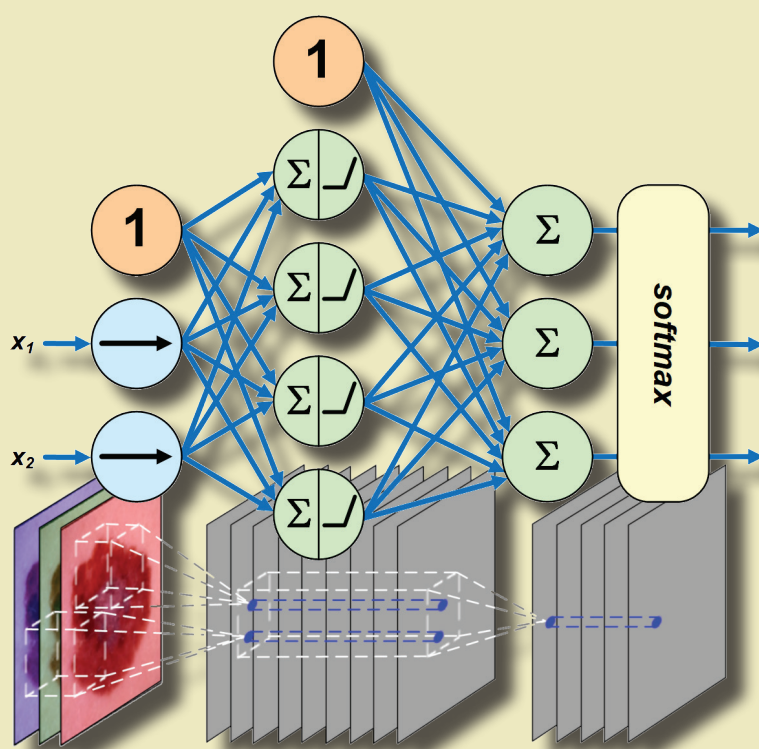




Zbigniew Omiotek

Wybrane problemy modelowania predykcyjnego w diagnostyce technicznej i medycznej



Wybrane problemy modelowania
predykcyjnego w diagnostyce
technicznej i medycznej

Monografie – Politechnika Lubelska



Politechnika Lubelska
Wydział Elektrotechniki i Informatyki
ul. Nadbystrzycka 38A
20-618 Lublin

Zbigniew Omiotek

Wybrane problemy modelowania predykcyjnego w diagnostyce technicznej i medycznej



Wydawnictwo Politechniki Lubelskiej
Lublin 2021

Recenzent:

prof. dr hab. inż. Ryszard Tadeusiewicz, Akademia Górniczo-Hutnicza

Publikacja wydana za zgodą Rektora Politechniki Lubelskiej

© Copyright by Politechnika Lubelska 2021

ISBN: 978-83-7947-455-4

Wydawca: Wydawnictwo Politechniki Lubelskiej
www.biblioteka.pollub.pl/wydawnictwa
ul. Nadbystrzycka 36C, 20-618 Lublin
tel. (81) 538-46-59

Druk: DjaF – 30-092 Kraków, ul. Kmietowicza 1/1
www.djaf.pl

Elektroniczna wersja książki dostępna w Bibliotece Cyfrowej PL www.bc.pollub.pl

Nakład: 50 egz.

Spis treści

1. Wstęp	11
2. Diagnostyka techniczna i medyczna	15
2.1. Diagnostyka techniczna	15
2.1.1. Istota diagnostyki technicznej	15
2.1.2. Diagnostyka procesów przemysłowych	16
2.1.3. Diagnostyka procesu spalania	19
2.2. Diagnostyka medyczna	20
2.2.1. Przedmiot i cel diagnostyki medycznej	20
2.2.2. Metody diagnostyki medycznej	21
2.2.3. Diagnostyka dermatologiczna	22
2.3. Podsumowanie	24
3. Podstawy uczenia maszynowego	27
3.1. Pojęcie i rodzaje uczenia maszynowego	27
3.2. Akwizycja obrazów	29
3.3. Wstępne przetwarzanie obrazów	30
3.4. Estymacja parametrów tekstury	31
3.4.1. Deskryptory jasności pikseli	32
3.4.2. Deskryptory wyznaczane z otoczeń pikseli	35
3.4.3. Deskryptory zbiorów par pikseli	36
3.4.4. Deskryptory pasm pikseli o jednakowym odcieniu	39
3.4.5. Deskryptory oparte na transformacie falkowej	40
3.5. Modelowanie predykcyjne	43
3.5.1. Wstępne przetwarzanie danych	43
3.5.2. Selekcja cech	44
3.5.3. Uczenie modelu	46
3.5.4. Ewaluacja modelu	47
3.5.5. Prognozowanie	52
4. Uczenie głębokie	55
4.1. Ogólne działanie i rodzaje sieci głębokich	55
4.2. Sztuczne sieci neuronowe	56

4.2.1. Preceptron	56
4.2.2. Perceptron wielowarstwowy	58
4.2.3. Szacowanie prawdopodobieństwa przynależności do klasy	60
4.2.4. Metoda gradientu prostego.....	62
4.2.5. Funkcje straty i uczenie modeli.....	64
4.2.6. Techniki wspierające uczenie modelu.....	65
4.2.7. Diagnostyka modelu na podstawie krzywych uczenia i walidacji ...	67
4.3. Sieci splotowe.....	70
4.3.1. Architektura sieci splotowej.....	70
4.3.2. Splot dwuwymiarowy	72
4.3.3. Filtry i mapy cech.....	73
4.3.4. Struktury danych przetwarzane przez sieć	77
5. Metodyka badań.....	83
5.1. Metody uczenia tradycyjnego.....	84
5.2. Metody uczenia głębokiego	85
6. Diagnostyka procesu współspalania pyłu węglowego i biomasy	89
6.1. Kontekst badań	89
6.2. Materiał do badań	91
6.3. Wstępne przetwarzanie obrazów płomienia	94
6.3.1. Analiza statystyczna składowych koloru	95
6.3.2. Adaptacyjna korekcja gamma	97
6.3.3. Dobór wielkości i ekstrakcja obszaru zainteresowania	98
6.4. Zastosowanie metod uczenia tradycyjnego oraz dużego zbioru danych.....	102
6.4.1. Wstępne przetwarzanie danych	102
6.4.2. Wykorzystanie pełnego zestawu cech.....	104
6.4.3. Wykorzystanie ograniczonego zestawu cech	108
6.5. Zastosowanie metod uczenia tradycyjnego oraz małego zbioru danych.....	110
6.6. Zastosowanie metod uczenia głębokiego oraz dużego zbioru danych ..	113
6.6.1. Wykorzystanie obrazów kolorowych.....	114
6.6.2. Wykorzystanie obrazów monochromatycznych	116
6.6.3. Wykorzystanie obrazów binarnych.....	117

6.7. Zastosowanie metod uczenia głębokiego oraz małego zbioru danych..	120
7. Diagnostyka zmian skórnych.....	123
7.1. Kontekst badań	123
7.2. Materiał do badań	126
7.3. Wstępne przetwarzanie obrazów zmian skórnych.....	127
7.4. Zastosowanie metod uczenia tradycyjnego oraz dużego zbioru danych.....	128
7.4.1. Wstępne przetwarzanie danych	128
7.4.2. Wykorzystanie pełnego zestawu cech.....	130
7.4.3. Wykorzystanie ograniczonego zestawu cech	133
7.5. Zastosowanie metod uczenia tradycyjnego oraz małego zbioru danych.....	134
7.6. Zastosowanie metod uczenia głębokiego oraz dużego zbioru danych ..	137
7.6.1. Wykorzystanie obrazów kolorowych.....	137
7.6.2. Wykorzystanie obrazów monochromatycznych	139
7.6.3. Wykorzystanie obrazów binarnych.....	141
7.7. Zastosowanie metod uczenia głębokiego oraz małego zbioru danych..	144
8. Analiza wyników badań.....	147
8.1. Wpływ liczby cech na dokładność klasyfikacji.....	147
8.2. Najlepsze metody selekcji i klasyfikacji.....	148
8.3. Najskuteczniejsze zestawy cech	151
8.4. Wrażliwość modeli na wielkość zbioru danych	155
8.5. Wpływ trybu koloru na dokładność klasyfikacji	158
8.6. Porównanie skuteczności modeli predykcyjnych.....	161
9. Podsumowanie	167
Literatura.....	171

Wykaz ważniejszych symboli i akronimów

Symbole

A	– metoda selekcji cech ANOVA
ADA	– metoda klasyfikacji AdaBoost
BAYES	– metoda klasyfikacji Gaussian Naive Bayes
F, FISHER	– metoda selekcji cech Fishera
GRAD	– metoda klasyfikacji Gradient Boosting
IRN	– model InceptionResNetV2
LGBM	– metoda klasyfikacji LightGBM
MN	– model MobileNetV2
R, RELIEF	– metoda selekcji cech Relief
RN	– model ResNet50
V16	– model VGG16
V19	– model VGG19
X	– model Xception

Akronimy

ACC	– <i>Accuracy</i>
AdaBoost	– <i>Adaptive Boosting</i>
AdaGrad	– <i>Adaptive Gradient Algorithm</i>
Adam	– <i>Adaptive Moment Estimation</i>
ANOVA	– <i>Analysis of Variance</i>
AUC	– <i>Area Under the Curve</i>
CNN	– <i>Convolutional Neural Network</i>
CNN	– <i>Convolutional Neural Network</i>
CT	– <i>Computed Tomography</i>
DBN	– <i>Deep Belief Networks</i>
DT	– <i>Decision Tree</i>
DWT	– <i>Discrete Wavelet Transform</i>
ERR	– <i>Error</i>
FDR	– <i>False Discovery Rate</i>
fMRI	– <i>functional Magnetic Resonance Imaging</i>
FPR	– <i>False Positive Rate</i>
GAN	– <i>Generative Adversarial Networks</i>
ISIC	– <i>International Skin Imaging Collaboration</i>
KNN	– <i>K-Nearest Neighbors</i>
LDA	– <i>Linear Discriminant Analysis</i>
LightGBM	– <i>Light Gradient Boosting Machine</i>
LR	– <i>Logistic Regression</i>

LTU	– <i>Linear Threshold Unit</i>
LUT	– <i>Look-Up Table</i>
MLP	– <i>Multi-Layer Perceptron</i>
MLP	– <i>Multi-Layer Perceptron</i>
MRI	– <i>Magnetic Resonance Imaging</i>
NPV	– <i>Negative Predictive Value</i>
NuSVM	– <i>Support Vector Machines</i> (z parametrem <i>nu</i> do sterowania liczbą wektorów nośnych)
PET	– <i>Positron Emission Tomography</i>
PPV	– <i>Positive Predictive Value</i>
PRE	– <i>Precision,</i>
QDA	– <i>Quadratic Discriminant Analysis</i>
RBM	– <i>Restricted Boltzmann Machines</i>
REC	– <i>Recall</i>
ReLU	– <i>Rectified Linear Unit</i>
RF	– <i>Random Forest</i>
RFE	– <i>Recursive Feature Elimination</i>
RL	– <i>Run-Length</i>
RLM	– <i>Run Length Matrix</i>
RMSprop	– <i>Root Mean Square Propagation</i>
RNN	– <i>Recurrent Neural Network</i>
ROC	– <i>Receiver Operating Characteristic Curve</i>
ROI	– <i>Region of Interest</i>
RTG	– <i>Rentgen</i>
SBS	– <i>Sequential Backward Selection</i>
SFS	– <i>Sequential Forward Selection</i>
SPECT	– <i>Single-Photon Emission Computed Tomography</i>
SVM	– <i>Support Vector Machines</i> (z parametrem regularyzacji <i>C</i>)
USG	– <i>Ultrasonography</i>

1. Wstęp

Tytuł niniejszej monografii – *Wybrane problemy modelowania predykcyjnego w diagnostyce technicznej i medycznej*, związany jest z czynnikami, które w istotny sposób wpływają na rozwój społeczny oraz komfort i jakość życia. Gwałtowny postęp w obszarze uczenia maszynowego (zwłaszcza uczenia głębokiego), jaki obserwujemy w ostatniej dekadzie, pozwala na budowę skutecznych modeli predykcyjnych, powszechnie wykorzystywanych w różnych systemach wspomagania decyzji. Modele te, z coraz większym powodzeniem, zastępują człowieka w zadaniach, w których liczy się szybkość, czas i niezawodność decyzji. Z kolei, diagnostyka techniczna i medyczna to dwa ważne obszary, w których trafna interpretacja stanu obiektów i zjawisk ma ogromne znaczenie.

Rozwój techniki sprawił, że zaawansowane systemy technologiczne wymagają szybkiej i precyzyjnej oceny zachodzących w nich procesów. W medycynie, różnorodne metody diagnostyczne dostarczają ogromnej ilości informacji, która powinna być skutecznie przetwarzana, w celu otrzymania trafnej decyzji diagnostycznej. W obu obszarach doskonale sprawdzają się uniwersalne narzędzia, jakimi są modele predykcyjne. Postęp w rozwoju metod i technik uczenia maszynowego sprawił, że są one coraz skuteczniejsze i stanowią coraz wartościowsze wsparcie decyzji podejmowanych przez człowieka. Diagnostyka techniczna i medyczna to obszary ściśle ze sobą powiązane. Pomimo innego przedmiotu zainteresowania, którym w pierwszym przypadku jest maszyna lub proces przemysłowy, a w drugim człowiek, w obu obszarach stosowane są takie same metody pomiarowe oraz metody wnioskowania. Wydaje się więc zasadne przedstawienie wybranych problemów modelowania predykcyjnego na przykładach z obszarów diagnostyki technicznej i medycznej.

W monografii można wyróżnić dwie części. Pierwsza stanowi wprowadzenie do obszernej dziedziny wiedzy, jaką jest uczenie maszynowe. Dokonano w niej przeglądu kluczowych problemów z tego obszaru. Na przykładzie klasyfikacji tekstur, scharakteryzowano poszczególne etapy modelowania, począwszy od akwizycji i wstępnego przetwarzania obrazów, poprzez estymację cech i wstępne przetwarzanie danych, do uczenia i ewaluacji modelu oraz prognozowania. Omówione zagadnienia stanowią kompleksowy opis procedury budowy modelu predykcyjnego. Podkreślono w nich kluczowy etap – wstępne przetwarzanie danych. Właściwe przygotowanie danych dla algorytmów uczących ma decydujący wpływ na budowę skutecznych modeli. Dużą uwagę poświęcono w monografii metodom uczenia głębokiego, które są rozszerzeniem metod tradycyjnych, zwanych często metodami klasycznymi lub płytkimi. Rozwinięciem części dotyczącej metod głębokich jest opis sieci splotowych, które wykorzystano w badaniach. Całość wspomnianej, pierwszej części monografii, wprowadza czytelnika w podstawową problematykę modelowania predykcyjnego, z wykorzystaniem

metod uczenia tradycyjnego i głębokiego. Jest to istotne, aby lepiej zrozumieć opisane w dalszej części badania.

Druga część monografii zawiera opis badań, w których na dwóch przykładach zademonstrowano proces modelowania predykcyjnego. Przykład pierwszy, z obszaru diagnostyki technicznej, dotyczy procesu współspalania pyłu węglowego i biomasy. Poruszony problem jest ważny z punktu widzenia ekonomiki i bezpieczeństwa procesu technologicznego, ale także z uwagi na przyjęte, krajowe i europejskie normy w zakresie ochrony środowiska. Drugi przykład, z dziedziny diagnostyki medycznej, związany jest z rozpoznawaniem zmian skórnych, a konkretnie dotyczy on rozróżniania czerniaka złośliwego skóry od niezłośliwych znamion melanocytowych. Waga tego zagadnienia, jako problemu diagnostycznego, jest bardzo duża. Czerniak jest najgroźniejszym nowotworem skóry. Stosunkowo często, jego źródłem są atypowe znamiona dysplastyczne o podłożu melanocytowym, w których z czasem mogą zachodzić zmiany prowadzące do rozwoju nowotworu. Kluczowe znaczenie ma wykrycie czerniaka w jak najwcześniejszym stadium rozwoju, co nadal stanowi duże wyzwanie dla współczesnych metod diagnostyki dermatologicznej.

W opisie przeprowadzonych badań zwrócono uwagę na wybrane techniki, mające istotny wpływ na skuteczność procesu modelowania. W przypadku metod uczenia tradycyjnego, są to procedury wstępnego przetwarzania danych oraz strojenia hiperparametrycznego modeli, z wykorzystaniem metody przeszukiwania siatki. Opisując wyniki metod uczenia głębokiego, podkreślono znaczenie podejścia określanego mianem *transfer learning*¹ oraz diagnostyki modeli w oparciu o krzywe trenowania i walidacji. Należy w tym miejscu zaznaczyć, iż wszystkie zaprezentowane badania, zaplanowano i zrealizowano wyłącznie na potrzeby monografii, a ich wyniki nie zostały dotychczas nigdzie opublikowane.

Przed niniejszym opracowaniem zostały postawione dwa cele. Pierwszym, jak wspomniano wcześniej, było wprowadzenie czytelnika w podstawową problematykę modelowania predykcyjnego. Drugim celem było znalezienie odpowiedzi na kilka pytań, które zdaniem autora wydają się istotne, zwłaszcza dla osób nie posiadających dużego doświadczenia w uczeniu maszynowym. Niżej wymieniono postawione problemy oraz krótki komentarz.

1. **W jaki sposób ograniczenie liczby cech diagnostycznych wpływa na skuteczność modeli uczenia tradycyjnego?** Optymalizacja hiperparametryczna modeli, z wykorzystaniem metody przeszukiwania siatki, pozwala w sposób automatyczny znaleźć optymalną liczbę cech, zapewniającą najwyższą skuteczność modelu dla danych walidacyjnych. Okazuje się jednak, że wykres krzywej skuteczności w funkcji liczby cech bardzo często wykazuje tendencję do nasycenia już dla niewielkiej ich liczby. Na podstawie tego wykresu można ograniczyć liczbę cech (np. z dwudziestu do dwóch lub trzech),

¹ *Transfer learning* to technika polegająca na wykorzystaniu modeli, które zostały wcześniej wytrenowane na dużym zbiorze danych.

zmniejszając złożoność modelu, ale jednocześnie tracąc nieco na jego skuteczności. Wiedza na temat wpływu liczby cech na dokładność modelu może pomóc w osiągnięciu kompromisu pomiędzy złożonością modelu, a jego jakością.

2. **Jakie są najskuteczniejsze metody selekcji i klasyfikacji, które można zastosować do rozwiązywania postawionych zadań diagnostycznych?** Opracowano dotychczas dużą liczbę algorytmów selekcji cech oraz klasyfikacji i trudno *a priori* zakładać większą skuteczność jednych metod, w stosunku do innych. Często, najlepszą praktyką jest zastosowanie możliwie dużej liczby różnych technik i porównanie ich wyników. Jakość klasyfikatora oceniamy na podstawie jego dokładności dla danych testowych, ale każdy klasyfikator zostaje wytrenowany na podstawie zestawu cech wyłonięnego przez konkretną metodę selekcji. Zatem, wybierając rozwiązanie końcowe, zawsze wskazujemy parę metod, odpowiedzialnych za wybór cech oraz uczenie modelu. Wskazanie takiej pary, w oparciu o dane odpowiadające obrazom płomienia i obrazom zmian skórnych, pozwoli je zarekomendować do zastosowań takich samych oraz podobnych, w kontekście własności wykorzystanych tekstur obrazu.
3. **Jakie są najskuteczniejsze zestawy cech, które można wykorzystać do budowy modeli klasyfikacyjnych?** Jest to w pewnym sensie kontynuacja wątku rozpoczętego w ramach poprzedniego pytania. Wskazanie najskuteczniejszej metody selekcji i optymalnej liczby cech daje możliwość jednoznacznego określenia zestawu cech. Ich znajomość pozwala dokonać implementacji modelu, w celu realizacji prognozowania nowych obserwacji. Podczas klasyfikacji obrazów teksturowych, początkowa liczba cech może być bliska trzystu, natomiast w wyniku redukcji wymiaru danych, liczba ta spada przeważnie do kilku lub kilkunastu.
4. **Jak wielkość zbioru danych wpływa na skuteczność modeli uczenia tradycyjnego i głębokiego?** Wielkość zbioru danych to bardzo ważny problem w praktyce uczenia maszynowego. Z jednej strony, zbiór danych powinien być dostatecznie duży, tak aby był reprezentatywny dla całej populacji obserwacji. Jednak z drugiej strony, nadmiernie duży zbiór może ograniczyć zdolność generalizacyjną modeli zbudowanych na jego podstawie. Istnieją wprawdzie zależności analityczne, pozwalające obliczyć wielkość zbioru treningowego, np. w oparciu o liczbę cech, ale zgromadzenie odpowiednio dużego zbioru danych, np. w obszarze badań technicznych, czy medycznych, jest często zadaniem bardzo trudnym lub wręcz niewykonalnym. W takiej sytuacji, świadomie korzystamy ze zbioru mniejszego, niż oczekiwany oraz wybieramy rozwiązania, które dobrze sprawdzają się dla mniejszej liczebności obserwacji. Do takich rozwiązań należy, np. technika *transfer learning* w uczeniu głębokim. Dlatego, aby podjąć decyzję dotyczącą wyboru metod i narzędzi, warto mieć wiedzę na temat wrażliwości różnych algorytmów uczenia maszynowego na wielkość zbioru danych.

5. **Jak tryb koloru obrazów wpływa na dokładność modeli uczenia głębokiego oraz które z tych modeli są najskuteczniejsze dla poszczególnych trybów koloru?** Do przetwarzania obrazów z wykorzystaniem algorytmów uczenia głębokiego stosowane są przeważnie sieci splotowe. Biblioteka Keras zawiera bogaty zbiór różnych architektur takich sieci. Charakteryzują się one różną złożonością oraz różną skutecznością dla tych samych danych wejściowych. Wiedza na ten temat, uzyskana na podstawie analizy obrazów płomienia oraz obrazów zmian skórnych, może pomóc w doborze architektury odpowiedniej dla danego zastosowania. Obrazy kolorowe, w porównaniu z innymi trybami koloru, mają największą nośność informacyjną. Dlatego, intuicyjnie spodziewamy się, że modele wyuczone na takich obrazach będą najskuteczniejsze. Często nie mamy wyboru i dysponujemy obrazami z konkretnym trybem koloru, gdyż jest on immanentnie związany z metodą ich pozyskiwania (np. obrazy monochromatyczne w obrazowaniu metodami RTG, CT, czy USG). Mogą być jednak sytuacje, gdy mamy wpływ na tryb koloru obrazów poddawanych akwizycji (np. w systemach monitorowania procesów przemysłowych). Wówczas, właściwy wybór trybu koloru oraz wiedza na temat skutecznej architektury sieci splotowej może pomóc w osiągnięciu lepszych wyników.
6. **Które podejście – oparte na uczeniu tradycyjnym, czy głębokim – jest skuteczniejsze w rozwiązywaniu postawionych zadań diagnostycznych?** Odpowiedź na to pytanie nie jest oczywista. Modele uczenia głębokiego charakteryzują się wysoką dokładnością, pozwalają także uprościć budowę systemu diagnostycznego, poprzez eliminację etapów estymacji cech i wstępnego przetwarzania danych. Z kolei, prawidłowo przeprowadzony proces wstępnego przygotowania danych dla metod uczenia tradycyjnego może sprawić, że skuteczność wyuczonych za ich pomocą modeli będzie porównywalna ze skutecznością modeli uczenia głębokiego. Nie można z góry zakładać, które podejście jest lepsze dla konkretnego zastosowania. Może się okazać, że liczebność obserwacji jest zbyt mała, aby uzyskać zadowalającą skuteczność modeli uczenia głębokiego, a modele uczenia tradycyjnego mogą sobie w takiej sytuacji poradzić lepiej. Uzyskanie odpowiedzi na postawione pytanie pozwoli rekomendować podejście odpowiednie do zastosowań takich samych oraz podobnych, ze względu na właściwości danych i liczebność obserwacji.
- Zakres eksperymentów zaplanowano tak, aby uzyskane wyniki oraz ich analiza pozwoliły odpowiedzieć na postawione pytania. Jednocześnie, należy mieć świadomość tego, że badania przeprowadzono z wykorzystaniem obrazów płomienia oraz zmian skórnych. Dlatego, sformułowane wnioski można uogólnić do takich zastosowań, w których zbiory danych mają podobne właściwości oraz liczebność, co dane wykorzystane w badaniach. Autor wyraża nadzieję, iż informacje zawarte w niniejszej monografii ułatwią potencjalnym czytelnikom podejmowanie decyzji istotnych z punktu widzenia praktyki modelowania predykcyjnego.

2. Diagnostyka techniczna i medyczna

2.1. Diagnostyka techniczna

2.1.1. Istota diagnostyki technicznej

Słowo **diagnostyka** pochodzi z języka greckiego, w którym *diagnostikos* oznacza umiejący rozpoznawać, a *diagnōsis* to rozpoznanie, diagnoza, decyzja. Według słownika wyrazów obcych i zwrotów obcojęzycznych Władysława Kopalińskiego, **diagnostyka** to dziedzina, która zajmuje się rozpoznawaniem badanego stanu rzeczy przez zaliczanie go do znanego typu lub gatunku, przez przyczynowe i całościowe wyjaśnienie tego stanu rzeczy, określenie jego fazy obecnej oraz przewidywanego dalszego rozwoju [1]. Początkowo, słowo to kojarzono z medycyną, a konkretnie, z jej działem zajmującym się metodami rozpoznawania chorób na podstawie ich objawów. Jednak, w miarę rozwoju techniki, znaczenie słowa diagnostyka rozszerzono także na ten obszar działalności człowieka. Powstało w ten sposób pojęcie **diagnostyki technicznej**, które według leksykonu Żółtowskiego i Ćwika, jest dziedziną wiedzy obejmującą całokształt zagadnień teoretycznych i praktycznych dotyczących identyfikacji i oceny przeszłych, aktualnych i przyszłych stanów obiektu technicznego, z uwzględnieniem jego otoczenia [2].

Przedmiotem diagnostyki technicznej mogą być maszyny lub procesy przemysłowe. W związku z tym, wyróżniamy dwie dziedziny:

1. **Diagnostyka maszyn** – zajmuje się oceną stanu urządzeń mechanicznych przez badania bezpośrednie ich własności i badania pośrednie procesów towarzyszących funkcjonowaniu tych urządzeń, tzw. procesów resztowych (mogą one mieć charakter mechaniczny, elektryczny, termiczny, itp.). Szczególną rolę odgrywają procesy wibroakustyczne (drgania i hałas) [3, 4, 5, 6, 7, 8].
2. **Diagnostyka procesów przemysłowych** – zajmuje się rozpoznawaniem zmian stanów tych procesów, których przyczynami są uszkodzenia i inne zdarzenia destrukcyjne (np. zużycie). Zadaniem jest wczesne wykrywanie i dokładne rozpoznawanie (rozdzielanie) powstających uszkodzeń [9, 10, 11, 12].

W ostatnich dziesięcioleciach nastąpił szybki rozwój technologiczny w wielu obszarach. Jego wynikiem jest stworzenie i eksploatacja systemów produkcyjnych, w których występują złożone procesy przemysłowe. Aby zapewnić ciągłą, bezawaryjną pracę systemów, dużą wagę przywiązuje się do rozwoju metod diagnostyki wspomnianych procesów. Głównym zadaniem diagnostyki procesów przemysłowych jest odpowiednio wczesne rozpoznanie objawów stanów awaryjnych. Ma to szczególne znaczenie tam, gdzie takie stany wiążą się z zagrożeniem bezpieczeństwa ludzi lub środowiska, czy też z dużymi stratami fi-

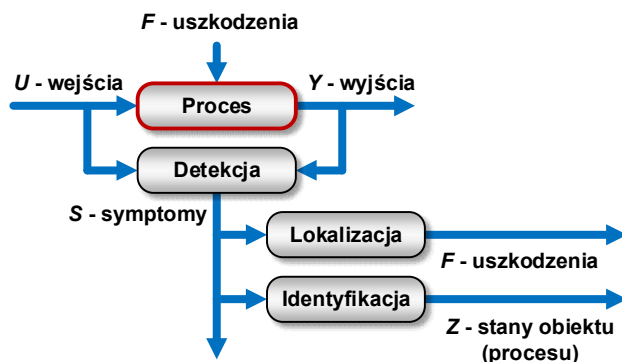
nansowymi. Obecnie, diagnostyka procesów przemysłowych realizowana jest głównie w sposób automatyczny. Dzięki temu, czas wykrycia i lokalizacji awarii może być istotnie skrócony, w porównaniu z diagnostyką realizowaną przez człowieka. Wyeliminowane zostają także ewentualne błędy interpretacji wielu różnych sygnałów pojawiających się jednocześnie, które mogłyby być popełnione przez ludzi. Poza tym, można zwiększyć efektywność ekonomiczną danego procesu, ograniczyć skutki awarii, uzyskać efekty tolerowania niektórych uszkodzeń oraz zmniejszyć koszty remontów.

2.1.2. Diagnostyka procesów przemysłowych

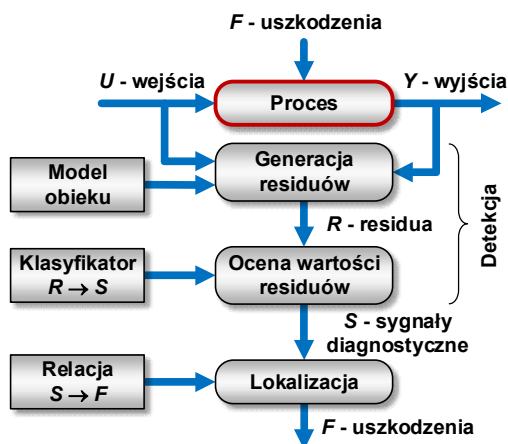
Podczas procesu diagnostycznego ma miejsce obserwacja oddziaływań pomiędzy otoczeniem oraz badanym obiektem. Obserwacja jest realizowana za pośrednictwem sygnałów. **Sygnałem** jest przebieg dowolnej wielkości fizycznej będącej nośnikiem informacji. W celu jej pozyskania, wyznaczane są wartości wybranych cech sygnałów, zwanych **zmiennymi procesowymi**. Na podstawie tych zmiennych można wyznaczyć **sygnały diagnostyczne** – są to przebiegi dowolnych wielkości będących nośnikami informacji o stanie diagnozowanego obiektu (lub procesu). Pojęcie **diagnozowanie** oznacza proces wykrywania i rozróżniania uszkodzeń obiektu, w wyniku zbierania, przetwarzania, analizy i oceny sygnałów diagnostycznych.

Ogólnie, w diagnostyce uszkodzeń można wyróżnić 3 fazy, są to: **detekcja**, **lokalizacja** oraz **identyfikacja**. Proces diagnostyczny jest najczęściej ograniczony do pierwszych dwóch faz. W fazie detekcji, przetwarzane są zmienne procesowe i w efekcie uszkodzenia są wykrywane. Następnie, w fazie lokalizacji, określany jest rodzaj uszkodzeń oraz miejsce i czas ich wystąpienia. Identyfikacja, jeżeli występuje, polega na określeniu rozmiaru uszkodzenia oraz jego zmienności w czasie. Zamiast fazy lokalizacji, może wystąpić faza rozpoznawania stanu procesu lub klasy stanów (rys. 2.1). Pod pojęciem **stan procesu**, rozumiemy zbiór ocen odchyłeń pomiędzy danym procesem, a procesem wzorcowym. Opracowano wiele metod realizacji procesu diagnostycznego. Najczęściej stosowane scharakteryzowano w dalszej części rozdziału.

Wykorzystanie modelu matematycznego. Schemat takiego podejścia, obejmującego fazę detekcji i lokalizacji, przedstawiono na rys. 2.2. Podczas fazy detekcji, generowane są residua z wykorzystaniem, np. modeli neuronowych, analitycznych, czy rozmytych. Ocena wartości residuów, rozumianych jako różnice między rzeczywistymi wartościami zmiennych procesowych, a wartościami estymowanymi przez model, odbywa się z wykorzystaniem klasyfikatora. Dzięki temu, można wykryć nieprawidłowy stan obiektu lub procesu.

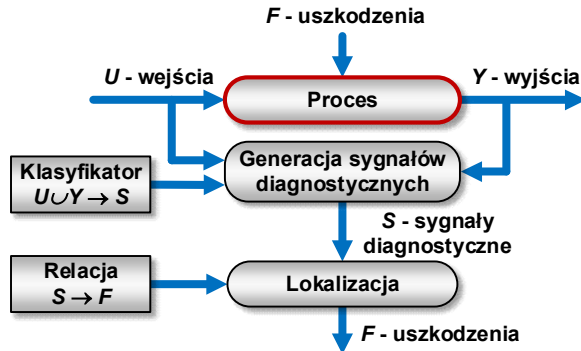


Rys. 2.1. Realizacja procesu diagnostycznego obejmująca fazę detekcji uszkodzeń oraz fazę lokalizacji lub rozpoznawania stanu procesu [10]



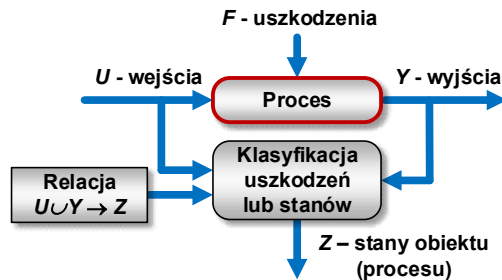
Rys. 2.2. Realizacja procesu diagnostycznego z wykorzystaniem modelu

Proces diagnostyczny bez użycia modelu. Diagnostowanie z detekcją bez użycia modelu stosuje się wówczas, gdy model procesu nie jest wystarczająco dokładny lub jest zbyt złożony (rys. 2.3). Stosowane są wówczas metody kontroli ograniczeń lub kontroli związków występujących między zmiennymi procesowymi.



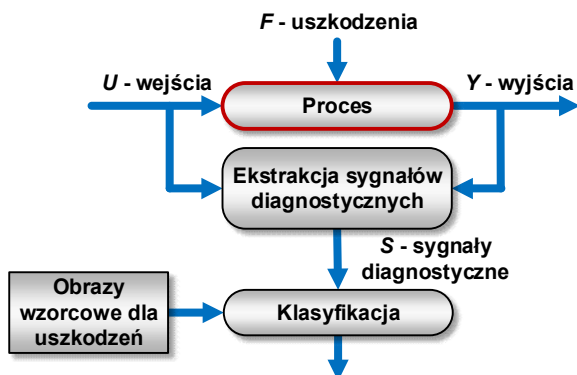
Rys. 2.3. Realizacja procesu diagnostycznego bez użycia modelu

Połączenie detekcji i lokalizacji. Jest to metoda wykorzystująca ciągle zmienne procesowe, na które składają się sygnały wejściowe i wyjściowe (rys. 2.4). Fazy detekcji i lokalizacji uszkodzeń są połączone, a klasyfikacja stanu obiektu lub procesu może być realizowana, np. z wykorzystaniem sieci neuronowych.



Rys. 2.4. Realizacja procesu diagnostycznego łączącego detekcję i lokalizację

Rozpoznawanie wzorców. W tym podejściu, w fazie detekcji uszkodzeń realizowana jest ekstrakcja symptomów, a w fazie lokalizacji uszkodzeń dokonuje się klasyfikacji uszkodzeń lub stanu obiektu (rys. 2.5). Ekstrakcja symptomów może się odbywać na podstawie, np. modeli obiektu, analizy widmowej wybranych sygnałów, analizy statystycznej wybranych sygnałów, czy kontroli ograniczeń.



Rys. 2.5. Realizacja procesu diagnostycznego z wykorzystaniem rozpoznawania wzorców

2.1.3. Diagnostyka procesu spalania

Proces produkcji energii elektrycznej jest znakomitym przykładem procesów złożonych o wielkiej skali, w przypadku których ogromną rolę odgrywa diagnostyka. W elektrowniach węglowych, pierwotnym źródłem energii jest węgiel, spalany najczęściej w palnikach pyłowych. Newralgicznym węzłem jest tam kocioł, w którym zachodzą procesy wpływające zarówno na ekonomikę wytwarzania energii, jak i zanieczyszczenie środowiska naturalnego. Aby proces produkcji energii spełniał założone kryteria, a jednocześnie był bezpieczny w eksploatacji, musi być osiągnięty kompromis pomiędzy różnymi (często przeciwnymi) wymaganiami [13].

Jakość realizacji procesu spalania jest obecnie oceniana subiektywnie przez operatorów, na podstawie obrazu płomienia widocznego w otworze inspekcyjnym. Stosowane są wprawdzie układy monitorowania, ale ich głównym przeznaczeniem jest wykrywanie stanów zaniku płomienia, grożących niekontrolowanym wybuchem pyłu węglowego. Diagnostyka płomienia pyłowego jest zadaniem trudnym, ponieważ prędkość mieszanki paliwowej na wylocie palnika jest tak duża, że spalanie zachodzi w płomieniu turbulentnym. Dotychczas, nie opracowano metod diagnostyki takiego płomienia, które pozwalałyby jednoznacznie i obiektywnie ocenić jakość procesu spalania. Dodatkowo, diagnostykę procesu spalania komplikuje coraz powszechniejsze stosowanie paliw alternatywnych, np. biomasy jako dodatku do pyłu węglowego. Wynika to z faktu, iż paliwa tego rodzaju charakteryzują się dużą zmiennością składu, co znacząco modyfikuje proces spalania w urządzeniach przemysłowych. W efekcie, istniejące metody diagnostyki muszą być dostosowane do innej mieszanki paliwowej.

Nakreślona wyżej problematyka, jest nadal aktywnym obszarem badawczym. W dalszym ciągu, wyzwaniem jest opracowanie skutecznej metody diagnostyki płomienia turbulentnego, zwłaszcza podczas spalania mieszanki pyłu węglowego.

go i biomasy. Należy też zauważyć, iż w rzeczywistości, jednocześnie może pracować nawet kilkadziesiąt palników (zależnie od mocy i konstrukcji kotła). Aby prawidłowo ocenić przebieg procesu spalania, każdy z tych palników powinien być wyposażony w układ monitorowania i diagnostyki, co dodatkowo komplikuje całą procedurę diagnostyczną. Wdrożenie indywidualnego sterowania palnikami dałoby możliwość sterowania procesem spalania w sposób bliski optymalnemu [14, 15]. Ciągłe istnieje wiele wyzwań w obszarze diagnostyki procesu spalania. Nowe metody powinny, z jednej strony, spełniać wymagania współczesnej energetyki w zakresie ekonomiki i ochrony środowiska, a z drugiej zabezpieczyć proces produkcji energii elektrycznej przed powstawaniem sytuacji awaryjnych.

2.2. Diagnostyka medyczna

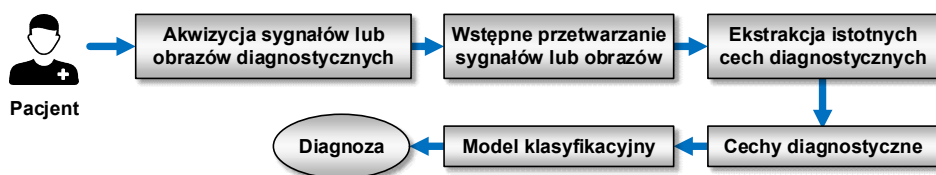
2.2.1. Przedmiot i cel diagnostyki medycznej

W przypadku diagnostyki medycznej, przedmiotem zainteresowania jest człowiek (pacjent). Podczas procedury diagnostycznej, muszą być brane pod uwagę takie rzeczy, jak jego grupa wiekowa, płeć, rasa, oraz stan zdrowia. Niezdeteminowanie pod względem występowania cech natury osobniczej pacjenta sprawia, że podczas wnioskowania diagnostycznego muszą być stosowane metody statystyczne.

Diagnostyka medyczna ma na celu rozpoznanie stanu zdrowia pacjenta na podstawie wyników badania lekarskiego. Przebieg procesu diagnostycznego jest w istocie podobny do tego, który występuje w diagnostyce technicznej. Jego zasadnicze etapy to:

- pomiar wartości sygnałów lub rejestracja obrazów diagnostycznych (stosowane są różne metody diagnostyczne);
- wnioskowanie diagnostyczne (interpretacja sygnałów lub obrazów);
- opracowanie diagnozy.

Rozwój technik komputerowych oraz metod sztucznej inteligencji sprawił, że coraz powszechniej stosowane są automatyczne systemy diagnostyczne, wspierające lekarzy w diagnozie przypadków trudnych do rozpoznania. Ogólną budowę takiego systemu przedstawiono na rysunku 2.6.



Rys. 2.6. Ogólna budowa automatycznego systemu diagnostyki medycznej

2.2.2. Metody diagnostyki medycznej

Metody pozyskiwania informacji do celów diagnostycznych są różnorakie. Ogólnie, można je podzielić na:

- badania podmiotowe (wywiad chorobowy);
- badania przedmiotowe (oglądanie, obmacywanie, opukiwanie i osłuchiwanie);
- badania dodatkowe (laboratoryjne, czynnościowe, obrazowe).

Wywiad chorobowy ma na celu pozyskanie wstępnych informacji o pacjencie. Ich źródłem może być sam pacjent, jego opiekunowie, a także dotychczasowa dokumentacja medyczna. Wyniki wywiadu pomagają opracować plan i zakres badań oraz ułatwiają opracowanie diagnozy.

Badania laboratoryjne (diagnostyka analityczna) są podstawową i zarazem jedną z najstarszych metod diagnostycznych. Polegają one na pobraniu i ocenie, w warunkach laboratoryjnych, fragmentu tkanki (narząd wewnętrzny, wycinek skóry) lub płynów ustrojowych (krew, mocz). Na podstawie ich składu jakościowego, ilościowego oraz chemicznego można uzyskać informacje o stanie i funkcjonowaniu poszczególnych układów wewnętrznych organizmu.

Badania czynnościowe, jak wskazuje nazwa, polegają na ocenie czynności wybranych organów. Istnieje bardzo szeroka gama takich badań, należą do nich, m.in.: elektroencefalografia, elektrokardiografia, elektromiografia, elektroneurografia, termografia podczerwona i mikrofalowa, echokardiografia. Do grupy badań czynnościowych można także zaliczyć **badania wydolnościowe** oraz **badania pod rzeczywistym obciążeniem**. Celem pierwszych jest pomiar wartości pewnych charakterystycznych parametrów (np. tętna, ciśnienia krwi, zużycia tlenu) podczas kontrolowanego obciążenia organizmu pacjenta. Wykorzystuje się w tym celu, np. rowery treningowe (cyklometry) i ruchome bieżnie. W przypadku badań pod rzeczywistym obciążeniem, rejestrowane są głównie podstawowe parametry pracy serca oraz ciśnienia tętniczego podczas codziennej aktywności pacjenta. Badania tego typu realizowane są metodą Holtera, polegającą na 24-godzinnej obserwacji organizmu pacjenta.

Badania obrazowe mają na celu pokazanie budowy wybranych fragmentów organizmu. Metody (modalności) obrazowania można podzielić, w zależności od wykorzystywanych własności fizycznych tkanek, na następujące grupy:

- metody wykorzystujące promieniowanie elektromagnetyczne w zakresie widzialnym i podczerwieni:
 - obrazowanie w promieniowaniu widzialnym (fotogrametria, koherentna tomografia optyczna, endoscopia, endoscopia kapsułkowa, funduskamera);
 - obrazowanie w promieniowaniu podczerwonym (termografia).
- metody wykorzystujące własności mechaniczne:
 - ultrasonografia (USG).
- obrazowanie w promieniowaniu jonizującym:
 - transmisyjne:
 - rentgenografia (RTG);

- tomografia komputerowa (CT);
- emisyjne:
 - scyntygrafia;
 - tomografia emisyjna pojedynczego fotonu (SPECT);
 - pozytonowa tomografia emisyjna (PET);
- obrazowanie magnetycznego rezonansu jądrowego (MRI).

Część wymienionych wyżej metod obrazowych pozwala wykonywać także badania czynnościowe. Należą do nich: SPECT, PET, PET-CT oraz fMRI (funkcjonalny rezonans magnetyczny). Podstawą działania tych metod jest fizjologiczny proces gromadzenia znacznika izotopowego w tkance badanego narządu. Dzięki rejestracji promieniowania emitowanego przez znacznik, można ocenić wielkość i położenie narządu. Stany chorobowe zaburzają proces gromadzenia znacznika, w efekcie zmienia się obraz struktury narządu. Zatem, czynność pełniona przez narząd pozwala uzyskać obrazową informację diagnostyczną.

Sygnały lub obrazy zarejestrowane przez metody diagnostyczne poddawane są procesowi wnioskowania, mającemu na celu opracowanie diagnozy. Do najbardziej znanych technik wnioskowania diagnostycznego należą tabele i grafy diagnostyczne, systemy ekspertowe oraz systemy wykorzystujące modele uczenia maszynowego. **Tabele diagnostyczne** zawierają zbiór objawów chorobowych wraz z przypisanymi im możliwymi przyczynami. Gdy jednej chorobie towarzyszy wiele objawów, tabela staje się nieczytelna i trudna do interpretacji. Stosowane są wówczas **grafy diagnostyczne**, pokazujące związki między wieloma objawami choroby i możliwymi przyczynami. **Systemy ekspertowe** to programy komputerowe zawierające zarówno bazę wiedzy w danym obszarze medycyny, jak również bazę reguł wiążących ze sobą poszczególne fakty i zjawiska. **Modele uczenia maszynowego** budowane są z wykorzystaniem algorytmów uczenia tradycyjnego i głębokiego. Dzięki dużemu potencjałowi, stają się one coraz bardziej popularne w medycznych systemach wspomagania decyzji. Przykładem są sztuczne sieci neuronowe, które są często stosowane jako moduły wnioskowania diagnostycznego w aparaturze medycznej.

2.2.3. Diagnostyka dermatologiczna

Pojawienie się dermatoskopii, jako narzędzia diagnostyki dermatologicznej, umożliwiło skuteczniejsze, niż dotychczas, wspomaganie lekarzy. Technika ta powiększa i rejestruje struktury występujące na skórze, eliminując jednocześnie odbłaski światła występujące na jej powierzchni (rys. 2.7). Pomaga to w odróżnieniu, np. czerniaka od innych rodzajów raka oraz niezłośliwych zmian skórnych. Możliwość ta jest szczególnie pożyteczna wówczas, gdy diagnozowana tkanka ma niewielkie rozmiary.



Rys. 2.7. Przykładowe stanowisko do badań dermatoskopowych². Na ekranie monitora widoczny jest powiększony obraz zmiany skórnej. Z prawej strony – zdjęcie wideodermatoskopu

Badania wykazały, że przy odpowiednim przeszkoleniu, dokładność diagnostyczna oparta na dermatoskopii wynosi ok. 75÷85% [16, 17, 18]. Dermatolodzy wykonują także fotografię skóry całego ciała. Ma to na celu wczesne wykrycie ewentualnych zmian tkanki oraz uniknięcie biopsji w sytuacji, gdy nie jest ona potrzebna. W celu poprawy skalowalności badania dermatoskopowego, opracowano różne algorytmy proceduralne, takie jak: metoda ABCDE [19], metoda Menziesa [20], analiza wzorców [17, 21], analiza tekstury [22] oraz 7-punktowa lista kontrolna [23]. Metody te dokonują porównania koloru, kształtu, symetrii oraz tekstury pomiędzy badaną tkanką, a skórą normalną [24]. Wielu lekarzy rezygnuje jednak z tych metod na rzecz osobistego doświadczenia [25].

