia Tools and Applications* 2020, 79.
- [104] Ali, K., Shaikh, Z.A, Khan, A.A, Laghari, A.A., Multiclass skin cancer classification using EfficientNets – a first step towards preventing skin cancer. *Neuroscience Informatics* 2022, 2 (4), 100034.
- [105] Jain, S., Singhanian, U., Tripathy, B., Nasr, E.A., Aboudaif, M.K., Kamrani, A.K., Deep learning-based transfer learning for classification of skin cancer. *Sensors* 2021, 21, 8142.

Załącznik A. Listing kodu do segmentacji zmian skórnych

```
import cv2, os, glob, shutil
import numpy as np
# Zmienne globalne
imageShape = (224, 224)
filePath = ''
fileIndex = 0
fileNumber = 0
path = 'OBRAZY'
left = right = top = bottom = 0
base_dir = './'
cropped = os.path.join(base_dir, 'CROPPED')
not_cropped = os.path.join(base_dir, 'NOT CROPPED')
def createDirectories() :
    if not os.path.exists(cropped) :
        os.mkdir(cropped)
    if not os.path.exists(not_cropped) :
        os.mkdir(not_cropped)
def rescaleImage(roi, nazwaPliku) :
    global fileIndex, fileNumber, left, right, top, bottom
    if left and right and top and bottom :
        cv2.imwrite(cropped + '/' + nazwaPliku, roi)
        print(str(fileIndex) + '/' + str(fileNumber) + ' - ' +
              nazwaPliku)
    else:
        shutil.move(filePath, not_cropped + '/' + nazwaPliku)
        print(str(fileIndex) + '/' + str(fileNumber) + ' - ' +
              nazwaPliku + ' - NOT CROPPED')
    fileName = 'Obszar zainteresowania.bmp'
    showImage(fileName, roi)
    cv2.imwrite(fileName, roi)
def showImage(fileName, imageName) :
    cv2.namedWindow(fileName, cv2.WINDOW_NORMAL)
    cv2.imshow(fileName, imageName)
    cv2.waitKey(0)
    cv2.destroyAllWindows()
def cropObject(nazwaPliku, image, contours, max_index, nrKolumny,
              nrWiersza, szerokosc, wysokosc, prostokat) :
    (h, w, dd) = image.shape
    global left, right, top, bottom
    left=right=top=bottom = False
    if szerokosc > wysokosc :
        delta = szerokosc-wysokosc
        wyrownanie = int(delta/2)
        margines = int(0.05*szerokosc)
        if nrKolumny-margines >=0 :
            startX = nrKolumny-margines
            left = True
        else :
            startX = 0
    if nrKolumny+szerokosc+margines <= w :
        endX = nrKolumny+szerokosc+margines
        right = True
    else :
        endX = w
    if nrWiersza-wyrownanie-margines >= 0 :
        startY = nrWiersza-wyrownanie-margines
        top = True
    else :
        startY = 0
```

```

        if nrWiersza+wysokosc+wyrownanie+margines <= h:
            endY = nrWiersza+wysokosc+wyrownanie+margines
            bottom = True
        else :
            endY = h
    elif szerokosc < wysokosc :
        delta = wysokosc-szerokosc
        wyrownanie = int(delta/2)
        margines = int(0.05*wysokosc)
        if nrKolumny-wyrownanie-margines >= 0 :
            startX = nrKolumny-wyrownanie-margines
            left = True
        else :
            startX = 0
        if nrKolumny+szerokosc+wyrownanie+margines <= w:
            endX = nrKolumny+szerokosc+wyrownanie+margines
            right = True
        else :
            endX = w
        if nrWiersza-margines >= 0:
            startY = nrWiersza-margines
            top = True
        else :
            startY = 0
        if nrWiersza+wysokosc+margines <= h :
            endY = nrWiersza+wysokosc+margines
            bottom = True
        else :
            endY = h