Ze względu na duży potencjał, coraz popularniejsze stają się techniki automatycznej oceny obrazów dermatoskopowych [26, 27, 28, 29]. Kluczowe etapy w tym obszarze to: akwizycja obrazu, przetwarzanie wstępne, segmentacja, ekstrakcja cech oraz klasyfikacja. Przetwarzanie wstępne stosuje się, m.in., w celu usunięcia z obrazów szumów i różnych artefaktów, takich jak włosy, linijki i inne oznaczenia oraz pęcherzyki powietrza. Przykładem takich metod jest algorytm usuwania owłosienia o nazwie DullRazor [30], metoda filtracji medianowej [31], czy algorytm wykorzystujący filtry kierunkowe [32]. Segmentacja polega na oddzieleniu zmienionej tkanki od otaczającej ją skóry, w celu przeprowadzenia analizy i skutecznej ekstrakcji cech. Na podstawie wyłonionych cech, moż-

² Źródło: <https://www.fotofinder.de/technologie/hautkrebsdiagnostik/vexia/>

liwa jest ocena rodzaju zmiany skórnej. Obecnie, odbywa się to najczęściej z wykorzystaniem reguły ABCD lub 7-punktowej listy kontrolnej, która dokładnie charakteryzuje zmianę odpowiadającą czerniakowi. Bardziej popularna jest jednak reguła ABCD. Jest ona oparta na takich cechach, jak: asymetria, nieregularność granic oraz zmienność koloru i średnicy zmiany skórnej. Często, wyodrębniane są także cechy kolorystyczne i teksturalne [33], dające większą dokładność oceny tkanki. Statystyki składowych koloru obejmują średnią i odchylenie standardowe, wariancję, histogram oraz asymetrię. Z kolei, cechy tekstury oparte są głównie na macierzy gradientu obrazu oraz macierzy przejść tonalnych [22, 34]. Do klasyfikacji stosowane są różne algorytmy, takie jak SVM [35, 36, 37], naiwny klasyfikator Bayes'a [22], metoda K-NN [38], regresja logistyczna [39] oraz drzewa decyzyjne. Przegląd wymienionych, tradycyjnych metod uczenia maszynowego można znaleźć, m.in., w pozycjach [22, 40, 41]. Niestety, ze względu na dużą wewnątrzklasową i małą międzyklasową wariancję cech czerniaka, sprawność diagnostyki opartej na ręcznej ekstrakcji cech jest nadal niezadowalająca [42]. Dokładność, możliwa do uzyskania przy wykorzystaniu tego podejścia, jest przeważnie w zakresie ok. 80÷90% [43, 44, 45] i rzadko go przekracza [46, 47, 48]. Ponadto, ocena i porównanie uzyskanych wyników jest często problematyczne, ponieważ poszczególne grupy badawcze wykorzystują własne bazy danych obrazów. Wszystko to sprawia, że poszukiwanie skutecznych metod w diagnostyce dermatologicznej jest nadal obszarem aktywnych badań. Wydaje się, że dużą rolę mogą tutaj odegrać metody uczenia głębokiego wykorzystujące sieci plotowe.

2.3. Podsumowanie

Podstawowa różnica, jaka występuje pomiędzy diagnostyką techniczną i medyczną, to przedmiot zainteresowania. Przypomnijmy, w pierwszym przypadku jest nim maszyna lub proces przemysłowy, a w drugim żywy człowiek. Jest jednak wiele podobieństw pomiędzy diagnostyką techniczną i medyczną. W obu przypadkach, przedmiot diagnostyki generuje podczas swojego życia szereg mierzalnych sygnałów, które mogą być wykorzystane jako sygnały diagnostyczne. Metody pomiarowe tych sygnałów wykorzystują te same zasady i prawa fizyczne. Taki sam jest również przebieg samego procesu diagnostycznego oraz metody wnioskowania prowadzące do opracowania diagnozy. Wydaje się zatem, że badania prowadzone we wspólnym obszarze zainteresowań, w którym mieszczą się, np. czujniki i przetworniki do celów diagnostycznych, metody pomiarowe, czy metody wnioskowania, mogą przyczynić się do jednoczesnego rozwoju zarówno diagnostyki technicznej, jak i medycznej. Na uwagę zasługuje praca J. Langmana i Z. Ślipka pt. *Diagnostyka techniczna i diagnostyka medyczna – podobieństwa i zapożyczenia*, w której autorzy dokonali interesującego porównania obu rodzajów diagnostyki (tabela 2.1).

Tabela 2.1. Porównanie diagnostyki technicznej i medycznej [49]

Kryterium	Diagnostyka techniczna	Diagnostyka medyczna
Diagnozowany obiekt	Maszyna lub proces – zdeterminowany, pozbawiony cech osobniczych	Żywy człowiek – niezdeterminowany, ma cechy osobnicze
Źródło informacji o obiekcie	Użytkownicy, książka serwisowa, elektroniczny rejestr zdarzeń	Pacjent, jego opiekunowie, dokumentacja medyczna
Diagnostyka analityczna	Analiza płynów i materiałów eksploatacyjnych	Analiza płynów ustrojowych
Diagnostyka obrazowa	Badania materiałów konstrukcyjnych i spoin spawalniczych promieniami X, defektoskopia ultradźwiękowa	Rentgenografia, ultrasonografia
Endoskopia	Obserwacja niedostępnych części i elementów urządzeń	Obserwacja rzeczywistego obrazu niektórych narządów człowieka
Metody elektryczne	Diagnoskopy silnikowe (analiza zmian napięcia w przewodach zasilających świece zapłonowe)	Elektrokardiografia, elektroencefalografia, itd.
Badania wydolnościowe	Hamownie silnikowe i podwoziowe dla pojazdów	Rowery treningowe, ruchome bieżnie
Pomiary pod rzeczywistym obciążeniem	Rejestratory podstawowych parametrów lotu (czarne skrzynki)	Rejestracja pracy serca i ciśnienia tętniczego metodą Holtera
Tabele i grafy diagnostyczne	Pomoc dla użytkownika urządzenia przy lokalizacji i usuwaniu prostych usterek	Diagnoza na podstawie powiązań między objawami choroby i możliwymi przyczynami
Systemy ekspertowe	Moduły wnioskowania w dużych, zautomatyzowanych systemach diagnostycznych	Systemy wspomagania decyzji medycznych
Sztuczne sieci neuronowe	Moduły wnioskowania w komputerach pokładowych urządzeń technicznych	Rozpoznawanie, eliminacja zakłóceń i poprawa jakości obrazów medycznych

3. Podstawy uczenia maszynowego

3.1. Pojęcie i rodzaje uczenia maszynowego

Uczenie maszynowe to nauka interdyscyplinarna, której celem jest budowa systemu zdolnego do automatycznego doskonalenia się i nabywania nowej wiedzy na podstawie zgromadzonych danych. Istotę tego pojęcia oddają poniższe definicje, sformułowane przez dwóch amerykańskich pionierów w dziedzinie sztucznej inteligencji:

1. Arthur Samuel (1959) – „Uczenie maszynowe to dziedzina nauki dająca komputerom możliwość uczenia się bez konieczności ich jawnego programowania”.
2. Tom Mitchell (1997) – „Mówimy, że program komputerowy uczy się na podstawie doświadczenia E, w odniesieniu do jakiegoś zadania T i pewnej miary wydajności P, jeśli jego wydajność (mierzona przez P) wobec zadania T wzrasta wraz z nabywaniem doświadczenia E”.

Istnieje wiele rodzajów uczenia maszynowego. Często, dokonuje się ich podziału ze względu na trzy niżej wymienione kryteria, które mogą być łączone w dowolny sposób [50]:

- nadzór człowieka w procesie trenowania (uczenie nadzorowane, nienadzorowane, uczenie przez wzmacnianie)³;
- możliwość uczenia się w czasie rzeczywistym (uczenie wsadowe i przyrostowe);
- sposób pracy (uczenie z przykładów i uczenie z modelu).

W uczeniu nadzorowanym (z nauczycielem), dane uczące zawierają także rozwiązania danego problemu (etykiety). Zadanie, w którym zmienna prognozowana jest zmienną kategorię to klasyfikacja (binarna lub wieloklasowa). Z kolei, jeżeli zmienna prognozowana ma wartość liczbową, mówimy o regresji (liniowej lub wielomianowej). Dane uczące mogą być także nieoznakowane. Taki rodzaj uczenia, nazywamy nienadzorowanym (bez nauczyciela). Popularne zadania tego rodzaju to: analiza skupień, redukcja wymiarowości i wykrywanie anomalii. Możemy mieć także do czynienia z uczeniem przez wzmacnianie. Wówczas, system obserwuje środowisko, dobiera i wykonuje czynności, a także otrzymuje nagrody lub kary. W kolejnym etapie, system musi nauczyć się najlepszej strategii, tzn. rodzaju działania w danej sytuacji tak, aby nagrody były jak największe. Ten rodzaj uczenia stosowany jest, m.in., w wielu modelach robotów do nauki chodzenia.

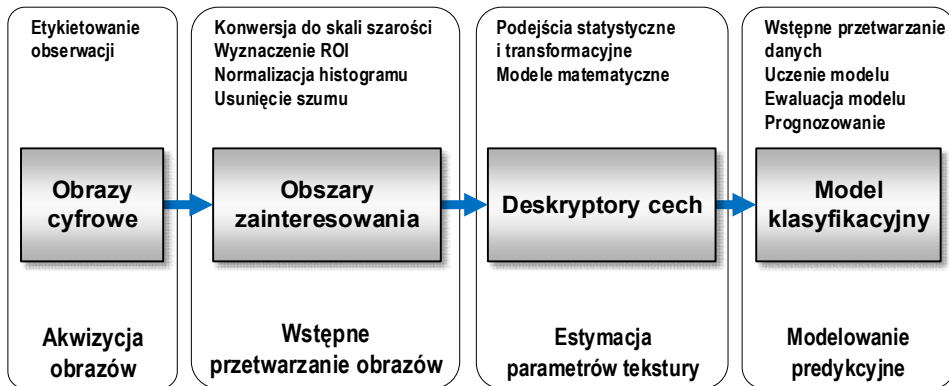
Uczenie wsadowe realizowane jest w taki sposób, że system najpierw jest uczony, a następnie wdrożony do określonego zastosowania. Od tego momentu,

³ Niekiedy, wyróżnia się także uczenie półnadzorowane. Występuje ono wówczas, gdy dane uczące są częściowo oznakowane.

system nie jest już więcej trenowany i opiera się wyłącznie na dotychczas zdobytych informacjach. Takie podejście nazywane również uczeniem offline. Inna sytuacja ma miejsce wówczas, gdy system jest uczony na bieżąco, poprzez cykliczne dostarczanie mu nowych danych. Mówimy wówczas o uczeniu przystosowym. Dane mogą być przekazywane pojedynczo lub w postaci małych pakietów. Dzięki temu, kolejne kroki uczenia nie są kosztowne obliczeniowo i system jest w stanie uczyć się na bieżąco (online), na podstawie nowych danych.

Ostatnie kryterium podziału dotyczy uogólniania. Podczas uczenia z przykładów, można powiedzieć, iż system uczy się ich „na pamięć”. Następnie, wykorzystując wybraną miarę podobieństwa, zdobyta wiedza jest porównywana z nowymi przypadkami. Inny sposób uogólniania wyników stosowany jest podczas uczenia z modelu. W tym podejściu, na podstawie danych uczących tworzony jest model, który w dalszej kolejności wykorzystywany jest do predykcji.

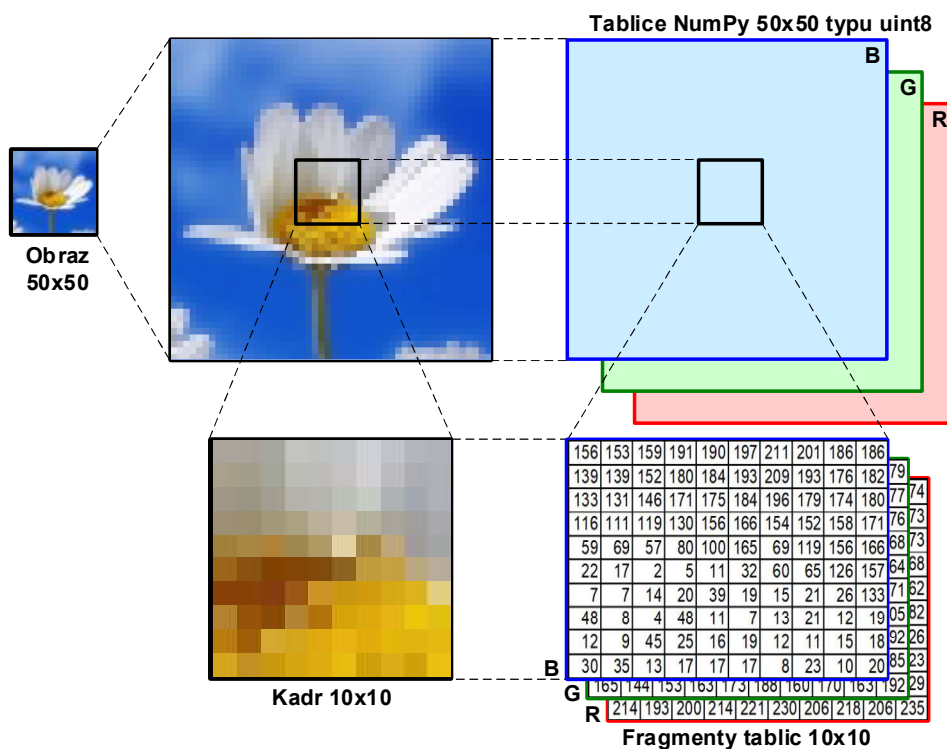
Na rys. 3.1 przedstawiono ogólny schemat systemu uczenia nadzorowanego, przeznaczonego do klasyfikacji tekstur obrazów. Danymi wejściowymi są obrazy cyfrowe, pozyskane jedną z wielu metod akwizycji, które zostały oznaczone etykietami klasy przez tzw. specjalistę dziedzinowego (np. lekarza o odpowiedniej specjalności). Wstępne przetwarzanie obrazów ma na celu uzyskanie obszarów zainteresowania (ang. *Region of Interest*, ROI). Oprócz samego wyznaczenia ROI, na etapie tym mogą być także wykonywane inne operacje, np. konwersja z trybu RGB do skali szarości, wyrównanie kontrastu, czy eliminacja szumów i artefaktów. Odpowiednio przygotowane obrazy są następnie poddawane analizie, prowadzącej do estymacji parametrów tekstury. Wyłoniony zbiór cech (najczęściej bardzo obszerny) zostaje wykorzystany do budowy modelu uczenia maszynowego, służącego do klasyfikacji obrazów.



Rys. 3.1. Schemat systemu uczenia nadzorowanego, przeznaczonego do klasyfikacji tekstur obrazów

3.2. Akwizycja obrazów

Akwizycja obrazów polega na wprowadzaniu informacji graficznych do komputera, z wykorzystaniem takich urządzeń, jak cyfrowe aparaty fotograficzne, kamery cyfrowe, skanery, czy inne urządzenia wytwarzające obraz w postaci cyfrowej. W systemie uczenia maszynowego, obiektem wyjściowym etapu akwizycji obrazów jest obraz cyfrowy (rys. 3.2). Po załadowaniu takiego obrazu do pamięci komputera, jest on reprezentowany w postaci tablic wypełnionych liczbami całkowitymi lub rzeczywistymi (zależnie od typu zmiennej tablicowej). Obrazom kolorowym odpowiadają 3 tablice, w których zapisane są intensywności składowych koloru poszczególnych pikseli. Obrazy monochromatyczne reprezentowane są przez jedną tablicę, zawierającą natężenie (luminancję) pikseli. Na rysunku 3.2 przedstawiono przykładowy obraz kolorowy oraz jego reprezentację w pamięci, przy wykorzystaniu tablic NumPy⁴ w języku Python.



Rys. 3.2. Obraz kolorowy o rozmiarze 50×50 pikseli i jego reprezentacja w pamięci

⁴ W bibliotece OpenCV, kolejność indeksowania tablic kolorów to BGR, a nie RGB.

Podczas akwizycji obrazów, muszą być określone warunki tego procesu oraz parametry wykorzystywanych urządzeń. Do najważniejszych należą [51]:

- oświetlenie (jasność, barwa, skupione, rozproszone, kąt padania, liczba źródeł);
- lokalizacja obszarów zainteresowania oraz wymiary kadru obrazu, zależne od zastosowanego powiększenia;
- dostrojenie układu optycznego, szczególnie ostrości;
- czas ekspozycji i związana z nim czułość przetwornika obrazu (jest to szczególnie ważne w przypadku obiektów ruchomych);
- rozdzielczość chromatyczna (liczba rozróżnianych kolorów lub odcieni szarości);
- rozdzielczość przestrzenna (liczba pikseli na jednostkę długości).

Przeważnie, proces akwizycji jest narażony na różne zakłócenia. Ich główne przyczyny to: szумы przetwornika obrazu oraz układów elektronicznych, wady oświetlenia (odbłaski, nierównomierność, natężenie), aberracje chromatyczne i geometryczne układu optycznego, różne artefakty (np. oznaczenia naniesione na obiekt przez człowieka, zabrudzenia układu optycznego). Większość zakłóceń można wyeliminować poprzez cyfrowe przetwarzanie obrazów. Jednak, niektóre błędy akwizycji, np. zbyt duży kontrast lub oświetlenie, mogą być przyczyną bezpowrotnej utraty części informacji. W związku z tym, podczas akwizycji obrazów, należy utrzymywać stałe wartości parametrów tego procesu tak, aby zapewnić porównywalność cech uzyskanych obrazów.

3.3. Wstępne przetwarzanie obrazów

Poprzez przetwarzanie obrazów, rozumiemy wszystkie operacje, które są na nich wykonywane, jeżeli ich efektem są także obrazy lub ich fragmenty. Przykładowe czynności wykonywane w ramach tego procesu to: usuwanie szumów i artefaktów, poprawa jakości obrazów (np. normalizacja zakresu jasności), konwersja trybu koloru (np. z RGB do skali szarości), uwydatnienie określonych cech, wyznaczenie obszaru zainteresowania.

Celem przetwarzania jest przygotowanie obrazów do analizy, prowadzącej do wyłonienia deskryptorów cech (estymacja cech). Analiza może dotyczyć różnych aspektów obrazów, np. ich tekstury, geometrii wyłoniętych obiektów, czy cech kolorymetrycznych. W związku z tym, w zależności od postawionego celu, dobierany jest odpowiedni zakres operacji. Szczegółowo, operacje przetwarzania obrazów zostaną opisane w dalszych rozdziałach, w których przedstawiono wyniki badań dotyczących obrazów płomienia oraz obrazów zmian skórnych. Pomimo faktu, iż w obu przypadkach analizowano teksturę obrazów, to specyfika każdego z kierunków badań uzasadniała zastosowanie nieco innego podejścia w procesie przetwarzania wstępnego.

Należy w tym miejscu zwrócić uwagę na bardzo ważną kwestię. Otóż, przetwarzanie obrazu, zmieniając go – może także zmienić wyniki analizy. W związku z tym, określone operacje należy stosować tylko wówczas, gdy są niezbędne. Często jest tak, iż proces akwizycji został wykonany prawidłowo,

a obrazy uzyskane w jego efekcie są dobrej jakości. Wówczas, nie są potrzebne dodatkowe operacje. Szczególną ostrożność należy także zachować w badaniach z wykorzystaniem obrazów medycznych. Może być tak, że jasność różnych obszarów obrazu zawiera ważną wskazówkę diagnostyczną (np. obszary o różnej echogeniczności na zdjęciu USG). Normalizacja histogramu obrazu, wykonana w takiej sytuacji, może doprowadzić do zmiany obrazu i utraty informacji. Są jednak sytuacje, gdy przeprowadzenie pewnych operacji jest niezbędne, np. z uwagi na zbyt słabą jakość obrazów. Wówczas, w celu uniknięcia zafałszowania wyników, należy zwrócić uwagę na to, aby stosować identyczne parametry wykonywanych operacji dla wszystkich obrazów.

3.4. Estymacja parametrów tekstury

Generalnie, analiza tekstur ma na celu znalezienie zbioru parametrów, zwanych także cechami teksturalnymi, z których każdy jest liczbową miarą (deskryptorem) określonej właściwości tekstury. Opisywanymi właściwościami mogą być: szorstkość (tekstura może być gładka lub ziarnista), kierunek i regularność ułożenia wzoru, jednorodność, lokalny kontrast oraz średni poziom jasności pikseli danego obszaru obrazu [51, 52, 53]. Analiza tekstur obrazów biomedycznych jest często stosowana do rozpoznawania różnych chorób. Jako wybrane przykłady, można wymienić diagnostykę choroby Hashimoto [54, 55, 56, 57, 58, 59], osteoporozy [60, 61], sarkoidozy [62], czy szpiczaka [63].

Istnieją cztery główne podejścia do ekstrakcji cech teksturalnych, które w połączeniu z odpowiednimi algorytmami klasyfikacji, znalazły szerokie zastosowanie, zarówno w diagnostyce technicznej, jak i medycznej. Są to podejścia:

1. Statystyczne – wykorzystywany jest histogram poziomów szarości, macierz zdarzeń (histogram drugiego rzędu), statystyki wyższego rzędu, macierz RL (ang. *Run-Length*), macierz gradientu [64, 65, 66].
2. Modele matematyczne – stosowane są losowe pola Markowa [67], modele autoregresji oraz modele fraktalne [68].
3. Transformacyjne – korzysta się z transformaty Fouriera, filtrów Gabora [69], transformaty falkowej [70].
4. Strukturalne – poszukiwany jest podstawowy, powtarzający się element tekstury (tzw. teksele) i określone są reguły rozmieszczenia tego elementu. Stosowane są do tego celu operacje morfologii matematycznej [71].

Dostępnych jest wiele narzędzi do estymacji parametrów tekstury. Jednym z godnych polecenia jest, zdaniem autora – program MaZda. Został on opracowany w Instytucie Elektroniki Politechniki Łódzkiej (autorem jest Piotr M. Szczypiński). Program ten jest bezpłatnie udostępniony w Internecie do celów naukowych [72]. Możliwości programu MaZda są bardzo szerokie. Między innymi, pozwala on analizować szare obrazy teksturowe i wyznaczać wartości liczbowe dla obszernego zbioru, ponad dwustu dziewięćdziesięciu cech. Ich dokładny opis można znaleźć w pozycjach [65, 66, 73, 74], a także

w dokumentacji programu MaZda. Zaletą programu jest fakt, iż oprócz podejścia statystycznego do analizy obrazu, stosuje także model matematyczny (model autoregresji) oraz podejście transformacyjne (transformata falkowa). Program MaZda wykorzystano do estymacji parametrów obrazów teksturowych wykorzystanych w badaniach zaprezentowanych w niniejszej monografii.

Zbiór cech analizowanych obrazów otrzymano na podstawie:

1. Wektora histogramu obrazu (9 cech): średnia, wariancja, skośność, kurtoza, percentyle 1%, 10%, 50%, 90% i 99%.
2. Macierzy gradientu obrazu (5 cech): średnia, wariancja, skośność, kurtoza, odsetek pikseli z niezerowym gradientem.
3. Macierzy rozkładu długości pasm (5 cech \times 4 kierunki): nierównomierność pasm, nierównomierność szarości, uwydatnienie krótkich pasm, uwydatnienie długich pasm, część obrazu w pasmach.
4. Macierzy przejść tonalnych (11 cech \times 4 kierunki \times 5 odległości): drugi moment kątowy, kontrast, korelacja, suma kwadratów, odwrotny moment różnicowy, sumaryczna średnia, sumaryczna wariancja, sumaryczna entropia, entropia, wariancja różnicowa, entropia różnicowa.
5. Modelu autoregresji (5 cech): parametry $\Theta_1, \Theta_2, \Theta_3, \Theta_4$, odchylenie standardowe.
6. Dyskretnej transformaty falkowej (24 cechy): energia współczynników falkowych (falki Haara) dla czterech podpasm (LL, LH, HL, HH) i sześciu skal.

3.4.1. Deskrytory jasności pikseli

Histogram obrazu szarego określa udział ilościowy w obrazie pikseli o poszczególnych odcieniach. Można go wyrazić matematycznie jako wektor H o Lh składowych odpowiadających poszczególnym jednakowym przedziałom o długości dG , na jakie podzielono nominalny zakres poziomów szarości. Każda składowa H_j reprezentuje liczbę pikseli obrazu posiadających odcień zawarty w odpowiadającym tej składowej przedziale:

$$H = [H_j], \quad (3.1)$$

gdzie: $j = 1, 2, \dots, Lh$,

$$H_j = \sum_{w=1}^{Lw} \sum_{k=1}^{Lk} L((j-1) \cdot dG < x_{w,k} \leq j \cdot dG)$$

Lw, Lk – liczba wierszy oraz liczba kolumn macierzy obrazu,

$dG = GR / Lh$ – długość przedziału,

GR – wektor nominalnych poziomów szarości,

$L(\text{wyrażenie})$ – funkcja wyznaczająca wartość logiczną wyrażenia
(1 – prawda, 0 – fałsz).

Histogram znormalizowany to wektor Hn , którego składowe Hn_j otrzymuje się przez podzielenie składowych H_j przez liczbę wszystkich pikseli obrazu Lx :

$$Hn_j = \frac{H_j}{Lx}. \quad (3.2)$$

Dzięki temu, histogram przyjmuje postać rozkładu prawdopodobieństwa występowania w obrazie poszczególnych odcieni.

Parametry statystyczne histogramu można traktować jako deskryptory cech obrazu, są to:

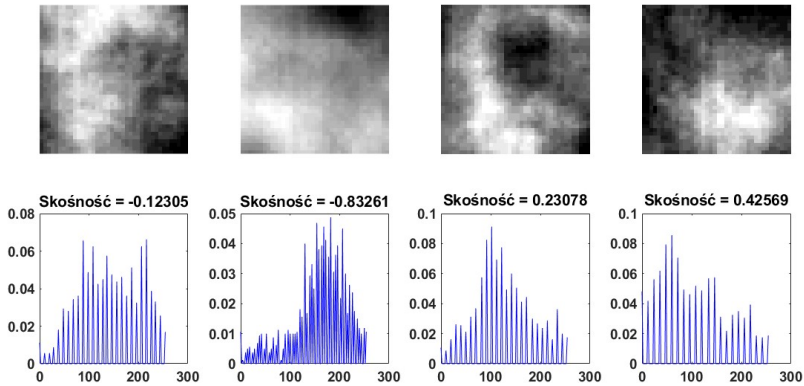
średnia
$$\mu_H = \sum_{j=1}^{Lh} j \cdot dG \cdot Hn_j, \quad (3.3)$$

wariancja
$$(\sigma_H)^2 = \sum_{j=1}^{Lh} (j \cdot dG - \mu_H)^2 \cdot Hn_j, \quad (3.4)$$

skośność
$$\mu_H^3 = \sigma_H^{-3} \cdot \sum_{j=1}^{Lh} (j \cdot dG - \mu_H)^3 \cdot Hn_j, \quad (3.5)$$

kurtoza
$$\mu_H^4 = \sigma_H^{-4} \cdot \sum_{j=1}^{Lh} (j \cdot dG - \mu_H)^4 \cdot Hn_j - 3. \quad (3.6)$$

Przy jednorodnym tle obrazu, skośność histogramu (μ_H^3) może służyć do detekcji przyrostu liczby zmian powierzchni, które mogą być charakterystyczne dla określonej klasy obrazów. Na rysunku 3.3 przedstawiono histogramy znormalizowane wraz z obliczoną skośnością dla przykładowych obrazów płomienia.



Rys. 3.3. Histogramy znormalizowane wraz z obliczoną skośnością dla przykładowych obrazów płomienia (40×40 pikseli)

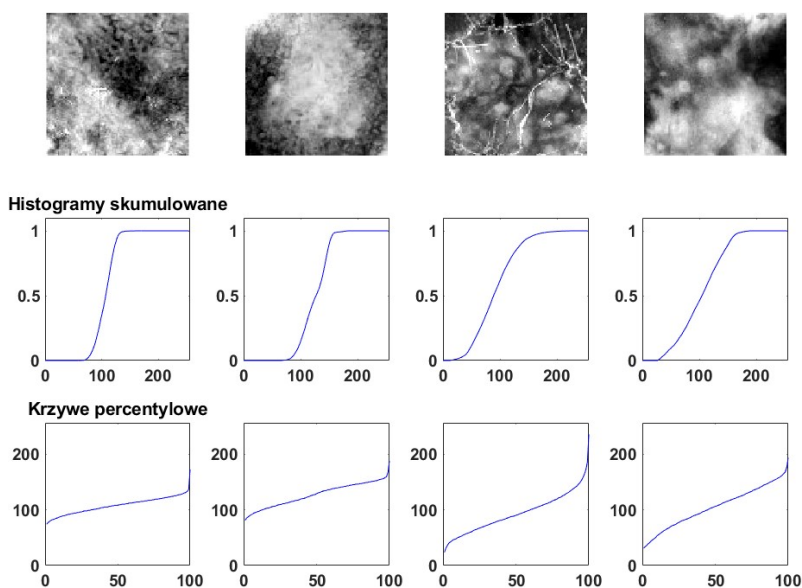
Histogram skumulowany Hns to dystrybucja rozkładu odcieni szarości. Można go określić następująco:

$$Hns = [Hns_j], \quad (3.7)$$

gdzie: $j = 1, 2, \dots, Lh$,

$$Hns_j = \sum_{i=1}^j Hn_i.$$

Krzywe percentylowe to histogramy skumulowane przedstawione w odpowiednim układzie współrzędnych. Jeśli ze zbioru wszystkich pikseli obrazu utworzymy wektor i uporządkujemy te piksele według rosnącej jasności, to n -ty percentyl obrazu jest jasnością piksela znajdującego się na takiej pozycji, że jej odległość od początku wektora, wyrażona w procentach długości całego wektora, wynosi $n\%$. Na rysunku 3.4 przedstawiono histogramy skumulowane i krzywe percentylowe dla przykładowych obrazów zmian skórnych. Na dolnych wykresach widać zależność jasności od położenia pikseli w uporządkowanych ciągach. Skalując oś odciętych w procentach całkowitej liczby pikseli, otrzymujemy krzywe percentylowe, których rzędne są percentylami obrazu.



Rys. 3.4. Histogramy skumulowane i krzywe percentylowe dla przykładowych obrazów zmian skórnych (100×100 pikseli)

3.4.2. Deskryptory wyznaczone z otoczeń pikseli

Macierz gradientu obrazu charakteryzuje różnice odcieni par pikseli sąsiadujących z badanym pikselem lub położonych od niego w odległości d (przeważnie $d = 1$ lub $d = 2$):

$$G_{w,k} = \sqrt{(x_{w+d,k} - x_{w-d,k})^2 + (x_{w,k+d} - x_{w,k-d})^2} . \quad (3.8)$$

Dla macierzy gradientu, podobnie jak w przypadku innych charakterystyk, można obliczać różne parametry statystyczne:

średni gradient
$$\mu_G = \frac{1}{m} \cdot \sum_{w,k} G_{w,k} , \quad (3.9)$$

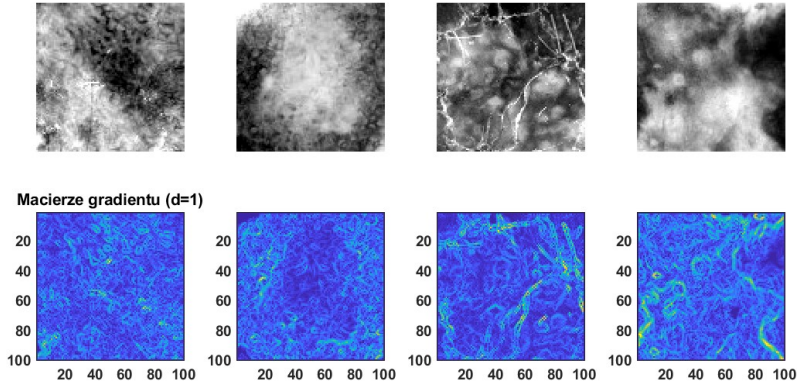
wariancja gradientu
$$(\sigma_G)^2 = \frac{1}{m} \cdot \sum_{w,k} (G_{w,k} - \mu_G)^2 , \quad (3.10)$$

skośność gradientu
$$\mu_G^3 = \frac{1}{m \cdot (\sigma_G)^3} \cdot \sum_{w,k} (G_{w,k} - \mu_G)^3 , \quad (3.11)$$

kurtoza gradientu
$$\mu_G^4 = \frac{1}{m \cdot (\sigma_G)^4} \cdot \sum_{w,k} (G_{w,k} - \mu_G)^4 - 3 , \quad (3.12)$$

gdzie: m – liczba elementów macierzy G .

Na rysunku 3.5 przedstawiono wizualizację macierzy gradientu dla przykładowych obrazów zmian skórnych.



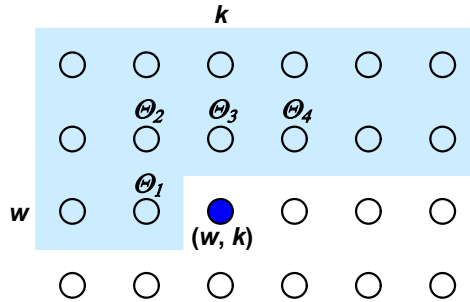
Rys. 3.5. Wizualizacje macierzy gradientu dla przykładowych obrazów zmian skórnych (100×100 pikseli)

Model autoregresji zakłada, że jasność danego piksela jest uzależniona od jasności pikseli z jego otoczenia (rys. 3.6). Dla otoczenia obejmującego tylko najbliższe piksele wyraża to zależność:

$$x_{w,k} = \Theta_1 \cdot x_{w,k-1} + \Theta_2 \cdot x_{w-1,k-1} + \Theta_3 \cdot x_{w-1,k} + \Theta_4 \cdot x_{w-1,k+1} + Er, \quad (3.13)$$

gdzie: Er – odchyłka będąca zmienną losową o rozkładzie normalnym, z wartością średnią równą zero i odchyleniem standardowym σ .

W przypadku prostego sąsiedztwa pikseli, składającego się z czterech sąsiadów (rys. 3.6), istnieje pięć nieznanymi parametrów modelu traktowanych jako cechy obrazu. Są to: odchylenie standardowe σ oraz wektor parametrów $\Theta = [\Theta_1, \Theta_2, \Theta_3, \Theta_4]$ wyznaczanych tak, aby zminimalizować odchyłkę Er [75]. Powierzchnia z niebieskim tłem na rysunku 3.6 wskazuje obszar, na którym może się znajdować sąsiedztwo piksela o współrzędnych (w, k) .



Rys. 3.6. Sąsiedztwo lokalne piksela znajdującego się w położeniu określonym przez numer wiersza w oraz numer kolumny k

3.4.3. Deskryptory zbiorów par pikseli

Macierz przejść tonalnych C , zwana też macierzą sąsiedztwa [76], macierzą zdarzeń [75, 77], macierzą współwystąpień [66, 78] lub histogramem drugiego rzędu wyraża relacje między odcieniami określonych par pikseli. Macierz C jest macierzą kwadratową o rozmiarach $Lg \times Lg$ wyznaczaną dla określonego kierunku ψ oraz odległości między parami pikseli. Element $C_{a,b}$ powstaje przez zliczanie wszystkich takich par pikseli, które posiadają odpowiednio odcienie a i b . Kierunkami ψ , dla których może być wyznaczana macierz C są: 0° , 45° , 90° i 135° . Odpowiada to wektorom skanowania macierzy $[dw, dk]$: $[0, d]$, $[d, d]$, $[d, 0]$, $[d, -d]$, gdzie: d – odległość, dw – przyrost numeru wiersza, dk – przyrost numeru kolumny. W programie MaZda, kierunki są oznaczane zgodnie z zasadą, że początek układu współrzędnych x, y znajduje się w lewym dolnym rogu obrazu. Macierze C są wyznaczane dla odległości: $d = (1, 2, \dots, 5)$.

Jeśli kierunek i odległość zdefiniujemy przy pomocy wektora przyrostu $[dw, dk]$, to element znormalizowanej macierzy $C(dw, dk)$ odpowiadający odcieniom a i b można określić wzorami:

$$C(dw, dk)_{a,b} = \frac{1}{m} \sum_w \sum_k pt(a, b, x_{w,k}, x_{w+dw,k+dk}), \quad (3.14)$$

gdzie: $a = 0, 1, 2, \dots, Lg$, $b = 0, 1, 2, \dots, Lg$,
 m – liczba wszystkich badanych par pikseli,
 pt – funkcja wyznaczająca wartość logiczną według poniższej zależności

$$pt(a, b, x_{w,k}, x_{w+dw,k+dk}) = \begin{cases} 1 & \text{gd}y(x_{w,k} = a) \wedge (x_{w+dw,k+dk} = b) \\ 0 & \text{w przeciwnym przypadku} \end{cases}.$$

Macierz przejść tonalnych może być znormalizowana przez podzielenie wartości jej elementów przez liczbę wszystkich badanych par pikseli m . Nadaje to macierzy C znaczenie rozkładu prawdopodobieństwa.

Ideę budowy macierzy przejść tonalnych pokazuje rysunek 3.7 na przykładzie prostego obrazu (rys. 3.7a) o trzech odcieniach szarości 0, 1, 2 i wymiarach 5×5 pikseli. Analizowane są sąsiadujące piksele ($d = 1$) w kierunku poziomym (0°). Macierz (rys. 3.7b) pokazuje liczbę przejść z odcieni 0, 1, 2 opisanych z lewej strony macierzy do odcieni opisanych nad macierzą.

a)

0	0	0	2	2
0	1	1	2	0
0	0	0	0	0
0	2	2	1	0
0	0	0	1	1

b)

	0	1	2
0	8	2	2
1	1	2	1
2	1	1	2

c)

	0	1	2
0	0,4	0,1	0,1
1	0,05	0,1	0,05
2	0,05	0,05	0,1

d)

	0	1	2
0	8	1	1
1	2	2	1
2	2	1	2

e)

	0	1	2
0	16	3	3
1	3	4	2
2	3	2	4

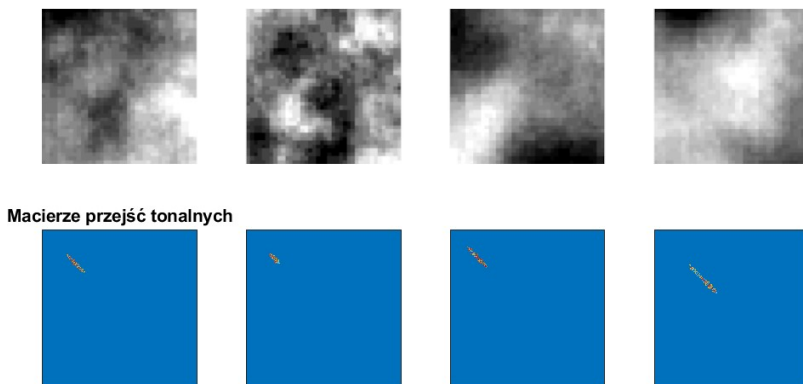
f)

	0	1	2
0	0,4	0,075	0,075
1	0,075	0,1	0,05
2	0,075	0,05	0,1

Rys. 3.7. Przykład macierzy przejść tonalnych: a) wartości pikseli obrazu; b) macierz (a) dla $dw=0$, $dk=1$; c) macierz (b) po normalizacji; d) transponowana macierz (b); e) suma macierzy (b) i (d); f) macierz (e) po normalizacji

Jak widać, jest to macierz asymetryczna. Aby jednakowo traktować kierunki skanowania i kierunki do nich przeciwne, dokonuje się symetryzacji macierzy

(rys. 3.7e) przez sumowanie macierzy pierwotnej (rys. 3.7b) i jej transpozycji (rys. 3.7d). Macierze z rysunków 3.7b,e znormalizowane przez podzielenie wartości wszystkich elementów przez liczbę wszystkich skanowanych par widać na rysunkach 3.7c,f (suma elementów macierzy znormalizowanych wynosi jeden). Na rysunku 3.8 przedstawiono wizualizacje macierzy C dla przykładowych obrazów płomienia. Jak widać, największe elementy tych macierzy skupiają się w pobliżu przekątnej głównej, a wykres wartości z tej przekątnej ma kształt podobny do histogramu.



Rys. 3.8. Wizualizacje macierzy przejść tonalnych dla przykładowych obrazów płomienia (40×40 pikseli)

Z macierzy C wyznaczane są deskryptory zwane parametrami lub deskryptorami Haralick'a zaproponowane przez niego po raz pierwszy w pracy [66]. Deskryptory Haralick'a wyznaczane z symetrycznej znormalizowanej macierzy przejść tonalnych przedstawiono poniżej:

$$\text{drugi moment kątowy (energia) } AngScMom = \sum_{a,b} (C_{a,b})^2, \quad (3.15)$$

$$\text{kontrast} \quad Contrast = \sum_{a,b} ((a-b)^2 \cdot C_{a,b}), \quad (3.16)$$

$$\text{korelacja} \quad Correlat = - \sum_{a,b} \frac{(a-\mu) \cdot (b-\mu)}{\sigma^2} \cdot C_{a,b}, \quad (3.17)$$

$$\text{wariancja (dyspersja) } SumOfSqs = \sum_{a,b} (a-\mu)^2 \cdot C_{a,b}, \quad (3.18)$$

$$\text{odwrotny moment różnicowy } InvDfMom = \sum_{a,b} \frac{C_{a,b}}{1+(a-b)^2}, \quad (3.19)$$

$$\text{sumaryczna średnia} \quad SumAverg = \sum_{k=2}^{2 \cdot Lg} k \cdot P_{x+y}(k), \quad (3.20)$$

$$\text{sumaryczna wariancja} \quad SumVar = \sum_{k=2}^{2 \cdot Lg} (k - SumAverg)^2 \cdot P_{x+y}(k), \quad (3.21)$$

$$\text{sumaryczna entropia} \quad SumEntrp = - \sum_{k=2}^{2 \cdot Lg} P_{x+y}(k) \cdot \log(P_{x+y}(k)), \quad (3.22)$$

$$\text{entropia} \quad Entropy = - \sum_{a,b} C_{a,b} \cdot \log(C_{a,b}), \quad (3.23)$$

$$\text{wariancja różnicowa} \quad DifVarnc = \sum_{k=0}^{Lg-1} k^2 \cdot P_{x-y}(k), \quad (3.24)$$

$$\text{entropia różnicowa} \quad DifEntrp = - \sum_{k=0}^{Lg-1} P_{x-y}(k) \cdot \log(P_{x-y}(k)), \quad (3.25)$$

$$\text{gdzie (dla macierzy symetrycznej): } P_{x+y}(k) = \sum_{a,b|a+b=k} C_{a,b}, \quad P_{x-y}(k) = \sum_{a,b|a-b=k} C_{a,b},$$

$$\text{średnia z sumy wierszy lub kolumn} \quad \mu = \mu_x = \mu_y = \sum_a a \sum_b C_{a,b} = \sum_b b \sum_a C_{a,b},$$

odchylenie standardowe sum wierszy lub kolumn

$$\sigma = \sigma_x = \sigma_y = \sum_a (a - \mu)^2 \cdot \sum_b C_{a,b}.$$

3.4.4. Deskryptory pasm pikseli o jednakowym odcieniu

Macierz rozkładu długości pasm jest charakterystyką zawierającą informacje o częstości występowania pasm o poszczególnych odcieniach oraz długościach (mierzonych liczbą pikseli). W literaturze jest ona określana akronimem RLM (ang. *Run Length Matrix*) lub macierzą R . Macierz ta jest wyznaczana dla określonego kierunku skanowania ψ , a każdy jej element $R_{g,j}$ zawiera informację ile razy wystąpiło pasmo pikseli o odcieniu g oraz długości j . Osobne macierze mogą być wyznaczane dla kierunków: 0° i 90° , a także 45° i 135° . Macierz R ma wymiary $Lg \times Lr$, gdzie Lr to maksymalna długość pasma (w pikselach), a Lg – liczba odcieni szarości.

Na podstawie macierzy R wyznaczone są następujące deskryptory [75]:
odwrotny moment uwydatnienia krótkich pasm

$$ShrtREmp = \left(\sum_{g=1}^{Lg} \sum_{j=1}^{Lr} \frac{R_{g,j}}{j^2} \right) / c, \quad (3.26)$$

moment uwydatnienia długich pasm

$$LngREmp = \left(\sum_{g=1}^{Lg} \sum_{j=1}^{Lr} j^2 \cdot R_{g,j} \right) / c, \quad (3.27)$$

niejednorodność pasm $RLNonUni = \left(\sum_{j=1}^{Lr} \left(\sum_{g=1}^{Lg} R_{g,j} \right)^2 \right) / c, \quad (3.28)$

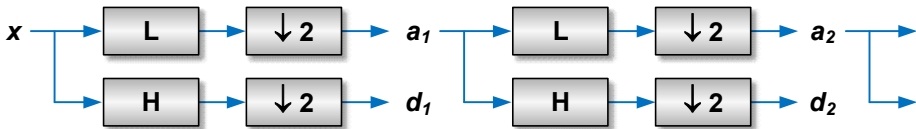
część obrazu w pasmach $Fraction = c / \left(\sum_{g=1}^{Lg} \sum_{j=1}^{Lr} j \cdot R_{g,j} \right), \quad (3.29)$

niejednorodność szarości $GLEvNonU = \left(\sum_{g=1}^{Lg} \left(\sum_{j=1}^{Lr} R_{g,j} \right)^2 \right) / c, \quad (3.30)$

przy czym współczynnik c jest zdefiniowany następująco: $c = \sum_{g=1}^{Lg} \sum_{j=1}^{Lr} R_{g,j}.$

3.4.5. Deskryptory oparte na transformacie falkowej

Dyskretna transformata falkowa (ang. *Discrete Wavelet Transform*, DWT) jest transformacją liniową, operującą na wektorze danych o długości stanowiącej całkowitą potęgę dwójki i przekształcającą go w inny wektor numeryczny o tej samej długości [79, 80]. Jest to narzędzie, które dzieli dane na składowe o różnych częstotliwościach, a następnie analizuje każdą składową z rozdzielczością dopasowaną do jej skali. DWT jest obliczana za pomocą algorytmu Mallata powszechnie znanego w środowiskach zajmujących się przetwarzaniem sygnałów jako dwukanałowy koder podpasмовy (rys. 3.9).

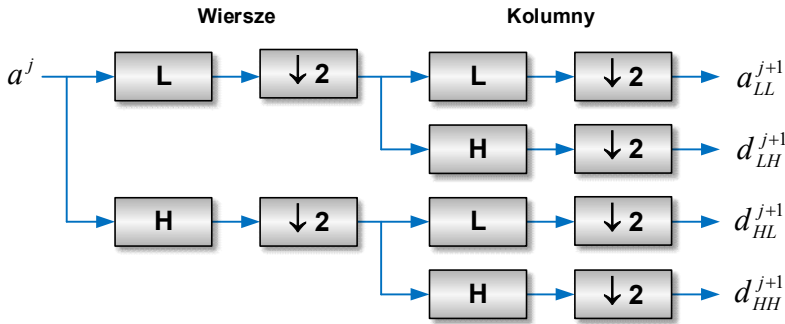


Literami H oraz L oznaczono filtry, odpowiednio górno i dolnoprzepustowe. Bloki znajdujące się po każdym filtrze oznaczają z kolei dwukrotne zmniejszenie częstotliwości próbkowania. Sygnały wyjściowe filtrów przedstawionych na rysunku 3.9 opisane są następującymi równaniami:

$$a_{j+1}[p] = \sum_{n=-\infty}^{+\infty} l[n-2p]a_j[n], \quad (3.31)$$

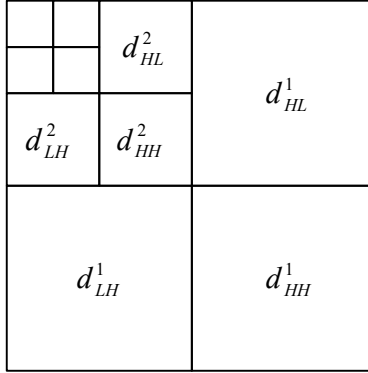
$$d_{j+1}[p] = \sum_{n=-\infty}^{+\infty} h[n-2p]a_j[n]. \quad (3.32)$$

Elementy a_j są wykorzystywane do następnego kroku (skali) transformacji, natomiast elementy d_j (współczynniki falkowe) oznaczają jej wynik. Z kolei, $l[n]$ oraz $h[n]$ oznaczają współczynniki filtrów, odpowiednio dolno i górnoprzepustowych. Można założyć, że w skali $j + 1$ występuje tylko połowa elementów a oraz d , w porównaniu ze skalą j . Sprawia to, że DWT może być wykonywana do momentu aż w analizowanym sygnale pozostaną tylko dwa elementy a_j . Elementy te zwane są współczynnikami skalowania funkcji. W przypadku obrazów, algorytm DWT jest realizowany w sposób podobny do przedstawionego na rysunku 3.10. Najpierw wykonywana jest DWT dla wszystkich wierszy obrazu, a następnie dla wszystkich jego kolumn.



Rys. 3.10. Dekompozycja falkowa obrazów

Podczas badań wykorzystanych w niniejszej monografii, jako falkowe cechy obrazu wykorzystano wektor zawierający energie współczynników falkowych obliczone w podpasmach dla kolejnych sześciu skal. W celu obliczenia cech falkowych, w pierwszym kroku wyznaczana jest falka Haara dla całego obrazu. Efektem tej transformacji są 4 obrazy podpasmostowe w każdej skali (rys. 3.11).

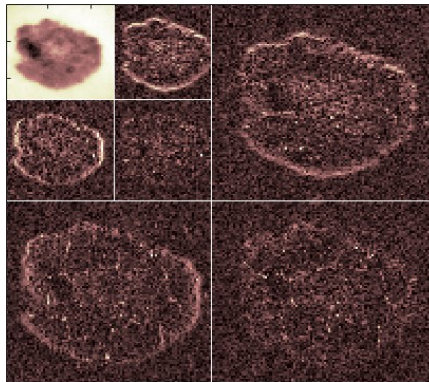


Rys. 3.11. Obrazy podpasmowe

Obraz podpasmowy a_{LL} jest wykorzystywany do obliczenia DWT w następnej skali. W następnym kroku, wyznaczone są energie współczynników a_{LL} , d_{LH} , d_{HL} oraz d_{HH} dla każdej rozważanej skali w zaznaczonych ROI:

$$E_{podpasmow,skala} = \frac{\sum_{x,y \in ROI} (d_{x,y}^{podpasmow})^2}{n}, \quad (3.33)$$

gdzie n jest liczbą pikseli w ROI, zarówno w danej skali, jak i w podpaśmie. Wymiary ROI są zmniejszane w kolejnych skalach, aby odpowiadały rozmiarom obrazów podpasmowych (rys. 3.12). Energia dla danej skali jest obliczana tylko wówczas, gdy ROI w tej skali zawiera co najmniej cztery punkty. Wynikiem działania powyższej procedury jest wektor cech zawierający energie współczynników falkowych, obliczone w podpasmach dla kolejnych skal.

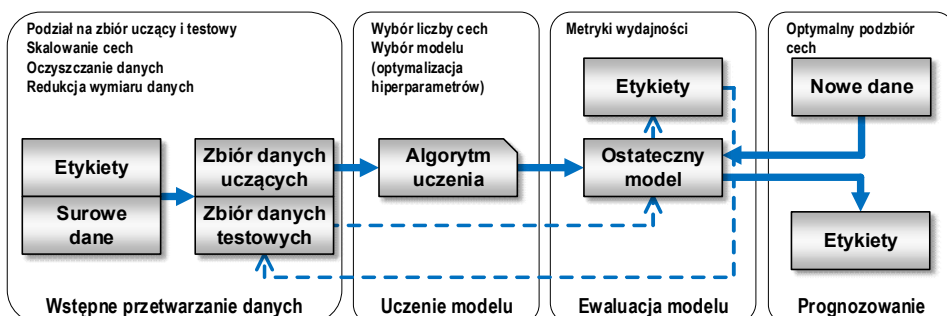


Rys. 3.12. Przykład obrazu zmiany skórnej zmniejszanego w kolejnych skalach

3.5. Modelowanie predykcyjne

Celem **modelowania predykcyjnego** jest budowa modelu służącego do wykonywania prognozy pojedynczej zmiennej wyjściowej na podstawie znanych wartości zmiennych wejściowych. Biorąc pod uwagę typ zmiennej prognozowanej, wyróżniamy zadanie klasyfikacji lub regresji. W modelowaniu predykcyjnym stosowane są dwa podejścia. W pierwszym, różne metody eksploracji danych wykorzystywane są osobno, niezależnie od siebie. Drugim jest podejście hybrydowe, polegające na połączeniu kilku metod o różnych sposobach działania, co często daje efekt synergii [81, 82, 83]. Interesujący przykład zastosowania podejścia hybrydowego do predykcji smaku wina można znaleźć w [84].

W badaniach, na których oparto niniejszą monografię, do rozwiązania zadania klasyfikacji zastosowano pierwsze podejście do modelowania predykcyjnego. Na rysunku 3.13 przedstawiono zasadnicze elementy systemu uczenia maszynowego oraz typowy przebieg procesów. Podczas budowy modelu, można wyróżnić cztery zasadnicze etapy: wstępne przetwarzanie danych, uczenie modelu, ewaluacja modelu, prognozowanie. W kolejnych rozdziałach podane zostaną najważniejsze informacje o każdej z tych faz.



Rys. 3.13. Typowy przebieg procesów podczas modelowania predykcyjnego [85]

3.5.1. Wstępne przetwarzanie danych

Pierwszym i zarazem jednym z najważniejszych etapów uczenia maszynowego jest wstępne przetwarzanie danych. Wynika to z faktu, iż dane surowe (nieprzetworzone) bardzo rzadko mają postać pozwalającą na uzyskanie optymalnej skuteczności algorytmów uczących. W dalszej części scharakteryzowano najczęstsze operacje wykonywane w ramach wstępnego przetwarzania danych.

Podział danych na zbiór uczący i testowy. Zbiór uczący służy do trenowania i optymalizacji modelu, a zbiór testowy wykorzystywany jest do oceny modelu ostatecznego.

Skalowanie cech. Algorytmy uczenia maszynowego często wymagają, aby zmienne wejściowe (cechy) były prezentowane w jednakowej skali. Skalowanie

odbywa się przeważnie poprzez transformację wartości liczbowych do zakresu $[0, 1]$ lub do standardowego rozkładu normalnego $N(0, 1)$.

Oczyszczanie danych. Obecność cech nieistotnych w zbiorze danych może powodować zmniejszenie dokładności wielu modeli klasyfikacyjnych i regresyjnych, zwłaszcza takich, które wykorzystują algorytmy liniowe. Operacja oczyszczania danych ma więc na celu eliminację takich cech, które mają znikomą wartość informacyjną, w kontekście realizowanej predykcji. Usuwane są cechy z wartościami stałymi i prawie stałymi, cechy zduplikowane (ich wartości są powielone), a także cechy bardzo silnie skorelowane ze sobą.

Redukcja wymiaru danych. Duża wymiarowość danych jest zjawiskiem niekorzystnym, ponieważ wychwycenie prawidłowości przy analizowaniu wszystkich cech jest najczęściej niemożliwe lub bardzo trudne do uzyskania. W takiej sytuacji, istnieje potrzeba redukcji wymiaru danych (zmniejszenia liczby używanych cech), która nie powoduje zbyt wielkich strat informacji, a czasem wręcz zwiększa wartość informacyjną danych. Operacja tego rodzaju, wykonana przed etapem uczenia modelu, polepsza jego możliwości generalizacyjne i w konsekwencji zmniejsza niebezpieczeństwo przeuczenia. Zbudowany model jest łatwiejszy do interpretacji oraz implementacji, ma krótszy czas uczenia i jest mniej podatny na błędy [86, 87, 88, 89]. Istnieją dwa zasadnicze podejścia do problemu redukcji wymiaru przestrzeni cech, pierwsze to ekstrakcja, a drugie selekcja.

Ekstrakcja cech ma na celu konstrukcję nowych cech na bazie tych istniejących. W tym przypadku, zbiór cech wejściowych zostaje poddany pewnej transformacji liniowej lub nieliniowej. Ocena uzyskanych wyników opierać się wówczas może, np. na analizie skategoryzowanych wykresów rozrzutu, prezentujących położenie obserwacji w dwuwymiarowym układzie nowych współrzędnych. Odnosząc to do zadania klasyfikacji, pożądane byłoby wówczas uzyskanie wyraźnych segmentów grupujących przypadki należące do różnych klas. Zbiór cech tworzących nowe współrzędne mógłby być wówczas wykorzystany do klasyfikacji. Przykładowe metody ekstrakcji cech to analiza składowych głównych, analiza czynnikowa oraz skalowanie wielowymiarowe [90, 91, 92].

Selekcja cech polega na wyborze pewnego podzbioru spośród zbioru cech oryginalnych [93]. Jest to jedna z najważniejszych operacji wykonywanych w ramach modelowania predykcyjnego, ponieważ od jej wyników zależy jakość przyszłego modelu. W badaniach, które stanowią podstawę napisania niniejszej monografii, do redukcji wymiaru danych zastosowano właśnie selekcję. Ze względu na znaczenie tej operacji, poświęcono jej kolejny podrozdział.

3.5.2. Selekcja cech

Selekcja cech polega na tym, że w sposób automatyczny wybierane są te cechy z całego zbioru, które mają największy wpływ na wynik predykcji. Metody

selekcji cech można podzielić na trzy grupy, są to metody filtra, opakowujące i wbudowane [94].

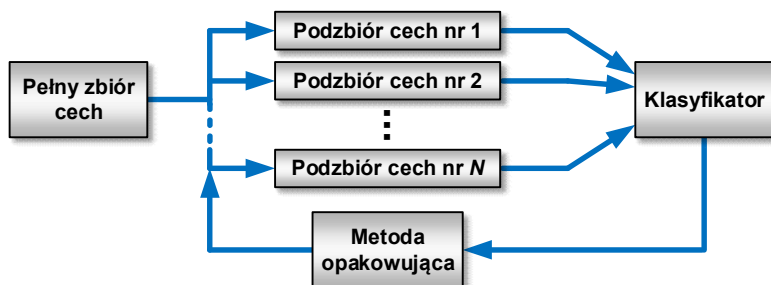
Metody filtra analizują wewnętrzne właściwości danych i wybierają podzbiór cech niezależnie od jakiegokolwiek klasyfikatora. Przypisują one punkty do każdej cechy, zależnie od wartości przyjętej miary statystycznej. Stosowane są w tym celu, np. współczynnik korelacji, wariancja, wskaźnik Fishera, informacja wzajemna, czy wynik testu chi-kwadrat. Na tej podstawie, cechy są porządkowane według wyniku (ranking), co ułatwia wybór ich podzbioru do dalszego wykorzystania (rys. 3.14).



Rys. 3.14. Schemat działania metod filtra

Metody tego typu mogą być jednowymiarowe lub wielowymiarowe. Metody jednowymiarowe rozpatrują daną cechę niezależnie od innych. Są one często stosowane do usuwania z danych cech o stałych i quasi-stałych wartościach. Ich wadą jest to, że mogą wybierać cechy nadmiarowe, ponieważ podczas podejmowania decyzji nie uwzględniają związku między poszczególnymi cechami. Z kolei, metody wielowymiarowe uwzględniają wzajemne relacje między cechami. Z tego powodu, są one często stosowane do usuwania cech zduplikowanych i skorelowanych między sobą.

Metody opakowujące opierają się na założeniu, że selekcja cech jest problemem wyszukiwania. Częścią procesu selekcji jest tutaj optymalizacja modelu zastosowanego do oceny podzbioru cech, dlatego przygotowywane są różne kombinacje cech, które są następnie oceniane i porównywane z innymi kombinacjami (rys. 3.15). Model predyktora ocenia te kombinacje i przypisuje im wynik na podstawie uzyskanej dokładności.



Rys. 3.15. Schemat działania metod opakowujących

Do wyszukiwania kombinacji cech stosowane są różne podejścia. Mogą one być heurystyczne (np. *przejsie do przodu* i *do tyłu* w celu dodania i usunięcia cech), stochastyczne (np. algorytm *hill-climbing*), czy systematyczne (np. algorytm *best-first search*). Metody opakowujące mogą dawać lepsze wyniki, niż metody filtra, ale są droższe obliczeniowo. Ponadto, optymalny zestaw cech wyszukany dla użytego modelu może nie być optymalny dla każdego innego modelu.

Metody wbudowane oceniają, które cechy najbardziej przyczyniają się do poprawy dokładności modelu podczas jego tworzenia. W przeciwieństwie do metod filtra i opakowujących, czynności uczenia klasyfikatora i wyboru cech nie mogą być rozdzielone (rys. 3.16). W związku z tym, znaleziony podzbiór cech może być wykorzystany tylko z klasyfikatorem, który był użyty przez daną metodę selekcji. Najpopularniejsze metody wbudowane to metody regularyzacyjne, zwane także metodami penalizacyjnymi. Wprowadzają one dodatkowe ograniczenia podczas optymalizacji algorytmu uczenia, dzięki czemu zmniejsza się złożoność modelu końcowego (mniejsza liczba współczynników).



Rys. 3.16. Schemat działania metod wbudowanych

Większość metod należących do wymienionych grup wykonuje dwie operacje, tj. ranking i wybór podzbioru cech. Podczas pierwszej czynności, oceniane jest znaczenie każdej pojedynczej cechy. Odbywa się to zazwyczaj poprzez pominięcie potencjalnego związku tej cechy z pozostałymi. Podczas drugiej czynności, dostarczany jest końcowy podzbiór cech. Niektóre metody, te dwie operacje (ranking i wybór podzbioru) wykonują sekwencyjnie [95]. Inne metody przeprowadzają tylko selekcję [96]. Ogólnie rzecz biorąc, wybór podzbioru cech jest zawsze nadzorowany, natomiast ranking może być wykonywany w sposób nadzorowany lub nie.

Dostępnych jest wiele bibliotek zawierających funkcje implementujące różne metody selekcji cech. Jako przykładowe, można wymienić *scikit-learn* dla Pythona [97], *caret* i *clusterSim* dla R [98, 99] oraz *FSLib* dla Matlaba [100]. Popularne jest także oprogramowanie o szerszym zastosowaniu, np. *MaZda* [72], czy *Weka* [101] oferujące różne metody selekcji cech.

3.5.3. Uczenie modelu

Drugi etap modelowania predykcyjnego to uczenie modelu. Zaprojektowano dotychczas wiele różnorodnych algorytmów uczenia maszynowego, służących do rozwiązywania problemów z różnych obszarów. Z punktu widzenia postaci

modelu wyjściowego, można je podzielić na dwie grupy [102, 103]. Do pierwszej należą, np.: liniowa i kwadratowa analiza dyskryminacyjna [104], naiwny klasyfikator Bayes'a, metoda K -najbliższych sąsiadów [105], sztuczne sieci neuronowe [106, 107], metoda wektorów nośnych [90, 91, 108, 109, 110], czy drzewa klasyfikacyjne [92, 111, 112, 113, 114]. Metody te pozwalają zbudować klasyfikator w postaci pojedynczego modelu.

Druga grupa metod wykorzystuje podejście wielomodelowe, polegające na łączeniu pewnej liczby n modeli bazowych (D_1, \dots, D_n) w jeden model zagregowany D^* . Jako przykłady, można tutaj wymienić: agregację bootstrapową [115], metodę wzmacniania [116], wiązanie modeli [117], czy lasy losowe [118]. Podejście wielomodelowe jest często stosowane do poprawy dokładności klasyfikacji, ponieważ model zagregowany jest zwykle bardziej precyzyjny, niż jakikolwiek z modeli pojedynczych, które wchodzi w jego skład [119].

Jeżeli nie określimy założeń dotyczących rozwiązywanego zadania, to żaden z algorytmów uczenia nie może być uznany za lepszy od pozostałych. Dlatego, rozsądną strategią jest wykorzystanie kilku różnych algorytmów, w celu wytrenowania wielu modeli i wybór najskuteczniejszego z nich.

Z pewnym uproszczeniem można powiedzieć, że podczas uczenia modelu generowanych jest wiele jego tymczasowych wersji. Jako ostateczny model wybrana zostaje ta wersja, która ma najlepsze zdolności generalizacyjne. Aby je ocenić, stosowane są często różnorodne techniki sprawdzianu krzyżowego. Ich zadaniem jest podział zbioru uczącego na część uczącą i walidacyjną. Ta ostatnia służy właśnie do oceny generalizacji modelu. Należy mieć także świadomość, że algorytmy uczenia maszynowego, zaimplementowane w różnych bibliotekach programistycznych, mają pewien zestaw parametrów (zwanych hiperparametrami) przyjmujących wartości domyślne. Są to elementy regulacyjne modelu, wykorzystywane do poprawy jego jakości. Dlatego, często stosowane są różne techniki optymalizacji hiperparametrów, w celu poprawy skuteczności modelu w rozwiązywaniu określonego problemu.

3.5.4. Ewaluacja modelu

Podczas ewaluacji, ocenie poddawany jest model ostateczny, a więc taki, który uzyskał najlepsze zdolności generalizacyjne dla danych walidacyjnych. Na tym etapie, szacowana jest skuteczność tego modelu dla nieznanymi danych testowych. Do oceny mogą być stosowane różne metryki wydajności. Podstawą ich obliczenia jest tzw. macierz pomyłek (rys. 3.17), w której zliczane są wyniki przewidywań poszczególnych klas.

Znaczenie akronimów zapisanych niebieską czcionką na rysunku 3.17, jest następujące:

- *TP* (ang. *True Positive*) – liczba przypadków *pozytywnych* zaklasyfikowanych jako *pozytywne*;
- *FN* (ang. *False Negative*) – liczba przypadków *pozytywnych* zaklasyfikowanych jako *negatywne*;
- *TN* (ang. *True Negative*) – liczba przypadków *negatywnych* zaklasyfikowanych jako *negatywne*;
- *FP* (ang. *False Positive*) – liczba przypadków *negatywnych* zaklasyfikowanych jako *pozytywne*.

Klasa rzeczywista	Prawdziwie pozytywne <i>TP</i>	Falszywie negatywne <i>FN</i>
	Falszywie pozytywne <i>FP</i>	Prawdziwie negatywne <i>TN</i>
	Klasa przewidywana	

Rys. 3.17. Macierz pomyłek

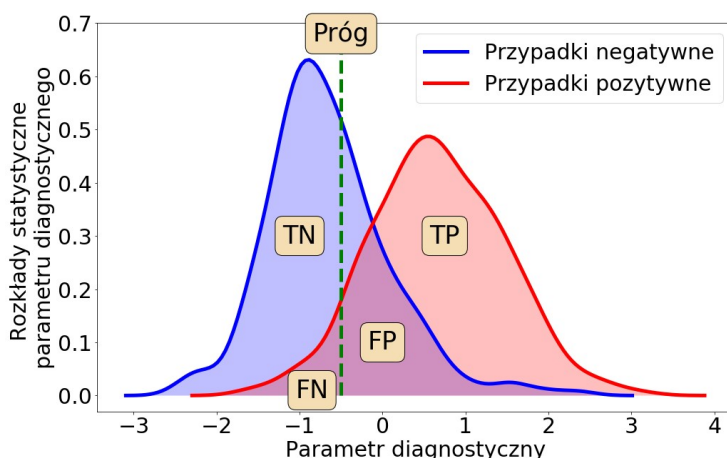
W zastosowaniach medycznych, przypadek *pozytywny* odpowiada zdiagnozowanej jednostce chorobowej, natomiast jako *negatywne*, określane są przypadki kontrolne (odpowiadające osobom zdrowym). Parametry diagnostyczne odpowiadające danym rzeczywistym nie dyskryminują analizowanych przypadków w sposób jednoznaczny. Podlegają one pewnym rozkładom statystycznym, których przykład przedstawiono na rysunku 3.18. Dlatego, relacje pomiędzy wynikami przewidywań poszczególnych klas zależą od wartości przyjętego progu. Przesunięcie progu w prawo powoduje, że większa część przypadków *pozytywnych* zostanie zakwalifikowana jako *negatywne* (rośnie *FN*). Jednocześnie, większa liczba przypadków *negatywnych* będzie zakwalifikowana prawidłowo (rośnie *TN*). Przeciwna sytuacja zaistnieje, gdy punkt odcięcia zostanie przesunięty w lewo.

Niżej podano definicje metryk wydajności stosowanych w systemach klasyfikacji binarnej, w których klasyfikator szacuje prawdopodobieństwo przynależności przypadku do klasy pozytywnej.

Czułość, wrażliwość (ang. *True Positive Rate, TPR*) lub **pełność** (ang. *Recall, REC*):

$$TPR = REC = \frac{TP}{TP + FN} . \quad (3.34)$$

Czułość daje oszacowanie prawdopodobieństwa poprawnej klasyfikacji przypadku jako *pozytywnego* pod warunkiem, że rzeczywiście należy on do klasy *pozytywnej*. Można także powiedzieć, że czułość (zwana również pełnością) określa, jak duża część trafnych obserwacji została wybrana. Odnosząc to do danych medycznych, czułość określa zdolność testu do poprawnego rozpoznania choroby.



Rys. 3.18. Przykładowe rozkłady statystyczne parametru diagnostycznego

Specyficzność, swoistość (ang. *True Negative Rate*, *TNR*):

$$TNR = \frac{TN}{TN + FP} = 1 - \frac{FP}{FP + TN} . \quad (3.35)$$

Specyficzność daje oszacowanie prawdopodobieństwa prawidłowej klasyfikacji przypadku jako *negatywnego*, pod warunkiem, że naprawdę należy on do klasy *negatywnej*. W medycynie, specyficzność określa zdolność testu do poprawnego wykluczenia choroby.

Wskaźnik fałszywie dodatni, fałszywy alarm (ang. *False Positive Rate*, *FPR*):

$$FPR = \frac{FP}{FP + TN} = 1 - TNR . \quad (3.36)$$

Wskaźnik fałszywie dodatni jest oszacowaniem prawdopodobieństwa błędu polegającego na przewidzeniu przypadku *pozytywnego*, gdy jest on *negatywny*.

Dla danych medycznych, fałszywy alarm określa liczbę fałszywie zdiagnozowanych pacjentów zdrowych, w stosunku do liczby pacjentów zdrowych.

Wartość predykcyjna dodatnia (ang. *Positive Predictive Value, PPV*):

$$PPV = \frac{TP}{TP + FP} . \quad (3.37)$$

Wartość predykcyjna dodatnia oznacza prawdopodobieństwo tego, że przypadek rzeczywiście należy do klasy *pozytywnej*, przy jego *pozytywnej* klasyfikacji. W medycynie, *PPV* jest prawdopodobieństwem tego, że pacjent jest chory, mając dodatni wynik testu. Jeśli więc pacjent otrzymał dodatni wynik testu, to *PPV*, daje informację na ile może być pewien, że faktycznie jest chory.

Wartość predykcyjna ujemna (ang. *Negative Predictive Value, NPV*):

$$NPV = \frac{TN}{TN + FN} . \quad (3.38)$$

Wartość predykcyjna ujemna jest prawdopodobieństwem tego, że przypadek rzeczywiście należy do klasy *negatywnej*, przy jego *negatywnej* klasyfikacji. Dla danych medycznych, *NPV* jest prawdopodobieństwem tego, że pacjent jest zdrowy mając ujemny wynik testu. Jeśli więc pacjent otrzymał wynik ujemny, to *NPV* daje informację na ile może być pewien, że faktycznie jest zdrowy.

Dokładność (ang. *Accuracy, ACC*):

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} . \quad (3.39)$$

Dokładność klasyfikacji jest prawdopodobieństwem prawidłowej klasyfikacji przypadków należących do obu kategorii. Odnosząc to do danych medycznych, dokładność określa liczbę prawidłowo zdiagnozowanych pacjentów w stosunku do liczby wszystkich pacjentów.

Błąd (ang. *Error, ERR*):

$$ERR = \frac{FP + FN}{TP + TN + FP + FN} = 1 - ACC . \quad (3.40)$$

Błąd klasyfikacji jest prawdopodobieństwem popełnienia błędu podczas klasyfikacji przypadków należących do obu kategorii. W medycynie, określa on liczbę błędnie zdiagnozowanych pacjentów w stosunku do liczby wszystkich pacjentów.

Precyzja (ang. *Precision, PRE*):

$$PRE = \frac{TP}{TP + FP} . \quad (3.41)$$

Precyzja określa, jak duża część wybranych obserwacji jest trafna.

Wskaźnik F1 (ang. *F1 Score, F1*):

$$F1 = 2 \cdot \frac{PRE \cdot REC}{PRE + REC}. \quad (3.42)$$

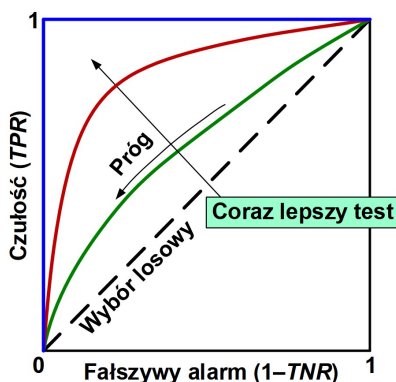
Wskaźnik F1 jest średnią harmoniczną precyzji i pełności. Będzie on duży, gdy zarówno precyzja, jak i pełność będą miały dużą wartość.

Wskaźnik fałszywego rozpoznania (ang. *False Discovery Rate, FDR*):

$$FDR = \frac{FP}{FP + TP}. \quad (3.43)$$

Wskaźnik fałszywego rozpoznania określa odsetek przypadków, które zostały fałszywie zaklasyfikowane do grupy *pozytywnej*.

Czułość i specyficzność są często stosowane do budowy tzw. **krzywej operacyjno-charakterystycznej odbiornika ROC** (ang. *Receiver Operating Characteristic Curve, ROC*). Obrazuje ona zależność pomiędzy czułością i fałszywym alarmem w pełnym zakresie pracy klasyfikatora i umożliwia porównanie różnych klasyfikatorów. Jak pokazano na rysunku 3.19, krzywa ROC jest miejscem geometrycznym punktów na płaszczyźnie o współrzędnych (*fałszywy alarm, czułość*).



Rys. 3.19. Krzywa ROC

Na podstawie krzywej ROC można obliczyć tzw. **obszar pod krzywą ROC** (ang. *Area Under the Curve, AUC*), opisujący skuteczność modelu klasyfikatora. *AUC* przyjmuje wartości z zakresu od 0 do 1, przy czym im jest większe, tym większa jest moc diagnostyczna testu. W diagnostyce medycznej, większość testów reprezentuje moc diagnostyczną wyrażającą się wartościami *AUC* pomiędzy 0,80 a 0,95.

Opisowo przyjmuje się:

- $AUC = 0,9 \div 1,0$ – test bardzo dobry;
- $AUC = 0,8 \div 0,9$ – test dobry;
- $AUC = 0,7 \div 0,8$ – test satysfakcjonujący;
- $AUC = 0,6 \div 0,7$ – test średni;
- $AUC = 0,5 \div 0,6$ – test niedostateczny.

W zadaniach klasyfikacji realizowanych w zastosowaniach medycznych, naturalnym oczekiwaniem jest, aby klasyfikator był czuły, czyli dawał wynik *pozytywny*, gdy pacjent jest chory, ale jednocześnie był specyficzny, czyli dawał wynik *negatywny*, gdy pacjent jest zdrowy. Niestety, czułość i specyficzność są wymaganiami przeciwnymi – zwiększenie jednego prowadzi do zmniejszenia drugiego. Dlatego też, klasyfikator wybierany jest zwykle na drodze kompromisu między obydwojema wymaganiami tak, by osiągnąć jak najmniejsze koszty błędnej klasyfikacji [90, 120, 121]. Z medycznego punktu widzenia (korzyść dla pacjenta), mniejsze koszty błędnej klasyfikacji wystąpią wówczas, gdy pacjent zdrowy zostanie zaklasyfikowany do grupy chorych.

W sytuacji, gdy stosowanych jest kilka algorytmów uczenia, ewaluacji podlega wiele różnych modeli. Spośród nich, wybrany zostaje model najskuteczniejszy, który następnie może być stosowany do prognozowania przyszłych, nowych danych. Należy w tym miejscu dodać, iż parametry procedur stosowanych do wstępnego przetwarzania danych, takich jak skalowanie cech, czy redukcja wymiarowości, określane są wyłącznie na podstawie zbioru uczącego. Następnie, są one wykorzystywane do przekształcania danych testowych oraz nowych danych.

3.5.5. Prognozowanie

Prognozowanie to etap wdrożenia do użytku modelu uczenia maszynowego. Na rysunku 3.20 pokazano schemat systemu do klasyfikacji tekstur. Na początku, nowe dane zostają poddane przetwarzaniu wstępnemu. Przykładowe operacje, które mogą być wykonywane na tym etapie to: konwersja trybu RGB do skali szarości, wyznaczenie obszaru zainteresowania, normalizacja histogramu obrazu, czy usunięcie szumu i artefaktów. Następnie, dokonywana jest analiza obrazu, mająca na celu estymację jego istotnych cech. Mamy tutaj na myśli tylko ten zbiór cech, który został wykorzystany do uczenia najskuteczniejszego modelu klasyfikacyjnego. W kolejnym kroku, wyłonione cechy zostają poddane skalowaniu (np. standaryzacji) z użyciem tych samych wektorów wartości średnich i wariancji, które były wykorzystane do skalowania zbioru uczącego. Ostatnim krokiem jest klasyfikacja nowych obrazów. Wykorzystany jest tutaj model klasyfikacyjny, który okazał się najskuteczniejszy na etapie ewaluacji, w odniesieniu do zbioru testowego. Na podstawie tego modelu oraz wartości deskryptorów cech, analizowanym obrazom przypisywane są etykiety klasy (prognozowanie).



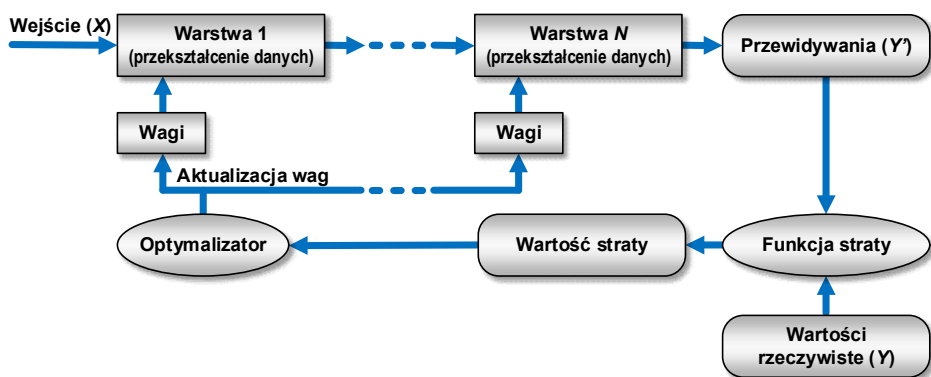
Rys. 3.20. Prognozowanie przynależności do klasy, w odniesieniu do nowych obrazów

4. Uczenie głębokie

4.1. Ogólne działanie i rodzaje sieci głębokich

Uczenie głębokie to obszar uczenia maszynowego, w którym kolejne warstwy modelu uczą się coraz lepszych reprezentacji na podstawie danych. Liczba tych warstw (głębokość) może wynosić od kilkudziesięciu do kilkuset. Jest to istotna różnica, w porównaniu z modelami uczenia tradycyjnego, które zawierają przeważnie jedną lub dwie warstwy.

W uczeniu głębokim, kolejne warstwy modelu przekształcają dane wejściowe na dane wyjściowe, przy czym operacje te są parametryzowane przez wagi danej warstwy. Na rysunku 4.1 przedstawiono architekturę wielowarstwowej sieci jednokierunkowej, w której sygnał biegnie tylko w jednym kierunku, od wejścia do wyjścia. Do kontroli działania takich sieci służy funkcja straty. Przyjmuje ona argumenty w postaci wartości przewidywanych przez sieć oraz wartości prawdziwych i oblicza, jak bardzo różnią się od siebie. Uzyskany wynik pełni rolę informacji zwrotnej, służącej do dostrojenia wartości wag w taki sposób, aby zmniejszyć stratę dla bieżącej obserwacji. Wagi dostrajane są przez zaimplementowany algorytm optymalizacji.



Rys. 4.1. Podstawowe elementy wielowarstwowej sieci jednokierunkowej

Istnieje wiele metod uczenia wielowarstwowych sieci jednokierunkowych, ale do najczęściej stosowanych należy algorytm wstecznej propagacji błędów [122, 123, 124]. Uczenie polega na tym, że na początku tego procesu, wagi inicjowane są małymi wartościami, zgodnie z wybraną strategią [125, 126]. Strata jest na tym etapie przeważnie duża, ponieważ wartości przewidywane mogą znacznie odbiegać od prawdziwych. W wyniku każdej kolejnej iteracji algorytmu (w pętli trenowania), wagi są nieco dostrajane i strata maleje. Wytrenowanie sieci polega na powtórzeniu algorytmu odpowiednią liczbę razy (przeważnie od

kilkudziesięciu do kilkuset). W efekcie, pozwala to na uzyskanie wag minimalizujących stratę i generowanie takich wartości przewidywanych, które są maksymalnie zbliżone do wartości prawdziwych.

Do najbardziej znanych architektur sieci głębokich należą:

- perceptrony wielowarstwowe (ang. *Multi-Layer Perceptron*, MLP) – realizują zadania klasyfikacji [127];
- ograniczone maszyny Boltzmanna (ang. *Restricted Boltzmann Machines*, RBM) – są składnikami sieci DBN [128, 129];
- głębokie sieci przekonań (ang. *Deep Belief Networks*, DBN) – redukują wymiar danych [130];
- autoenkodery (stosowe, odsumiające, rzadkie, wariacyjne) – redukują wymiar danych, odsumiają dane wejściowe, można w nich nakładać warunek rzadkości na warstwę kodowania, losowo generują nowe dane przypominające dane uczące [131, 132, 133];
- splotowe sieci neuronowe (ang. *Convolutional Neural Network*, CNN) – rozpoznają obrazy, rozpoznają mowę, przetwarzają język naturalny [134, 135, 136, 137, 138];
- rekurencyjne sieci neuronowe (ang. *Recurrent Neural Network*, RNN) – analizują szeregi czasowe, wykrywają anomalie w danych, są stosowane w pojazdach autonomicznych, automatycznym tłumaczeniu, przetwarzaniu mowy na tekst, analizie sentymentów [139, 140, 141, 142, 143, 144];
- generatywne sieci przeciwstawne lub generatywne sieci z przeciwnikiem (ang. *Generative Adversarial Networks*, GAN) – generują realistycznie wyglądające, sztuczne obrazy [145].

Wyniki badań przedstawione w dalszej części monografii uzyskano wykorzystując sieci splotowe do rozwiązania zadania klasyfikacji binarnej. Z tego tytułu, informacje przedstawione w kolejnych rozdziałach będą ukierunkowane głównie na zastosowanie sieci splotowych do klasyfikacji obrazów.

4.2. Sztuczne sieci neuronowe

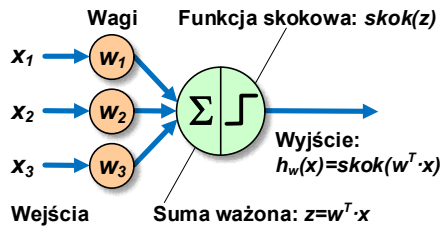
4.2.1. Preceptron

Jedną z najprostszych architektur sztucznych sieci neuronowych jest **perceptron**, zaproponowany przez Franka Rosenblatta w 1957 r. Powstał on w wyniku modyfikacji sztucznego neuronu, zwanego też jednostką liniową z progiem (ang. *Linear Threshold Unit*, LTU). Wartościami wejść oraz wyjść takiego elementu są liczby, a każde połączenie ma przypisaną wagę (rys. 4.2). Wspomniana jednostka oblicza sumę ważoną sygnałów wejściowych z :

$$z = w_1x_1 + w_2x_2 + \dots + w_nx_n = w^T \cdot x, \quad (4.1)$$

a następnie, wobec tej sumy zostaje zastosowana funkcja skokowa dająca wynik:

$$h_w(x) = \text{skok}(z) = \text{skok}(w^T \cdot x). \quad (4.2)$$



Rys. 4.2. Jednostka liniowa z progiem

W perceptronach, jako funkcje skokowe, najczęściej używane są funkcje Heaviside'a (4.3) lub signum (4.4):

$$\text{Heaviside}(z) = \begin{cases} 0 & \text{dla } z < 0 \\ 1 & \text{dla } z \geq 0 \end{cases}, \quad (4.3)$$

$$\text{sgn}(z) = \begin{cases} -1 & \text{dla } z < 0 \\ 0 & \text{dla } z = 0 \\ +1 & \text{dla } z > 0 \end{cases}. \quad (4.4)$$

Scharakteryzowana wcześniej jednostka liniowa z progiem może rozwiązywać proste zadanie klasyfikacji. Realizuje go w ten sposób, że oblicza liniową kombinację danych wejściowych i jeżeli wynik przekracza określony próg, klasyfikuje obserwację do klasy pozytywnej, w przeciwnym wypadku – do negatywnej.

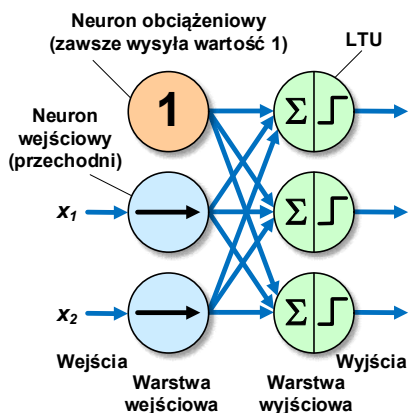
Perceptron zbudowany jest z jednej warstwy jednostek liniowych (neuronów), w której każdy element jest połączony ze wszystkimi wejściami (rys. 4.3). Połączenia są realizowane przez neurony wejściowe (zwane też przechodnikami), których zadaniem jest przekazywanie dalej wszystkich dostarczonych im danych wejściowych. Ponadto, w strukturze perceptronu występuje tzw. neuron obciążeniowy. Reprezentuje on dodatkową zmienną wejściową x_0 (tzw. wagę obciążenia) o wartości 1.

Perceptrony uczone są za pomocą odmiany reguły Hebba, według której waga połączenia pomiędzy dwoma neuronami wzrasta za każdym razem, gdy mają ten sam stan na wyjściu. Perceptron jednocześnie przetwarza jedną próbkę uczącą i oblicza dla niej prognozy. Dla każdego neuronu wyjściowego, zwracającego błędny wynik, zwiększane są wagi połączeń ze wszystkimi wejściami, które przyczyniają się do prognozy prawidłowej. Zatem, w procesie uczenia uwzględniany jest błąd popełniany przez sieć, dzięki czemu, nie są wzmacniane takie

połączenia, które prowadzą do nieprawidłowych wyników. Powyższa zasada została ujęta w równaniu:

$$w_{i,j}^{(\text{następny krok})} = w_{i,j} + \eta(y_j - \hat{y}_j)x_i, \quad (4.5)$$

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



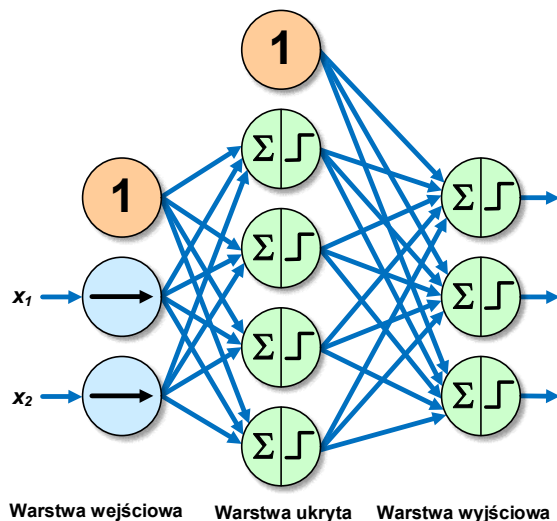
Rys. 4.3. Budowa perceptronu

4.2.2. Perceptron wielowarstwowy

Opisany model perceptronu ma wiele wad, związanych z brakiem możliwości rozwiązywania pewnych problemów (np. klasyfikacja XOR). Część ograniczeń perceptronu można jednak wyeliminować, stosując wiele warstw takich elementów. Powstała w ten sposób architektura to **perceptron wielowarstwowy** (MLP). Podstawowe elementy składowe MLP to jedna warstwa wejściowa, co najmniej jedna warstwa jednostek LTU (warstwy ukryte) oraz ostatnia warstwa jednostek LTU (warstwa wyjściowa). Z wyjątkiem warstwy wyjściowej, każda warstwa zawiera neuron obciążeniowy oraz jest w pełni połączona z warstwą następną (rys. 4.4). Jeżeli sieć tego typu zawiera przynajmniej dwie warstwy ukryte, wówczas nazywamy ją **głębką siecią neuronową**.

Perceptrony wielowarstwowe uczone są za pomocą algorytmu propagacji wstecznej. Idea jego działania polega na tym, że dla każdej próbki uczącej, algorytm najpierw oblicza prognozę (przebieg w przód), mierzy błąd, następnie cofa się w kierunku początku sieci, sprawdzając wkład każdego połączenia w zmie-

rzony błąd (przebieg odwrotny), a na końcu nieznacznie modyfikuje wagi, w celu zredukowania błędu.



Rys. 4.4. Perceptron wielowarstwowy

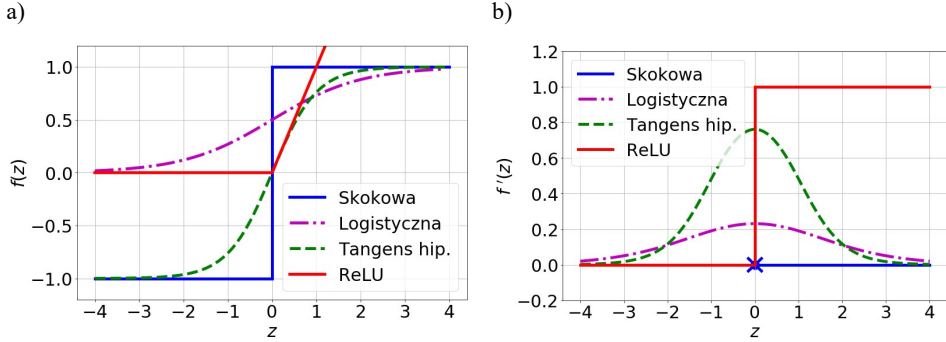
Wprowadzenie algorytmu propagacji wstecznej wymusiło kluczową zmianę w architekturze MLP. Zrezygnowano z funkcji skokowej, ponieważ jej przebieg zawiera tylko odcinki płaskie, które uniemożliwiają korzystanie z gradientu (algorytm nie może poruszać się po płaskim przebiegu funkcji). Zaczęto stosować inne funkcje, np. funkcję logistyczną (sigmoidalną):

$$\sigma(z) = \frac{1}{1 + e^{-z}}, \quad (4.6)$$

Pochodna funkcji logistycznej w każdym punkcie jest niezerowa. Oprócz funkcji logistycznej, algorytm propagacji wstecznej może być również używany z innymi funkcjami aktywacji (rys. 4.5). Do najczęściej stosowanych należą: funkcja tangensa hiperbolicznego (4.7) oraz funkcja ReLU (ang. *Rectified Linear Unit*) (4.8):

$$\tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}} = 2\sigma(2z) - 1, \quad (4.7)$$

$$\text{ReLU}(z) = \max(0, z). \quad (4.8)$$



Rys. 4.5. Na rysunku przedstawiono wykresy: a) funkcji aktywacji; b) pochodnych funkcji aktywacji; z – suma ważona sygnałów wejściowych

4.2.3. Szacowanie prawdopodobieństwa przynależności do klasy

Klasyfikacja binarna. Jedno z zastosowań perceptronu wielowarstwowego to rozwiązywanie zadania klasyfikacji binarnej, gdzie każde wyjście odpowiada innej klasie (np. *zdrowy/chory*). Wówczas, w warstwie wyjściowej, jako funkcja aktywacji stosowana jest funkcja logistyczna. Szacuje ona prawdopodobieństwo \hat{p} przynależności próbki x do klasy pozytywnej:

$$\hat{p} = h_{\theta}(x) = \sigma(\theta^T \cdot x), \quad (4.9)$$

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.

Na tej podstawie może być obliczona prognoza \hat{y} :

$$\hat{y} = \begin{cases} 0 & \text{gdy } \hat{p} < 0,5 \\ 1 & \text{gdy } \hat{p} \geq 0,5 \end{cases}. \quad (4.10)$$

Zauważmy, że $\sigma(z) < 0,5$, gdy $z < 0$ oraz $\sigma(z) \geq 0,5$, gdy $z \geq 0$, zatem model przewiduje 1, jeżeli wynik $\theta^T \cdot x$ jest dodatni oraz 0, gdy jest on ujemny.

Klasyfikacja wieloklasowa. Perceptrony wielowarstwowe są stosowane także w sytuacji, gdy poszczególne klasy wykluczają się wzajemnie (klasyfikacja wieloklasowa). Wówczas, warstwa wyjściowa zostaje zmodyfikowana w taki sposób, że pojedyncze funkcje aktywacji zostają zastąpione funkcją *softmax*. W tym przypadku, dla danej próbki x obliczony zostaje wynik $s_k(x)$ dla każdej klasy k :

$$s_k(x) = (\theta^{(k)})^T \cdot x, \quad (4.11)$$

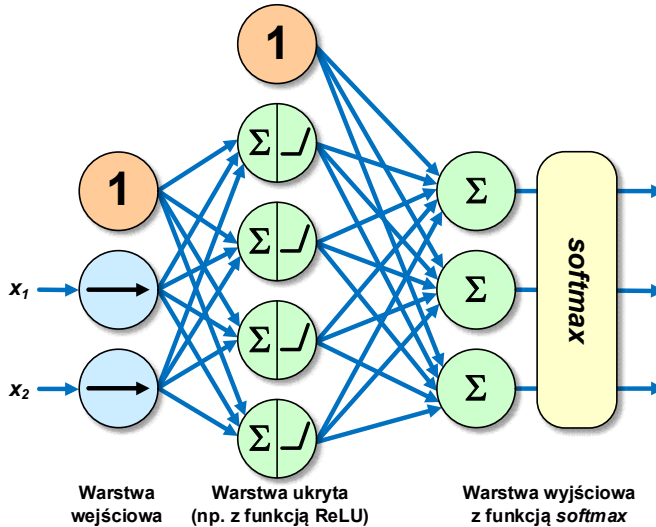
gdzie $\theta^{(k)}$ jest własnym, wyspecjalizowanym wektorem parametrów klasy k . Po obliczeniu wyników każdej klasy dla próbki x , można oszacować prawdopodobieństwo \hat{p}_k przynależności tej próbki do klasy k :

$$\hat{p}_k = \sigma(s(x))_k = \frac{e^{s_k(x)}}{\sum_{j=1}^K e^{s_j(x)}}, \quad (4.12)$$

gdzie: K – liczba klas; $s(x)$ – wektor zawierający wyniki każdej klasy dla próbki x ; $\sigma(s(x))_k$ – szacowane prawdopodobieństwo przynależności próbki x do klasy k przy znanych wynikach wszystkich klas dla tego przykładu. Wynik warstwy wyjściowej jest normalizowany, dlatego suma zawsze wynosi 1. W związku z tym, wynik każdego neuronu odpowiada oszacowanemu prawdopodobieństwu przynależności do danej klasy (rys. 4.6). Na tej podstawie obliczona zostaje prognoza klasy o najwyższym wyniku:

$$\hat{y} = \arg \max_k \sigma(s(x))_k = \arg \max_k s_k(x) = \arg \max_k ((\theta^{(k)})^T \cdot x), \quad (4.13)$$

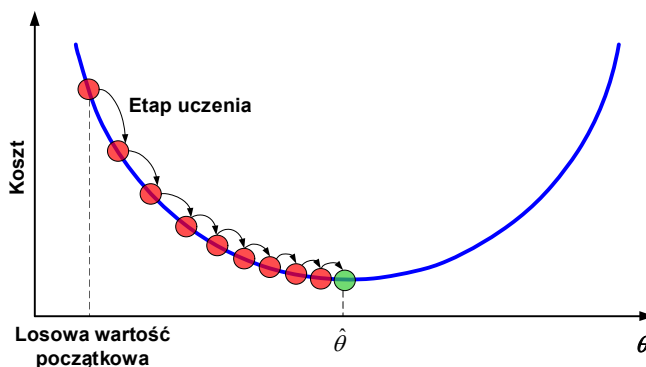
gdzie *argmax* – operator zwracający wartość zmiennej maksymalizującej funkcję. W równaniu (4.13) zwraca on wartość parametru k maksymalizującą szacowane prawdopodobieństwo $\sigma(s(x))_k$.