    else :
        margines = int(0.05*szerokosc)
        startX = nrKolumny-margines
        endX = nrKolumny+szerokosc+margines
        startY = nrWiersza-margines
        endY = nrWiersza+wysokosc+margines
        left=right=top=bottom = True
# Obraz kontrolny
control_image1 = image.copy()
maska1 = cv2.drawContours(control_image1, contours, max_index,
                          (0,0,0), -1)
maska2 = cv2.subtract(image, maska1)
maska3 = cv2.add(maska1, maska2)
kopia_maska3 = maska3.copy()
kopia_maska3 = cv2.rectangle(prostokat, (startX, startY),
                             (endX, endY), (0,0,255), 3)
fileName = 'Obraz z kwadratem ograniczajacym zmiane.bmp'
showImage(fileName, kopia_maska3)
cv2.imwrite(fileName, kopia_maska3)
roi = maska3[startY:endY, startX:endX]
return roi
def segmentImage(image) :
    global threshold
    # Konwersja do skali szarosci
    gray_image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
    fileName = 'Obraz w skali szarosci.bmp'
    showImage(fileName, gray_image)
    cv2.imwrite(fileName, gray_image)
    # Filtracja medianowa
    median_image = cv2.medianBlur(gray_image, 5)
    fileName = 'Obraz po filtracji medianowej.bmp'
    showImage(fileName, median_image)
    cv2.imwrite(fileName, median_image)

```

```

# Progowanie
ret, binary_image = cv2.threshold(median_image, 0, 255,
                                 cv2.THRESH_BINARY_INV + cv2.THRESH_OTSU)
fileName = 'Obraz po progowaniu.bmp'
showImage(fileName, binary_image)
cv2.imwrite(fileName, binary_image)
# Zamknięcie
mask = binary_image.copy()
mask = cv2.dilate(mask, None, iterations = 8)
mask = cv2.erode(mask, None, iterations = 8)
# Otwarcie
mask = cv2.erode(mask, None, iterations = 8)
mask = cv2.dilate(mask, None, iterations = 8)
fileName = 'Obraz po operacjach morfologicznych.bmp'
showImage(fileName, mask)
cv2.imwrite(fileName, mask)
# Wykrycie wszystkich konturów
contours, hierarchy = cv2.findContours(mask, cv2.RETR_TREE,
                                       cv2.CHAIN_APPROX_SIMPLE)
# Wyszukanie indeksu konturu o największej powierzchni
areas = [cv2.contourArea(c) for c in contours]
max_index = np.argmax(areas)
kontur = image.copy()
najwiekszyObiekt = cv2.drawContours(kontur, contours, max_index,
                                   (0,255,255), 3)
fileName = 'Obraz z konturem największego obiektu.bmp'
showImage(fileName, kontur)
cv2.imwrite(fileName, najwiekszyObiekt)
# Znalezienie prostokata opisanego na konturze
nrKolumny, nrWiersza, szerokosc, wysokosc =
    cv2.boundingRect(contours[max_index])
startX = nrKolumny
startY = nrWiersza
endX = nrKolumny + szerokosc
endY = nrWiersza + wysokosc
prostokat = cv2.rectangle(kontur, (startX, startY), (endX, endY),
                          (255,0,0), 3)
fileName = 'Obraz z prostokatem ograniczajacym kontur.bmp'
showImage(fileName, prostokat)
cv2.imwrite(fileName, prostokat)
return contours, max_index, nrKolumny, nrWiersza, szerokosc,
        wysokosc, prostokat
def main():
    global fileIndex, fileNumber, filePath
    createDirectories()
    fileList = glob.glob(path + '/*')
    for filePath in fileList:
        fileNumber = len(fileList)
        # Wczytanie obrazu zrodlowego
        _, nazwaPliku = os.path.split(filePath)
        image = cv2.imread(filePath, -1)
        fileIndex += 1
        fileName = 'Obraz zrodlowy.bmp'
        showImage(fileName, image)
        cv2.imwrite(fileName, image)