Rys. 4.6. Perceptron wielowarstwowi z funkcjami ReLU i *softmax*

4.2.4. Metoda gradientu prostego

Zaprojektowano dotychczas wiele algorytmów optymalizacji, służących do uczenia sztucznych sieci neuronowych. Jako przykład, zostanie scharakteryzowana **metoda gradientu prostego**, która jest wykorzystywana do wyszukiwania optymalnych rozwiązań dla bardzo szerokiej gamy problemów. Na tej podstawie, przybliżona zostanie idea optymalizacji oraz główne problemy związane z tym procesem. Metoda gradientu prostego zaczyna swoje działanie od inicjalizacji wektora wag modelu θ . Następnie, mierzony jest lokalny gradient funkcji straty i poprawiane są wagi tak, aby uzyskać mniejszą wartość gradientu. Operacja aktualizacji wag wykonywana jest wielokrotnie, a po każdym jej etapie osiąga się coraz mniejszą wartość funkcji straty. Uzyskanie zerowej wartości gradientu (zbieżność algorytmu) oznacza minimum funkcji straty i tym samym ustawienie optymalnych wartości wag (rys. 4.7).

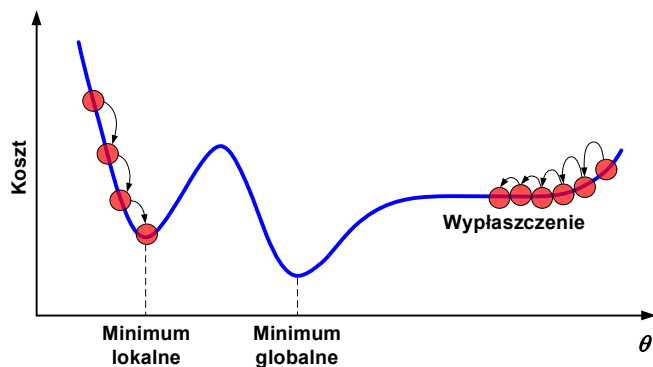


Rys. 4.7. Metoda gradientu prostego

W metodzie tej, istotne znaczenie ma współczynnik uczenia (η), który określa jak bardzo w poszczególnych krokach będą zmieniały się wagi modelu. Zbyt mała wartość tego parametru powoduje, że uzyskanie zbieżności będzie czasochłonne, ponieważ wykonana będzie duża liczba kroków. Z kolei, zbyt duży współczynnik uczenia sprawia, że algorytm może stać się rozbieżny. Wynikać to może z ustawienia wag odpowiadających przeciwnemu wzniesieniu w przebiegu funkcji straty (przeoczenie minimum).

Inne problemy związane z metodą gradientu prostego wynikają z tego, że w rzeczywistym przebiegu funkcji straty mogą występować nieregularności (doliny, pagórki, wypłaszczenia), znacznie utrudniające uzyskanie zbieżności (rys. 4.8). W sytuacji, gdy inicjalizacja ustawi wagi odpowiadające lewej stronie wykresu, algorytm osiągnie minimum lokalne, a nie globalne. Jeżeli algorytm rozpocznie pracę po prawej stronie wykresu, przejście przez wypłaszczenie będzie bardzo czasochłonne. Algorytm może się także zbyt wcześnie zatrzymać

i nigdy nie osiągnąć minimum globalnego. Należy także pamiętać, iż podczas korzystania z metody gradientu prostego wszystkie cechy powinny być przeskalowane. W przeciwnym wypadku, osiągnięcie zbieżności może zająć znacznie więcej czasu.



Rys. 4.8. Potencjalne problemy metody gradientu prostego

Istnieją trzy podstawowe implementacje metody gradientu prostego:

1. **Wsadowy gradient prosty** – wykorzystuje pełny zbiór danych uczących w każdym przebiegu. W efekcie, algorytm jest bardzo powolny w przypadku dużych zbiorów danych.
2. **Stochastyczny spadek wzdłuż gradientu** – podczas każdego przebiegu dobierana jest tylko jedna, losowa próbka ucząca, za pomocą której obliczane są gradienty. Dzięki temu, dla dużych zbiorów danych, algorytm ten jest znacznie szybszy od poprzedniego.
3. **Schodzenie po gradiencie z minigrupami** – w każdym przebiegu, gradient obliczany jest na podstawie niewielkich grup, losowo dobieranych danych. Algorytm ten pozwala na znaczne przyspieszenie operacji na macierzach, szczególnie podczas wykorzystywania procesorów graficznych.

Algorytmy gradientu prostego, np. stochastyczny spadek wzdłuż gradientu, mogą być stosowane do uczenia głębokich sieci neuronowych przetwarzających duże zbiory danych. Jednak w rzeczywistości, optymalizowana funkcja może nie być wypukła, a liczba parametrów może być bardzo duża. W takich warunkach, rośnie ryzyko natrafienia na tzw. punkty siodłowe (zamiast minimumów lokalnych), co może spowolnić proces uczenia na niemal płaskiej powierzchni funkcji. Wówczas, algorytm nie osiąga konfiguracji optymalnej i traci możliwość dalszych poprawek. W celu uniknięcia podobnych problemów, zaprojektowano wiele algorytmów optymalizacyjnych, których celem jest przyspieszenie uzyskiwania zbieżności (szczególnie dla bardzo małych gradientów) oraz zapobieganie niestabilnościom. Najpopularniejsze z nich to algorytmy momentum

i *Nesterova*, *RMSProp*, *Adam* [146], *AdaGrad* [147] i *AdaDelta* [148]. Algorytmy te dokładnie scharakteryzowano w pozycji [149].

4.2.5. Funkcje straty i uczenie modeli

W zależności od rodzaju funkcji używanej w ostatniej warstwie modelu, stosowana jest nieco inna postać funkcji straty. W przypadku funkcji logistycznej (klasyfikacja binarna) jest to logarytmiczna funkcja straty:

$$J(\theta) = -\frac{1}{m} \sum_{i=1}^m \left[y^{(i)} \log(\hat{p}^{(i)}) + (1 - y^{(i)}) \log(1 - \hat{p}^{(i)}) \right]. \quad (4.14)$$

Jak widać na przykładzie równania (4.14), funkcja straty dla całego zbioru uczącego jest średnią stratą obliczoną dla każdej próbki w zbiorze.

Nie ma jawnego wzoru pozwalającego obliczyć wartość θ minimalizującą funkcję straty. Natomiast, funkcja $J(\theta)$ jest wypukła, co daje szansę na znalezienie minimum lokalnego (np. za pomocą metody gradientu prostego). Za pomocą równania (4.15) zapisano pochodne cząstkowe funkcji straty $J(\theta)$ dla j -tego parametru θ_j :

$$\frac{\partial}{\partial \theta_j} J(\theta) = \frac{1}{m} \sum_{i=1}^m \left(\sigma(\theta^T \cdot x^{(i)}) - y^{(i)} \right) x_j^{(i)} \quad (4.15)$$

Dla każdej próbki obliczony zostaje błąd predykcji. Następnie, jest on pomnożony przez wartość j -tej cechy, po czym obliczana jest wartość średnia ze wszystkich próbek uczących. Po otrzymaniu wektora gradientów, zawierającego wszystkie pochodne cząstkowe, może on być użyty przez algorytm optymalizacji.

Jeżeli w ostatniej warstwie modelu stosowana jest funkcja *softmax*, to jako funkcji straty używa się entropii krzyżowej:

$$J(\theta) = -\frac{1}{m} \sum_{i=1}^m \sum_{k=1}^K y_k^{(i)} \log(\hat{p}_k^{(i)}). \quad (4.16)$$

Jest ona często wykorzystywana do sprawdzania, w jakim stopniu zbiór oszacowanych prawdopodobieństw zgadza się z klasami docelowymi. W równaniu (4.16), $y_k^{(i)}$ jest równe 1, jeżeli k jest klasą docelową dla i -tej próbki, w przeciwnym wypadku jego wartość wynosi 0. Zauważmy, że w przypadku klasyfikacji binarnej ($K=2$), funkcja entropii krzyżowej przyjmuje postać logarytmicznej funkcji straty (4.14).

Wektor gradientów entropii krzyżowej dla klasy k można zapisać następująco:

$$\nabla_{\theta^{(k)}} J(\theta) = \frac{1}{m} \sum_{i=1}^m (\hat{p}_k^{(i)} - y_k^{(i)}) x^{(i)}, \quad (4.17)$$

gdzie operator ∇ (nabla) oznacza wszystkie pochodne cząstkowe funkcji straty (po jednej dla każdego parametru modelu). Na tej podstawie można obliczyć wektor gradientów dla każdej klasy, a następnie, używając, np. algorytmu gradientu prostego – wyszukać macierz parametrów θ minimalizującą funkcję straty $J(\theta)$:

$$\theta_{(\text{kolejny krok})}^{(k)} = \theta^{(k)} - \eta \nabla_{\theta^{(k)}} J(\theta) \quad (4.18)$$

Wektor gradientów wskazuje pewien punkt na wzniesieniu funkcji $J(\theta)$. Zatem, aby rozpocząć schodzenie w dół funkcji, należy skierować się w przeciwną stronę, niż wskazuje gradient⁵. Matematycznie, należy to interpretować jako odjęcie wartości $\nabla_{\theta^{(k)}} J(\theta)$ od $\theta^{(k)}$. Współczynnik uczenia (η) przemnożony przez wektor gradientów określa długość kroku.

4.2.6. Techniki wspierające uczenie modelu

Wyjaśnijmy na początku dwa ważne pojęcia uczenia maszynowego – **optymalizację** i **uogólnianie**. Optymalizacja polega na dostrajaniu modelu, w celu uzyskania jak najlepszej wydajności dla danych uczących. Z kolei, uogólnianie określa, jak dobrze model sprawdza się podczas przetwarzania nowych danych (walidacyjnych). Kluczowy problem uczenia maszynowego sprowadza się właśnie do uzyskania kompromisu pomiędzy optymalizacją, a uogólnianiem [150]. Na początku procesu uczenia ma miejsce zbyt słabe dopasowanie modelu do danych uczących (niedotrenowanie). Objawia się to korelacją pomiędzy optymalizacją i uogólnianiem, tzn. im mniejsza będzie strata dla danych uczących, tym mniejsza będzie ona także dla danych walidacyjnych. Po wykonaniu pewnej liczby iteracji (epok) algorytmu uczącego, uogólnianie dla danych walidacyjnych osiąga stały poziom, a następnie (przeważnie) zaczyna się pogarszać. Oznacza to, że ma miejsce nadmierne dopasowanie modelu do danych uczących (przetrenowanie).

W ramach walki z nadmiernym dopasowaniem stosowane są różne metody, które ogólnie nazywamy **metodami regularyzacji**. Jedna z nich polega na **zgrupowaniu większego zbioru uczącego**, niestety w praktyce, nie zawsze jest to możliwe. Inny sposób, to **redukcja pojemności modelu** (liczby parametrów), która zależy od liczby warstw oraz liczby neuronów, które je tworzą. Nie ma

⁵ Wynika to z faktu, iż gradient jest zawsze skierowany w stronę najbliższego maksimum funkcji, więc kierunek przeciwny wskazuje jej najbliższe minimum.

jednak żadnej formuły określającej właściwą liczbę warstw oraz ich rozmiar (liczbę neuronów). W praktyce, w celu znalezienia modelu optymalnego z punktu widzenia analizowanych danych, należy sprawdzić działanie zestawu różnych architektur. Kolejna technika regularyzacji określana jest jako **wczesne zatrzymywanie**. Polega ona na przerywaniu procesu uczenia w momencie spadku wydajności modelu wobec zbioru walidacyjnego. Implementacja tej techniki może polegać, np. na ocenie wydajności modelu dla zbioru walidacyjnego w regularnych odstępach czasu.

Następne podejście, którego celem jest uniknięcie nadmiernego dopasowania, określane jest jako **regularyzacja wag** [122]. Implementuje się go poprzez dodanie do funkcji straty kosztu związanego z dużymi wartościami wag. Istnieją trzy metody regularyzacji wag:

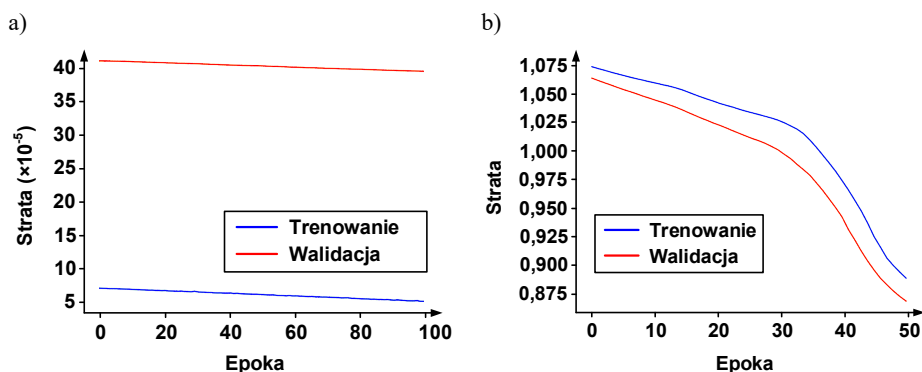
- **regularyzacja L1** – dodawany koszt jest proporcjonalny do bezwzględnej wartości współczynników wag;
- **regularyzacja L2** – dodawany koszt jest proporcjonalny do kwadratu wartości współczynników wag;
- **regularyzacja max-norm** – dla każdego neuronu, wagi połączeń wejściowych w są ograniczane tak, że $\|w\|_2 \leq r$, gdzie r jest hiperparametrem metody max-norm, a $\|\cdot\|_2$ jest członem regularyzacji L2. Przeważnie, ograniczenie to jest implementowane w taki sposób, że po każdym przebiegu uczenia, obliczane jest $\|w\|_2$ i (w razie potrzeby) zmniejszana jest wartość w ($w \leftarrow w(r/\|w\|_2)$).

Kolejna metoda, przeciwdziałająca przetrenowaniu modelu, to **porzucanie** [151, 152]. Jest to jedna z najbardziej skutecznych i najpopularniejszych technik regularyzacji. Polega ona na losowym wybieraniu pewnej liczby cech wyjściowych warstwy podczas uczenia i zastępowaniu tych wartości zerami. Ułamek określający część wyzerowanych cech nazywamy współczynnikiem porzucania. Przeważnie, przyjmuje on wartość z przedziału $0,2 \div 0,5$. Po zakończeniu uczenia, neurony przestają być porzucane.

Ostatnia technika regularyzacji, która ma duże znaczenie szczególnie w kontekście klasyfikacji obrazów, to **dogenerowanie danych** (ang. *Data Augmentation*). Jej celem jest wygenerowanie większej liczby elementów zbioru uczącego na drodze losowych przekształceń, zwracających obrazy wyglądające wiarygodnie. Wykonywany jest obrót obrazu, przekształcenia pionowe i poziome, przycinanie, przybliżanie i oddalanie fragmentów oraz odbicie połowy obrazu w płaszczyźnie poziomej i pionowej. Jednocześnie, stosowane są różne strategie wypełniania nowo utworzonych pikseli, które mogą powstać w wyniku obrotu lub przesunięcia. W efekcie tych przekształceń, trenowany model nigdy nie będzie przetwarzać dwukrotnie tego samego obrazu.

4.2.7. Diagnostyka modelu na podstawie krzywych uczenia i walidacji

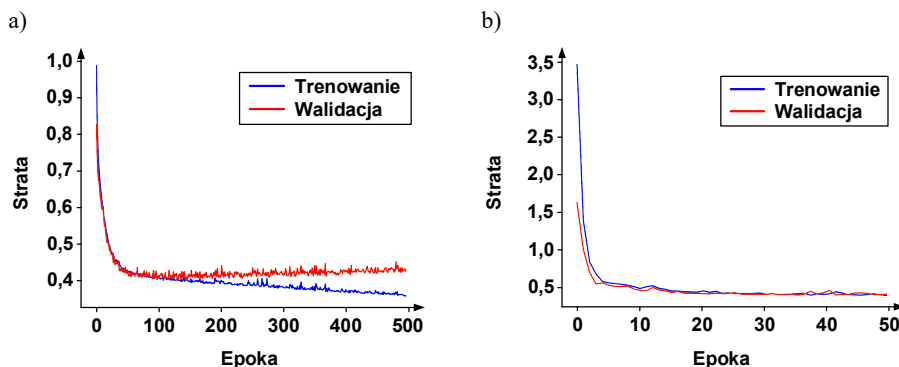
Na podstawie wykresu krzywych uczenia i walidacji można wnioskować o występowaniu jednego z trzech stanów modelu, w stosunku do danych uczących, tj. niedotrenowania, przetrenowania oraz dopasowania. **Niedotrenowanie** występuje wówczas, gdy model nie jest w stanie uzyskać wystarczająco niskiej wartości błędu dla zbioru treningowego. Można ten stan zidentyfikować wyłącznie na podstawie wykresu krzywej straty trenowania. Taki wykres może mieć postać linii płaskiej lub szumu o dużej amplitudzie zmian. Oznacza to wówczas, że model nie jest w stanie nauczyć się reprezentacji danych występujących w zbiorze treningowym. Taka sytuacja występuje, gdy model nie ma odpowiedniej pojemności, w stosunku do złożoności zbioru danych (rys. 4.9a). Niedotrenowanie występuje również wówczas, gdy krzywa straty trenowania cały czas wykazuje tendencję opadającą, także pod koniec wykresu (rys. 4.9b). Taka sytuacja wskazuje, że model jest zdolny do dalszego uczenia się i ewentualnej poprawy wydajności, a proces trenowania zatrzymano przedwcześnie.



Rys. 4.9. Przykład niedotrenowania: a) model nie ma wystarczającej pojemności; b) model wymaga dalszego trenowania

Przetrenowanie występuje wówczas, gdy model jest za bardzo dopasowany do danych uczących, w tym także do szumu statystycznego oraz fluktuacji losowych, które w nich występują. Problem polega na tym, że im bardziej model jest wyspecjalizowany w odniesieniu do danych uczących, tym mniejsza jest jego zdolność do generalizacji nowych danych. Powoduje to wzrost błędu generalizacji, który można mierzyć na podstawie wydajności modelu dla zbioru walidacyjnego. Taka sytuacja może wystąpić, gdy model ma pojemność (a tym samym elastyczność) większą, niż jest to wymagane dla danego problemu. Inną przyczyną przetrenowania może być zbyt długie uczenie modelu. Przetrenowanie można stwierdzić wówczas, gdy krzywa straty trenowania cały czas wykazuje tendencję opadającą, a krzywa straty walidacji opada do pewnego momentu,

a następnie zaczyna narastać (rys. 4.10a). Miejsce przegięcia krzywej straty walidacji jest tym punktem, w którym trenowanie powinno być zatrzymane, ponieważ po jego przekroczeniu pojawiają się oznaki przetrenowania.

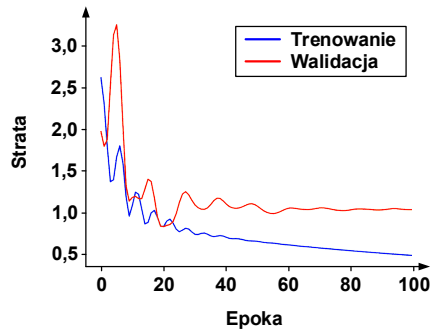


Rys. 4.10. Wykresy krzywej straty dla: a) przetrenowania; b) dopasowania modelu

Dopasowanie jest celem algorytmu uczenia i występuje pomiędzy stanami niedotrenowania i przetrenowania. Można ten stan zidentyfikować na podstawie krzywych straty trenowania i walidacji, które na początku wykazują silną tendencję opadającą. Następnie, na pewnym odcinku ich przebieg jest ustabilizowany, a na końcu wykresu występuje bardzo mała różnica pomiędzy wartościami straty trenowania i walidacji (rys. 4.10b). Strata jest prawie zawsze mniejsza dla zbioru treningowego, niż walidacyjnego. Oznacza to, że należy spodziewać się pewnego odstępu pomiędzy przebiegiem krzywych straty trenowania i walidacji. Ten odstęp jest często określany jako „luka generalizacyjna”.

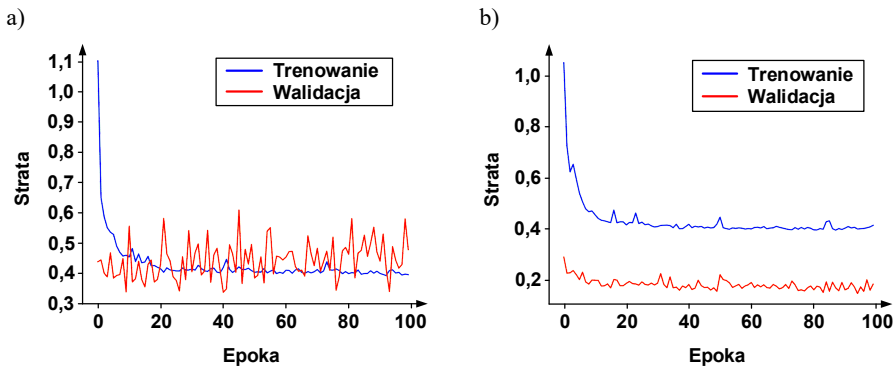
Oprócz możliwości wykrycia stanów niedotrenowania, przetrenowania i dopasowania, wykresy krzywych trenowania i walidacji pozwalają także wnioskować o tym, czy zbiory danych są niereprezentatywne. **Zbiór niereprezentatywny** to taki, który nie uwzględnia cech statystycznych innego zbioru, pochodzącego z tej samej dziedziny problemu. Taka sytuacja może wystąpić, gdy liczba próbek zbioru danych jest zbyt mała w stosunku do liczebności innego zbioru. Niereprezentatywność zbioru danych może mieć miejsce zarówno w odniesieniu do zbioru treningowego, jak i walidacyjnego.

Zbiór treningowy jest niereprezentatywny wówczas, gdy nie dostarcza on dostatecznej ilości informacji potrzebnej do nauki modelu, w porównaniu ze zbiorem walidacyjnym. Może to mieć miejsce wówczas, gdy dane treningowe zawierają zbyt mało przykładów, w porównaniu z danymi walidacyjnymi. W takiej sytuacji, krzywe straty trenowania i walidacji wykazują tendencję opadającą, ale występuje pomiędzy nimi bardzo duży odstęp (rys. 4.11).



Rys. 4.11. Przykład sytuacji, gdy zbiór treningowy jest niereprezentatywny (zbiór treningowy jest zbyt mały, w porównaniu ze zbiorem walidacyjnym)

Zbiór walidacyjny jest niereprezentatywny, gdy nie dostarcza dostatecznej ilości informacji potrzebnej do oceny zdolności generalizacyjnej modelu. Sytuacja taka może wystąpić, gdy zbiór walidacyjny ma zbyt mało przykładów, w porównaniu ze zbiorem treningowym. Krzywa straty trenowania ma wówczas przebieg podobny do przypadku dopasowania, natomiast krzywa straty walidacji ma kształt szumu oscylującego wokół krzywej straty trenowania (rys. 4.12a). Sytuacja, gdy zbiór walidacyjny jest niereprezentatywny ma miejsce także wówczas, gdy zbiór danych walidacyjnych jest łatwiejszy do predykcji przez model, niż zbiór danych treningowych. W tym wypadku, krzywa straty walidacji występuje na wykresie znacznie niżej, w porównaniu z krzywą straty trenowania (rys. 4.12b).



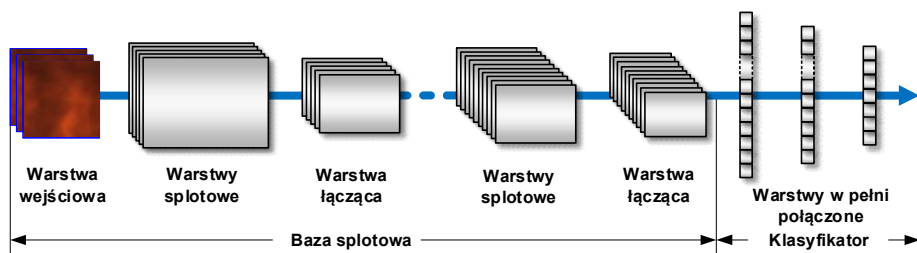
Rys. 4.12. Przykład sytuacji, gdy zbiór walidacyjny jest niereprezentatywny: a) zbiór walidacyjny jest zbyt mały, w porównaniu ze zbiorem treningowym; b) zbiór walidacyjny jest łatwiejszy do predykcji, niż zbiór treningowy

4.3. Sieci splotowe

4.3.1. Architektura sieci splotowej

Model uczenia głębokiego jest w istocie skierowanym acyklicznym grafem warstw. Jego najpopularniejszą formą jest sekwencyjny stos warstw przypisujący jeden element wyjściowy do jednego elementu wejściowego. Bardziej zaawansowane topologie wykorzystują sieci o dwóch rozgałęzieniach, sieci typu *multihead* oraz bloki iniepcji. Obszerny przegląd takich rozwiązań można znaleźć w [122]. Opracowano pewne ogólne zasady doboru architektury do charakteru rozwiązywanego problemu, ale w praktyce, wybór ten zależy bardziej od doświadczenia osoby prowadzącej badania.

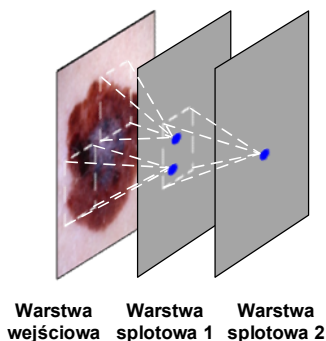
Na rysunku 4.13 przedstawiono typową, sekwencyjną architekturę sieci splotowej, przeznaczoną do klasyfikacji obrazów. Składa się ona zazwyczaj z kilku warstw splotowych (najczęściej z funkcją aktywacji ReLU), warstwy łączącej (ang. *Pooling*) po niej znowu kilku warstw splotowych, kolejnej warstwy łączącej, itd. Obraz stopniowo maleje, przechodząc przez kolejne warstwy sieci, ale jednocześnie zwiększa się jego głębokość. Na końcu sieci znajduje się klasyczna sieć neuronowa, zawierająca kilka warstw w pełni połączonych. Ostatnia z tych warstw oblicza prognozy. Może to być, np. warstwa *softmax* oszacowująca prawdopodobieństwo przynależności próbki do danej klasy.



Rys. 4.13. Typowa architektura splotowej sieci neuronowej

Sieci splotowe stosowane do klasyfikacji obrazów składają się z dwóch części. Część pierwsza – baza modelu, składa się z serii warstw splotowych oraz łączących. Warstwy splotowe są zbudowane w taki sposób, że neurony w pierwszej warstwie nie są połączone z każdym pikselem obrazu wejściowego, lecz wyłącznie z pikselami znajdującymi się w ich polu recepcyjnym. Z kolei, każdy neuron w drugiej warstwie łączy się wyłącznie z neuronami znajdującymi się w niewielkim obszarze pierwszej warstwy (rys. 4.14). Dzięki temu, warstwy splotowe uczą się wzorców lokalnych, odnajdywanych właśnie w ich polach recepcyjnych (np. o rozmiarze 3×3). W efekcie, sieć może koncentrować się na ogólnych cechach w pierwszej warstwie ukrytej, następnie łączyć je w bardziej

złożone kształty w drugiej warstwie ukrytej, itd. Reprezentacje danych wyuczone przez bazę splotową są bardziej ogólne, z poziomem ogólności zależnym od głębokości danej warstwy. Początkowe warstwy wyłaniają ogólne cechy (charakteryzujące, np. krawędzie, kolory, tekstury), a warstwy dalsze wyłaniają cechy bardziej abstrakcyjne (dotyczące, np. kształtów).



Rys. 4.14. Warstwy splotowe z lokalnymi polami recepcyjnymi

Warstwy łączące realizują podpróbkowanie map cech, w celu zmniejszenia obciążenia obliczeniowego, wykorzystania pamięci oraz liczby parametrów (ograniczając tym samym ryzyko przetrenowania). Nie zawierają one żadnych wag, ich zadaniem jest tylko gromadzenie danych wejściowych za pomocą wybranej funkcji agregacyjnej. Podpróbkowanie jest najczęściej realizowane z wykorzystaniem okien ekstrakcji przetwarzających mapy cech o wymiarach 2×2 , przy kroku równym 2 (ma to na celu dwukrotne zredukowanie mapy cech). Okna te zwracają wartości maksymalne lub średnie każdego z kanałów.

Druga część sieci splotowej, znajdująca się na jej końcu, to klasyfikator zbudowany z warstw w pełni połączonych. Warstwy te uczą się cech parametrów globalnych w swoich przestrzeniach wejściowych (są to wzorce związane ze wszystkimi pikselami obrazu). Reprezentacje danych wyuczone przez klasyfikator są specyficzne dla zbioru klas, na których model został wytrenowany – zawierają one jedynie informacje pozwalające określić prawdopodobieństwo, że dana klasa jest widoczna na obrazie.

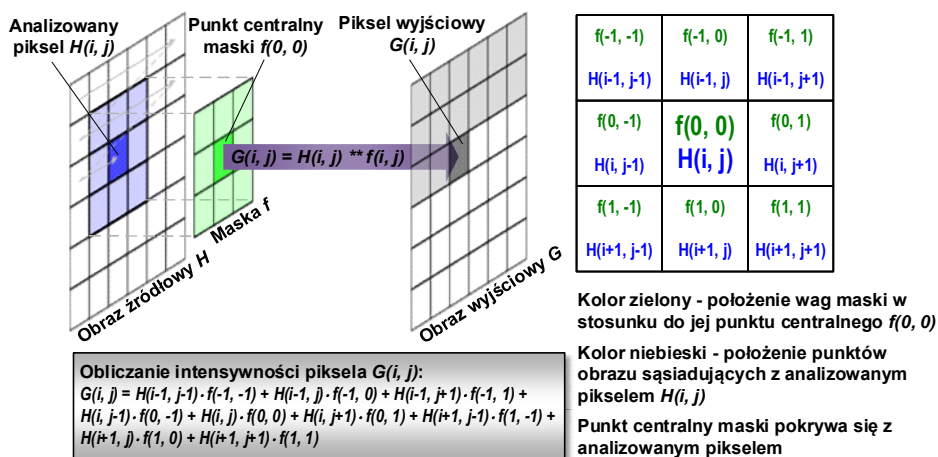
Podsumowując, każda warstwa modelu działa jak filtr, na wyjściu którego tworzona jest reprezentacja danych. Dzięki takiej architekturze, głębokie splotowe sieci neuronowe mogą rozwiązywać zadania klasyfikacji na podstawie surowych danych uzyskanych z obrazów wejściowych. W porównaniu z tradycyjnymi algorytmami uczenia maszynowego, zadanie klasyfikacji może być uproszczone, ponieważ zostają wyeliminowane etapy estymacji cech i wstępnego przetwarzania danych.

4.3.2. Splot dwuwymiarowy

Operacja splotu realizuje filtrację liniową polegającą na tym, że intensywność piksela obrazu wyjściowego jest liniową kombinacją intensywności pikseli z jego otoczenia w obrazie źródłowym. Operacja tego rodzaju jest zdefiniowana następująco:

$$G(i, j) = H(i, j) ** f(i, j), \quad (4.19)$$

gdzie: $G(i, j)$ – obraz wyjściowy, $H(i, j)$ – obraz źródłowy, $f(i, j)$ – funkcja definiująca maskę (filtr). Operator $**$ oznacza dwuwymiarowy splot (konwolucję). Na rysunku 4.15 przedstawiono zasadę wyznaczania intensywności piksela obrazu wyjściowego $G(i, j)$, w wyniku splotu maski f o rozmiarze 3×3 i obrazu źródłowego H . Podczas tego procesu, obliczana jest suma ważona intensywności pikseli znajdujących się w obszarze maski. Istotą maski jest właśnie zdefiniowanie zbioru wag dla każdej pozycji w jej granicach. Operację splotu można sobie wyobrazić w ten sposób, że nakładamy maskę na obraz tak, że jej punkt centralny pokrywa się z pikselem znajdującym się w lewym górnym rogu i obliczamy sumę ważoną. Następnie, przesuwamy maskę o pewien krok (równy, np. 1 lub 2) w wierszach od lewej strony do prawej, w kolumnach od góry do dołu i wykonujemy obliczenia dla każdej pozycji maski.



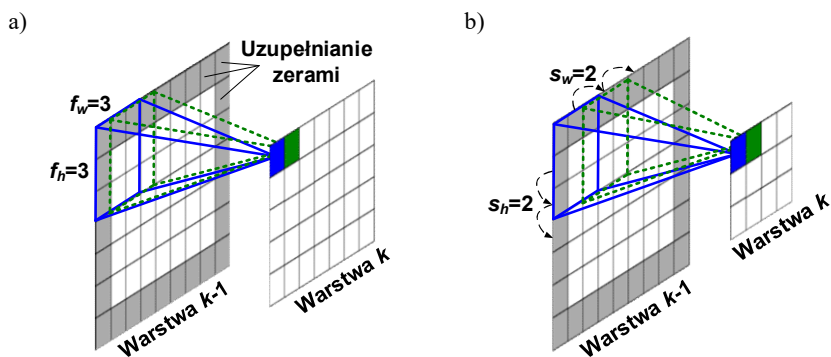
Rys. 4.15. Splot obrazu źródłowego z maską o rozmiarze 3×3 . Elementy każdego iloczynu cząstkowego to waga danej pozycji w masce oraz intensywność piksela źródłowego o tych samych współrzędnych, co wspomniana pozycja maski

Podczas wykonywania operacji splotu występują, tzw. efekty graniczne, związane z brzegiem obrazu. W zależności od zastosowanej metody, może ulec zmianie rozmiar obrazu wyjściowego. Jedną z metod polega na tym, że brakują-

ce piksele obrazu źródłowego, niezbędne do obliczenia intensywności pikseli obrazu wyjściowego, są powielane spośród pikseli znajdujących się w sąsiedztwie analizowanego piksela. Inna metoda jest taka, że nie są uwzględniane piksele znajdujące się na brzegu obrazu. W wyniku tego, zmniejsza się rozmiar obrazu, np. o 2 wiersze i 2 kolumny, w przypadku maski o rozmiarach 3×3 . Jest to dość istotne nie tylko podczas przetwarzania obrazów o bardzo małych rozmiarach, ale także w przypadku sekwencyjnego wykonywania kilku iteracji, ponieważ obraz wyjściowy może mieć znacząco mniejszy rozmiar od obrazu źródłowego. Kolejna metoda stosuje operację dodawania zer wokół obrazu źródłowego. Metoda ta jest wykorzystywana przez sieci spłotowe, które będą omówione w kolejnych rozdziałach. Operacja dodawania zer wokół warstwy występującej wcześniej pozwala zachować takie same rozmiary warstw, które w istocie nie przetwarzają obrazów (poza 1-szą warstwą) lecz mapy cech.

4.3.3. Filtry i mapy cech

Na rysunku 4.16a przedstawiono sposób połączenia neuronów znajdujących się w sąsiednich warstwach spłotowych. Załóżmy, że f_h i f_w oznaczają, odpowiednio, wysokość i szerokość pola recepcyjnego. Neuron, który znajduje się w wierszu i oraz kolumnie j następnej warstwy jest połączony z wyjściami neuronów warstwy poprzedniej, znajdującymi się w wierszach od i do $i+f_h-1$ i kolumnach od j do $j+f_w-1$. Aby zachować takie same rozmiary warstw, najczęściej stosuje się operację dodawania zer wokół warstwy występującej wcześniej.

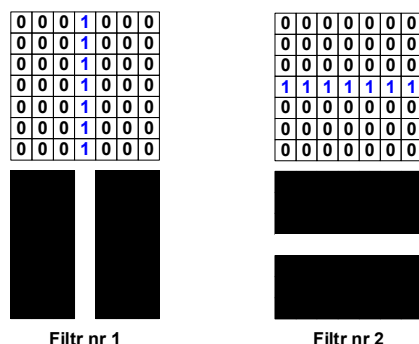


Rys. 4.16. Sposób połączenia neuronów w sąsiednich warstwach spłotowych: a) parametr *krok* równy 1; b) parametr *krok* równy 2

Na rysunku 4.16a, odległość pomiędzy kolejnymi polami recepcyjnymi (*krok*) jest równa 1. Zwiększenie tego parametru pozwala łączyć ze sobą kolejne warstwy, które mogą się znacznie różnić rozmiarami. Załóżmy, że wartość kroku pionowego (s_h) i poziomego (s_w) jest równa 2 (rys. 4.16b). Wówczas, neuron

znajdujący się wierszu i oraz kolumnie j warstwy następnej jest połączony z wyjściami neuronów warstwy poprzedniej znajdującymi się w wierszach od $i \times s_h$ do $i \times s_h + f_h - 1$ i w kolumnach od $j \times s_w$ do $j \times s_w + f_w - 1$.

Ważnym pojęciem, związanym z działaniem warstwy spłotowej, jest filtr, zwany inaczej jądrem spłotowym. **Filtr** jest utworzony przez zbiór wag neuronu. Można go także przedstawić w postaci małego obrazu o rozmiarze pola recepcyjnego. Załóżmy, że zdefiniowane są filtry, których budowę przedstawiono na rysunku 4.17.

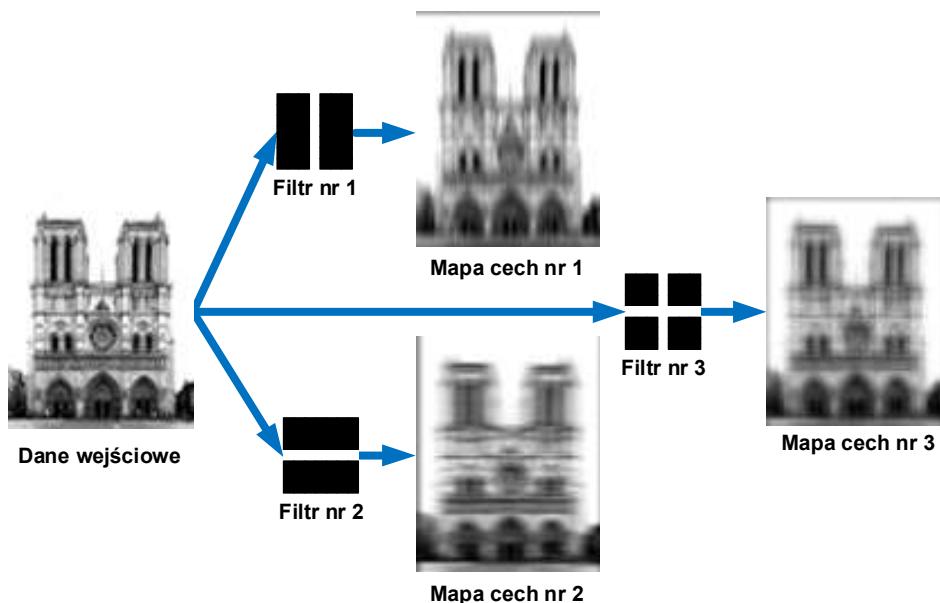


Rys. 4.17. Przykładowe filtry. W górnym wierszu przedstawiono reprezentację w pamięci poszczególnych filtrów, natomiast w dolnym – odpowiadające im obrazy

Zgodnie z działaniem spłotu, neurony ignorują te dane wejściowe, którym w filtrze odpowiada 0. W efekcie, są one słabiej widoczne na obrazie wyjściowym. Natomiast, uwzględniane są dane, którym w filtrze odpowiada 1. Dzięki temu, na obrazie wyjściowym są one wyraźniejsze. W praktyce, wszystkie neurony w danej warstwie wykorzystują ten sam filtr. Dlatego, na całym obrazie wyjściowym uwypuklone zostają te elementy, które są specyficzne dla danego filtra. Efekt działania filtrów z rysunku 4.17, w odniesieniu do przykładowego obrazu, pokazano na rysunku 4.18. Po zastosowaniu filtra nr 1, na obrazie wyjściowym wyraźniejsze stają się linie pionowe. Natomiast, dla filtra nr 2, uwypuklone są linie poziome.

W tym momencie, możemy wprowadzić kluczowe pojęcie tego rozdziału, a mianowicie **mapę cech**. Jest to warstwa spłotowa z neuronami wykorzystującymi ten sam filtr. Ważną rolą mapy cech jest to, że pozwala nam dostrzec w obrazie wejściowym te elementy, które najbardziej przypominają zastosowany filtr. Pokazuje ona występowanie filtrowanego wzorca w różnych miejscach danych wejściowych. Jedna warstwa spłotowa ma wiele takich map, a każda z nich wykrywa inną cechę. Sieć spłotowa, podczas uczenia, wyszukuje takie filtry (dobiera wagi), które są najbardziej przydatne do wykonania danego zadania. Następnie, dalsze warstwy sieci (w strukturze hierarchicznej) uczą się łą-

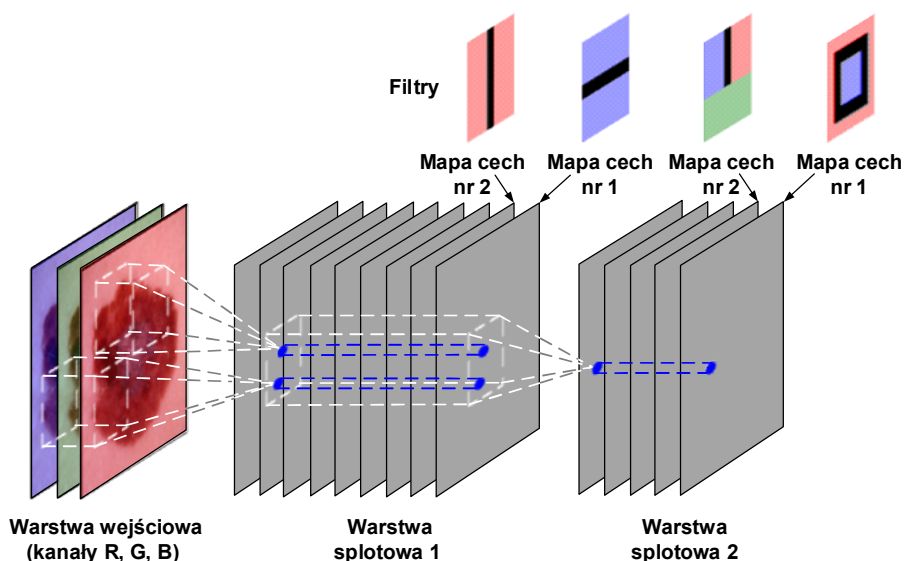
czyć te filtry w bardziej złożone wzorce, którymi mogą być, np. krzyże tam, gdzie aktywne są filtry nr 1 i 2 (np. filtr nr 3 na rys. 4.18).



Rys. 4.18. Efekt działania filtrów z rys. 4.17 w odniesieniu do przykładowego obrazu

Weźmiemy jako przykład, sieć wykrywającą twarze. Pierwsza warstwa splotowa może uczyć się wykrywać proste cechy, np. kąty ostre, przecięcia się linii, itp. Kolejna warstwa, na podstawie prostych cech z poprzedniej warstwy, może wykrywać bardziej skomplikowane przedmioty, np. oczy, uszy, nos. Dalsza warstwa splotowa, na podstawie wcześniej wykrytych części twarzy, może wykonywać detekcję całych twarzy. Taki sposób wyłaniania cech przez hierarchiczną strukturę warstw splotowych, od prostych do coraz bardziej abstrakcyjnych, nosi nazwę **ekstrakcji cech**. Należy w tym miejscu zauważyć, iż można wizualizować wartości wag z niższych warstw sieci na dwuwymiarowych obrazach i w ten sposób przeglądać, jakie cechy są wykrywane. Dobrym przykładem są filtry z rysunku 4.18. Z kolei, bardzo abstrakcyjne cechy (z dalszych warstw) są znacznie trudniejsze do interpretacji przez człowieka.

W dotychczasowym opisie, w celu zapewnienia większej czytelności omawianych treści, warstwy splotowe przedstawiane były w uproszczeniu, w postaci pojedynczych dwuwymiarowych warstw. W rzeczywistości, warstwa splotowa jest strukturą trójwymiarową, którą tworzy pewna liczba map cech o takich samych rozmiarach (rys. 4.19).



Rys. 4.19. Rzeczywista budowa sieci splotowej, w której poszczególne warstwy mają strukturę trójwymiarową

Wszystkie neurony znajdujące się w danej mapie cech mają ustawione takie same parametry, tzn. wagi (także wagę obciążenia). Z kolei, neurony znajdujące się w innej mapie cech mają ustawione inne parametry. Jeżeli weźmiemy pod uwagę neuron znajdujący się w określonej warstwie, to jego pole recepcyjne nie zmienia się. Pokrywa ono wszystkie mapy cech warstw poprzednich (rys. 4.19). Stwierdziłismy wcześniej, że warstwa splotowa zbudowana jest z wielu map cech, tym samym stosuje ona wiele różnych filtrów na wejściach. Dlatego, jest w stanie wykrywać wiele różnych cech, w różnych miejscach obrazu. Istotne znaczenie ma fakt, że wszystkie neurony w jednej mapie cech mają ustawione takie same parametry. Dzięki temu, gdy sieć splotowa nauczy się rozpoznawać określony wzorek w jednym miejscu na obrazie, będzie w stanie go rozpoznawać także w innych miejscach.

Także warstwę wejściową może w rzeczywistości tworzyć więcej warstw składowych, niż jedna. W przypadku obrazu kolorowego, występują 3 kanały barw – czerwony, zielony i niebieski. Wówczas, warstwa wejściowa będzie zbudowana z trzech warstw, a każda z nich będzie reprezentować inny kanał barw. Jak można się domyśleć, gdy obrazem wejściowym będzie obraz monochromatyczny, wystąpi tylko jedna warstwa na wejściu, reprezentująca luminancję obrazu.

Biorąc pod uwagę rzeczywistą, trójwymiarową strukturę warstw splotowych, należy podkreślić, że wszystkie neurony, które znajdują się w tym samym wierszu i oraz kolumnie j , ale w innych mapach cech są połączone z wyjściami do-

kładnie tych samych neuronów warstwy poprzedniej. W związku z tym, jeżeli rozpatrzmy neuron znajdujący się w wierszu i oraz kolumnie j mapy cech k w warstwie splotowej l , to jest on połączony z neuronami warstwy wcześniejszej $l-1$ znajdującymi się wierszach od $i \times s_h$ do $i \times s_h + f_h - 1$ i kolumnach od $j \times s_w$ do $j \times s_w + f_w - 1$ we wszystkich mapach cech tej warstwy. Uwzględniając powyższe, sformułujmy zależność na sumę ważoną wszystkich danych wejściowych wraz z neuronem obciążenia (wartość wyjściowa neuronu w warstwie splotowej):

$$z_{i,j,k} = b_k \sum_{u=0}^{f_h-1} \sum_{v=0}^{f_w-1} \sum_{k'=0}^{f_n-1} x_{i',j',k'} \cdot w_{u,v,k',k} \quad \text{dla} \quad \begin{cases} i' = i \times s_h + u \\ j' = j \times s_w + v \end{cases}, \quad (4.20)$$

gdzie:

- $z_{i,j,k}$ – wyjście neuronu znajdującego się w wierszu i , kolumnie j i mapie cech k warstwy splotowej l ;
- b_k – neuron obciążenia dla mapy cech k w warstwie l ;
- f_h, f_w – wysokość i szerokość pola recepcyjnego;
- f_n – liczba map cech w warstwie poprzedniej $l-1$;
- $x_{i',j',k'}$ – wyjście neuronu znajdującego się w warstwie $l-1$, wierszu i' , kolumnie j' , mapie cech k' (lub kanale k' , jeżeli poprzednia warstwa była warstwą wejściową);
- $w_{u,v,k',k}$ – waga połączenia pomiędzy dowolnym neuronem w mapie cech k warstwy l , a jego wejściem mieszczącym się w wierszu u , kolumnie v (względem pola recepcyjnego neuronu) w mapie cech k' ;
- s_h, s_w – kroki pionowy i poziomy.

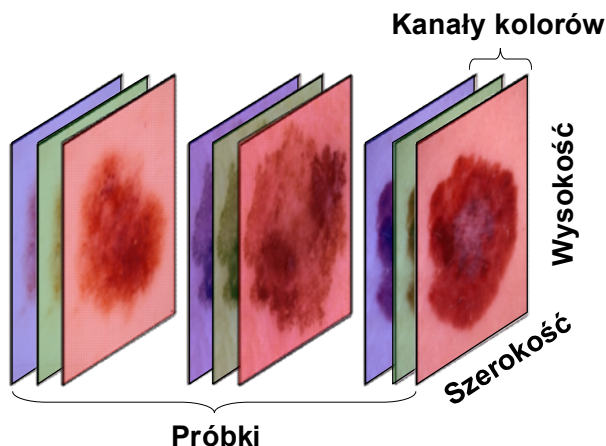
4.3.4. Struktury danych przetwarzane przez sieć

We wszystkich współczesnych systemach uczenia maszynowego, podstawową strukturą danych są **tensor**y. Są one uogólnieniem macierzy w przestrzeniach o dowolnej liczbie wymiarów. Praktycznie, zawsze są to struktury przeznaczone do przechowywania liczb. W przypadku obrazów, dane przechowywane są w tensorach czterowymiarowych o kształcie (*próbki, wysokość, szerokość, kanały*)⁶ (rys. 4.20). Dla obrazów RGB, rozmiar osi kanałów (głębina) wynosi 3 (kolor czerwony, zielony i niebieski), a dla obrazów monochromatycznych jest on równy 1 (luminancja). W przypadku warstw splotowych, głębina jest parametrem danej warstwy i określa liczbę zastosowanych filtrów (liczbę map cech).

Modele uczenia głębokiego nie przetwarzają naraz całego zbioru danych, jest on dzielony na małe porcje, tzw. **wsady**. Wielkość wsadu, to najczęściej wielokrotność liczby 8. W przypadku tensora wsadowego, pierwsza oś nazywana jest osią wsadu lub wymiarem wsadu. Na przykład, wsad 64 obrazów kolorowych o rozdzielczości 100×100 pikseli można zapisać w tensorze o kształcie (64, 100,

⁶ W przypadku biblioteki TensorFlow, wymiar *kanaly* występuje na końcu notacji, natomiast w bibliotece Theano na jej początku.

100, 3), a wsad 32 obrazów monochromatycznych o tej samej rozdzielczości można zapisać w tensorze o kształcie (32, 100, 100, 1).



Rys. 4.20. Czwierowymiarowy tensor danych obrazu kolorowego

Podczas budowy modelu, musi być zapewniona kompatybilność łączonych ze sobą warstw. Wynika to z faktu, że każdy rodzaj warstwy przyjmuje tensory wejściowe o określonym kształcie i generuje tensory wyjściowe mające także określony kształt. Na przykład, jeżeli utworzymy warstwę zwracającą tensor, którego pierwszy wymiar ma długość 32, to może ona być połączona z kolejną warstwą, która oczekuje⁷ na wejściu wektora o długości 32.

Przeanalizujmy fragment kodu w języku Python, odpowiedzialnego za budowę prostej sieci splotowej (listing 1). W wierszach nr 1 i 2 zaimportowano niezbędne klasy biblioteki Keras. W wierszu nr 3 utworzono model sekwencyjny, który składa się ze stosu warstw splotowych (`Conv2D`) i łączących (`MaxPooling2D`). Poszczególne warstwy są dodawane do modelu za pomocą metody `add()`. Sieć przyjmuje tensor wejściowy o kształcie określonym przez wysokość, szerokość i kanały koloru (dla uproszczenia zapisu nie wliczamy tutaj wymiaru próbki). W związku z tym, przetwarzane są obiekty wejściowe (obrazy) o rozmiarze (40, 40, 3). Dlatego też, do pierwszej warstwy sieci przekazujemy argument `input_shape=(40, 40, 3)` w wierszu nr 5. Pierwszy argument przekazany konstruktorowi klasy `Conv2D`, równy 32 – oznacza liczbę kanałów (filtrów). Argument drugi to krotka (3, 3), definiująca rozmiar pola recepcyjnego dla operacji splotu (3×3). Na trzeciej pozycji występuje nazwa funkcji aktywacji. W naszym przykładzie, wszystkie warstwy (oprócz ostatniej) wyko-

⁷ W pakiecie Keras, warstwy tworzące model są budowane dynamicznie, w celu dopasowania ich do warstwy poprzedniej.

rzystując funkcję ReLU, która jest najczęściej stosowana. Parametr ostatni (`input_shape`) definiuje rozmiar obrazu wejściowego. Stosujemy obrazy kolorowe (3 kanały koloru) o rozmiarze 40×40 pikseli (wysokość, szerokość). Jeżeli tensorem wejściowym jest mapa cech, a nie obraz, argument `input_shape` jest pomijany.

Listing 1. Kod w języku Python odpowiedzialny za utworzenie prostej sieci splotowej

```
1  from keras import layers
2  from keras import models
3  model = models.Sequential()
4  model.add(layers.Conv2D(32, (3, 3), activation='relu', \
5      input_shape=(40, 40, 3)))
6  model.add(layers.MaxPooling2D((2, 2)))
7  model.add(layers.Conv2D(64, (3, 3), activation='relu'))
8  model.add(layers.MaxPooling2D((2, 2)))
9  model.add(layers.Conv2D(64, (3, 3), activation='relu'))
10 model.add(layers.Flatten())
11 model.add(layers.Dense(64, activation='relu'))
12 model.add(layers.Dense(1, activation='sigmoid'))
13 model.summary()
```

Przyjrzyjmy się architekturze sieci wyświetlonej za pomocą metody `summary()` w wierszu nr 13 (rys. 4.21). Na wyjściu pierwszej warstwy splotowej pojawia się tensor o kształcie (38, 38, 32). Jest on mniejszy o 2 pola wzdłuż każdego z wymiarów, względem obrazu wejściowego. Przyczyną tego jest występowanie efektów granicznych, o których wspomniano w rozdziale 4.3.2. Gdybyśmy chcieli uzyskać tensor wyjściowy o takim samym rozmiarze, jak obraz wejściowy, należałoby skorzystać z techniki dopełniania, polegającej na dodaniu odpowiedniej liczby wierszy i kolumn po każdej stronie obrazu wejściowego. W przypadku warstw `Conv2D`, dopełnianie konfiguruje się za pomocą argumentu `padding` (jego wartość domyślna wyłącza dopełnianie).

W wierszu nr 6 dodano warstwę łączącą, której zadaniem jest zmniejszenie o połowę rozmiaru wejściowej mapy cech. Na podstawie architektury sieci (rys. 4.21) widać, iż przed przejściem przez pierwszą warstwę `MaxPooling2D`, mapa cech miała rozmiar 38×38, a operacja *max-pooling* zmniejszyła ten rozmiar do 19×19. Operacja *max-pooling* działa w ten sposób, że w oknach o rozmiarze 2×2 (przy kroku równym 2) zwracane są wartości maksymalne każdego z kanałów. W identyczny sposób działa druga warstwa łącząca, którą dodano w wierszu nr 8.

Druga warstwa splotowa została dołączona do modelu w wierszu nr 7. Na swoim wyjściu tworzy ona tensor o kształcie (17, 17, 64). Zwróćmy uwagę, że wskutek efektów granicznych, nastąpiło ponowne zmniejszenie rozmiaru wyj-

ściowej mapy cech (17×17), w stosunku do wejściowej (19×19). Druga warstwa `MaxPooling2D` (dodana w wierszu nr 8) zredukowała rozmiar mapy cech z (17, 17, 64) do (8, 8, 64).

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 38, 38, 32)	896
max_pooling2d_1 (MaxPooling2D)	(None, 19, 19, 32)	0
conv2d_2 (Conv2D)	(None, 17, 17, 64)	18496
max_pooling2d_2 (MaxPooling2D)	(None, 8, 8, 64)	0
conv2d_3 (Conv2D)	(None, 6, 6, 64)	36928
flatten_1 (Flatten)	(None, 2304)	0
dense_1 (Dense)	(None, 64)	147520
dense_2 (Dense)	(None, 1)	65
Total params: 203,970		
Trainable params: 203,970		
Non-trainable params: 0		

Rys. 4.21. Architektura sieci utworzonej za pomocą kodu z listingu nr 1

Ostatnią warstwę spłotową dołączono w wierszu nr 9. Na jej wyjściu tworzo-
ny jest tensor o kształcie (6, 6, 64). Jest to tensor wyjściowy, który zostaje prze-
kazany do klasyfikatora, w postaci warstw w pełni połączonych (stos warstw
`Dense`). Klasyfikator tego typu przetwarza wektory (jeden wymiar), natomiast
tensor wyjściowy jest trójwymiarowy. W związku z tym, dane trójwymiarowe
muszą zostać spłaszczone do jednowymiarowych. Operacja ta jest realizowana
za pomocą warstwy `Flatten`, dołączonej do modelu w wierszu nr 10. Warstwa
ta generuje wektor zawierający 2 304 elementy (6×6×64).

W wierszach nr 11 i 12 dodano 2 warstwy w pełni połączone, tworzące kla-
syfikator. W tym przypadku, realizujemy zadanie klasyfikacji binarnej, mającej
na celu dokonanie przyporządkowania obrazu wejściowego do jednej dwóch
klas. W związku z tym, w ostatniej warstwie wykorzystujemy jeden neuron
z logistyczną funkcją aktywacji (*sigmoid*), która szacuje prawdopodobieństwo
przynależności próbki do klasy pozytywnej. Jeżeli wykonywane byłoby zadanie

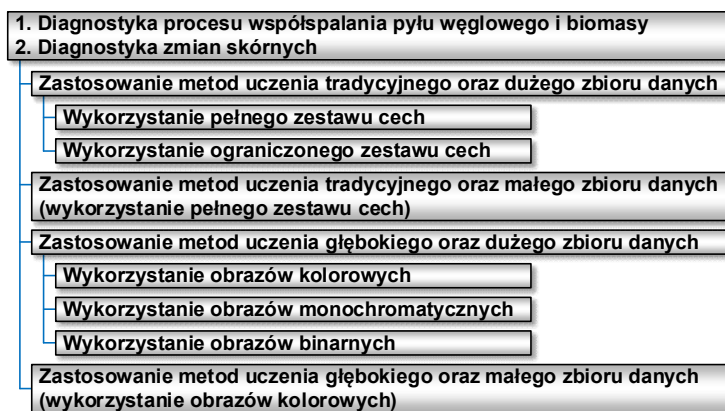
klasyfikacji wieloklasowej, ostatnia warstwa zawierałaby liczbę neuronów równą liczbie klas oraz funkcję *softmax*, szacującą prawdopodobieństwo wystąpienia każdej klasy.

Zwróćmy uwagę na pewien aspekt dotyczący kształtu tensorów przetwarzanych przez sieć. Na wyjściu każdej warstwy Conv2D i MaxPooling2D pojawia się tensor o kształcie (wysokość, szerokość, kanały). Wraz z zagłębianiem się w sieć, wysokość i szerokość zmniejszają się dwukrotnie, przechodząc przez kolejne warstwy. Natomiast zwiększa się dwukrotnie liczba kanałów (filtrów). W pierwszej warstwie splotowej (wiersz nr 4) zastosowano 32 kanały, a w warstwach następnych liczba ta wynosi 64 (wiersze 7 i 9). W bardziej złożonych sieciach, np. takich, które zastosowano w badaniach zaprezentowanych w dalszych rozdziałach, liczba kanałów może dochodzić nawet do 512. Typowym podejściem jest podwajanie liczby kanałów zastosowanych w pierwszej warstwie splotowej, wynoszącej przeważnie 32 lub 64.

W ostatniej kolumnie podsumowania architektury sieci zamieszczona jest liczba parametrów (wag) poszczególnych warstw. Widać tam, iż warstwy łączące i spłaszczająca nie mają żadnych parametrów, a posiadają je wyłącznie warstwy splotowe i w pełni połączone. Na przykład, pierwsza warstwa Dense ma 147 520 parametrów (2 304 wejścia po 64 neurony każde oraz 64 wagi obciążenia). Z kolei, warstwa ostatnia ma 65 parametrów (64 wejścia po jednym neuronie każde oraz jedna waga obciążenia).

5. Metodyka badań

Zakres wykonanych badań przedstawiono na rysunku 5.1. Przedmiotem analizy były obrazy płomienia oraz obrazy zmian skórnych. Obrazy płomienia wykorzystano do diagnostyki procesu współspalania pyłu węglowego i biomasy, mającej na celu rozróżnienie dwóch wariantów spalania. Pierwszy odpowiadał warunkom normalnym, a drugi nadmiarowi powietrza. Diagnostyka zmian skórnych miała na celu odróżnienie czerniaków złośliwych skóry od niezłośliwych znamion melanocytowych. W dalszych rozdziałach bardziej szczegółowo scharakteryzowano kontekst obu kierunków badań.

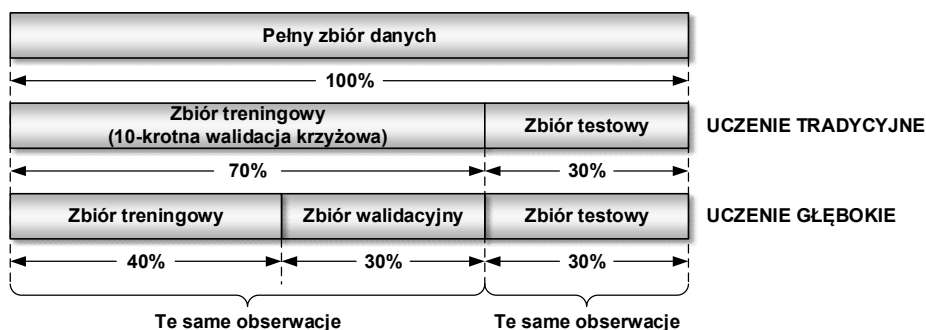


Rys. 5.1. Zakres badań

Scenariusz wykonanych eksperymentów był taki sam dla obrazów płomienia oraz obrazów zmian skórnych (rys. 5.1). Modele budowane były za pomocą metod uczenia tradycyjnego i głębokiego, przy wykorzystaniu zbiorów danych określanych umownie jako **duże** i **małe**. Liczebność obserwacji zbiorów małych była pięciokrotnie mniejsza, niż zbiorów dużych. W uczeniu tradycyjnym zastosowano 2 duże zbiory danych (z pełnym i ograniczonym zestawem cech) oraz 1 mały zbiór (z pełnym zestawem cech). Podczas uczenia głębokiego zastosowano 3 duże zbiory danych (z obrazami kolorowymi, monochromatycznymi i binarnymi) oraz 1 mały zbiór (z obrazami kolorowymi). W sumie, wykonano 14 eksperymentów, a w każdym z nich zastosowano inne zestawy danych.

Sposób przygotowania zbiorów danych zaprezentowano na rysunku 5.2. W przypadku metod uczenia tradycyjnego, pełny zbiór podzielono w ten sposób, że 70% stanowiło część treningową, a 30% testową. Należy dodać, iż do celów walidacyjnych wykorzystana była część zbioru treningowego (zastosowano 10-krotną walidację krzyżową). W przypadku metod uczenia głębokiego, wyko-

rzystany został ten sam (utworzony wcześniej) zbiór testowy. Natomiast, zbiór treningowy został podzielony tak, że 40% obserwacji przeznaczono na zbiór treningowy, a 30% na walidacyjny. Taki sposób przygotowania zestawów danych zapewnił porównywalność uzyskanych wyników. Zarówno podczas uczenia tradycyjnego, jak i głębokiego, do uczenia i testowania modeli wykorzystano te same zbiory obserwacji.



Rys. 5.2. Sposób przygotowania zbiorów danych

5.1. Metody uczenia tradycyjnego

Podczas uczenia tradycyjnego, wykorzystano 9 metod selekcji cech. W nawiasach podano ich akronimy, które są stosowane w dalszych rozdziałach.

Metody filtra:

- jednowymiarowe – metoda Fishera (FISHER) oraz metoda analizy wariancji (ANOVA);
- wielowymiarowe – metoda Relief (RELIEF).

Metody opakowujące:

- sekwencyjny wybór do przodu (SFS);
- sekwencyjny wybór do tyłu (SBS);
- rekurencyjna eliminacja cech wraz z estymatorem regresji logistycznej (RFE).

Metody wbudowane:

- metoda *SelectFromModel* z estymatorem regresji logistycznej (LR);
- metoda *SelectFromModel* z estymatorem *AdaBoost* (ADA);
- metoda *SelectFromModel* z estymatorem *LightGBM* estimator (LGBM).

W procesie uczenia wykorzystano metodę przeszukiwania siatki, która pozwoliła określić optymalne wartości hiperparametrów każdego modelu. Zastosowano do tego celu metodę *GridSearchCV*, która jest dostępna w module *model_selection* biblioteki *scikit-learn*. Ocena modeli dokonywana była na podstawie 10-krotnej walidacji krzyżowej. Każdy model, uznany za optymalny dla

danej liczby cech, zapisywany był w postaci pliku na dysku. Następnie, z dostępnego zbioru, wybierany był model najlepszy, tzn. taki, który zapewniał najwyższą dokładność klasyfikacji przy minimalnej liczbie cech. Wybór najlepszego modelu pozwolił jednocześnie określić optymalny zestaw cech, który został użyty w procesie uczenia. Powyższą procedurę zastosowano dla wszystkich metod filtra i opakowujących. W przypadku metod wbudowanych, trenowano i dostrajano tylko te modele, które były zastosowane w procesie selekcji cech.

Cechy wyłonione przez metody filtra i opakowujące wykorzystane były przez 11 metod uczenia nadzorowanego. Były to: liniowa i kwadratowa analiza dyskryminacyjna (LDA, QDA), naiwny klasyfikator Bayes'a (BAYES), maszyna wektorów nośnych wykorzystująca parametr regularyzacji C (SVM), maszyna wektorów nośnych wykorzystująca parametr ν do sterowania liczbą wektorów nośnych (NuSVM), metoda K -najbliższych sąsiadów (KNN), drzewa decyzyjne (DT) oraz perceptron wielowarstwowy (MLP). Ponadto, zastosowano 3 metody zespołowe: lasy losowe (RF), wzmacnianie gradientowe (GRAD) oraz metodę łączenia modeli *AdaBoost* (ADA). W przypadku cech wybranych za pomocą metod wbudowanych, zastosowano 3 metody uczenia: *AdaBoost*, regresję logistyczną (LR) oraz *LightGBM* (LGBM). Wszystkie zastosowane algorytmy (oprócz *LightGBM*) zaimplementowane są w bibliotece *scikit-learn*.

5.2. Metody uczenia głębokiego

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 [153]. Jeżeli zbiór danych użytych do wytrenowania modelu był wystarczająco duży i ogólny, przestrzenna hierarchia wyuczonych cech może skutecznie pełnić rolę ogólnego modelu do przetwarzania obrazu. 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.

Podczas badań, do budowy modeli wykorzystano komputer z 64-bitowym systemem Windows 10, procesorem Intel Core i5-3470 3.20 GHz i pamięcią RAM 32 GB. Zastosowano platformę TensorFlow 1.13.1 [154], bibliotekę Keras 2.2.4 [155] oraz język programowania Python 3.7.3. Obliczenia wykonywane były z wykorzystaniem karty graficznej (NVIDIA GeForce GTX 1060 3 GB), platformy CUDA 9.0 oraz biblioteki cuDNN 7.0.

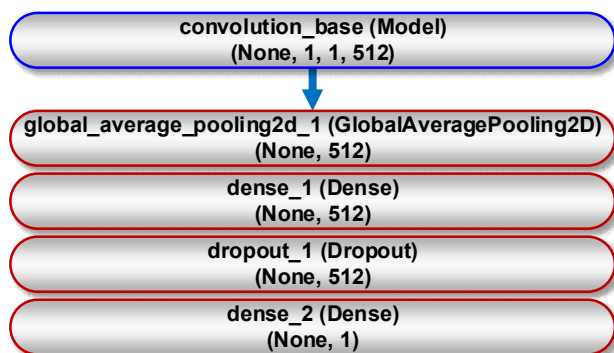
Biblioteka Keras zawiera wiele modeli do przetwarzania obrazów, m.in. Xception [156], VGG16, VGG19 [157], ResNet [158], ResNetV2 [159], InceptionV3 [160], InceptionResNetV2 [161], MobileNet [162], MobileNetV2 [163], DenseNet [164], NASNet [165]. Wszystkie te modele zostały wytrenowane na zbiorze ImageNet zawierającym ok. 1,4 miliona obrazów podzielonych na 1 000 klas [166]. W badaniach wykorzystano 6 modeli charakteryzujących się różną głębokością topologiczną (tabela 5.1).

Tabela 5.1. 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	155	136÷155
5	ResNet50	175	150÷175
6	InceptionResNetV2	780	631÷780

Modele składały się z dwóch części. Pierwszą były bazy splotowe, a druga gęsto połączone 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 dwóch warstw gęsto połączonych. Reprezentacje, których się one wyuczyły były specyficzne dla badanego zestawu klas, na których modele trenowano. Rysunek 5.3 przedstawia ogólny schemat implementacji sieci głębokiej, która składa się z bazy splotowej oraz nowego klasyfikatora binarnego (2 warstwy *Dense*) dodanego na jej koniec.

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. Następnie, dodano 2 warstwy w pełni połączone. Pierwsza warstwa, zawierająca 512 neuronów na wyjściu oraz funkcję aktywacji ReLU, interpretuje cechy wyłonięne przez bazę splotową. Druga warstwa w pełni połączona, z jednym neuronem na wyjściu oraz logistyczną funkcją aktywacji, to warstwa wyjściowa. Jej zadaniem jest prognozowanie przynależności bieżącej obserwacji do klasy pozytywnej. Pomiędzy warstwami w pełni połączonymi znajduje się warstwa porzucania (*Dropout*), ze współczynnikiem porzucania *rate*=0,3. Jej zadaniem jest przeciwdziałanie nadmiernemu dopasowaniu modelu do danych uczących.



Rys. 5.3. 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; *Dropout* – warstwa porzucania (przeciwdziała przetrenowaniu modelu). Mapa cech ma kształt (*próbki, wysokość, szerokość, kanały*). Pierwszy wymiar, równy *None*, oznacza dowolną liczbę próbek

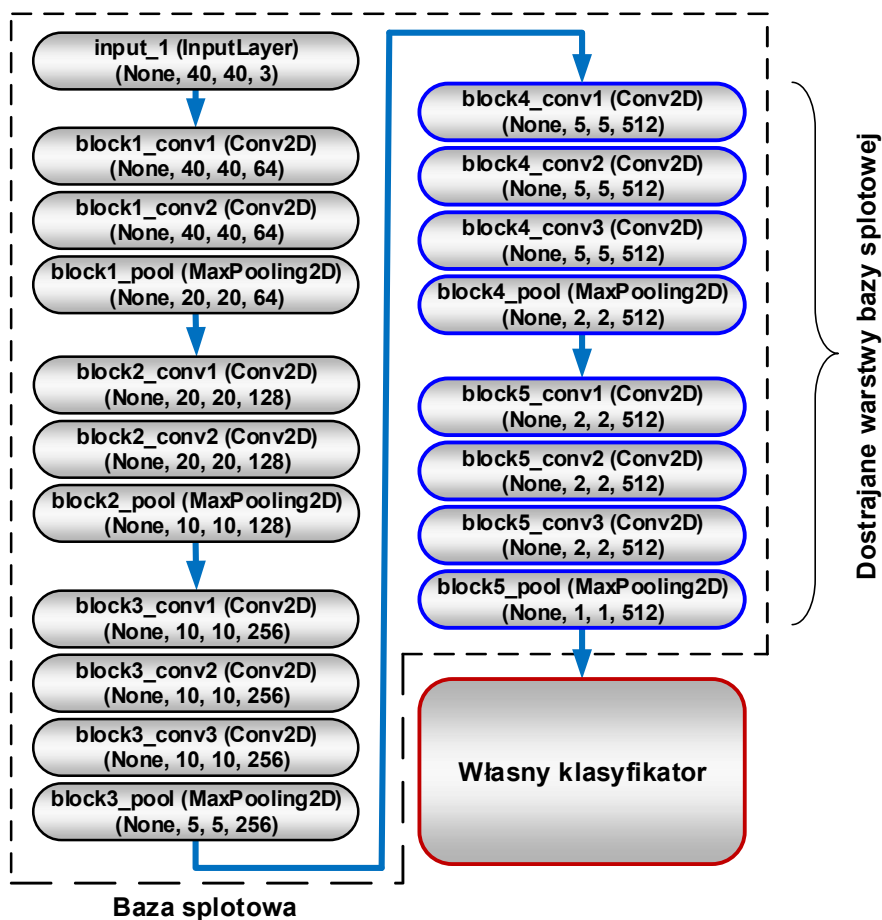
Trenowanie modelu realizowane było w dwóch fazach:

1. **Ekstrakcja cech** – baza splotowa została zamrożona, a dodane warstwy w pełni połączone (tworzące nowy klasyfikator) zostały losowo zainicjowane i wytrenowane przez 100 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 200 epok razem z nowymi warstwami w pełni połączonymi (także z wykorzystaniem dogenerowania danych). Na końcu, cała baza splotowa została odmrożona.

Podczas uczenia modeli, jako funkcję straty, zastosowano binarną entropię krzyżową. Jako algorytmu optymalizacji, użyto RMSprop z niską wartością współczynnika uczenia. Wynosił on 2×10^{-5} dla fazy ekstrakcji cech oraz 10^{-5} dla fazy dostrajania. Niska wartość tego współczynnika wynikała z tego, że modyfikacje reprezentacji dostrajanych warstw bazy splotowej powinny być minimalne. Zbyt duże zmiany tych wartości mogłyby zaszkodzić reprezentacji danych. Jako miarę skuteczności procesu uczenia, zastosowano dokładność klasyfikacji. W ostatniej warstwie klasyfikatora, jako funkcję aktywacji, zastosowano funkcję logistyczną. Wytrenowane modele zapisywane były na dysku w postaci plików w formacie HDF5.

Należy w tym miejscu podkreślić, że strojenie modeli ograniczono do określenia optymalnej liczby epok trenowania, na podstawie wykresów krzywych uczenia i walidacji. Zakres badań nie obejmował poszukiwania optymalnej licz-

by warstw spłotowych podlegających dostrajaniu – zawsze odpowiadała ona dwóm ostatnim blokom bazy spłotowej. Oczywiście, liczba dostrajanych warstw zależała od architektury modelu, co można zaobserwować w ostatniej kolumnie tabeli 5.1. Jako przykład, na rysunku 5.4 przedstawiono architekturę sieci VGG16 wykorzystaną w badaniach, na której zaznaczono bloki podlegające dostrajaniu.



Rys. 5.4. Architektura sieci VGG16. Linia przerywana obejmuje bloki bazy spłotowej. Dwa ostatnie bloki (linia niebieska) są dostrajane. Własny klasyfikator (o strukturze pokazanej na rys. 5.3) zaznaczono linią czerwoną. Znaczenie poszczególnych klas biblioteki Keras jest następujące: *InputLayer* – warstwa wejściowa przyjmująca tensor o kształcie określonym przez format obrazu wejściowego (w tym wypadku jest to obraz RGB o rozmiarze 40×40pikseli); *Conv2D* – warstwa spłotowa z funkcją aktywacji ReLU; *MaxPooling2D* – warstwa łącząca, która zmniejsza o połowę rozmiar mapy cech (zwraca wartości maksymalne każdego z kanałów)

6. Diagnostyka procesu współspalania pyłu węglowego i biomasy

6.1. Kontekst badań

Jedną z najpopularniejszych metod ograniczania emisji gazów cieplarnianych, szczególnie CO₂, jest stosowanie paliw odnawialnych, m.in. biomasy. Spalanie biomasy jest korzystniejsze dla środowiska, niż spalanie paliw kopalnych, ponieważ zawartość szkodliwych pierwiastków (głównie siarki i azotu) w biomase jest niższa, niż w przeciętnym węglu. Ponadto, biomasa pochodzenia roślinnego ma teoretycznie korzystniejszy bilans dwutlenku węgla od paliw kopalnych, ponieważ rosnąc – pochłania jego część w procesie fotosyntezy. Współspalanie biomasy z węglem to stosunkowo tania alternatywa pozyskiwania energii cieplnej do produkcji energii elektrycznej dla istniejących kotłów energetycznych, którą można zastosować po odpowiednim przystosowaniu istniejących instalacji.

Współspalanie biomasy i węgla ma jednak pewne wady, takie jak zmniejszenie sprawności spalania, zwiększenie ilości żużla, czy korozja stalowych elementów kotła. Fizyczne i chemiczne parametry biomasy nie są stałe w czasie, co więcej, różnią się od parametrów węgla. Wraz z paliwem, do komory spalania zawsze dostaje się pewna niekontrolowana ilość powietrza, która zmienia normalną (efektywną) wartość współczynnika nadmiaru powietrza. W wyniku tego zmieniają się warunki i paliwo przestaje być spalane całkowicie. Może to zmienić stan procesu w sposób nieprzewidywalny, a w konsekwencji zwiększyć ilość zanieczyszczeń emitowanych do atmosfery. Dlatego, konieczne jest utrzymywanie procesu w pożądanym stanie. Wymaga to zastosowania systemu, który w sposób możliwie nieinwazyjny i bez opóźnień odczytywałby informacje o zachodzącym procesie.

Nie istnieją metody bezpośredniego pomiaru parametrów świadczących o jakości procesu spalania zachodzącego w jednym palniku. Dlatego też, stosowane są metody pośrednie, które obejmują przede wszystkim metody optyczne i akustyczne [167]. Są to metody nieinwazyjne, a ponadto pozwalają one na uzyskiwanie selektywnej przestrzennej informacji o zachodzącym procesie spalania praktycznie, bez żadnych opóźnień. Wiele z tych metod wykorzystuje różne cechy optycznej emisji płomienia, jako źródła informacji o procesie spalania. Wykorzystywane są do tego celu czujniki wąskopasmowe i szerokopasmowe [168, 169, 170], pracujące w zakresie promieniowania ultrafioletowego, widzialnego i podczerwonego [167, 169]. Opracowane dotychczas metody diagnostyczne wiążą mierzone sygnały optyczne z ważnymi parametrami spalania, takimi jak: stechiometria płomieniowa [168], emisja zanieczyszczeń [169, 170], stabilność płomienia [171, 172, 173], właściwości paliw [174], warunki pracy i stany awaryjne. W wyniku przetwarzania danych optycznych, wyodrębniane są

wybrane cechy czasowe sygnału, takie jak średnia intensywność (najczęściej stosowana) i jej odchylenie standardowe, częstotliwości charakterystyczne, a także inne parametry związane z częstotliwością [168, 169, 170].

Oprócz sygnału pozyskiwanego wskutek optycznej emisji płomienia, jako źródło informacji o procesie spalania, wykorzystywany jest także obraz płomienia, w połączeniu z różnymi metodami komputerowego przetwarzania danych [175, 176, 177]. W ostatnich latach nastąpił gwałtowny rozwój algorytmów uczenia maszynowego, szczególnie metod uczenia głębokiego. Wiele z nich bardzo dobrze sprawdza się w procesie rozpoznawania obrazów, w tym płomieni. Na przykład, spłotowe sieci neuronowe zapewniły wysoką dokładność (94,4%) wykrywania pożaru, na podstawie nagrań wideo zarejestrowanych przez system monitoringu obiektów [178, 179]. Algorytmy uczenia głębokiego są także stosowane w diagnostyce procesu spalania. Abdurakipov i in. badali możliwość monitorowania trybów spalania z wykorzystaniem obrazów płomienia palnika gazowego [180]. Zastosowali oni sieć spłotową, która pozwoliła im uzyskać średnią dokładność klasyfikacji dla wszystkich trybów na poziomie 97,9 %. Lyu i in. zastosowali głęboką sieć przekonań, w celu wykrycia nietypowych warunków w eksperymentalnym systemie spalania [181]. Osiągnęli oni wskaźnik wykrywalności usterek na poziomie 95,6%. Wang i in. zaproponowali metodę identyfikacji stanu spalania (w tym ognia olejowego, spalania proszku i spalania normalnego) oraz pomiaru szybkości wydzielania się ciepła [182]. Używając sieci spłotowej, autorzy osiągnęli średnią dokładność przewidywania stanu spalania na poziomie 99,91%. Gangopadhyay i in. wykorzystali obrazy wideo płomienia nagrane z dużą szybkością oraz dane dotyczące ciśnienia akustycznego, do wykrywania niestabilności w eksperymentalnym systemie spalania [183]. Zastosowali oni sieć spłotową i osiągnęli dokładność od 69% do 100% (zależnie od wariantu spalania). Qiu i in. wykorzystali obrazy płomieni, w celu rozpoznania stanów spalania węgla sproszkowanego (stabilnego, półstabilnego, niestabilnego) w elektrowni ciepłej [184]. Ich metoda opierała się na autoenkoderze, analizie głównych składowych oraz ukrytym modelu Markowa i oferowała dokładność klasyfikacji na poziomie 97,36%. Liu i in. wykorzystali głęboką sieć przekonań do przewidywania zawartości tlenu na wylocie systemu spalania [185]. Głębokie, w pełni połączone sieci neuronowe zostały wykorzystane w badaniach przeprowadzonych przez Tóth i in. [186]. Autorzy zaproponowali system, który z półgodzinnym wyprzedzeniem oraz dokładnością $\pm 1^{\circ}\text{C}$, może przewidywać temperaturę wody w kotle na biomasę z rusztem skokowym. Li i in. zbudowali model oparty na głębokim spłotowym autoenkoderze do przewidywania emisji NO_x w systemie spalania biomasy i gazu [187]. Na podstawie obrazów rodników obecnych w płomieniu, zarejestrowanych za pomocą spektroskopowego systemu obrazowania, autorzy uzyskali względne błędy emisji NO_x pomiędzy wartościami przewidywanymi i referencyjnymi, w granicach $1,7\div 2,6\%$. Przedmiotem badań było także wczesne wykrywanie niestabilnego

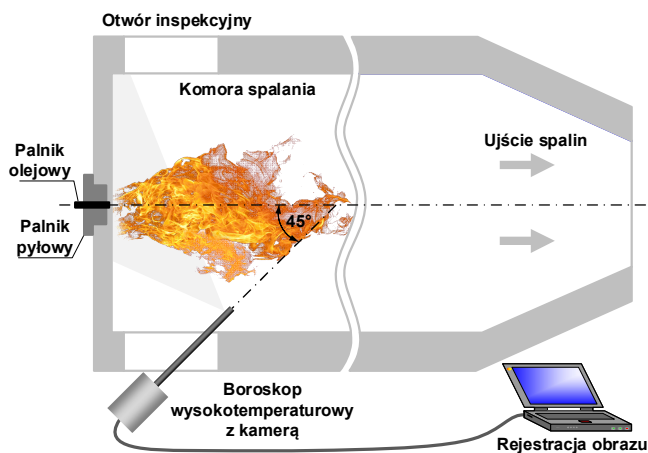
spalania w silnikach z turbiną gazową za pomocą obrazów wideo płomienia, nagranych z dużą szybkością. Akintayo i in. zastosowali w tym celu selektywny autoenkoder [188], Sarkar i in. wykorzystali głęboką sieć przekonań oraz sieć spłotową, po których wykonywana była analiza symboliczna szeregów czasowych [189, 190]. W literaturze nie znaleziono wyników badań, w których autorzy zastosowaliby algorytmy uczenia głębokiego do rozpoznawania stanu spalania pyłu węglowego i jego mieszaniny z biomasą, w oparciu o obrazy płomienia.

Podczas badań, podjęto próbę rozwiązania problemu polegającego na wykryciu niepożądanych stanów procesu spalania. Każdy z takich stanów charakteryzuje się inną wartością współczynnika nadmiaru powietrza. Wraz ze zmianą tego parametru, zmieniają się warunki spalania i może dojść do sytuacji, że paliwo nie będzie spalane całkowicie. Wykrycie stanu niepożądanego może oznaczać nieprawidłową pracę instalacji spalania. Uzyskanie informacji o pojawieniu się nowego stanu niepożądanego pozwoliłoby operatorowi lub systemowi sterowania na przywrócenie stanu normalnego. W wyniku przeprowadzonych badań, taką informację otrzymano na podstawie analizy i klasyfikacji obrazu płomienia za pomocą metod uczenia tradycyjnego i głębokiego. W ramach badań, porównano także oba sposoby uczenia, m.in. pod względem skuteczności wykrywania stanów spalania. Na podkreślenie zasługuje użycie metod głębokich, ponieważ jest to nowe podejście w badaniach nad diagnostyką współspalania węgla kamiennego i biomasy. Autorowi monografii nie są znane wyniki badań, w których do tego celu byłyby stosowane metody uczenia głębokiego. Metoda diagnostyki oparta na uczeniu głębokim znacznie upraszcza cały cykl budowy modelu, ponieważ, w porównaniu z tradycyjnymi algorytmami uczenia, wyeliminowane są etapy analizy obrazu i wstępnego przetwarzania danych. Głębokie sieci spłotowe są w stanie ujawnić wewnętrzne cechy poszczególnych stanów spalania w oparciu o surowe dane uzyskane z obrazów, zapewniając jednocześnie wysoką skuteczność klasyfikacji [191, 192].

6.2. Materiał do badań

Materiał wykorzystany w monografii został zgromadzony przez pracowników Katedry Elektroniki Politechniki Lubelskiej (obecnie Katedra Elektroniki i Technik Informacyjnych) podczas eksperymentów przeprowadzonych w Instytucie Energetyki w Warszawie. Na tej podstawie opracowano dotychczas wiele artykułów naukowych i monografii, m.in. [13, 172, 173, 176, 177, 193]. Testy spalania pyłu węglowego i rozdrobnionej biomasy przeprowadzono na stanowisku zbudowanym z cylindrycznej komory spalania o długości ok. 2,5 m i średnicy ok. 0,7 m (rys. 6.1). Wewnątrz komory znajdował się model wirowego palnika pyłowego, wykonany w skali 1:10 w stosunku do palników przemysłowych oraz palnik olejowy wraz z zapalarką gazową. Za pomocą pyłoprzewodu do palnika dostarczana była mieszanina paliwowa wraz z powietrzem pierwotnym. Istniała możliwość niezależnej regulacji przepływu paliwa oraz powie-

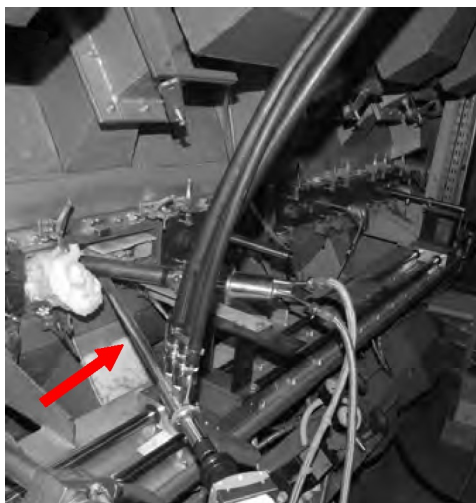
trza. Zmiana ilości dostarczanego węgla oraz biomasy odbywała się poprzez regulację prędkości obrotowej podajnika. Podczas pracy stanowiska, komputerowy system akwizycji danych rejestrował szereg parametrów, w tym natężenie przepływu powietrza pierwotnego i wtórnego, średnią temperaturę w komorze, skład gazu w odległości 1 m od czoła palnika oraz na wylocie z komory paleniskowej. Po obu stronach komory spalania znajdowały się otwory inspekcyjne umożliwiające obserwację płomienia. Do rejestracji obrazu wykorzystano kamerę cyfrową MC1311 firmy Mikrottron GmbH z matrycą z pełnym przenoszeniem ramki, wyposażoną w interfejs CamerLink. Kamera przymocowana była do boroskopu wysokotemperaturowego zainstalowanego w otworze inspekcyjnym pod kątem około 45° do osi palnika (rys. 6.2). Aby zabezpieczyć część optyczną przed zabrudzeniem (sadza, popiół) oraz wpływem wysokiej temperatury, światłowod był chłodzony wodą oraz przedmuchiwany sprężonym powietrzem. Rejestrowano obrazy o rozdzielczości 800×800 pikseli, w trybie RGB, z 24-bitową głębią kolorów. Utrzymywano stały czas naświetlania oraz częstotliwość odświeżania 500 klatek na sekundę. Nagrania wideo dla poszczególnych wariantów spalania zapisywane były w pamięci masowej komputera, w postaci plików AVI, przy czym czas trwania każdego zapisu wynosił 300 sekund.



Rys. 6.1. Schemat poglądowy pokazujący położenie kamery w komorze spalania

Podczas testów realizowane było współspalanie mieszanek sproszkowanego węgla i biomasy w postaci rozdrobnionej słomy. Pomiary wykonywano przy stałym przepływie paliwa, tak aby obliczona moc cieplna palnika (P_{th}) była utrzymywana na stałym poziomie. Ilość doprowadzanego powietrza regulowano, co miało wpływ na wartość współczynnika nadmiaru powietrza (λ). Współczynnik λ określa się jako stosunek rzeczywistej ilości powietrza, w której spalane

jest paliwo, do ilości niezbędnej do jego całkowitego spalania (ilość stechiometryczna). Ilość stechiometryczną można z kolei obliczyć na podstawie analizy równań reakcji chemicznych występujących podczas spalania określonego paliwa.



Rys. 6.2. Zdjęcie komory spalania z zaznaczonym boroskopem (czerwona strzałka) [193]

Podczas współspalania mieszanki sproszkowanego węgla i biomasy, całkowite spalanie odbywało się dla ilości powietrza niższej, niż ilość stechiometryczna, tj. $\lambda \approx 0,75$, dlatego za wartość współczynnika nadmiaru powietrza uznano wartość odpowiadającą „normalnym” warunkom spalania. Na tej podstawie, podczas testów przyjęto 3 wartości współczynnika λ . Pierwsza z nich odpowiadała warunkom normalnym ($\lambda = 0,75$). Druga wartość odpowiadała niedomiarowi powietrza ($\lambda = 0,65$), a trzecia jego nadmiarowi ($\lambda = 0,85$). Dla mocy cieplnej $P_{th} = 250$ kW zbadano wszystkie trzy przypadki, dla $P_{th} = 300$ kW były to przypadki normalnego spalania i nadmiaru powietrza, natomiast dla $P_{th} = 400$ kW zbadano przypadki normalnego spalania i niedomiaru powietrza. Testy przeprowadzono dla siedmiu wariantów spalania, z różnymi ustawieniami P_{th} i λ oraz przy 10% zawartości biomasy. Wartości poszczególnych parametrów, dla każdego z wariantów, zestawiono w tabeli 6.1. Rejestracja sekwencji video rozpoczynała się po ustabilizowaniu się procesu w komorze spalania. Moment ten był określany na podstawie wskazań analizatora gazów oraz opinii specjalisty z wieloletnim doświadczeniem w eksploatacji przemysłowych komór spalania sproszkowanego węgla i rozdrobnionej biomasy.

Tabela 6.1. Warianty procesu spalania (P_{th} – moc cieplna, λ – współczynnik nadmiaru powietrza)

Wariant	P_{th} [kW]	λ	Paliwo [kg/h]	Powietrze wtórne [Nm ³ /h]
1	250	0,85	35,2	103,6
2	250	0,75	36,0	73,2
3	250	0,65	39,4	62,9
4	300	0,85	44,2	132,5
5	300	0,75	43,5	96,4
6	400	0,75	59,7	181,3
7	400	0,65	56,8	152,8

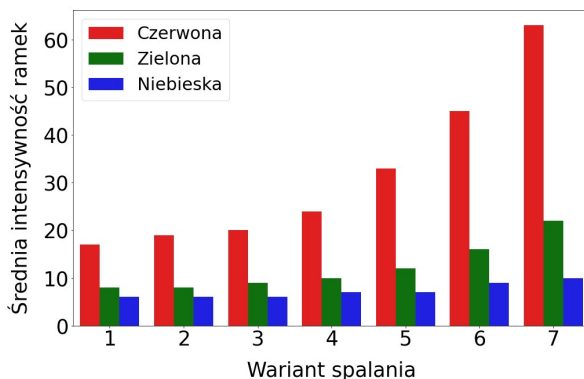
6.3. Wstępne przetwarzanie obrazów płomienia

W diagnostyce procesu spalania z wykorzystaniem metod analizy obrazu stosowane są dwa podstawowe podejścia. Pierwsze z nich wykorzystuje kształt płomienia, a drugie jego teksturę. Wyznaczenie kształtu płomienia nie jest sprawą prostą. Występowanie płomienia jest związane z miejscem w przestrzeni, w którym zachodzi egzotermiczna reakcja utleniania. Trudno jest jednoznacznie wskazać granice, w ramach których zachodzi spalanie, ponieważ stężenia reagentów nie zmieniają się skokowo. W przypadku płomienia pyłowego, głównym źródłem promieniowania jest chmura cząstek stałych o wysokiej temperaturze (pył węglowy, sadza, popiół, itp.). Ich koncentracja także nie zmienia się skokowo, dlatego nie można dokładnie i jednoznacznie określić krawędzi płomienia, jak np. w przypadku ciała stałego. W płomieniu występują również zjawiska pochłaniania i rozpraszania promieniowania, które wprowadzają dodatkowe niejednoznaczności przy wyznaczaniu jego granicy. Problem z jednoznacznym określeniem krawędzi płomienia komplikuje cały proces analizy. Biorąc to pod uwagę, w badaniach zastosowano podejście oparte na teksturze płomienia, a nie na jego krawędzi. W tym przypadku znika problem dokładnej segmentacji płomienia, w celu ekstrakcji jego cech geometrycznych. Drugą, ważniejszą zaletą wykorzystania tekstury jest fakt, iż dostarcza ona znacznie więcej informacji o obiekcie analizy, niż jego kształt.

W dalszej części przedstawiono metodykę postępowania, która pozwoliła na zautomatyzowanie procesu wyznaczania obszaru zainteresowania (ROI), stanowiącego podstawę do ekstrakcji teksturalnych cech obrazu. Zastosowano ROI o kształcie kwadratu, którego środek pokrywał się z centroidem konturu płomienia. W związku z tym, istniała potrzeba wyznaczenia granic płomienia. Należy jednak podkreślić, że służyło to jedynie znalezieniu centroidu konturu, a nie obliczeniu cech geometrycznych. W związku z tym, precyzja wyznaczenia granic płomienia, jakkolwiek ważna, nie była w tym wypadku zagadnieniem kluczowym. Wszystkie operacje, których wyniki przedstawiono w niniejszym rozdziale, wykonano z wykorzystaniem języka Python 3.7.3 oraz bibliotek OpenCV 3.4.1 i NumPy 1.16.4.