        contours, max_index, nrKolumny, nrWiersza, szerokosc,
            wysokosc, prostokat = segmentImage(image)
        roi = cropObject(nazwaPliku, image, contours, max_index,
                        nrKolumny, nrWiersza, szerokosc, wysokosc, prostokat)
        rescaleImage(roi, nazwaPliku)
if __name__ == '__main__':
    main()

```


Załącznik B. Listing przykładowego kodu do trenowania modeli

```
import os
from keras import optimizers
from keras.preprocessing.image import ImageDataGenerator
from keras.models import Sequential
from keras.layers import Dense, GlobalAveragePooling2D
from keras.callbacks import ModelCheckpoint
import warnings
# *** USTAWIENIA *****
warnings.filterwarnings('ignore')
MODEL = 'VGG16'
VARIANT = 'NEVUS-MELANOMA'
CLASS_MODE = 'binary'
LOSS_TYPE = 'binary_crossentropy'
MODEL_NAME = MODEL + '_' + VARIANT + '.h5'
CONV_BASE_NAME = 'CONV_BASE_' + MODEL + '_' + VARIANT + '.h5'
BASE_DATASET_FOLDER = './' + VARIANT
MODEL_PATH = os.path.join(MODEL_NAME)
CONV_BASE_PATH = os.path.join(CONV_BASE_NAME)
TRAIN_FOLDER = os.path.join(BASE_DATASET_FOLDER, 'train')
VALIDATION_FOLDER = os.path.join(BASE_DATASET_FOLDER, 'validation')
TEST_FOLDER = os.path.join(BASE_DATASET_FOLDER, 'test')
CLASSES_NUMBER = 0
TRAIN_BATCH_SIZE = 8
VAL_BATCH_SIZE = 8
TRAINING_EPOCHS = 50
TUNNING_EPOCHS = 50
STOP_TRAINING = 0
TUNNING_HISTORY_DIAGRAMS = True
IMAGE_SIZE = (224, 224)
INPUT_SHAPE = (224, 224, 3)
LR = 0
ACC=VAL_ACC=LOSS=VAL_LOSS=SZYBKOSC_UCZENIA=None
def trainGenerator():
    print('\n*** BUDOWA GENERATORA DANYCH TRENINGOWYCH ***')
    train_datagen = ImageDataGenerator(
        rescale=1./255,
        rotation_range=40,
        width_shift_range=0.2,
        height_shift_range=0.2,
        shear_range=0.2,
        zoom_range=0.2,
        horizontal_flip=True,
        fill_mode='nearest')
    train_generator = train_datagen.flow_from_directory(
        TRAIN_FOLDER,
        target_size=IMAGE_SIZE,
        batch_size=TRAIN_BATCH_SIZE,
        class_mode=CLASS_MODE,
        shuffle=True)
    return train_generator
def validationGenerator():
    print('\n*** BUDOWA GENERATORA DANYCH WALIDACYJNYCH ***')
    val_datagen = ImageDataGenerator(rescale=1./255)
    val_generator = val_datagen.flow_from_directory(
        VALIDATION_FOLDER,
        target_size=IMAGE_SIZE,
        batch_size=VAL_BATCH_SIZE,
        class_mode=CLASS_MODE,
```

```

        shuffle=True)
    return val_generator
def printClasses(train_generator):
    global CLASSES_NUMBER
    classes = {v: k for k, v in
               train_generator.class_indices.items()}
    CLASSES_NUMBER = len(classes)
    print('\n*** PODZIAŁ DANYCH NA KLASY ***')
    print('LICZBA KLAS: ' + str(CLASSES_NUMBER))
    print(classes)
    print()
def freezeModel(convBase):
    print('\n*** ZAMROŻENIE MODELU BAZOWEGO ***')
    for layer in convBase.layers:
        layer.trainable = False
def unfreezeModel(convBase, layer):
    print('\n*** ODMROŻENIE GÓRNYCH WARSTW MODELU BAZOWEGO ***')
    for layer in convBase.layers[layer:]:
        layer.trainable = True
def compileModel(model, LR):
    model.compile(loss=LOSS_TYPE,
                  optimizer=optimizers.Adam(lr=LR),
                  metrics=['acc'])
def modelTraining(model, train_generator, val_generator):
    print('\n*** TRENOWANIE NOWYCH WARSTW, CAŁY MODEŁ BAZOWY JEST
          ZAMROŻONY ***')
    # Zapisanie wag najlepszej epoki treningowej