6.3.1. Analiza statystyczna składowych koloru

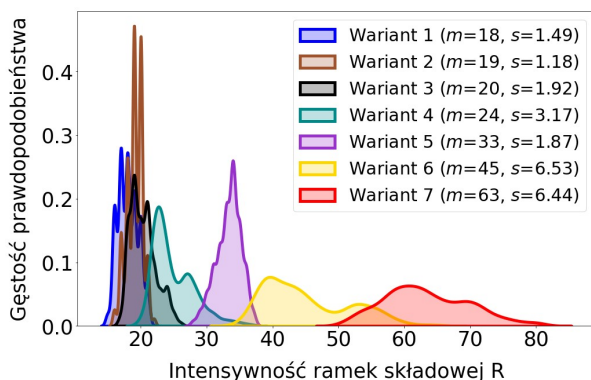
Podczas testów założono stacjonarność warunków, w jakich zachodził proces spalania oraz ustabilizowanie się parametrów emisji. Na tej podstawie uznano, że fragment zapisu obrazu płomienia o czasie trwania około 3 sekund jest wystarczający do wykorzystania go w dalszych badaniach. Fragmenty plików wideo, odpowiadające różnym wariantom procesu spalania, zapisano w postaci obrazów o rozmiarach 800×800 pikseli. Każdy plik wideo został skonwertowany do 1676 plików graficznych w formacie PNG, z 24-bitową głębią kolorów. Do tej konwersji użyto oprogramowania VirtualDubMod 1.5.10.2. Analiza udziału poszczególnych składowych koloru w tworzeniu obrazu płomienia wykazała, że składowa koloru czerwonego (R) odgrywa kluczową rolę. Zilustrowano to na wykresach słupkowych pokazanych na rysunku 6.3. Wykresy, które wykonano dla każdego wariantu spalania, przedstawiają wartości średnie intensywności poszczególnych składowych dla całego zestawu danych.



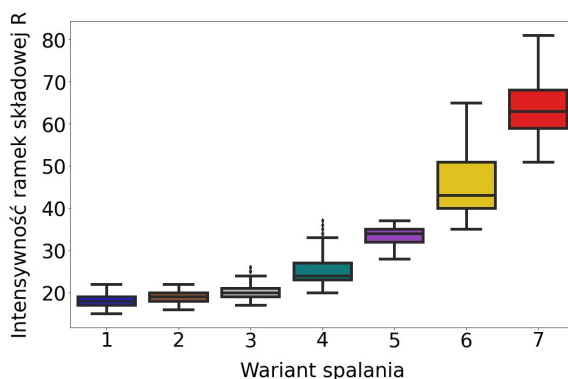
Rys. 6.3. Wartość średnia intensywności składowych koloru dla poszczególnych wariantów procesu spalania. Przedstawione wyniki uzyskano na podstawie analizy całego zestawu, liczącego dla każdego wariantu 1676 ramek obrazu

Dominującą rolę odgrywała składowa R, dlatego w dalszej kolejności, dla każdego obrazu obliczono jej intensywność, rozumianą jako wartość średnia wszystkich pikseli obrazu. Uzyskany parametr stał się podstawą do obliczenia funkcji gęstości prawdopodobieństwa (rys. 6.4) oraz budowy wykresów pudełkowych (rys. 6.5). Wyniki przedstawione na rysunkach 6.4 i 6.5 pokazują, że występują znaczne różnice natężenia składowej R pomiędzy poszczególnymi mocami cieplnymi P_{th} . Im większa moc, tym większe jest natężenie składowej R ramek obrazu. Z kolei, przy tej samej mocy cieplnej, natężenie ramek przyjmuje różne wartości, zależnie od współczynnika nadmiaru powietrza. Zwiększenie nadmiaru powietrza, w stosunku do wartości normalnej, zmniejsza intensywność obrazu i odwrotnie, zmniejszenie nadmiaru powietrza powoduje zwiększenie

intensywności. Należy zauważyć, że poziom intensywności ramek jest ściśle związany z jasnością tła obrazu, z którego wydzielany jest kontur płomienia. Dlatego też, w procesie segmentacji konieczne było uwzględnienie zmian intensywności ramek obrazu, w zależności od bieżącej mocy cieplnej oraz współczynnika nadmiaru powietrza.



Rys. 6.4. Wykresy funkcji gęstości prawdopodobieństwa intensywności ramek składowej R dla poszczególnych wariantów spalania (m – średnia arytmetyczna, s – odchylenie standardowe)



Rys. 6.5. Wykresy rozproszenia intensywności ramek składowej R wokół mediany

W celu zapewnienia lepszych warunków do segmentacji konturu płomienia, konieczne było wyeliminowanie efektu tła, którego jasność zmieniała się, w zależności od wartości parametrów P_{th} i λ . Do tego celu, zastosowano nieliniową korekcję gamma. Polegała ona na tym, że jaśniejsze piksele obrazu (należące do płomienia) były tylko nieznacznie przyciemniane, natomiast piksele

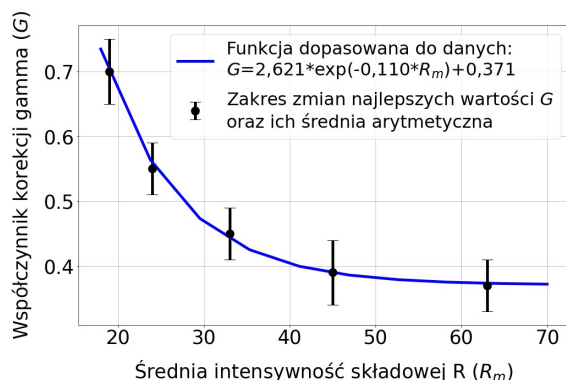
ciemniejsze (należące do tła) były przyciemniane w znacznie większym stopniu. W ten sposób, wpływ tła na jakość segmentacji konturu płomienia został osłabiony. Wielkość zmiany natężenia poszczególnych pikseli zależał od wartości współczynnika korekcji gamma G . Zastosowano adaptacyjny dobór tej wartości na podstawie natężenia składowej R bieżącej ramki obrazu. Postać funkcji służącej do obliczenia wymaganej wartości G określono empirycznie. Należy zaznaczyć, że korekcja gamma nie miała wpływu na wartość deskryptorów cech i pełniła tylko funkcję pomocniczą, w możliwie dokładnym określeniu granic płomienia. Otrzymany kontur był następnie nakładany na obraz oryginalny (bez korekty) i na jego podstawie uzyskiwano ROI, którego środek pokrywał się z centroidem konturu. Zatem, precyzja określenia granic płomienia nie wpływała na cechy diagnostyczne, ponieważ uzyskiwane one były na podstawie tekstury centralnej części płomienia, a nie na podstawie jego konturu.

6.3.2. Adaptacyjna korekcja gamma

Na początku algorytmu realizującego korekcję gamma, budowana jest tablica odwzorowań LUT (ang. *Look-up table*). Podczas tej operacji, elementy tablicy wejściowej o wartościach z zakresu $[0, 255]$ zostają przeskalowane do zakresu $[0, 1]$. Nowe wartości są następnie podnoszone do potęgi o wykładniku równym odwrotności współczynnika korekcji gamma. Operacja ta przeprowadzana jest zgodnie z formułą: $O = I^{1/G}$, gdzie: I – wartość elementu macierzy wejściowej; O – wartość elementu macierzy wyjściowej; G – współczynnik korekcji gamma. Na końcu, wszystkie elementy macierzy wyjściowej zostają z powrotem przeskalowane do zakresu $[0, 255]$ i zapisywane w tablicy LUT. Istota korekcji gamma polega więc na tym, że dla każdego piksela obrazu wejściowego, w tablicy LUT znajduje się jego odwzorowanie (nowa wartość intensywności). Zbiór wszystkich odwzorowań tworzy obraz wyjściowy (po korekcji). Wartości współczynnika $G > 1$ rozjaśniają obraz, dla $G = 1$ nie wykonywane są żadne zmiany, natomiast dla $G < 1$ obraz jest przyciemniany.

Zastosowano adaptacyjny dobór wartości współczynnika korekcji gamma, w zależności od natężenia składowej R bieżącej ramki obrazu. Analiza statystyczna pozwoliła określić wartości średnie intensywności ramek (R_m) dla poszczególnych wariantów spalania: wariant 1 – $18 \pm 1,49$; wariant 2 – $19 \pm 1,18$; wariant 3 – $20 \pm 1,92$; wariant 4 – $24 \pm 3,17$; wariant 5 – $33 \pm 1,87$; wariant 6 – $45 \pm 6,53$; wariant 7 – $63 \pm 6,44$. Na tej podstawie, dla każdego wariantu wybrano po 10 ramek, dla których intensywność składowej R była równa, określonej wcześniej, wartości średniej. Ze względu na zbliżone wartości R_m dla wariantów nr 1, 2 i 3, przyjęto dla nich jedną wartość, równą 19. Następnie, w odniesieniu do ramek wybranych dla poszczególnych wariantów, wykonano segmentację płomienia, podczas której arbitralnie określono najlepsze wartości współczynnika G .

Uśrednienie 10 wyników znalezionych empirycznie, pozwoliło uzyskać wartości współczynnika G , które uznano za optymalne dla poszczególnych wariantów. W ten sposób otrzymano współrzędne 5 punktów (R_m , G), do których dopasowano funkcję wykładniczą postaci: $G = 2,621 \cdot \exp(-0,110 \cdot R_m) + 0,371$ (rys. 6.6). Funkcję tę zastosowano następnie w procesie segmentacji płomienia, do adaptacyjnego doboru wartości współczynnika G , w zależności od intensywności składowej R bieżącej ramki obrazu.



Rys. 6.6. Optymalne wartości współczynnika G dla średnich intensywności składowej R poszczególnych wariantów (czarne punkty), znalezione doświadczalnie na podstawie jakości segmentacji. Linia niebieska pokazuje przebieg funkcji wykładniczej dopasowanej do danych empirycznych

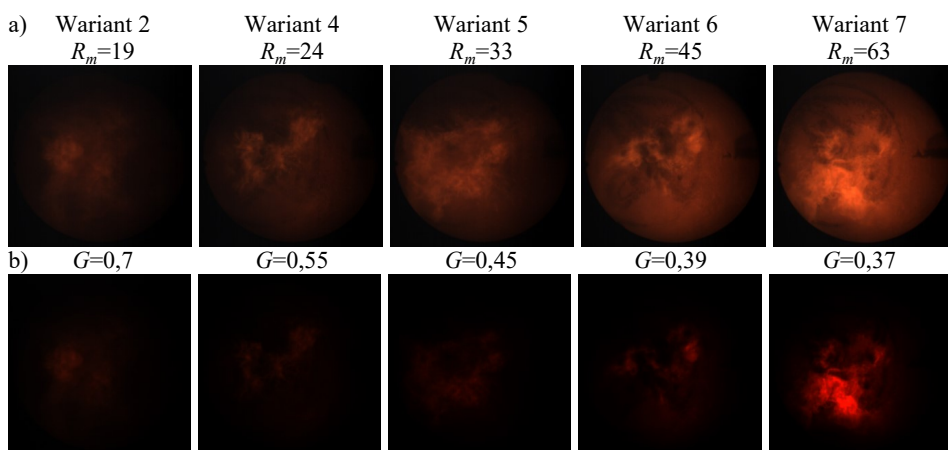
Na rysunku 6.7 zaprezentowano uzyskane wyniki dla przykładowych obrazów płomieni należących do różnych wariantów. Z kolei, na rysunku 6.8 przedstawiono różnice w wynikach segmentacji tego samego obrazu płomienia bez korekty gamma i po jej wykonaniu.

6.3.3. Dobór wielkości i ekstrakcja obszaru zainteresowania

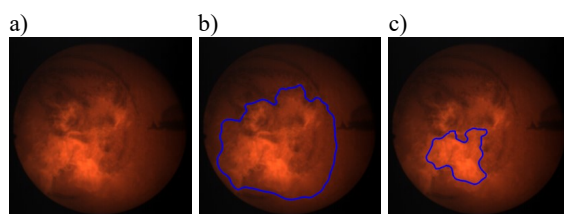
Celem etapu przetwarzania obrazu było wyznaczenie kwadratowego obszaru zainteresowania, którego środek pokrywał się z centroidem konturu płomienia. Kształt ROI wynikał z własności zastosowanych klasyfikatorów. W przypadku wielkości ROI, należało uwzględnić dwa przeciwstawne wymagania. Z jednej strony, analizowany obszar powinien być jak największy, tak aby zawierał maksymalną ilość informacji pozwalających skutecznie rozróżnić warianty spalania. Jednak z drugiej strony, im mniejszy jest obszar zainteresowania, tym krótszy jest czas przetwarzania całego systemu oraz mniejsze straty ramek obrazu.

Straty ramek występują wówczas, gdy nie jest spełniony warunek ekstrakcji, tzn. ROI nie zawiera się całkowicie w konturze płomienia. Aby oszacować odsetek ramek obrazu, dla których nie był spełniony warunek ekstrakcji, przetesto-

wano algorytm segmentacji dla różnych wielkości ROI. Uzyskane wyniki zamieszczono w tabeli 6.2. Problem strat ramek jest bardzo ważny, ponieważ istnieje różnica pomiędzy płomieniami dla poszczególnych wariantów spalania. W przypadku wariantów 1÷3 obszar płomienia jest stosunkowo duży i regularny, a obraz wykazuje małą dynamikę zmian tekstury. Zmiany te zachodzą na dużej powierzchni, więc aby uchwycić różnice pomiędzy wariantami, wielkość ROI musi być dostatecznie duża.



Rys. 6.7. Wyniki korekcji gamma dla przykładowych obrazów płomienia należących do różnych wariantów: a) obrazy wejściowe; b) obrazy po korekcji gamma. Adaptacyjny wybór współczynnika korekcji gamma skutecznie eliminuje tło obrazu, niezależnie od średniej intensywności składowej R . Dzięki temu utworzone zostają lepsze warunki do segmentacji obszaru płomienia



Rys. 6.8. Wpływ korekcji gamma na jakość segmentacji: a) przykładowy obraz płomienia należący do wariantu nr 7; b) wynik segmentacji bez korekcji gamma (wyłoniony obiekt zawiera również znaczny obszar tła); c) wynik segmentacji po korekcji gamma. Wyniki pokazują, że zastosowanie korekcji gamma znacznie poprawia jakość segmentacji obszaru płomienia.

Tabela 6.2. Odsetek ramek obrazu odrzuconych z powodu niespełnienia warunku ekstrakcji ROI dla różnych wielkości obszaru zainteresowania (liczby wyrażone są w %). Warunek ten oznacza, że ROI musi być całkowicie zawarty w konturze płomienia

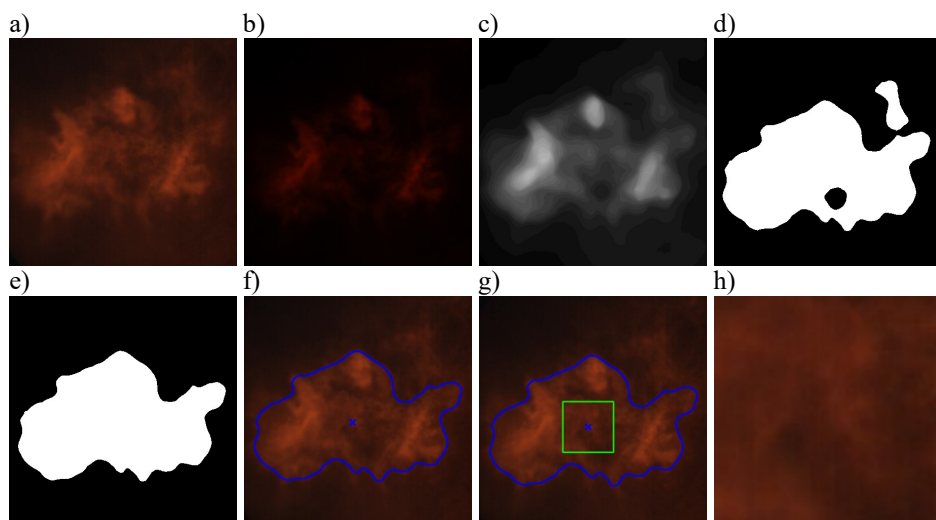
Wariant	100×100	80×80	60×60	40×40
1	0,12	0,00	0,00	0,00
2	0,06	0,00	0,00	0,00
3	6,03	2,27	0,89	0,36
4	36,58	23,75	14,86	7,88
5	2,45	1,37	0,78	0,12
6	44,81	30,43	21,54	14,14
7	39,74	27,15	18,08	11,75

Dla wariantów 4÷7, obszar płomienia wykazuje większą nieregularność oraz dynamikę zmian tekstury. Dlatego, aby straty ramek były mniejsze, wielkość ROI musi być mniejsza, niż dla wariantów 1÷3. W przypadku wariantów 4÷7 zmiany tekstury pomiędzy kolejnymi ramami występują na stosunkowo małej powierzchni, dlatego mniejszy ROI zawiera wystarczającą ilość informacji, aby zapewnić skuteczną klasyfikację. Przyjęto założenie, że straty ramek nie mogą przekraczać 15%. Na tej podstawie oraz biorąc pod uwagę wcześniejsze rozważania, przyjęto wielkość ROI równą 100×100 pikseli dla wariantów 1÷3 oraz 40×40 pikseli dla wariantów 4÷7.

Niżej wymieniono najważniejsze operacje wykonane podczas wstępnego przetwarzania obrazu.

1. Wczytanie obrazu źródłowego w formacie PNG, z 24-bitową głębią kolorów i wielkości 800×800 pikseli (Rys. 6.9a).
2. Obliczenie intensywności składowej R obrazu jako wartości średniej intensywności wszystkich jego pikseli.
3. Przeprowadzenie adaptacyjnej korekcji gamma zgodnie z procedurą opisaną w podrozdziale 6.3.2 (rys. 6.9b).
4. Konwersja obrazu do 8-bitowej skali szarości.
5. Normalizacja histogramu obrazu.
6. Filtracja medianowa z maską o rozmiarze 33×33 piksele (rys. 6.9c).
7. Progowanie obrazu z wykorzystaniem metody Otsu (rys. 6.9d).
8. Operacje morfologiczne (zamknięcie, otwarcie) z wykorzystaniem elementu strukturalnego o rozmiarze 3×3 piksele. Wykrycie wszystkich konturów na obrazie.
9. Znalezienie konturu o największej powierzchni (rys. 6.9e). Wykorzystano obserwację polegającą na tym, że statystycznie, spośród wszystkich konturów występujących na obrazie, ten należący do płomienia ma największą powierzchnię.

10. Obliczenie współrzędnych centroidu konturu (rys. 6.9f). Wykorzystano następujące zależności: $x=m_{10}/m_{00}$, $y=m_{01}/m_{00}$, gdzie: m_{00} – moment normalny rzędu zerowego, m_{10} , m_{01} – momenty normalne rzędu pierwszego.
11. Znalezienie kwadratu o wielkości 100×100 lub 40×40 pikseli (zależnie od wariantu spalania), którego środek pokrywa się z centroidem konturu (rys. 6.9g).
12. Skopiowanie z obrazu źródłowego (bez korekcji gamma) ROI wyznaczonego przez znaleziony kwadrat pod warunkiem, że jest on całkowicie zawarty w konturze płomienia (rys. 6.9h). Jeśli warunek ten nie jest spełniony, ekstrakcja ROI nie jest wykonywana.



Rys. 6.9. Przykładowe wyniki najważniejszych operacji wykonywanych podczas wstępnego przetwarzania obrazu: a) obraz źródłowy; b) wynik korekcji gamma zgodnie funkcją $G=f(R_m)$ (rys. 6.6); c) obraz w skali szarości po normalizacji histogramu i filtracji medianowej; d) obraz binarny po progowaniu metodą Otsu; e) obraz po operacjach morfologicznych (zamknięcie, otwarcie) i znalezieniu konturu o największej powierzchni; f) obraz z zaznaczonym konturem płomienia oraz jego centroidem (niebieski krzyżyk); g) obraz z zaznaczonym ROI (zielony kwadrat o wielkości 100×100 lub 40×40 pikseli), którego środek pokrywa się z centroidem konturu; h) ROI skopiowany z obrazu źródłowego (bez korekcji gamma). Dla lepszej czytelności, na powyższych rysunkach przedstawiono centralną część przetwarzanych obrazów, zawierającą płomień. Rozmiary wszystkich obrazów (oprócz ROI) wynosiły 800×800 pikseli.

6.4. Zastosowanie metod uczenia tradycyjnego oraz dużego zbioru danych

6.4.1. Wstępne przetwarzanie danych

Pełny zbiór danych zawierał 3 tys. obserwacji, przy czym połowa z nich należała do wariantu nr 5, a połowa do wariantu nr 4. Zbiór ten został losowo podzielony na część treningową i testową w taki sposób, że pierwsza część stanowiła 70%, a druga 30% zbioru pełnego. Szczegółowe dane dotyczące liczebności podzbiorów zamieszczono w tabeli 6.3.

Tabela 6.3. Liczebność podzbiorów otrzymanych w wyniku podziału dużego zbioru danych

Zbiór danych	Liczba obserwacji	Część zbioru pełnego	Wariant nr 5	Wariant nr 4
Pełny	3000	1,0	1500	1500
Treningowy	2100	0,7	1050	1050
Testowy	900	0,3	450	450

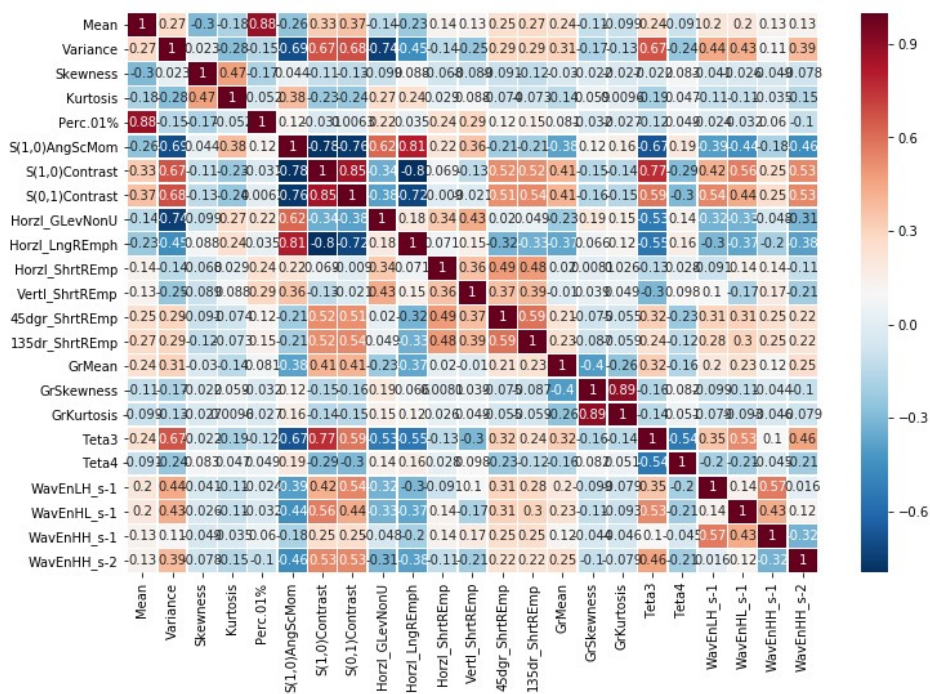
W wyniku estymacji parametrów tekstury otrzymano 290 deskryptorów cech [72]. Kolejnym etapem było wstępne przetwarzanie danych. Ze względu na fakt, iż deskryptory cech mierzone były w różnych skalach (przedziałowa, ilorazowa), cechy zostały przeskalanowane z wykorzystaniem standaryzacji klasycznej: $z_{ij} = (x_{ij} - \bar{x}_j) / s_j$, gdzie: x_{ij} – wartość cechy j dla obserwacji i ; z_{ij} – wartość standaryzowana cechy j dla obserwacji i ; \bar{x}_j – średnia arytmetyczna cechy j ; s_j – odchylenie standardowe cechy j . Po standaryzacji, dla wszystkich cech obowiązywała skala przedziałowa oraz rozkład normalny $N(0,1)$. Skalowanie przeprowadzono jednorazowo, dla danych treningowych. Następnie, w podobny sposób przekształcono dane testowe. Podczas tego procesu, dla każdej cechy wykorzystano wektory wartości średnich i wariancji uzyskane podczas standaryzacji danych treningowych.

Kolejnym krokiem była czteroetapowa procedura oczyszczania danych:

1. Usunięcie cech przyjmujących wartości stałe (wariancja równa 0).
2. Usunięcie cech przyjmujących wartości prawie stałe (wariancja mniejsza, niż 0,01).
3. Usunięcie cech przyjmujących wartości zduplikowane.
4. Usunięcie cech wzajemnie skorelowanych. Zastosowano współczynnik korelacji Pearsona (r), który wykrywa zależności liniowe pomiędzy cechami oraz zakłada normalność ich rozkładu. Usunięto cechy, dla których $|r| > 0,9$, co wskazuje na występowanie silnej korelacji.

W wyniku oczyszczania danych, liczba cech została zmniejszona z 290 do 23 (rys. 6.10). Wstępne przetwarzanie danych przeprowadzono z wykorzystaniem

biblioteki *scikit-learn* i języka programowania Python, który zastosowano w dalszych eksperymentach.



Rys. 6.10. Macierz korelacji pomiędzy cechami stanowiącymi wynik oczyszczania dużego zbioru danych

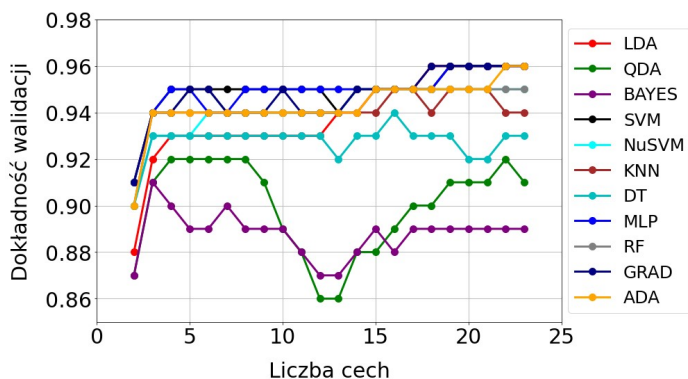
W kolejnym etapie wstępnego przetwarzania danych wykonano selekcję cech. Ogólnie rzecz biorąc, celem tej operacji jest ograniczenie początkowego (pełnego) zbioru cech do pewnego podzbioru, zawierającego cechy ważne z punktu widzenia zastosowanego kryterium. W wyniku oczyszczania danych, pełny zbiór 290 cech został zredukowany do 23 cech. Na tak znaczną redukcję wpłynęło głównie występowanie dużej liczby cech z silną korelacją wzajemną ($|r| > 0,9$). W związku z tym, na etapie selekcji zrezygnowano z dodatkowej redukcji, a jako cel przyjęto budowę listy rankingowej przez każdą z zastosowanych metod selekcji. Listy rankingowe zawierały zestaw cech w kolejności odzwierciedlającej ich ważność w procesie rozróżniania obserwacji należących do poszczególnych wariantów spalania. Liczba cech występujących na listach zbudowanych przez poszczególne metody była następująca: metody filtra i opakowujące: 23, metody wbudowane: regresja logistyczna – 11, AdaBoost – 8, LGBM – 10. Wyboru optymalnej liczby cech dla każdej z zastosowanych metod

selekcji dokonano na kolejnym etapie, którym było trenowanie modeli klasyfikacyjnych. Aby uniknąć przetrenowania modeli, selekcję cech przeprowadzono wyłącznie w odniesieniu do danych treningowych. Następnie, jej wyniki wykorzystano do przekształcenia danych testowych. Selekcję cech zrealizowano z wykorzystaniem bibliotek *scikit-learn*, *scikit-feature*, *ReliefF*, *MLxtend* i *LightGBM*.

6.4.2. Wykorzystanie pełnego zestawu cech

W procesie trenowania modeli klasyfikacyjnych wykorzystano 9 zbiorów treningowych (po jednym dla każdej metody selekcji) uporządkowanych zgodnie z rankingiem cech. Należy zaznaczyć, iż w tym scenariuszu badań zachowano pełny zestaw cech, to znaczy taki, jaki zwróciła dana metoda selekcji. Podczas uczenia modeli, liczba cech zbioru treningowego zmieniała się zgodnie z listą rankingową. Dla metod filtra i opakowujących, liczba ta zmieniała się od 2 do 23, dla regresji logistycznej – od 2 do 11, dla metody AdaBoost – od 2 do 8, zaś dla metody LGBM – od 2 do 10. Podczas trenowania, zastosowano metodę przeszukiwania siatki do znalezienia optymalnych wartości hiperparametrów danego modelu (dostrajanie). Wykorzystano do tego celu metodę *GridSearchCV*, dostępną w module *model_selection* biblioteki *scikit-learn*. Oceny modeli dokonywano na podstawie 10-krotnej walidacji krzyżowej. Na tej podstawie, dla każdego podzbioru cech określono najskuteczniejszy klasyfikator oraz optymalne wartości jego hiperparametrów. W dalszej części, taki model będzie określany jako optymalny.

Na rysunku 6.11 przedstawiono przykładowe wyniki walidacji dla metody SFS. Każda z 11 krzywych znajdujących się na wykresie odpowiada innej metodzie klasyfikacji. Z kolei, punkty zaznaczone na krzywych odzwierciedlają dokładność walidacji modelu, który okazał się optymalny dla danej liczby cech. Jak wynika z wykresu, dla każdej metody klasyfikacji zbudowano 22 modele optymalne. W sumie, dla wszystkich metod, zbudowano i zapisano w postaci plików na dysku 242 takie modele.



Rys. 6.11. Wyniki walidacji modeli optymalnych dla metody SFS

Na podstawie wyników przedstawionych na rys. 6.11, dla każdej metody klasyfikacji wybrano model najskuteczniejszy, tzn. taki, który zapewniał największą dokładność walidacji, przy minimalnej liczbie cech zbioru treningowego. Wybór najskuteczniejszego modelu jednoznacznie określał optymalny zestaw cech, który był wykorzystany do jego budowy.

W tabeli 6.4. zamieszczono wyniki wyboru modeli najskuteczniejszych dla poszczególnych metod klasyfikacji. Na podstawie analizy tych danych, jako modele najlepsze dla metody SFS, uznano modele LDA i GRAD (wiersze podświetlone szarym tłem w tabeli 6.4).

Tabela 6.4. Wyniki wyboru modeli, które okazały się najskuteczniejsze dla poszczególnych metod klasyfikacji, przy wykorzystaniu metody SFS

Metoda klasyfikacji	Dokładność walidacji	Optymalna liczba cech	Optymalne parametry modelu
LDA	0,96	18	<i>solver='svd'</i>
QDA	0,92	4	<i>tol=1e-05</i>
BAYES	0,91	3	<i>var_smoothing=1e-09</i>
SVM	0,96	19	<i>C=1.0, gamma='scale', kernel='linear'</i>
NuSVM	0,95	14	<i>gamma='scale', kernel='rbf', nu=0.3</i>
KNN	0,95	16	<i>n_neighbors=5</i>
DT	0,94	16	<i>criterion='gini', max_depth=7</i>
MLP	0,96	19	<i>activation='relu', alpha=0.0001, hidden_layer_sizes=(3,), max_iter=1000, solver='lbfgs'</i>
RF	0,95	5	<i>max_depth=9, n_estimators=100</i>
GRAD	0,96	18	<i>loss='exponential', n_estimators=100</i>
ADA	0,96	22	<i>n_estimators=75</i>

Kryterium wyboru najlepszego modelu było takie samo, jak w przypadku wyboru modelu optymalnego, tzn. największa dokładność walidacji, przy minimalnej liczbie cech zbioru treningowego. Dla obu modeli, dokładność walidacji wyniosła 0,96, a do ich budowy wykorzystano 18 cech.

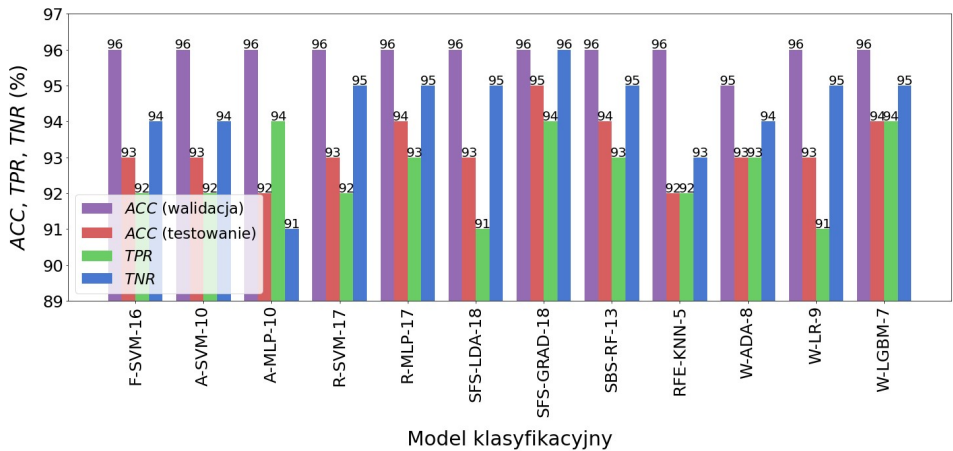
Procedurę wyboru najlepszych modeli, którą scharakteryzowano wcześniej na przykładzie metody SFS, zastosowano dla wszystkich metod filtra i opakujących. W przypadku metod wbudowanych, konstruowane i dostrajane były tylko te klasyfikatory, które zastosowano w procesie selekcji. Uzyskane wyniki zamieszczono w tabeli 6.5. Dla każdego modelu podano metodę selekcji zastosowaną do utworzenia rankingu cech, metodę klasyfikacji oraz liczbę wykorzystanych cech. Oznaczenia występujące w symbolu modelu należy interpretować następująco: 1 pozycja – metoda selekcji (A – ANOVA, F – FISHER, R – RELIEF); 2 pozycja – metoda klasyfikacji; 3 pozycja – liczba użytych cech. Znaczenie symboli metod selekcji i klasyfikacji występujących w tabeli 6.5 podano w wykazie symboli i akronimów na początku monografii.

Tabela 6.5. Najlepsze modele klasyfikacyjne dla dużego zbioru danych i pełnego zestawu cech.
Litera "W" występująca w symbolu modelu oznacza metodę wbudowaną

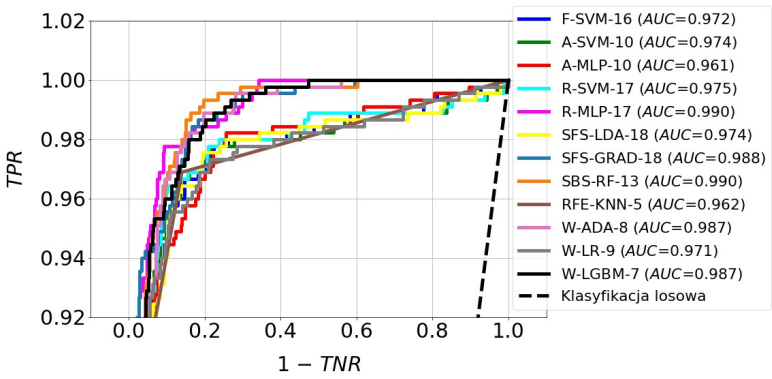
Symbol modelu	Metoda selekcji cech	Metoda klasyfikacji	Liczba cech
F-SVM-16	FISHER	SVM	16
A-SVM-10	ANOVA	SVM	10
A-MLP-10	ANOVA	MLP	10
R-SVM-17	RELIEF	SVM	17
R-MLP-17	RELIEF	MLP	17
SFS-LDA-18	SFS	LDA	18
SFS-GRAD-18	SFS	GRAD	18
SBS-RF-13	SBS	RF	13
RFE-KNN-5	RFE	KNN	5
W-ADA-8	W	ADA	8
W-LR-9	W	LR	9
W-LGBM-7	W	LGBM	7

Wyniki walidacji i testowania modeli wymienionych w tabeli 6.5 zaprezentowano na rysunku 6.12. Największą dokładność klasyfikacji dla zbioru testowego miały modele: R-MLP-17 ($ACC = 94\%$), SFS-GRAD-18 ($ACC = 95\%$), SBS-RF-13 ($ACC = 94\%$) i W-LGBM-7 ($ACC = 94\%$). Dokładność testowania pozostałych modeli była niewiele gorsza i wahała się w granicach $92\div 93\%$. Zdecydowana większość modeli uzyskała dokładność walidacji na poziomie 96%. Należy także zauważyć, iż niemal wszystkie modele uzyskały wyższą wartość indeksu TNR , w porównaniu z TPR . Oznacza to, że wariant nr 5 (normalne warunki spalania) rozpoznawany był z większą dokładnością, niż wariant nr 4 (nadmiar powietrza). Wysoką jakość, wymienionych wcześniej, czterech

modeli potwierdzają także krzywe ROC na rysunku 6.13. Na wykresie ($1-TNR$, TPR) przebiegają one znacznie bliżej punktu o współrzędnych (0, 1), oznaczającego idealną klasyfikację, w porównaniu z adekwatnymi krzywymi należącymi do pozostałych modeli. Ten fakt potwierdzają wielkości obszarów pod krzywymi ROC, wyrażone za pomocą parametrów AUC . Dla wymienionych wyżej czterech najlepszych modeli, wartość AUC waha się w zakresie $0,987 \div 0,990$, natomiast dla pozostałych modeli, wspomniany zakres to $0,961 \div 0,975$.



Rys. 6.12. Wyniki walidacji i testowania najlepszych modeli klasyfikacyjnych dla dużego zbioru danych i pełnego zestawu cech



Rys. 6.13. Krzywe ROC wraz z parametrami AUC dla najlepszych modeli klasyfikacyjnych zbudowanych z wykorzystaniem pełnego zestawu cech dla dużego zbioru danych

6.4.3. Wykorzystanie ograniczonego zestawu cech

Istota badań, których wyniki zaprezentowano w niniejszym rozdziale polegała na tym, że na podstawie rankingu, liczbę cech ograniczono do takiej wielkości, której zwiększanie nie powodowało już istotnego wzrostu dokładności walidacji. Przyjęto, iż ten *istotny* wzrost nie może być większy, niż 0,05. Na rysunku 6.14, na przykładzie metody SFS i klasyfikatora GRAD, przedstawiono ideę wyboru najlepszego modelu dla pełnego i ograniczonego zestawu cech.



Rys. 6.14. Wybór najlepszego modelu dla pełnego i ograniczonego zestawu cech dla metody SFS i klasyfikatora GRAD

Jak widać na podstawie przebiegu dokładności walidacji, dla zestawu pełnego uzasadnione jest przyjęcie liczebności $n = 18$, ponieważ jest to minimalna liczba, przy której uzyskana zostaje maksymalna, możliwa do osiągnięcia dokładność walidacji (0,96). Z kolei, dla scenariusza z ograniczonym zestawem cech, wykorzystano pierwsze 3 cechy z listy rankingowej, dla których dokładność walidacji wyniosła 0,94. Jak widać na rysunku 6.14, dalsze zwiększanie liczebności zestawu cech nie wiąże się już z istotnym wzrostem tej dokładności. Wybrano zatem model, którego dokładność walidacji jest tylko o 0,02 gorsza od maksymalnej dokładności klasyfikatora GRAD dla metody SFS. Kosztem nieznacznego pogorszenia dokładności, uzyskano jednak znaczną redukcję złożoności modelu, polegającą na zmniejszeniu liczby wykorzystywanych cech z 18 do 3. Taka sytuacja korzystnie wpływa na implementację modelu oraz wydajność potencjalnego systemu predykcji.

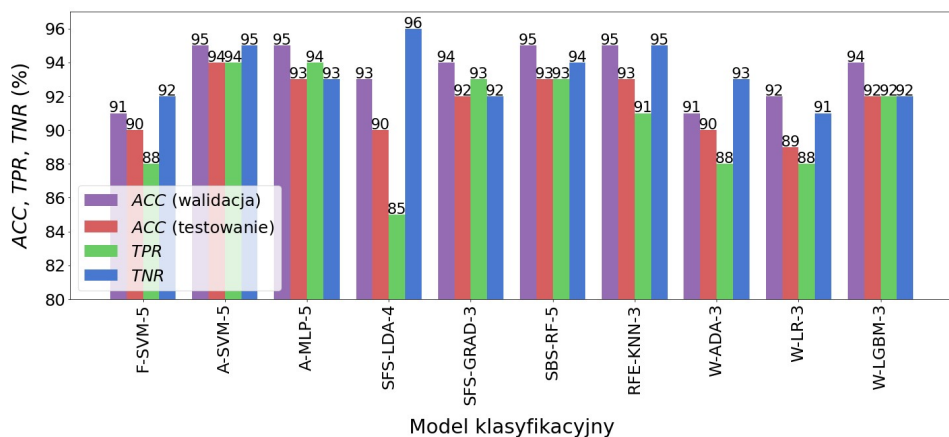
Opisaną wyżej metodę wyboru najlepszych modeli zastosowano dla pozostałych metod selekcji. Uzyskane wyniki zamieszczono w tabeli 6.6. Tabela nie zawiera informacji o modelach zbudowanych z wykorzystaniem metody Relief. W tym przypadku, nie udało się określić liczby cech mniejszej, niż dla zestawu pełnego (17), przy której dokładność walidacji zmniejszyłaby się mniej, niż

wartość dopuszczalna, tj. 0,05. Jak wynika z danych zamieszczonych w tabeli 6.6, liczebność ograniczonych zestawów cech wahała się w granicach 3÷5.

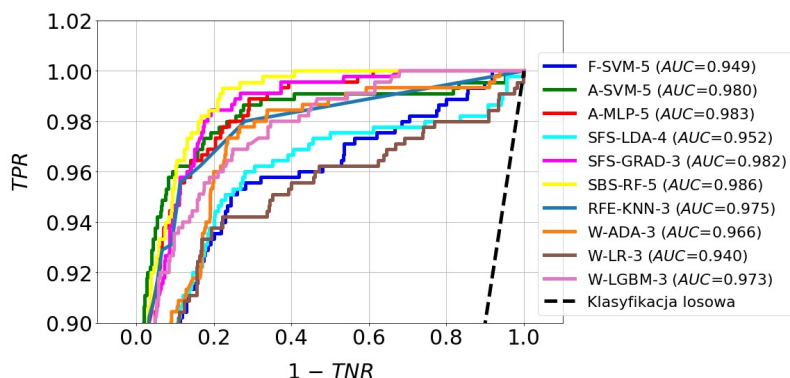
Na rysunku 6.15 zaprezentowano wyniki walidacji i testowania modeli wymienionych w tabeli 6.6. Największą dokładność klasyfikacji dla zbioru testowego uzyskały modele: A-SVM-5 ($ACC = 94\%$), A-MLP-5 ($ACC = 93\%$), SBS-RF-5 ($ACC = 93\%$) i RFE-KNN-3 ($ACC = 93\%$). Niewiele mniejsza była dokładność pozostałych modeli, zmieniała się ona w granicach 90÷92%. Na podstawie wyników testowania, wykreślono krzywe ROC przedstawione na rysunku 6.16. Wartość parametru AUC wymienionych wcześniej modeli oraz modelu SFS-GRAD-3 wahała się w zakresie 0,975÷0,986. W pozostałych przypadkach, ten zakres to 0,940÷0,973.

Tabela 6.6. Najlepsze modele klasyfikacyjne dla dużego zbioru danych i ograniczonego zestawu cech. W kolumnie "Różnica" podano wartość, o jaką zmniejszyła się dokładność walidacji, w porównaniu z modelem zbudowanym za pomocą tej samej metody klasyfikacji i pełnego zestawu cech

Symbol modelu	Metoda selekcji cech	Metoda klasyfikacji	Liczba cech	Dokładność walidacji	Różnica
F-SVM-5	FISHER	SVM	5	0,91	0,05
A-SVM-5	ANOVA	SVM	5	0,95	0,01
A-MLP-5	ANOVA	MLP	5	0,95	0,01
SFS-LDA-4	SFS	LDA	4	0,93	0,03
SFS-GRAD-3	SFS	GRAD	3	0,94	0,02
SBS-RF-5	SBS	RF	5	0,95	0,01
RFE-KNN-3	RFE	KNN	3	0,95	0,01
W-ADA-3	W	ADA	3	0,91	0,04
W-LR-3	W	LR	3	0,92	0,04
W-LGBM-3	W	LGBM	3	0,94	0,02



Rys. 6.15. Wyniki walidacji i testowania najlepszych modeli klasyfikacyjnych dla dużego zbioru danych i ograniczonego zestawu cech



Rys. 6.16. Krzywe ROC wraz z parametrami AUC dla najlepszych modeli klasyfikacyjnych zbudowanych z wykorzystaniem ograniczonego zestawu cech dla dużego zbioru danych

6.5. Zastosowanie metod uczenia tradycyjnego oraz małego zbioru danych

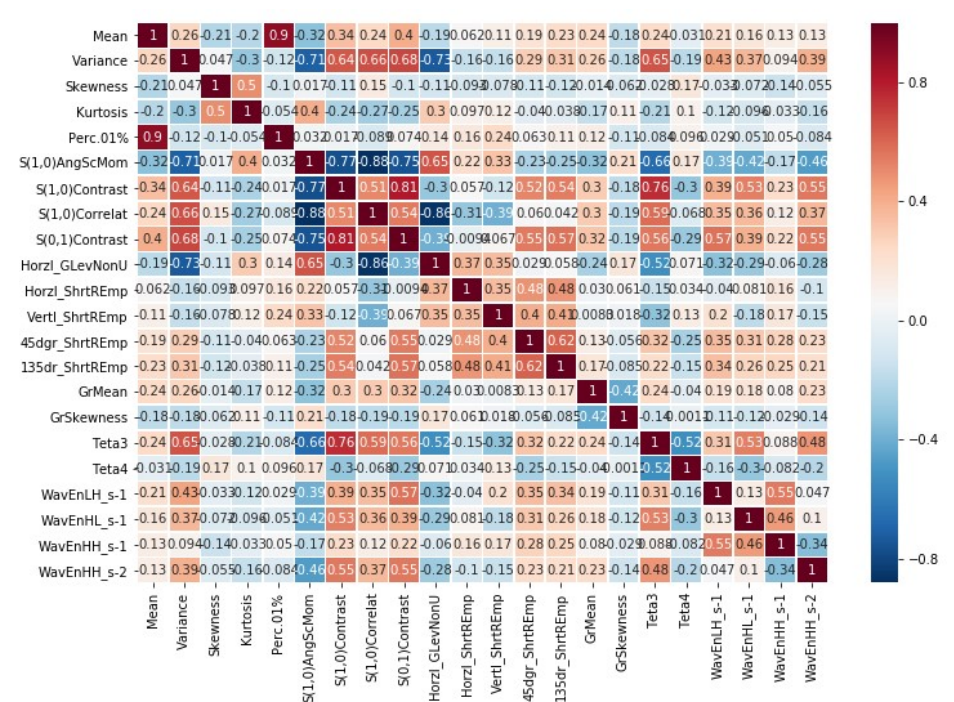
Mały zbiór danych zawierał 600 obserwacji, przy czym połowa z nich należała do wariantu nr 5, a połowa do wariantu nr 4. Zbiór ten został losowo podzielony na część treningową i testową w taki sposób, że pierwsza część stanowiła 70%, a druga 30% zbioru pełnego. Należy podkreślić, że zbiory treningowy i testowy nie były wydzieloną częścią podzbiorów zbioru pełnego, lecz powstały

wskutek zupełnie nowego losowania. Szczegółowe dane dotyczące liczebności podzbiorów zamieszczono w tabeli 6.7.

Tabela 6.7. Liczebność podzbiorów otrzymanych w wyniku podziału małego zbioru danych

Zbiór danych	Liczba obserwacji	Część zbioru pełnego	Wariant nr 5	Wariant nr 4
Pełny	600	1,0	300	300
Treningowy	420	0,7	210	210
Testowy	180	0,3	90	90

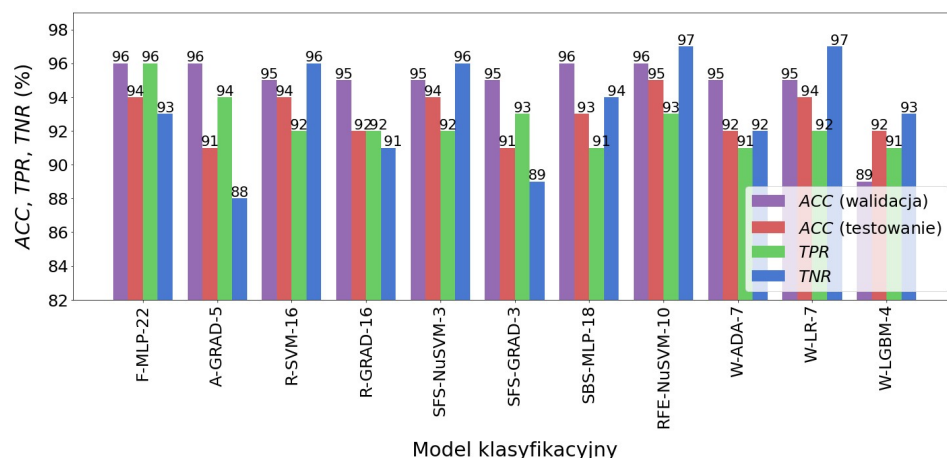
W wyniku oczyszczania danych, liczba cech została zmniejszona z 290 do 22 (rys. 6.17). Liczebność cech występujących na listach rankingowych zbudowanych przez poszczególne metody selekcji była następująca: metody filtra i opakowujące: 22, metody wbudowane: regresja logistyczna i AdaBoost – 9, LGBM – 8. Efekty trenowania i wyboru najlepszych modeli zamieszczono w tabeli 6.8, natomiast rysunek 6.18 prezentuje wyniki ich walidacji i testowania, a na rysunku 6.19 pokazano wykresy krzywych ROC.



Rys. 6.17. Macierz korelacji pomiędzy cechami stanowiącymi wynik oczyszczania małego zbioru danych

Tabela 6.8. Najlepsze modele klasyfikacyjne dla małego zbioru danych i pełnego zestawu cech.
Litera "W" występująca w symbolu modelu oznacza metodę wbudowaną

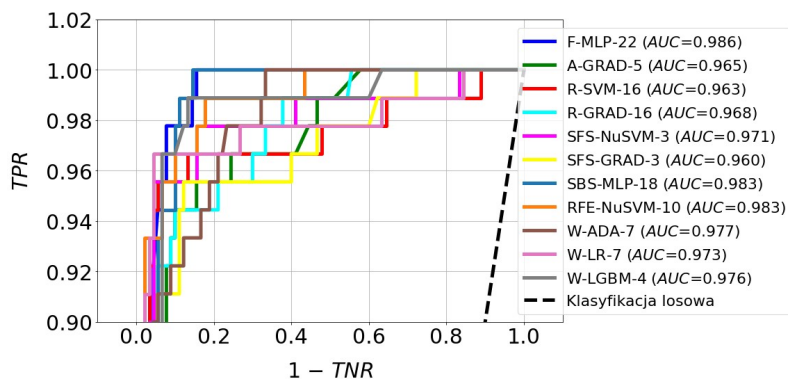
Symbol modelu	Metoda klasyfikacji	Metoda selekcji cech	Liczba cech
F-MLP-22	MLP	FISHER	22
A-GRAD-5	GRAD	ANOVA	5
R-SVM-16	SVM	RELIEF	16
R-GRAD-16	GRAD	RELIEF	16
SFS-NuSVM-3	NuSVM	SFS	3
SFS-GRAD-3	GRAD	SFS	3
SBS-MLP-18	MLP	SBS	18
RFE-NuSVM-10	NuSVM	RFE	10
W-ADA-7	W	ADA	7
W-LR-7	W	LR	7
W-LGBM-4	W	LGBM	4



Rys. 6.18. Wyniki walidacji i testowania najlepszych modeli klasyfikacyjnych dla małego zbioru danych i pełnego zestawu cech

Następujące modele wyróżniają się pod względem dokładności klasyfikacji dla zbioru testowego: F-MLP-22 ($ACC = 94\%$), R-SVM-16 ($ACC = 94\%$), SFS-NuSVM-3 ($ACC = 94\%$), RFE-NuSVM-10 ($ACC = 95\%$) i W-LR-7 ($ACC = 94\%$). Należy zauważyć, że model SFS-NuSVM-3 uzyskał tak wysoką dokładność (94%) przy wykorzystaniu zaledwie trzech cech. Na uwagę zasługuje również fakt, iż model F-MLP-22 pozwolił uzyskać wysoką dokładność rozpoznawania wariantu nr 4 (nadmiar powietrza). Indeks TPR wynosił w tym przypadku 96% . Natomiast modele RFE-NuSVM-10 i W-LR-7 uzyskały bardzo dobrą dokładność rozpoznawania wariantu nr 5 (normalne warunki spalania). Wartość indeksu TNR była dla nich równa 97% . Na podstawie przebiegu krzy-

wych ROC, przedstawionych na rysunku 6.19, można wskazać 3 modele, których skuteczność jest wyraźnie lepsza od pozostałych. Należą do nich: F-MLP-22 ($AUC = 0,986$), SBS-MLP-18 ($AUC = 0,983$) i RFE-NuSVM-10 ($AUC = 0,983$).



Rys. 6.19. Krzywe ROC wraz z parametrami AUC dla najlepszych modeli klasyfikacyjnych zbudowanych z wykorzystaniem pełnego zestawu cech dla małego zbioru danych

6.6. Zastosowanie metod uczenia głębokiego oraz dużego zbioru danych

Zbiór danych dla metod uczenia głębokiego przygotowano w taki sposób, że zbiór treningowy, który był wcześniej wykorzystany przez metody uczenia tradycyjnego, podzielono losowo na część treningową i walidacyjną. Z kolei, zbiór testowy wykorzystano w takiej samej postaci, jak w przypadku metod uczenia tradycyjnego. Dzięki takiemu podejściu, zarówno metody uczenia tradycyjnego, jak i głębokiego korzystały z tych samych obserwacji na etapie uczenia i walidacji modeli oraz podczas ich ewaluacji. Ten sposób podziału pozwolił ocenić skuteczność obu rodzajów metod dla tych samych danych. Szczegółowe informacje na temat liczebności poszczególnych podzbiorów zamieszczono w tabeli 6.9.

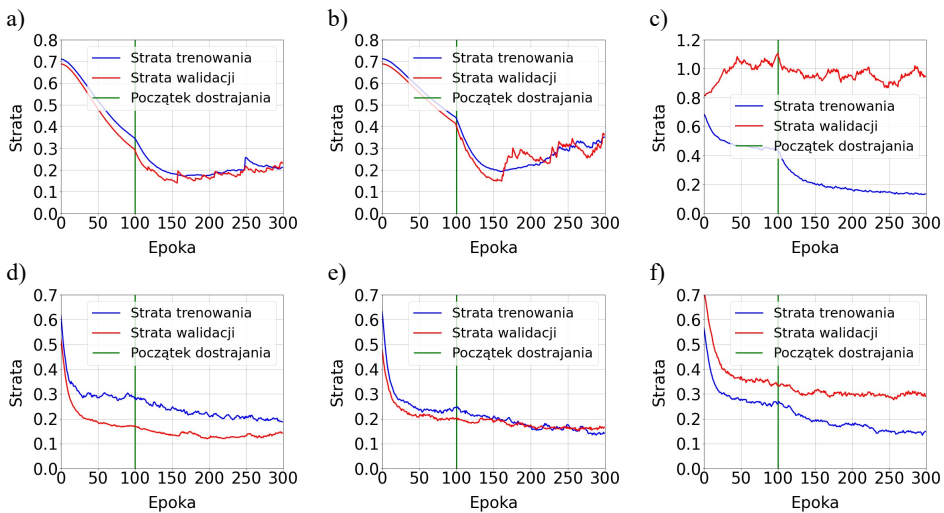
Tabela 6.9. Liczebność podzbiorów otrzymanych w wyniku podziału dużego zbioru danych

Zbiór danych	Liczba obserwacji	Część zbioru pełnego	Wariant nr 5	Wariant nr 4
Pełny	3000	1,0	1500	1500
Treningowy	1200	0,4	600	600
Walidacyjny	900	0,3	450	450
Testowy	900	0,3	450	450

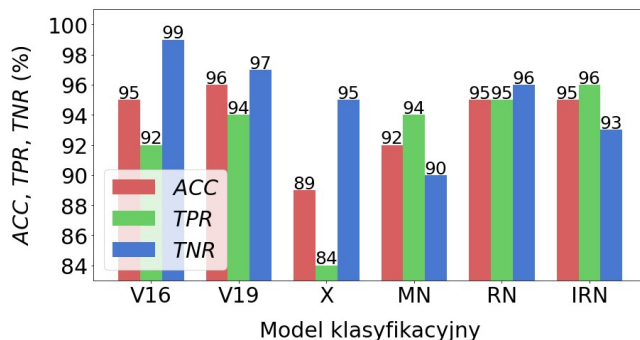
6.6.1. Wykorzystanie obrazów kolorowych

Na rysunku 6.20 przedstawiono krzywe straty trenowania i walidacji. Na ich podstawie widać, że model Xception wyróżnił się słabszym dopasowaniem do danych, niż modele pozostałe. Można to stwierdzić na podstawie największej różnicy pomiędzy wartością straty trenowania i walidacji. Przyczyną tego faktu mogła być za mała pojemność modelu. W pozostałych przypadkach, dopasowanie do danych było lepsze, co znalazło odzwierciedlenie w wynikach ewaluacji. Ponadto, dla dwóch modeli można zauważyć cechy przetrenowania wynikającego ze zbyt długiego czasu uczenia. Dla modelu VGG16 przetrenowanie jest małe, natomiast dla VGG19 jest ono zdecydowanie większe. W obu przypadkach straty zaczynają narastać powyżej 150-ciu epok, dlatego granicę tę przyjęto jako czas ponownego trenowania obu modeli.

Na rysunku 6.21 zaprezentowano wyniki ewaluacji. Największą dokładność klasyfikacji dla zbioru testowego uzyskały modele: VGG16 ($ACC = 95\%$), VGG19 ($ACC = 96\%$), ResNet50 ($ACC = 95\%$) i InceptionResNetV2 ($ACC = 95\%$). Dokładność testowania modelu MobileNetV2 wyniosła 92% , natomiast zdecydowanie najgorzej wypadł model Xception ($ACC = 89\%$), co wynikało z najslabszego dopasowania do danych (rys. 6.20c).

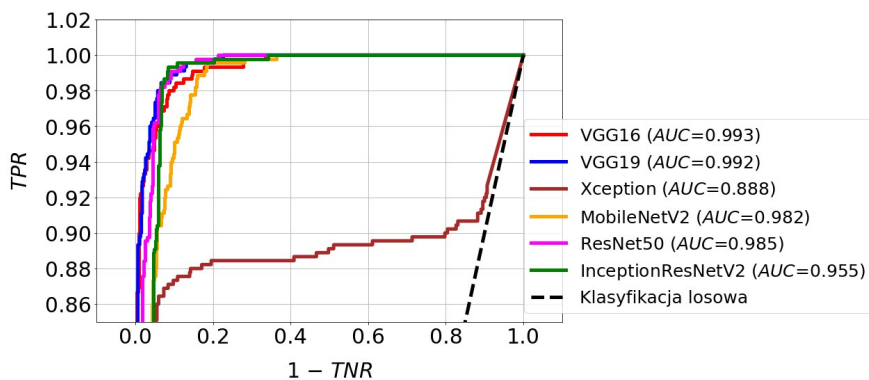


Rys. 6.20. Wykresy straty trenowania i walidacji modeli dla dużego zbioru danych i obrazów kolorowych: a) VGG16; b) VGG19; c) Xception; d) MobileNetV2; e) ResNet50; f) InceptionResNetV2



Rys. 6.21. Wyniki testowania modeli dla dużego zbioru danych i obrazów kolorowych. Zastosowane symbole: V16 – VGG16, V19 – VGG19, X – Xception, MN – MobileNetV2, RN – ResNet50, IRN – InceptionResNetV2

Należy podkreślić wysoką dokładność rozpoznawania wariantu nr 4 (nadmiar powietrza) przez model InceptionResNetV2 ($TPR = 96\%$) oraz bardzo dobrą skuteczność rozpoznawania normalnych warunków spalania (wariant nr 5) przez model VGG16 ($TNR = 99\%$). Krzywe ROC przedstawione na rysunku 6.22 potwierdzają wysoką skuteczność modeli VGG16, VGG19, ResNet50 oraz InceptionResNetV2. Parametr AUC przyjmuje dla nich wartość z zakresu $0,955 \div 0,993$. Nieco słabiej wypada na ich tle model MobileNetV2 ($AUC = 0,982$), a zdecydowanie najgorsza skuteczność odpowiada krzywej ROC należącej do modelu Xception ($AUC = 0,888$).

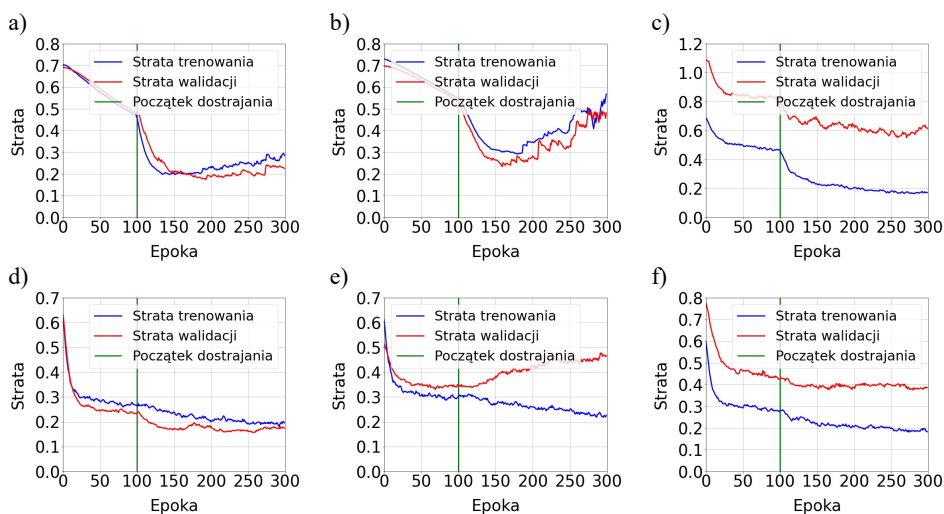


Rys. 6.22. Krzywe ROC wraz z parametrami AUC dla modeli zbudowanych z wykorzystaniem dużego zbioru danych i obrazów kolorowych

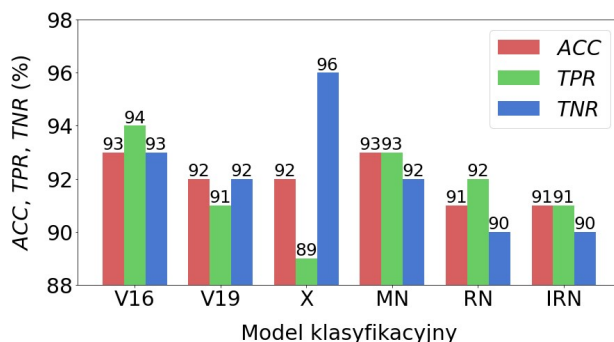
6.6.2. Wykorzystanie obrazów monochromatycznych

Analizując przebieg krzywych straty trenowania i walidacji modeli zbudowanych na podstawie obrazów monochromatycznych (rys. 6.23), można ocenić, iż ich własności są podobne do modeli zbudowanych na podstawie obrazów kolorowych. Także i w tym przypadku Xception jest najslabiej dopasowany do danych, a VGG16 i VGG19 wykazują oznaki przetrenowania (wyuczono je ponownie przez okres 150-ciu epok). Dodatkowo, efekt przetrenowania można zaobserwować również dla modelu ResNet50. W tym przypadku, punkt przebiecia krzywej straty walidacji występuje w okolicy epoki nr 100, zatem taki czas uczenia przyjęto w ponownym trenowaniu tego modelu. Z kolei, bardzo dobre dopasowanie prezentują krzywe strat należące do modelu MobileNetV2. Odstęp pomiędzy tymi krzywymi systematycznie zmniejsza się podczas uczenia i na końcu tego procesu osiąga wartość najmniejszą spośród wszystkich modeli.

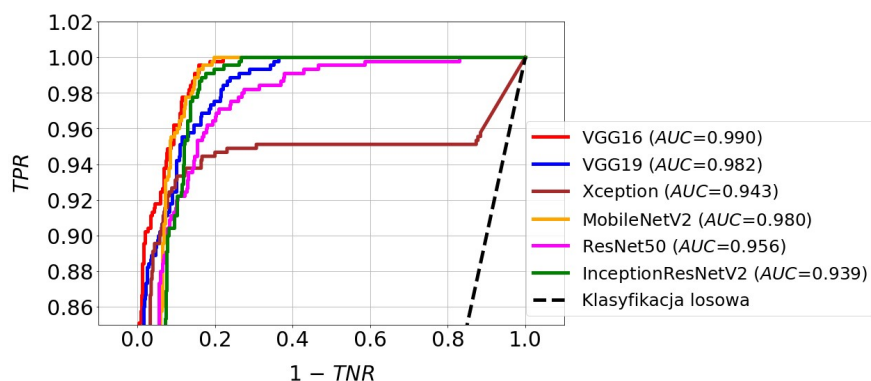
Wszystkie modele zbudowane w oparciu o obrazy monochromatyczne uzyskały zbliżone wyniki testowania (rys. 6.24). Wśród nich, nieznacznie skuteczniejsze od innych okazały się modele VGG16 ($ACC = 93\%$) i MobileNetV2 ($ACC = 93\%$). Ten fakt potwierdza przebieg krzywych ROC na rysunku 6.25, dla których wartość parametru AUC wyniosła 0,990 dla modelu VGG16 i 0,980 dla MobileNetV2. Z kolei, krzywa ROC dla modelu Xception ($AUC = 0,943$) odzwierciedla jego najslabsze dopasowanie do danych.



Rys. 6.23. Wykresy straty trenowania i walidacji modeli dla dużego zbioru danych i obrazów monochromatycznych: a) VGG16; b) VGG19; c) Xception; d) MobileNetV2; e) ResNet50; f) InceptionResNetV2



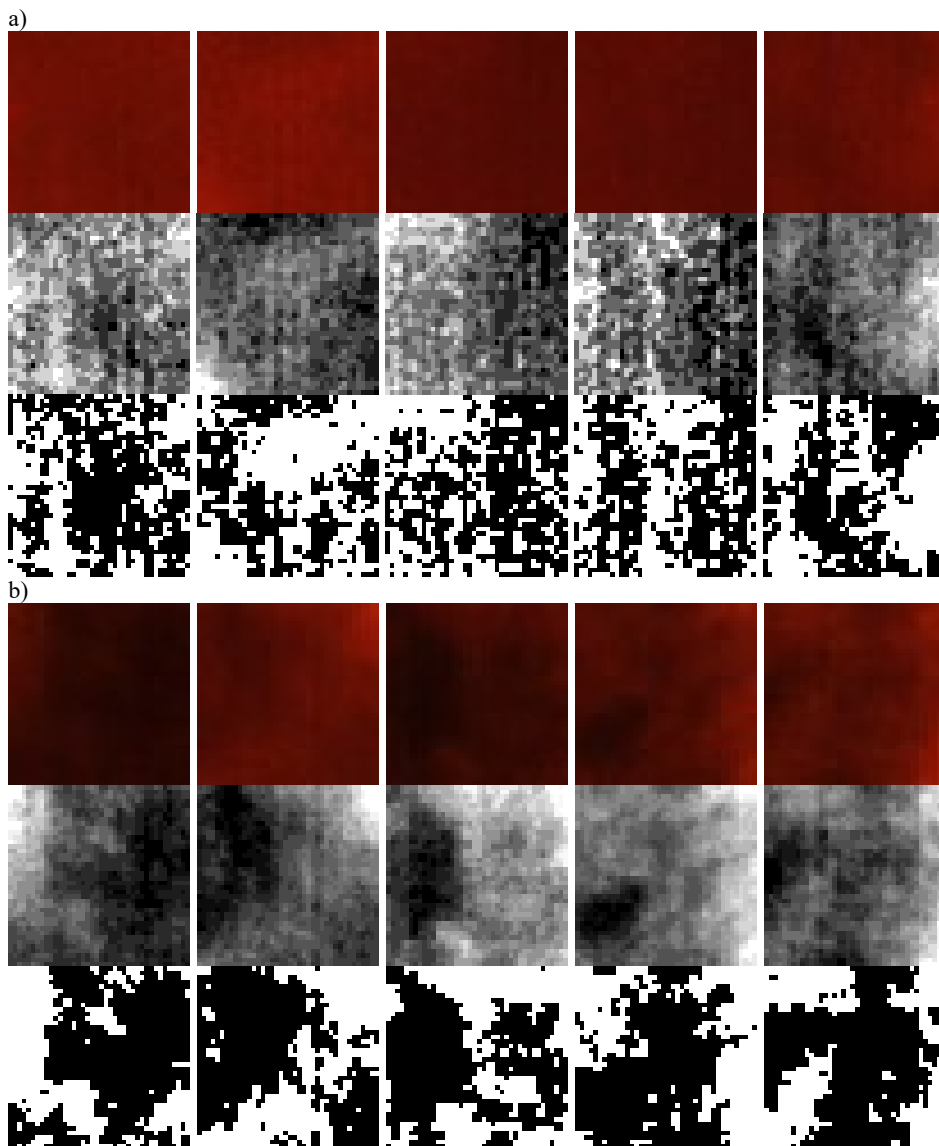
Rys. 6.24. Wyniki testowania modeli dla dużego zbioru danych i obrazów monochromatycznych. Zastosowane symbole: V16 – VGG16, V19 – VGG19, X – Xception, MN – MobileNetV2, RN – ResNet50, IRN – InceptionResNetV2



Rys. 6.25. Krzywe ROC wraz z parametrami AUC dla modeli zbudowanych z wykorzystaniem dużego zbioru danych i obrazów monochromatycznych

6.6.3. Wykorzystanie obrazów binarnych

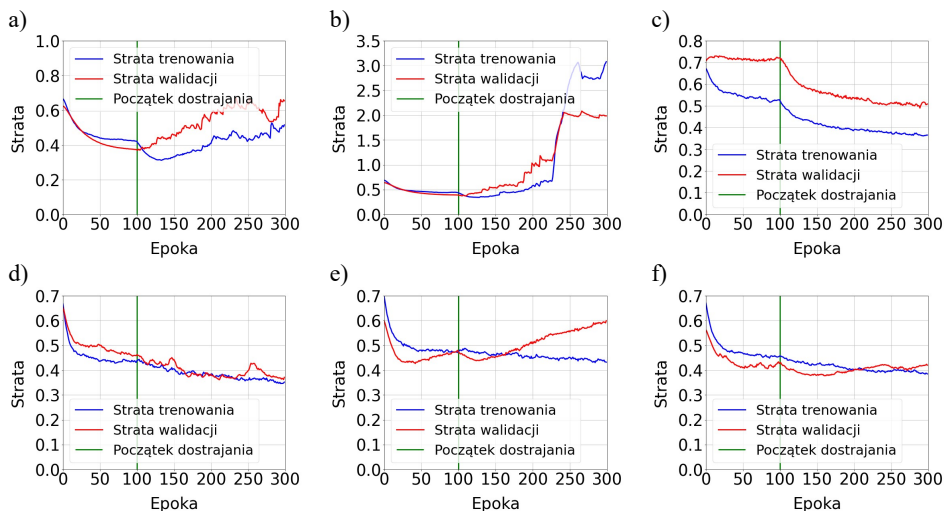
Obrazy monochromatyczne poddano operacji normalizacji histogramu. Dzięki poprawie kontrastu, stworzono lepsze warunki do segmentacji obrazu. Przetestowano kilka metod realizujących tę operację, należały do nich: lokalne i globalne progowanie Otsu, lokalne progowanie Bradleya, metoda zbiorów poziomowych, progowanie adaptacyjne oraz metoda aktywnego konturu. Podstawowym kryterium jakości segmentacji było wierne odtworzenie granic jaśniejszych i ciemniejszych obszarów widocznych na analizowanych obrazach. Najlepsze efekty osiągnięto za pomocą progowania lokalnego metodą Bradleya. W związku z tym, metodę tę zastosowano do segmentacji obrazów monochromatycznych. Przykładowe wyniki tej operacji zaprezentowano na rysunku 6.26.



Rys. 6.26. Przykładowe obrazy wykorzystane w badaniach: a) wariant nr 5; b) wariant nr 4. Górne wiersze przedstawiają wydzielone obszary zainteresowania (40×40 pikseli). Wiersze środkowe prezentują obrazy monochromatyczne po normalizacji histogramu, natomiast w dolnych wierszach pokazano wynik progowania lokalnego metodą Bradleya

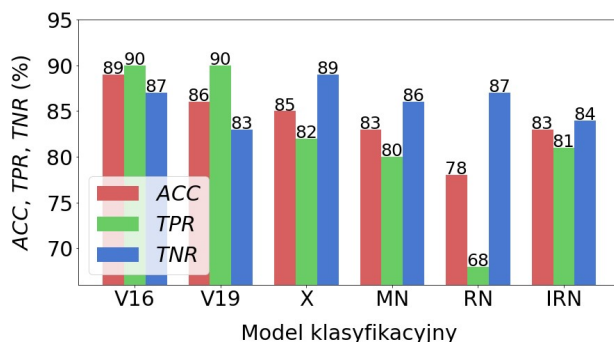
Przebiegi krzywych straty trenowania i walidacji przedstawione na rysunku 6.27 świadczą o dobrym dopasowaniu do danych modeli MobileNetV2 oraz InceptionResNetV2. Tak samo, jak dla obrazów monochromatycznych, modele

VGG16, VGG19 i ResNet50 również wykazują oznaki przetrenowania. Wszystkie trzy modele wytrenowano ponownie, przy czym przyjęto liczbę epok uczenia równą 125.

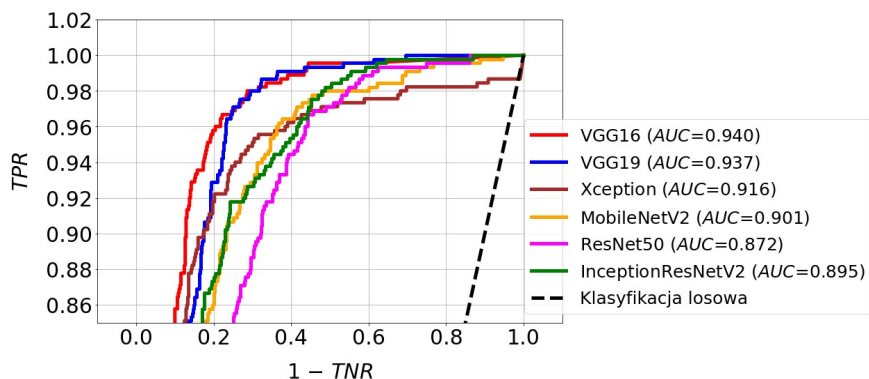


Rys. 6.27. Wykresy straty trenowania i walidacji modeli dla dużego zbioru danych i obrazów binarnych: a) VGG16; b) VGG19; c) Xception; d) MobileNetV2; e) ResNet50; f) InceptionResNetV2

Wyniki ewaluacji przedstawione na rysunku 6.28 pokazują, że modele VGG16 i VGG19 uzyskały największą dokładność testowania. Wyniosła ona 89% dla VGG16 i 86% dla VGG19. Należy także podkreślić, że w obu przypadkach osiągnięto największą dokładność rozpoznawania wariantu nr 4 (nadmiar powietrza). Dla obu modeli wartość indeksu TPR wyniosła 90%. Skuteczność stwierdzona na podstawie ewaluacji znajduje także potwierdzenie w przebiegu krzywych ROC oraz wielkościach pól pod tymi krzywymi (parametr AUC) przedstawionych na rysunku 6.29. Wartości parametrów AUC są największe dla modelu VGG16 (0,940) oraz VGG19 (0,937). Z kolei, dla najmniej skutecznego modelu ResNet50 ($ACC = 78\%$), obszar pod krzywą ROC był najmniejszy i wyniósł 0,872.



Rys. 6.28. Wyniki testowania modeli dla dużego zbioru danych i obrazów binarnych. Zastosowane symbole: V16 – VGG16, V19 – VGG19, X – Xception, MN – MobileNetV2, RN – ResNet50, IRN – InceptionResNetV2



Rys. 6.29. Krzywe ROC wraz z parametrami AUC dla modeli zbudowanych z wykorzystaniem dużego zbioru danych i obrazów binarnych

6.7. Zastosowanie metod uczenia głębokiego oraz małego zbioru danych

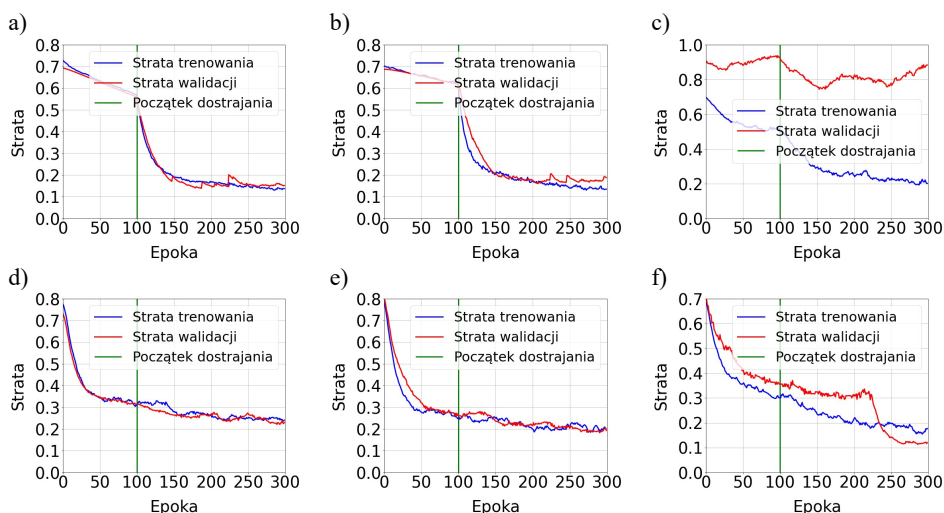
Zasada podziału małego zbioru danych dla metod uczenia głębokiego była taka sama, jak dla zbioru dużego. Także w sposób losowy podzielono zbiór treningowy, wykorzystany wcześniej przez metody uczenia tradycyjnego, na część treningową i uczącą, natomiast bez żadnych zmian wykorzystano ten sam zbiór testowy. Informację dotyczącą liczebności podzbiorów zamieszczono w tabeli 6.10. Należy w tym miejscu przypomnieć, iż w scenariuszu badań wykorzystującym mały zbiór danych zastosowano tylko obrazy kolorowe.

Tabela 6.10. Liczebność podzbiorów otrzymanych w wyniku podziału małego zbioru danych

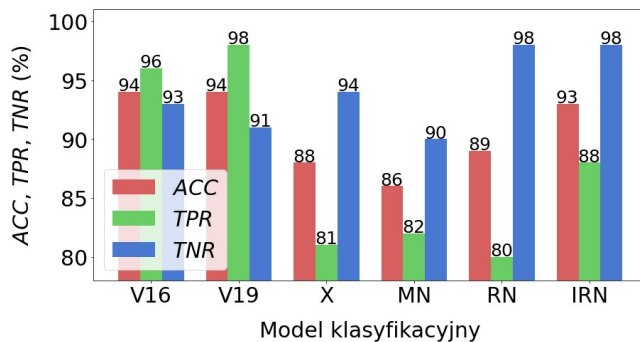
Zbiór danych	Liczba obserwacji	Część zbioru pełnego	Wariant nr 5	Wariant nr 4
Pełny	600	1,0	300	300
Treningowy	240	0,4	120	120
Walidacyjny	180	0,3	90	90
Testowy	180	0,3	90	90

Na rysunku 6.30 przedstawiono wykresy krzywych straty trenowania i walidacji. Ich przebieg świadczy o dobrym dopasowaniu do danych modeli VGG16, VGG19, MobileNetV2 i ResNet50. Nieco gorsze dopasowanie charakteryzuje model InceptionResNetV2, natomiast najgorzej na tym tle wypadł model Xception, dla którego obserwujemy największy odstęp pomiędzy krzywymi trenowania i walidacji. Dla żadnego z modeli nie wystąpiły oznaki przetrenowania.

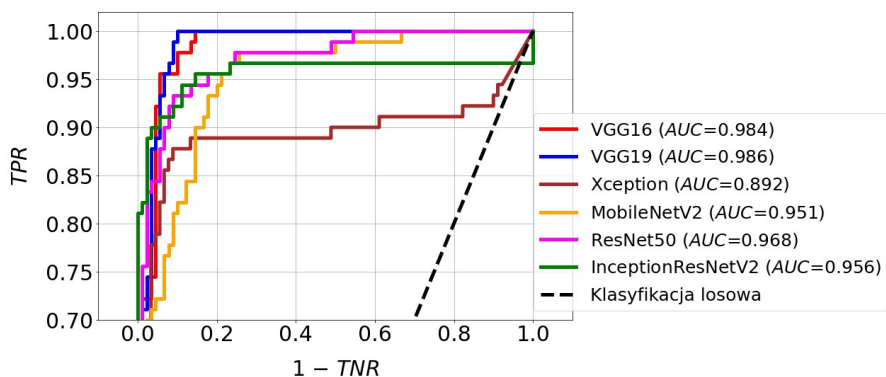
Wyniki ewaluacji zaprezentowano na rysunku 6.31. Zdecydowanie najsukcesowniej okazały się modele: VGG16 i VGG19, dla których wskaźnik *ACC* osiągnął wartość 94%. Modele te wyróżniły się także wysoką dokładnością rozpoznawania wariantu nr 4 (*TPR*), która dla VGG16 wyniosła 96%, a dla VGG19 była równa 98%. Krzywe ROC przedstawione na rysunku 6.32 potwierdzają wysoką skuteczność modeli VGG16 i VGG19. Parametr *AUC* przyjmuje dla nich wartość odpowiednio: 0,984 oraz 0,986. Dla modeli pozostałych obszar pod krzywą ROC jest znacznie mniejszy i zmienia się w zakresie 0,892÷0,968.



Rys. 6.30. Wykresy straty trenowania i walidacji modeli dla małego zbioru danych i obrazów kolorowych: a) VGG16; b) VGG19; c) Xception; d) MobileNetV2; e) ResNet50; f) InceptionResNetV2



Rys. 6.31. Wyniki testowania modeli dla małego zbioru danych i obrazów kolorowych. Zastosowane symbole: V16 – VGG16, V19 – VGG19, X – Xception, MN – MobileNetV2, RN – ResNet50, IRN – InceptionResNetV2

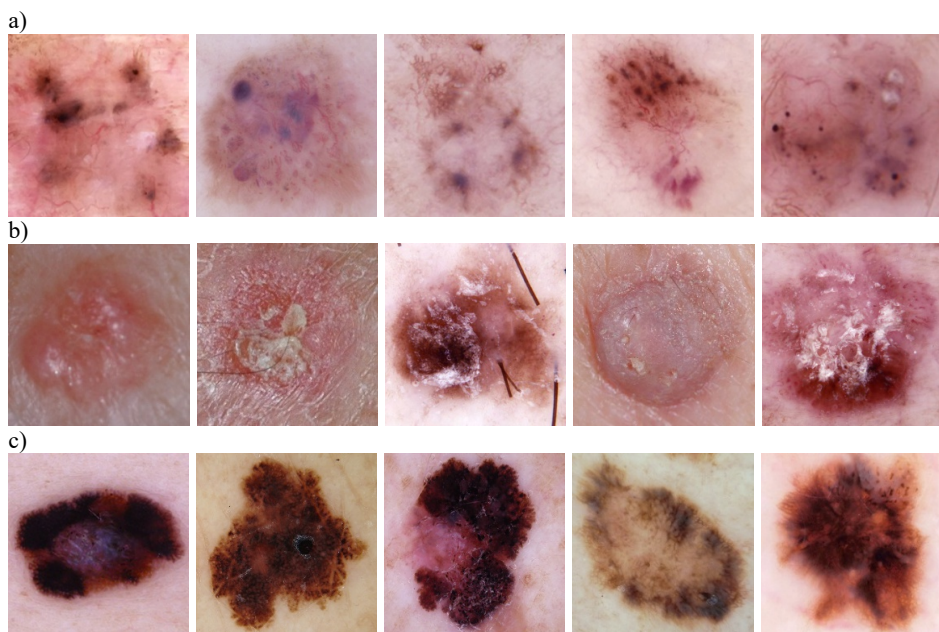


Rys. 6.32. Krzywe ROC wraz z parametrami AUC dla modeli zbudowanych z wykorzystaniem małego zbioru danych i obrazów kolorowych

7. Diagnostyka zmian skórnych

7.1. Kontekst badań

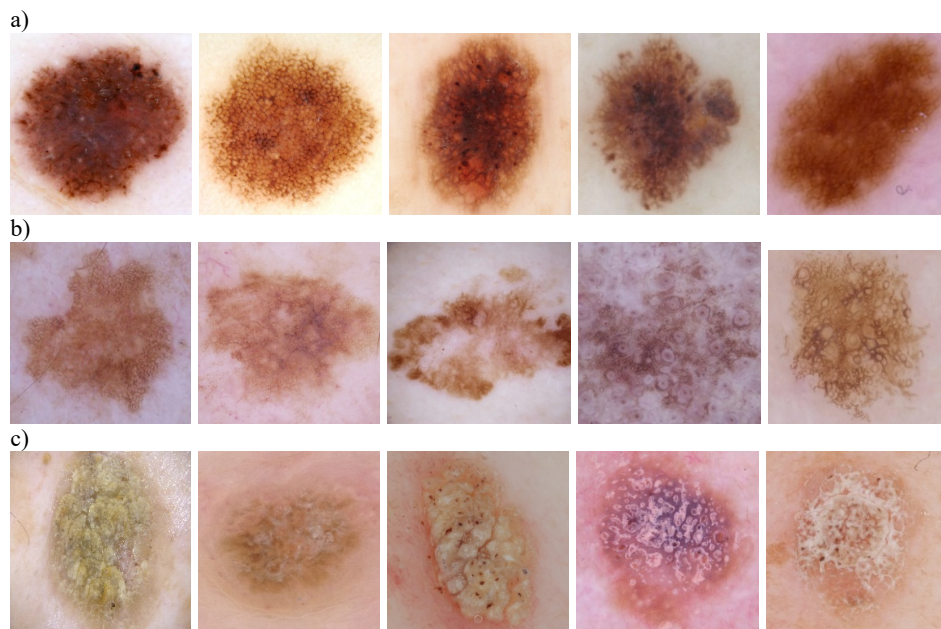
Najlichnieszą grupą nowotworów złośliwych skóry są raki skóry. Jest to niejednolita grupa nowotworów pochodzenia nabłonkowego. Jest wiele rodzajów raka skóry, jednak najczęściej spotykane to rak podstawnkomórkowy, rak kolczystokomórkowy oraz czerniak złośliwy (rys. 7.1).



Rys. 7.1. Najczęściej spotykane raki skóry: a) rak podstawnkomórkowy; b) rak kolczystokomórkowy; c) czerniak złośliwy

Najbardziej niebezpiecznym z nich jest czerniak, który powstaje w wyniku szybkiego wzrostu komórek barwnikowych skóry (melanocytów). Jest to spowodowane głównie ekspozycją na promieniowanie ultrafioletowe. Czerniak może pojawić się na skórze jako nowa zmiana, może też rozwinąć się z już posiadanej zmiany niezłośliwej (rys. 7.2). Do zmian, które najczęściej mogą ulegać zezłośliwieniu i transformacji w kierunku czerniaka, należą znamiona barwnikowe. Szczególnie niebezpieczne są znamiona dysplastyczne (atypowe). Charakteryzują się one asymetrycznym kształtem, rozmiarem powyżej 5 mm, nieregularnym brzegiem oraz różnorodnością kolorów. Znamiona barwnikowe rozwijają się na skutek niewłaściwego rozwoju skóry w życiu płodowym. Mogą po-

jawić się bezpośrednio po narodzinach, w okresie młodzieńczym lub w wieku dojrzałym. Ich obecność nie wskazuje jednoznacznie na występowanie choroby, istnieje jednak zagrożenie, że może być ona uśpiona nawet na długie lata. Człowiek posiada średnio około 20 znamion tego typu. Istotny wpływ na ryzyko wystąpienia czerniaka ma liczba znamion występujących na kończynach [194].



Rys. 7.2. Najczęściej spotykane, niezłośliwe zmiany skórne: a) znamię barwnikowe (melanocytowe); b) łagodne rogowacenie barwnikowe; c) brodawka łojotokowa

Kluczowe znaczenie ma wykrycie czerniaka we wczesnym stadium, ponieważ w stadiach późniejszych coraz bardziej wnika on w głąb skóry oraz rozprzestrzenia się do innych części ciała. W związku z tym, odpowiednio wczesne wykrycie tej choroby może znacznie obniżyć śmiertelność dotkniętych nią pacjentów. W ciągu ostatnich 10 lat, zachorowalność na czerniaka na całym świecie zwiększyła się dwukrotnie – najwyższa występuje w Australii, gdzie choruje 29 na 100 tys. osób. W Polsce, co roku nowotwór ten diagnozuje się u ok. 3,5 tys. osób. Czerniak stanowi zaledwie 7% wszystkich nowotworów złośliwych skóry, jest jednak przyczyną blisko 90% zgonów pacjentów chorych na raka skóry. W Polsce, rocznie z jego powodu umiera ok. 1,1 tys. osób. Pomimo tego, iż pacjenci jako pierwsi zgłaszają podejrzenie występowania czerniaka [195], to dokładność diagnostyczna badań wzrokowych wykonywanych przez specjalistów wynosi ok. 60% [196]. Wynika to z faktu, iż cechy charakterystyczne zarówno łagodnych, jak i złośliwych zmian barwnikowych skóry mogą być bardzo

podobne. W efekcie, różnicowanie takich zmian jest zadaniem bardzo trudnym, a dokładność diagnozy zależy w głównej mierze od doświadczenia dermatologa [197].

Jako alternatywę dla tradycyjnych algorytmów uczenia maszynowego, od pewnego już czasu proponowane są metody oparte na spłotowych sieciach neuronowych. Pozwalają one zautomatyzować proces wykrywania, segmentacji i klasyfikacji obiektów, w odniesieniu do najbardziej złożonych obrazów medycznych [198, 199, 200, 201]. Sieci tego typu mogą wydobyć głębokie cechy wysokiego poziomu bezpośrednio z surowych danych wejściowych, osiągając przy tym znacznie lepsze wyniki klasyfikacji, niż metody tradycyjne [202]. Jak zaznaczono wcześniej, większość dotychczasowych badań oparta była na bazach obrazów niedostępnych dla szerszej społeczności. Pewną próbą poprawy tej sytuacji było utworzenie publicznego archiwum PH² [203, 204]. Niestety, w zbiorze tym zgromadzono tylko 200 obrazów, co było jego zasadniczym mankamentem.

Sytuacja poprawiła się dopiero wówczas, gdy organizacja ISIC (ang. *International Skin Imaging Collaboration*) zaczęła gromadzić na dużą skalę publicznie dostępne obrazy dermatoskopowe. Obecnie, w zbiorze danych znajduje się ponad 20 000 obrazów pochodzących z wiodących, międzynarodowych ośrodków klinicznych. Zbiór danych ISIC stanowił podstawę do organizacji w 2016 r. pierwszego publicznego konkursu w zakresie analizy obrazów dermatoskopowych [205, 206]. Ideą tego przedsięwzięcia było dostarczenie szerszej społeczności zbioru obrazów, w celu wsparcia rozwoju algorytmów automatycznej diagnostyki czerniaka w ramach 3 zadań: segmentacji, wykrywania i ekstrakcji cech dermatoskopowych oraz klasyfikacji. Podobne cele przyświecały kolejnym edycjom tego konkursu, przeprowadzonym w latach 2017-2018 [207, 208, 209]. W roku 2019 zorganizowano konkurs, którego głównym zadaniem była klasyfikacja obrazów dermatoskopowych należących do dziewięciu różnych kategorii diagnostycznych.

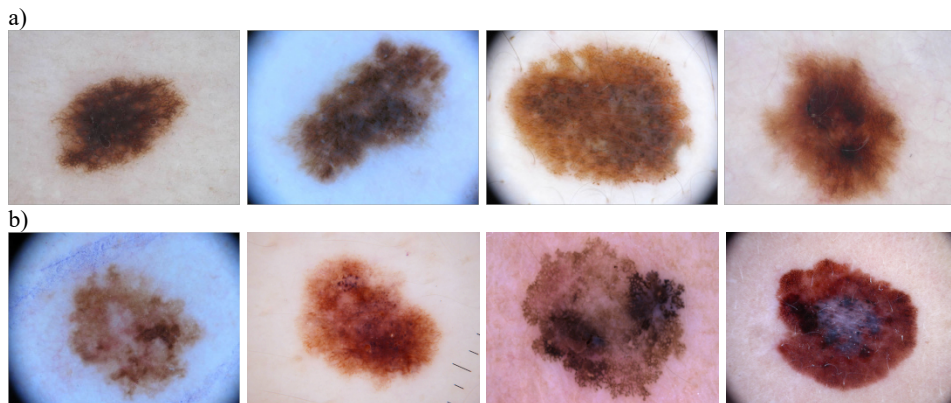
Konkursy organizowane przez ISIC, dzięki swej popularności oraz dużej liczbie udostępnionych obrazów, wyznaczają główne kierunki rozwoju metod diagnostyki zmian skórnych, wśród których główną rolę odgrywają algorytmy uczenia głębokiego. Od momentu udostępnienia szerszej społeczności bogatego archiwum ISIC, obserwuje się wzrost zainteresowania metodami uczenia głębokiego, w kontekście diagnostyki zmian skórnych. Bardzo popularne w tym zastosowaniu są spłotowe sieci neuronowe. Istnieje wiele publikacji opisujących zastosowanie tych sieci do rozwiązywania różnych problemów z zakresu automatycznej diagnostyki chorób skóry. Na przykład, w [210] opisano efekt zastosowania cech wyłonionych za pomocą wstępnie wytrenowanej sieci spłotowej, w połączeniu z cechami uzyskanymi innymi metodami. Porównanie efektów identyfikacji wzorców w obrazie dermatoskopowym, z wynikami uzyskanymi tradycyjnymi metodami uczenia maszynowego zamieszczono w pracy [211]. W pozycjach [212, 213] podano wyniki zastosowania CNN do klasyfikacji

5 i 10 klasowej, dla różnych rozdzielczości obrazów. W innych pracach opisano wyniki zastosowania sieci rezydualnych do segmentacji i klasyfikacji zmian chorobowych [214], a także porównano, m.in. różne strategie wstępnego trenowania i dokładnego dostrajania [215] oraz wyniki działania sieci wytrenowanej na podstawie 129 tys. obrazów z diagnozami postawionymi przez 29 dermatologów [216]. Generalnie, modele klasyfikacyjne zbudowane z wykorzystaniem metod uczenia głębokiego pozwalają uzyskać lepsze efekty, niż modele uczenia tradycyjnego. W zależności od konkretnego zadania, dokładność możliwa do uzyskania w ich przypadku, często osiąga poziom 90÷95% i wyższy.

7.2. Materiał do badań

W Internecie dostępnych jest wiele zbiorów obrazów dermatoskopowych. Część z nich to typowe atlasy chorób skóry, pełniące głównie funkcję edukacyjną [217, 218, 219]. Inne, to bazy danych udostępnione do celów badawczych [220, 221]. Jednak zasadniczą wadą tych zasobów jest ich mała objętość. Na przykład, baza MED-NODE zawiera 100 obrazów czerniaka i 70 obrazów znamion. Z kolei, wspomnianą w poprzednim rozdziale, bazę PH², tworzy 200 obrazów, wśród których jest 40 obrazów czerniaka i 160 obrazów innych zmian skórnych. Obecnie, największą i najbardziej reprezentatywną bazą obrazów dermatoskopowych jest baza ISIC, zawierająca ponad 20 000 obrazów różnych zmian skórnych pogrupowanych na różne kategorie [222].