    top_weights_path = './top_model_weights.h5'
    mc = ModelCheckpoint(top_weights_path, monitor='val_acc',
                        mode='max', verbose=0, save_best_only=True)
    trainingHistory = model.fit_generator(train_generator,
                                         steps_per_epoch=
                                         train_generator.samples//train_generator.batch_size,
                                         epochs=TRAINING_EPOCHS,
                                         validation_data=val_generator,
                                         validation_steps=
                                         val_generator.samples//val_generator.batch_size,
                                         verbose=1,
                                         callbacks=[mc])
    # Załadowanie do modelu wag najlepszej epoki treningowej
    model.load_weights(top_weights_path)
    trainedModel = model
    os.remove(top_weights_path)
    global ACC, VAL_ACC, LOSS, VAL_LOSS, STOP_TRAINING
    ACC = trainingHistory.history['acc']
    VAL_ACC = trainingHistory.history['val acc']
    LOSS = trainingHistory.history['loss']
    VAL_LOSS = trainingHistory.history['val_loss']
    STOP_TRAINING = len(ACC)
    print('\n*** TRENOWANIE NOWYCH WARSTW ZOSTAŁO ZAKOŃCZONE ***')
    return trainingHistory, trainedModel
def modelTunning(trainedModel, trainingHistory, train_generator,
                 val_generator):
    print('\n*** DOSTRAJANIE MODELU, GÓRNE WARSTWY MODELU BAZOWEGO SA
          ODMROŻONE ***')
    # Zapisanie wag najlepszej epoki treningowej
    final_weights_path = './model_weights.h5'
    mc = ModelCheckpoint(final_weights_path, monitor='val_acc',
                        mode='max', verbose=0, save_best_only=True)
    TOTAL_EPOCHS = TRAINING_EPOCHS + TUNNING_EPOCHS
    tunningHistory = trainedModel.fit_generator(
        train_generator,

```

```

        steps_per_epoch=
            train_generator.samples//train_generator.batch_size,
        epochs=TOTAL_EPOCHS,
        initial_epoch = trainingHistory.epoch[-1]+1,
        validation_data=val_generator,
        validation_steps=
            val_generator.samples//val_generator.batch_size,
        verbose=1,
        callbacks=[mc])
# Załadowanie do modelu wag najlepszej epoki treningowej
trainedModel.load_weights(final_weights_path)
os.remove(final_weights_path)
tunnedModel = trainedModel
print('\n*** DOSTRAJANIE MODELU ZOSTAŁO ZAKOŃCZONE ***')
return tunningHistory, tunnedModel
def budowaModelu(convBase):
    print('\n*** DODAWANIE NOWYCH WARSTW DO MODELU BAZOWEGO ***')
    model = Sequential()
    model.add(convBase)
    model.add(GlobalAveragePooling2D())
    if VARIANT == 'ALL_SKIN_LESIONS':
        model.add(Dense(CLASSES_NUMBER, activation='softmax'))
    else:
        model.add(Dense(1, activation='sigmoid'))
    return model
def main():
    global convBase
    print('\n*** ROZPOCZĘTO BUDOWĘ MODELU ' + MODEL_NAME + ' ***')
    # BUDOWA GENERATORÓW DANYCH
    train_generator = trainGenerator()
    val_generator = validationGenerator()
    # BUDOWA MODELU
    print('\n*** POBIERANIE MODELU BAZOWEGO ***')
    if MODEL == 'VGG16':
        from keras.applications import VGG16
        convBase = VGG16(weights='imagenet', include_top=False,
            input_shape=INPUT_SHAPE)
    model = budowaModelu(convBase)
    # ZAMROŻENIE MODELU BAZOWEGO
    freezeModel(convBase)
    compileModel(model, 1e-3)
    # TRENOWANIE MODELU
    trainingHistory, trainedModel = modelTraining(model,
        train_generator, val_generator)
    # ODMROŻENIE GÓRNYCH WARSTW MODELU BAZOWEGO
    if MODEL == 'VGG16':
        layer = 11
    unfreezeModel(convBase, layer)
    compileModel(model, 1e-5)
    # DOSTRAJANIE MODELU
    tunningHistory, tunnedModel = modelTunning(trainedModel,
        trainingHistory, train_generator, val_generator)
    tunnedModel.save(MODEL_PATH)
    convBase.save(CONV_BASE_PATH)
    print('\n*** MODEL ZOSTAŁ ZAPISANY POD NAZWĄ: ' + MODEL_NAME + '
        ***')
if __name__ == "__main__":
    main()

```

Załącznik C. Listing przykładowego kodu do ewaluacji modeli

```
import os
from sklearn.metrics import confusion_matrix, classification_report
import numpy as np
import matplotlib.pyplot as plt
from tensorflow import keras
from keras.preprocessing.image import ImageDataGenerator
import itertools
# *** USTAWIENIA ****
MODEL = 'VGG16'
VARIANT = 'NEVUS-MELANOMA'
CLASS_MODE = 'binary'
MODEL_NAME = MODEL + '_' + VARIANT + '.h5'
BASE_DATASET_FOLDER = './' + VARIANT
MODEL_PATH = os.path.join(MODEL_NAME)
TEST_FOLDER = os.path.join(BASE_DATASET_FOLDER, 'test')
IMAGE_SIZE = (224, 224)
INPUT_SHAPE = (224, 224, 3)
CLASS_MODE = None
classes = ['ZNAMIE', 'CZERNIAK']
def plotCM(cm):
    global classes
    plt.figure(figsize=(4, 4))
    plt.imshow(cm, interpolation='nearest', cmap=plt.cm.Blues)
    tick_marks = np.arange(len(classes))
    plt.xticks(tick_marks, classes, horizontalalignment="center",
               fontsize=16)
    plt.yticks(tick_marks, classes, rotation=90,
               verticalalignment="center", fontsize=16)
    thresh = cm.max() / 2.
    for i, j in itertools.product(range(cm.shape[0]),
                                  range(cm.shape[1])):
        plt.text(j, i, format(cm[i, j]),
                 horizontalalignment="center",
                 verticalalignment="center",
                 color="white" if cm[i, j] > thresh else "black",
                 fontsize=30)
    plt.ylabel('Klasa rzeczywista', fontsize=22)
    plt.xlabel('Klasa przewidywana', fontsize=22)
    plt.show()
def testGenerator():
    test_datagen = ImageDataGenerator(rescale=1./255)
    test_generator = test_datagen.flow_from_directory(
        TEST_FOLDER,
        target_size=IMAGE_SIZE,
        batch_size=1,
        class_mode=CLASS_MODE,
        shuffle=False)
    return test_generator
def main():
    global classes
    print("\n*** ROZPOCZĘTO TESTOWANIE MODELU " + MODEL_NAME + "
        ***\n")
    model = keras.models.load_model('./' + MODEL_NAME)
    test_generator = testGenerator()
    labels = (test_generator.class_indices)
    labels = dict((v, k) for k, v in labels.items())
    steps=test_generator.samples//test_generator.batch_size
    test_generator.reset()
```

```

Y_pred = model.predict_generator(test_generator, steps,
                                verbose=1)
y_pred=[1*(x[0]>=0.5) for x in Y_pred]
cm = confusion_matrix(test_generator.classes, y_pred)
plotCM(cm)
print('')
print(classification_report(test_generator.classes, y_pred,
                            target_names=classes))
TN, FP, FN, TP = confusion_matrix(test_generator.classes,
                                  y_pred).ravel()

TPR = TP/(TP+FN)
TNR = TN/(TN+FP)
print('TPR: {0:d}'.format(int(round(100*TPR))))
print('TNR: {0:d}\n'.format(int(round(100*TNR))))
if __name__ == "__main__":
    main()

```