W badaniach, których wyniki przedstawiono w niniejszej monografii, wykorzystano obrazy zamieszczone w archiwum ISIC. Obecnie (19.08.2020), oprócz innych kategorii, baza ta zawiera 18 566 zdjęć znamion barwnikowych oraz 2 169 zdjęć czerniaka. Z bazy tej wybrano 1 200 zdjęć, przy czym 600 przedstawiało znamiona oraz 600 czerniaki skóry. Część zdjęć zamieszczonych w archiwum zawiera różne artefakty (włosy, pęcherzyki powietrza, linijki, dodatkowe oznaczenia nanoszone przez personel), które zakłócają proces budowy modelu klasyfikacyjnego. Zwrócono uwagę, aby obrazy wykorzystane w badaniach były pozbawione wyżej wymienionych elementów. Na rysunku 7.3. przedstawiono przykładowe obrazy znamion i czerniaków pobrane z bazy ISIC.



Rys. 7.3. Przykładowe obrazy pobrane z bazy ISIC: a) znamiona; b) czerniaki

7.3. Wstępne przetwarzanie obrazów zmian skórnych

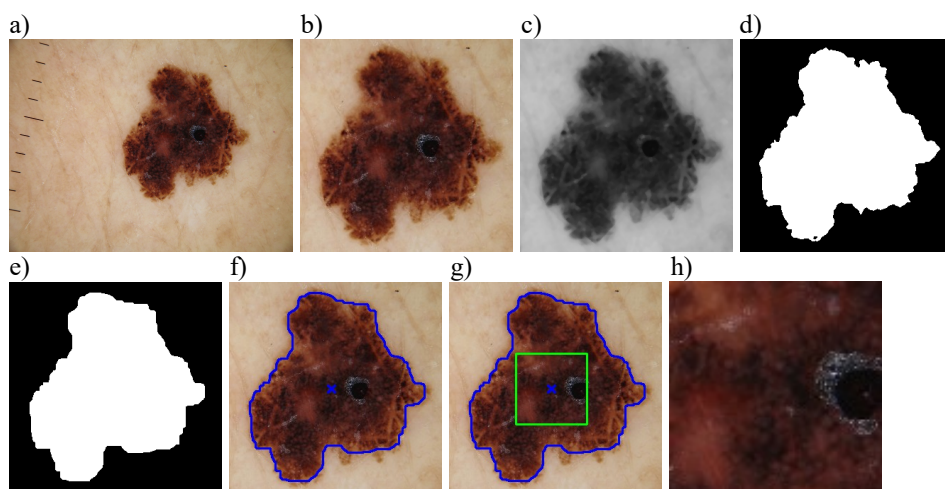
Podczas badań zmian skórnych, podobnie jak dla obrazów płomienia, do ekstrakcji cech obrazu wykorzystano jego teksturę. Pozwoliło to uzyskać porównywalność wyników dla obu rodzajów badanych obrazów, a ponadto, ten sam obszar zainteresowania mógł zostać wykorzystany zarówno przez algorytmy uczenia tradycyjnego, jak i głębokiego. Zastosowano ROI o kształcie kwadratu (100×100 pikseli), którego środek znajdował się w centroidzie konturu wykrytego obiektu. Metodyka postępowania mająca na celu wyłonienie ROI, a także wykorzystane narzędzia i biblioteki były podobne, jak w przypadku obrazów płomienia.

Niżej scharakteryzowano główne etapy wstępnego przetwarzania obrazu.

1. Wczytanie obrazu źródłowego, z 24-bitową głębią kolorów (rys. 7.4a).
2. Wstępne kadrowanie (ręczne), mające na celu eliminację artefaktów oraz dużych obszarów skóry wokół obiektu zainteresowania.
3. Przeskalowanie do rozmiaru 300×300 pikseli (rys. 7.4b). Konieczność przeskalowania wynikała z faktu, iż baza ISIC zawiera obrazy o różnej rozdzielczości. Rozmiar 300×300 pikseli wybrano arbitralnie, dopasowując go do obrazów o najniższej rozdzielczości.
4. Konwersja obrazu do 8-bitowej skali szarości.
5. Filtracja medianowa z maską o rozmiarze 5×5 pikseli (rys. 7.4c).
6. Progowanie obrazu z wykorzystaniem metody Otsu (rys. 7.4d).
7. Operacje morfologiczne (zamknięcie, otwarcie) z wykorzystaniem elementu strukturalnego o rozmiarze 3×3 piksele. Wykrycie wszystkich konturów na obrazie.
8. Znalezienie konturu o największej powierzchni (rys. 7.4e). W tym przypadku także wykorzystano obserwację polegającą na tym, że statystycznie, spo-

śród wszystkich konturów występujących na obrazie, ten należący do obiektu zainteresowania ma największą powierzchnię.

9. Obliczenie współrzędnych centroidu konturu (rys. 7.4f). Wykorzystano następujące zależności: $x=m_{10}/m_{00}$, $y=m_{01}/m_{00}$, gdzie: m_{00} – moment normalny rzędu zerowego, m_{10} , m_{01} – momenty normalne rzędu pierwszego.
10. Znalezienie kwadratu o wielkości 100×100 pikseli, którego środek pokrywa się z centroidem konturu (rys. 7.4g).
11. Skopiowanie z obrazu źródłowego ROI wyznaczonego przez znaleziony kwadrat pod warunkiem, że jest on całkowicie zawarty w konturze zmiany skórnej (rys. 7.4h). Jeśli warunek ten nie jest spełniony, ekstrakcja ROI nie jest wykonywana.



Rys. 7.4. Przykładowe wyniki najważniejszych etapów wstępnego przetwarzania obrazu: a) obraz źródłowy (1024×768 pikseli); b) obraz po wstępnym kadrowaniu i przeskalowaniu do rozmiaru 300×300 pikseli; c) obraz w skali szarości po filtracji medianowej; d) obraz binarny po progowaniu metodą Otsu; e) obraz po operacjach morfologicznych (zamknięcie, otwarcie) i znalezieniu konturu o największej powierzchni; f) obraz z zaznaczonym konturem zmiany skórnej oraz jej centroidem (niebieski krzyżyk); g) obraz z zaznaczonym ROI (zielony kwadrat o wielkości 100×100 pikseli), którego środek pokrywa się z centroidem konturu; h) ROI skopiowany z obrazu źródłowego.

7.4. Zastosowanie metod uczenia tradycyjnego oraz dużego zbioru danych

7.4.1. Wstępne przetwarzanie danych

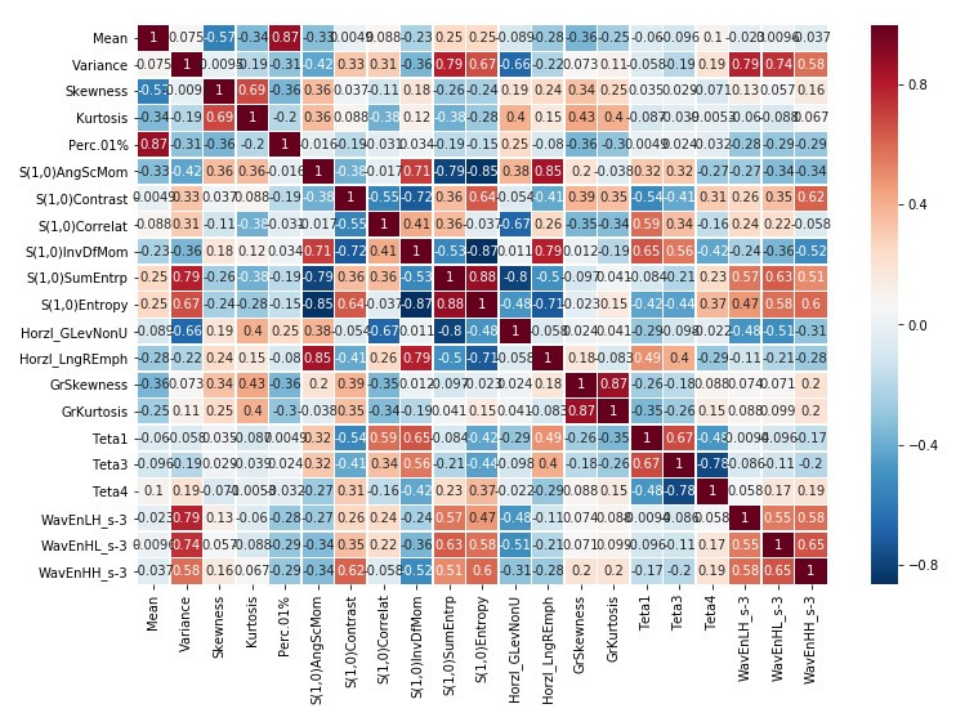
Pełny zbiór danych zawierał 1200 obserwacji, przy czym połowa z nich należała do grupy kontrolnej (klasa *znamię*), a połowa do grupy z rozpoznaniem

czerniaka (klasa *czerniak*). Zbiór ten podzielono losowo na część treningową i testową tak, że część pierwsza stanowiła 70%, a druga 30% zbioru pełnego. Szczegółowe dane dotyczące liczebności podzbiorów zamieszczono w tabeli 7.1.

Tabela 7.1. Liczebność podzbiorów otrzymanych w wyniku podziału dużego zbioru danych

Zbiór danych	Liczba obserwacji	Część zbioru pełnego	Znamię	Czerniak
Pełny	1200	1,0	600	600
Treningowy	840	0,7	420	420
Testowy	360	0,3	180	180

Estymację parametrów tekstury przeprowadzono z wykorzystaniem tego samego narzędzia, co w przypadku obrazów płomienia [72]. W wyniku analizy otrzymano 294 deskryptory cech. Dalsze etapy przetwarzania wstępnego danych, tj. skalowanie cech i oczyszczanie danych przeprowadzono zgodnie z procedurą opisaną w rozdziale 6.4.1. W wyniku tego procesu, liczba cech została zmniejszona z 294 do 21 (rys. 7.5).

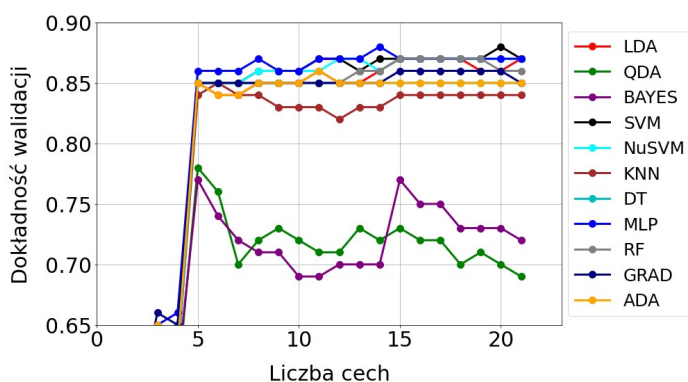


Rys. 7.5. Macierz korelacji pomiędzy cechami stanowiącymi wynik oczyszczania dużego zbioru danych

Kolejnym etapem wstępnego przetwarzania danych była selekcja cech. Listy rankingowe zbudowane w wyniku selekcji zawierały następującą liczbę cech: metody filtra i opakowujące – 21, metody wbudowane: regresja logistyczna – 9, AdaBoost i LGBM – 7. Podobnie, jak dla obrazów płomienia, wyboru optymalnej liczby cech dla każdej z zastosowanych metod selekcji dokonano na kolejnym etapie, tj. podczas trenowania modeli klasyfikacyjnych.

7.4.2. Wykorzystanie pełnego zestawu cech

Trenowanie modeli oraz wybór modelu optymalnego przeprowadzono w ten sam sposób, co dla obrazów płomienia (rozdział 6.4.2). Na rysunku 7.6 przedstawiono przykładowe wyniki walidacji dla metody Fishera.



Rys. 7.6. Wyniki walidacji modeli optymalnych dla metody Fishera

Na podstawie takich wyników (rys. 7.6), dla każdej metody klasyfikacji wybrany został model najskuteczniejszy, zapewniający największą dokładność walidacji, przy minimalnej liczbie cech zbioru treningowego. W tabeli 7.2. zamieszczono wyniki wyboru modeli najskuteczniejszych dla poszczególnych metod klasyfikacji. Na podstawie analizy tych danych, jako model najlepszy dla metody Fishera, uznano model MLP (wiersz podświetlony szarym tłem w tabeli 7.2). Przypomnijmy, że kryterium wyboru najlepszego modelu była największa dokładność walidacji, przy minimalnej liczbie cech zbioru treningowego. Dla modelu MLP, dokładność walidacji wyniosła 0,88, a do jego budowy wykorzystano 14 cech.

Procedurę wyboru najlepszych modeli, którą scharakteryzowano wcześniej na przykładzie metody Fishera, zastosowano dla wszystkich metod filtra i opakowujących. W przypadku metod wbudowanych, konstruowane i dostrajane były tylko te klasyfikatory, które zastosowano w procesie selekcji. Uzyskane wyniki zamieszczono w tabeli 7.3. Znaczenie symboli metod selekcji i klasyfikacji występujących w tabeli podano w wykazie symboli i akronimów.

Tabela 7.2. Wyniki wyboru modelu, który okazał się najskuteczniejszy dla poszczególnych metod klasyfikacji, przy wykorzystaniu metody Fishera

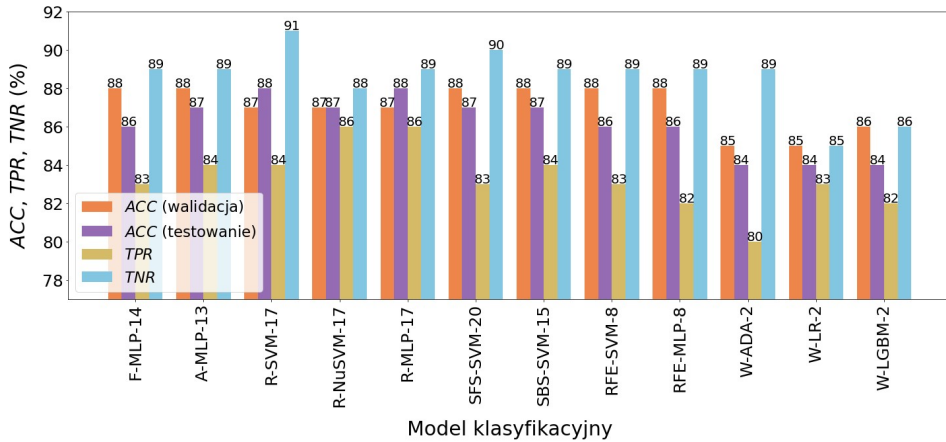
Metoda klasyfikacji	Dokładność walidacji	Optymalna liczba cech	Optymalne parametry modelu
LDA	0,87	15	<i>solver='svd'</i>
QDA	0,78	5	<i>tol=1e-05</i>
BAYES	0,77	5	<i>var_smoothing=1e-09</i>
SVM	0,88	20	<i>C=1.0, gamma='scale', kernel='linear'</i>
NuSVM	0,87	12	<i>gamma='scale', kernel='linear', nu=0.4</i>
KNN	0,85	6	<i>n_neighbors=10</i>
DT	0,85	5	<i>criterion='gini', max_depth=3</i>
MLP	0,88	14	<i>activation='relu', alpha=0.1, hidden_layer_sizes=(2,), max_iter=1000, solver='lbfgs'</i>
RF	0,87	15	<i>max_depth=4, n_estimators=10</i>
GRAD	0,86	15	<i>loss='exponential', n_estimators=50</i>
ADA	0,86	11	<i>n_estimators=20</i>

Tabela 7.3. Najlepsze modele klasyfikacyjne dla dużego zbioru danych i pełnego zestawu cech. Litera "W" występująca w symbolu modelu oznacza metodę wbudowaną

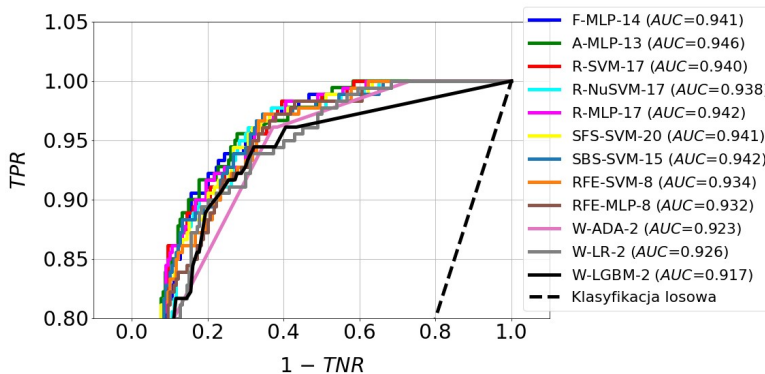
Symbol modelu	Metoda selekcji cech	Metoda klasyfikacji	Liczba cech
F-MLP-14	FISHER	MLP	14
A-MLP-13	ANOVA	MLP	13
R-SVM-17	RELIEF	SVM	17
R-NuSVM-17	RELIEF	NuSVM	17
R-MLP-17	RELIEF	MLP	17
SFS-SVM-20	SFS	SVM	20
SBS-SVM-15	SBS	SVM	15
RFE-SVM-8	RFE	SVM	8
RFE-MLP-8	RFE	MLP	8
W-ADA-2	W	ADA	2
W-LR-2	W	LR	2
W-LGBM-2	W	LGBM	2

Wyniki walidacji i testowania modeli wymienionych w tabeli 7.3 zaprezentowano na rysunku 7.7. Największą dokładność klasyfikacji dla zbioru testowego uzyskały modele: R-SVM-17 ($ACC = 88\%$) i R-MLP-17 ($ACC = 88\%$). Dokładność testowania czterech innych modeli (A-MLP-13, R-NuSVM-17, SFS-SVM-20, SBS-SVM-15) była niewiele gorsza i wynosiła 87%. Wszystkie modele uzyskały wyższą wartość indeksu TNR , w porównaniu z TPR . Oznacza to, że obrazy z grupy kontrolnej (należące do osób zdrowych) rozpoznawane były z większą dokładnością, niż obrazy z rozpoznaniem czerniaka. Taka sytuacja nie jest oczywiście korzystna dla pacjenta z medycznego punktu widzenia. Należy

dążyć do tego, aby czułość była większa od specyficzności lub, aby wartości obu indeksów były zbliżone. Wykresy krzywych ROC analizowanych modeli przedstawiono na rysunku 7.8. Przebiegi tych krzywych świadczą o tym, iż jakość większości modeli jest podobna. Na tym tle odstają nieco modele, do budowy których zastosowano wbudowane metody selekcji cech. Zauważmy, iż przyjmując parametr AUC jako główny wskaźnik jakości modelu, jako najskuteczniejszy należy uznać model A-MLP-13 ($AUC = 0,946$).



Rys. 7.7. Wyniki walidacji i testowania najlepszych modeli klasyfikacyjnych dla dużego zbioru danych i pełnego zestawu cech



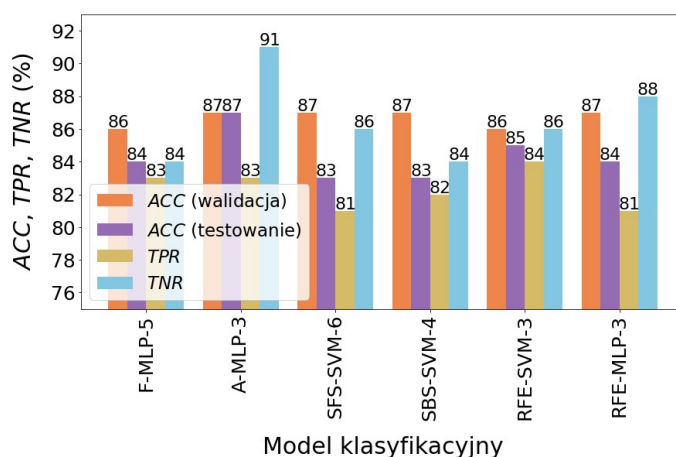
Rys. 7.8. Krzywe ROC wraz z parametrami AUC dla najlepszych modeli klasyfikacyjnych zbudowanych z wykorzystaniem pełnego zestawu cech dla dużego zbioru danych

7.4.3. Wykorzystanie ograniczonego zestawu cech

Ograniczenie zestawu cech przeprowadzono zgodnie z metodą opisaną w rozdziale 6.4.3 i na rysunku 6.14. Uzyskane wyniki zamieszczono w tabeli 7.4. Tabela nie zawiera informacji o modelach zbudowanych z wykorzystaniem metody Relief oraz metod wbudowanych. W tych przypadkach, nie udało się określić liczby cech mniejszej, niż dla zestawu pełnego, przy której dokładność walidacji zmniejszyłaby się mniej, niż wartość dopuszczalna, tj. 0,05. Jak wynika z danych zamieszczonych w tabeli 7.4, liczebność ograniczonych zestawów cech wahała się w granicach 3÷6. Na rysunku 7.9 zaprezentowano wyniki walidacji i testowania modeli wymienionych w tabeli 7.4.

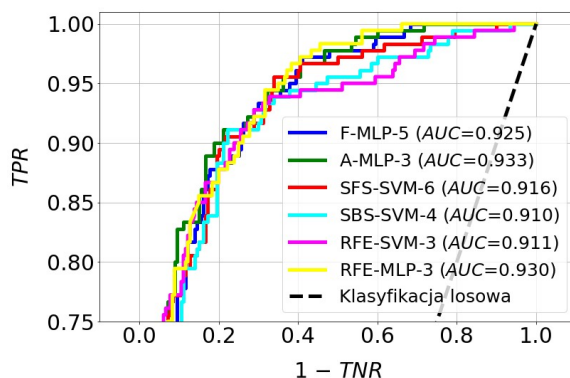
Tabela 7.4. Najlepsze modele klasyfikacyjne dla dużego zbioru danych i ograniczonego zestawu cech. W kolumnie "Różnica" podano wartość, o jaką zmniejszyła się dokładność walidacji, w porównaniu z modelem zbudowanym za pomocą tej samej metody klasyfikacji i pełnego zestawu cech

Symbol modelu	Metoda selekcji cech	Metoda klasyfikacji	Liczba cech	Dokładność walidacji	Różnica
F-MLP-5	FISHER	MLP	5	0,86	0,02
A-MLP-3	ANOVA	MLP	3	0,87	0,01
SFS-SVM-6	SFS	SVM	6	0,87	0,01
SBS-SVM-4	SBS	SVM	4	0,87	0,01
RFE-SVM-3	RFE	SVM	3	0,86	0,02
RFE-MLP-3	RFE	MLP	3	0,87	0,01



Rys. 7.9. Wyniki walidacji i testowania najlepszych modeli klasyfikacyjnych dla dużego zbioru danych i ograniczonego zestawu cech

Największą dokładność klasyfikacji dla zbioru testowego uzyskały modele: A-MLP-3 ($ACC = 87\%$) oraz RFE-SVM-3 ($ACC = 85\%$). Niewiele mniejsza była dokładność pozostałych modeli, zmieniała się ona w granicach $83\div 84\%$. Krzywe ROC przedstawiono na rysunku 7.10. Dobrą skuteczność modeli A-MLP-3 i RFE-SVM-3 potwierdzają parametry AUC , które przyjmują wartości odpowiednio 0,933 i 0,930. W pozostałych przypadkach, parametr ten zmieniał się w zakresie $0,910\div 0,925$.



Rys. 7.10. Krzywe ROC wraz z parametrami AUC dla najlepszych modeli klasyfikacyjnych zbudowanych z wykorzystaniem ograniczonego zestawu cech dla dużego zbioru danych

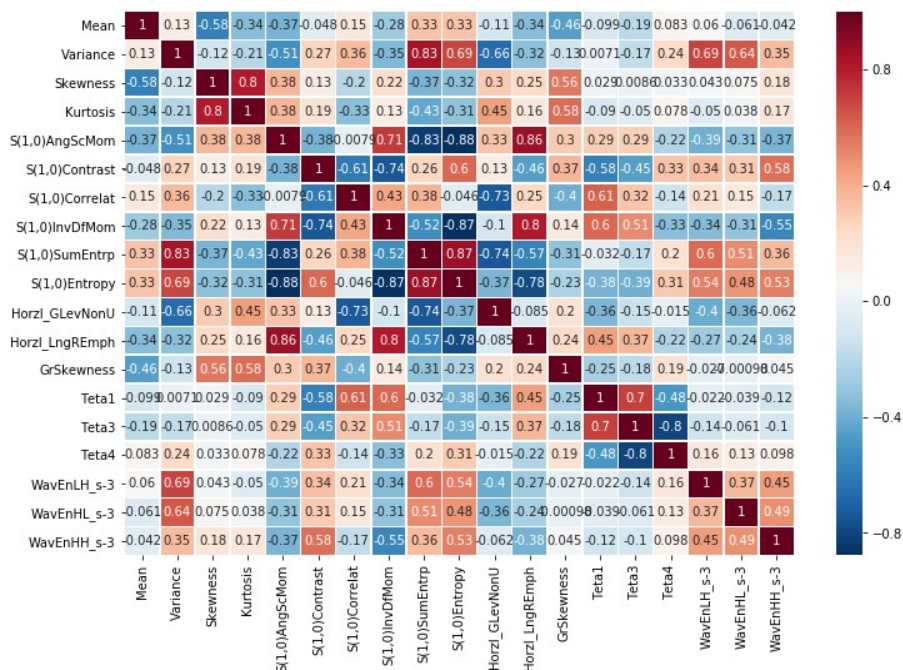
7.5. Zastosowanie metod uczenia tradycyjnego oraz małego zbioru danych

Mały zbiór danych zawierał 240 obserwacji, przy czym połowa z nich należała do grupy kontrolnej (klasa *znamię*), a połowa do grupy z rozpoznaniem czerniaka (klasa *czerniak*). Zbiór ten podzielono losowo na część treningową i testową tak, że część pierwsza stanowiła 70%, a druga 30% zbioru pełnego. Zbiory treningowy i testowy nie były wydzieloną częścią podzbiorów zbioru pełnego, lecz powstały wskutek zupełnie nowego losowania. Szczegółowe dane dotyczące liczebności podzbiorów zamieszczono w tabeli 7.5.

Tabela 7.5. Liczebność podzbiorów otrzymanych w wyniku podziału małego zbioru danych

Zbiór danych	Liczba obserwacji	Część zbioru pełnego	Znamię	Czerniak
Pełny	240	1,0	120	120
Treningowy	168	0,7	84	84
Testowy	72	0,3	36	36

W wyniku oczyszczania danych, liczba cech została zmniejszona z 294 do 19 (rys. 7.11). Liczebność cech występujących na listach rankingowych zbudowanych przez poszczególne metody selekcji była następująca: metody filtra i opakowujące – 19, metody wbudowane: regresja logistyczna – 5, AdaBoost – 8, LGBM – 19.

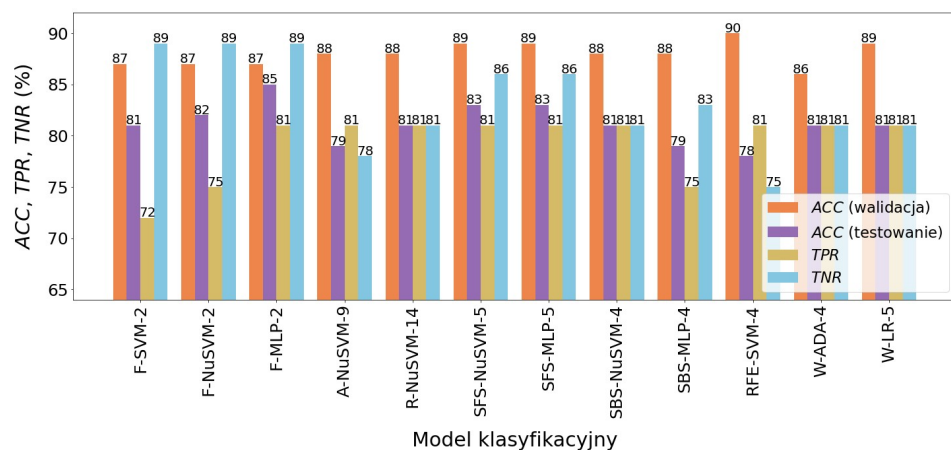


Rys. 7.11. Macierz korelacji pomiędzy cechami stanowiącymi wynik oczyszczania małego zbioru danych

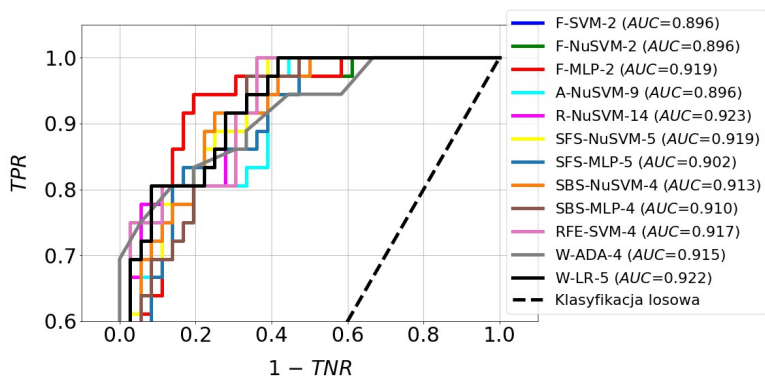
W tabeli 7.6 zamieszczono efekty trenowania i wyboru najlepszych modeli, natomiast rysunek 7.12 prezentuje wyniki ich walidacji i testowania. Pod względem dokładności klasyfikacji dla zbioru testowego, wyróżniają się następujące modele: F-MLP-2 ($ACC = 85\%$), SFS-NuSVM-5 ($ACC = 83\%$) oraz SFS-MLP-5 ($ACC = 83\%$). Zauważmy, że model F-MLP-2 uzyskał dość wysoką dokładność (85%) przy wykorzystaniu tylko dwóch cech. Przebiegi krzywych ROC przedstawione na rysunku 7.13 nie pozwalają na jednoznaczne wskazanie najskuteczniejszego modelu. W kilku przypadkach wartości parametru AUC są bardzo zbliżone: F-MLP-2 ($AUC = 0,919$), R-NuSVM-14 ($AUC = 0,923$), SFS-NuSVM-5 ($AUC = 0,919$), W-LR-5 ($AUC = 0,922$).

Tabela 7.6. Najlepsze modele klasyfikacyjne dla małego zbioru danych i pełnego zestawu cech.
Litera "W" występująca w symbolu modelu oznacza metodę wbudowaną

Symbol modelu	Metoda klasyfikacji	Metoda selekcji cech	Liczba cech
F-SVM-2	SVM	FISHER	2
F-NuSVM-2	NuSVM	FISHER	2
F-MLP-2	MLP	FISHER	2
A-NuSVM-9	NuSVM	ANOVA	9
R-NuSVM-14	NuSVM	RELIEF	14
SFS-NuSVM-5	NuSVM	SFS	5
SFS-MLP-5	MLP	SFS	5
SBS-NuSVM-4	NuSVM	SBS	4
SBS-MLP-4	MLP	SBS	4
RFE-SVM-4	SVM	RFE	4
W-ADA-4	ADA		4
W-LR-5	LR		5
W-LGBM-2	LGBM		2



Rys. 7.12. Wyniki walidacji i testowania najlepszych modeli klasyfikacyjnych dla małego zbioru danych i pełnego zestawu cech



Rys. 7.13. Krzywe ROC wraz z parametrami AUC dla najlepszych modeli klasyfikacyjnych zbudowanych z wykorzystaniem pełnego zestawu cech dla małego zbioru danych

7.6. Zastosowanie metod uczenia głębokiego oraz dużego zbioru danych

Sposób podziału zbioru danych dla metod uczenia głębokiego był taki sam, jak w przypadku obrazów płomienia (rozdział 6.6). Dzięki temu, zarówno metody uczenia tradycyjnego, jak i głębokiego korzystały z tych samych obserwacji na etapie uczenia i walidacji modeli oraz podczas ich ewaluacji. W efekcie, możliwa była ocena skuteczności obu rodzajów metod dla tych samych danych. Szczegółowe informacje na temat liczebności poszczególnych podzbiorów zamieszczono w tabeli 7.7.

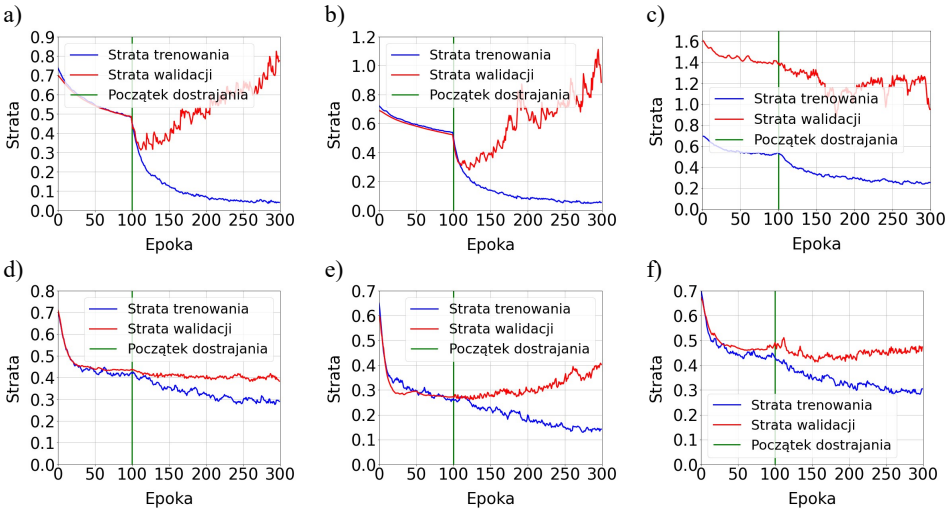
Tabela 7.7. Liczebność podzbiorów otrzymanych w wyniku podziału dużego zbioru danych

Zbiór danych	Liczba obserwacji	Część zbioru pełnego	Znamię	Czerniak
Pełny	1200	1,0	600	600
Treningowy	480	0,4	240	240
Walidacyjny	360	0,3	180	180
Testowy	360	0,3	180	180

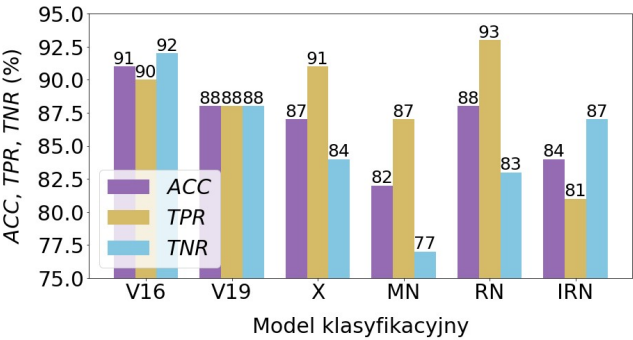
7.6.1. Wykorzystanie obrazów kolorowych

Wykresy straty trenowania i walidacji przedstawiono na rysunku 7.14. W przypadku modelu Xception widać oznaki niedotrenowania, czego przyczyną mogła być za mała pojemność modelu. W pozostałych przypadkach obserwujemy lekkie przetrenowanie, które najwyraźniej widać na przykładzie modeli VGG16 i VGG19. Przedstawione wykresy dostarczają nam bardzo ważnej informacji na temat wyboru czasu uczenia, który powinniśmy zastosować dla każdego modelu. Czas ten powinien się kończyć wówczas, gdy straty walidacji

zaczynają narastać. Na przykład, dla modeli VGG16, VGG19 i ResNet50 czas ten powinien obejmować ok. 120 epok, a dla modelu InceptionResNetV2 powinien on być równy ok. 150 epok. Uwzględniając powyższe uwagi, wszystkie 4 modele wytrenowano ponownie. Wyniki ewaluacji zaprezentowano na rysunku 7.15.

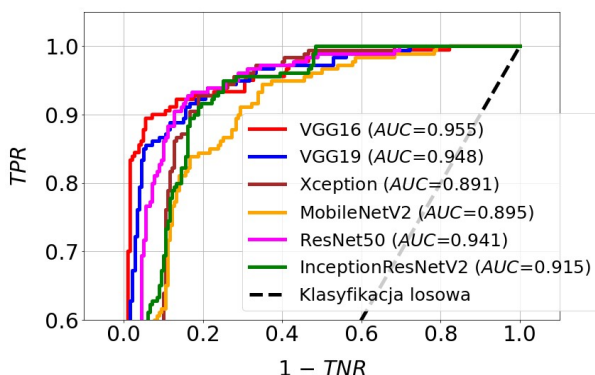


Rys. 7.14. Wykresy straty trenowania i walidacji modeli dla dużego zbioru danych i obrazów kolorowych: a) VGG16; b) VGG19; c) Xception; d) MobileNetV2; e) ResNet50; f) InceptionResNetV2



Rys. 7.15. Wyniki testowania modeli dla dużego zbioru danych i obrazów kolorowych. Zastosowane symbole: V16 – VGG16, V19 – VGG19, X – Xception, MN – MobileNetV2, RN – ResNet50, IRN – InceptionResNetV2

Największą dokładność klasyfikacji dla zbioru testowego uzyskał model VGG16 ($ACC = 91\%$). Z kolei, zdecydowanie najgorzej wypadł MobileNetV2 ($ACC = 82\%$). Taką ocenę skuteczności potwierdzają krzywe ROC przedstawione na rysunku 7.16. Parametr AUC przyjmuje największą wartość, równą 0,955, dla modelu VGG16.

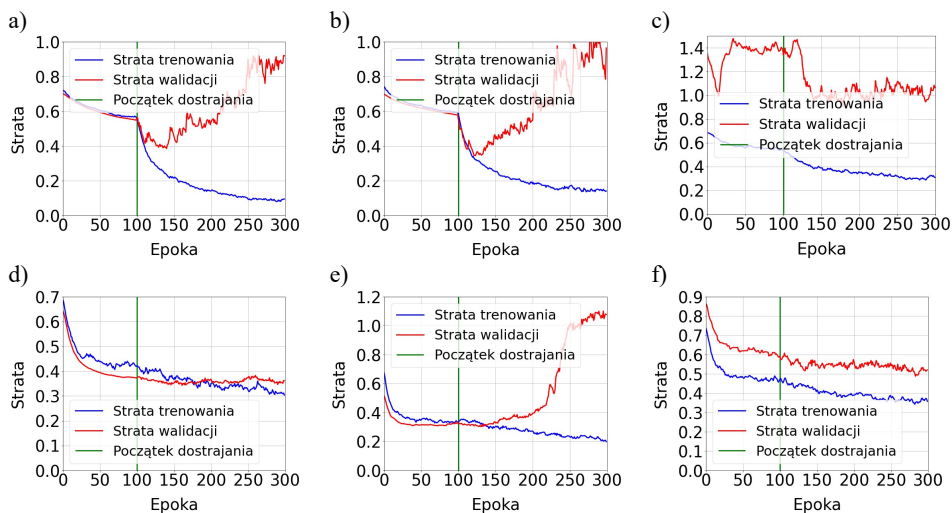


Rys. 7.16. Krzywe ROC wraz z parametrami AUC dla modeli zbudowanych z wykorzystaniem dużego zbioru danych i obrazów kolorowych

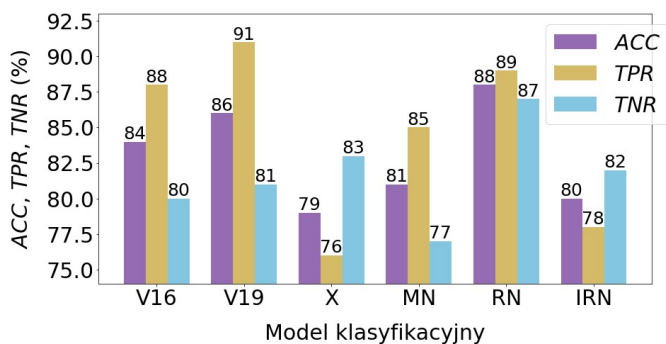
7.6.2. Wykorzystanie obrazów monochromatycznych

Krzywe straty trenowania i walidacji modeli zbudowanych na podstawie obrazów monochromatycznych przedstawiono na rysunku 7.17. Własności tych modeli są podobne do własności modeli zbudowanych na podstawie obrazów kolorowych. Model Xception jest także najslabiej dopasowany do danych, a modele VGG16, VGG19 oraz ResNet50 wykazują oznaki przetrenowania. Z kolei, dość dobre dopasowanie prezentują krzywe strat należące do modelu MobileNetV2. Biorąc pod uwagę przebieg krzywych strat walidacji przedstawionych na rysunku 7.17, modele VGG16, VGG19 i ResNet50 wytrenowano ponownie przez okres 125 epok.

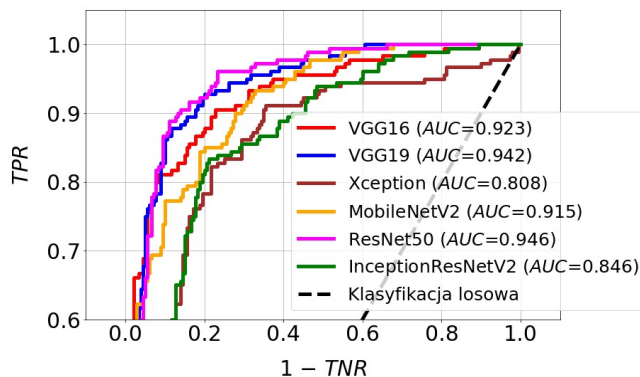
Wyniki testowania modeli przedstawiono na rysunku 7.18. Największą skuteczność osiągnął model ResNet50 ($ACC = 88\%$), natomiast nieznacznie gorszy był VGG19 ($ACC = 86\%$). Przebieg krzywych ROC pokazanych na rysunku 7.19 potwierdza wysoką jakość obu wymienionych wyżej modeli, dla ResNet50 wartość parametru AUC wyniosła 0,946, zaś dla VGG19 była ona równa 0,942. W pozostałych przypadkach, wartość AUC wahała się w granicach 0,808÷0,923.



Rys. 7.17. Wykresy straty trenowania i walidacji modeli dla dużego zbioru danych i obrazów monochromatycznych: a) VGG16; b) VGG19; c) Xception; d) MobileNetV2; e) ResNet50; f) InceptionResNetV2



Rys. 7.18. Wyniki testowania modeli dla dużego zbioru danych i obrazów monochromatycznych. Zastosowane symbole: V16 – VGG16, V19 – VGG19, X – Xception, MN – MobileNetV2, RN – ResNet50, IRN – InceptionResNetV2



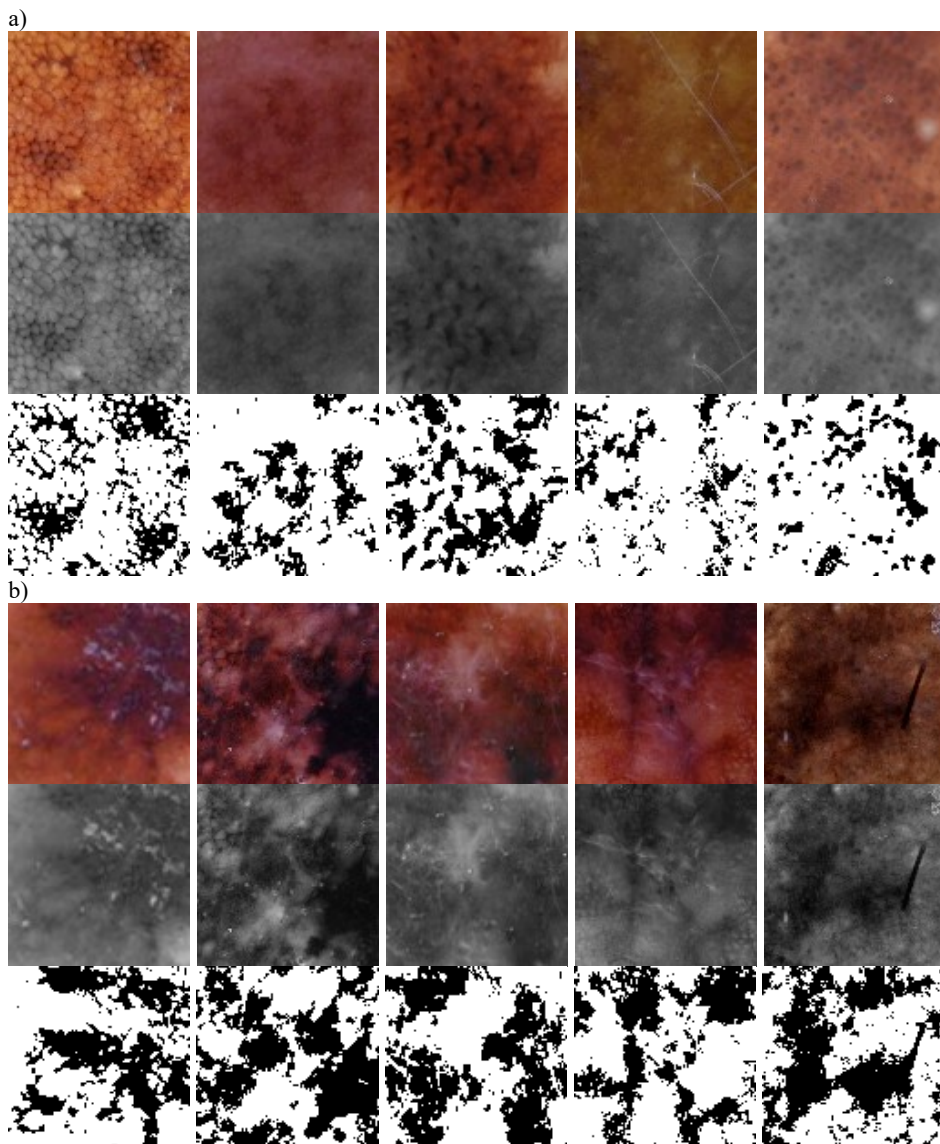
Rys. 7.19. Krzywe ROC wraz z parametrami AUC dla modeli zbudowanych z wykorzystaniem dużego zbioru danych i obrazów monochromatycznych

7.6.3. Wykorzystanie obrazów binarnych

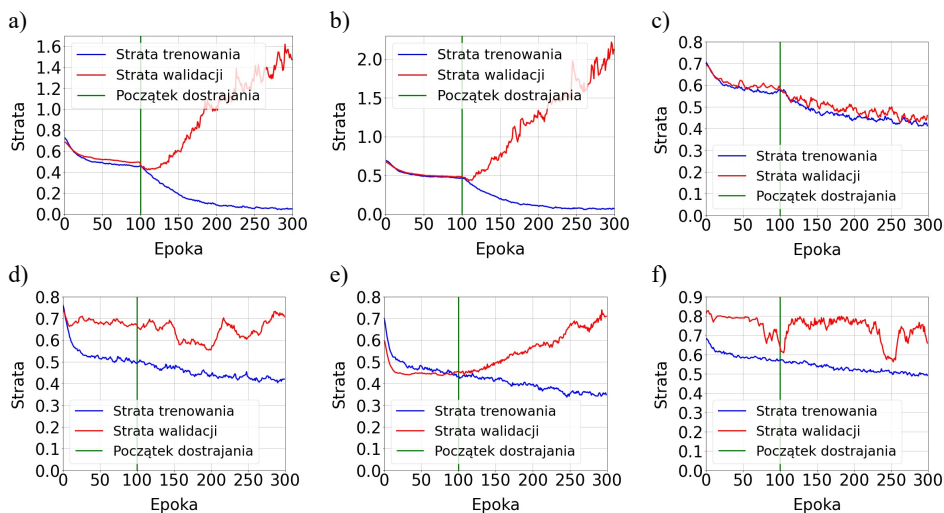
Do segmentacji obrazów monochromatycznych zastosowano progowanie lokalne metodą Bradleya. Metoda ta, podobnie jak dla obrazów płomienia, okazała się najlepsza spośród testowanych metod segmentacji. W odróżnieniu od obrazów płomienia, w przypadku obrazów zmian skórnych nie zastosowano normalizacji histogramu. Operacja ta poprawia kontrast obrazów, ale należy z niej korzystać ostrożnie, ponieważ zmieniając intensywność pikseli, powoduje ona zniekształcenie oryginalnej informacji zawartej w obrazie. Jest to szczególnie istotne dla obrazów medycznych, w przypadku których, różnica intensywności pikseli pomiędzy obiektami występującymi na obrazie może zawierać ważną informację diagnostyczną. Na rysunku 7.20 zaprezentowano przykładowe wyniki operacji progowania.

Wykresy krzywych straty trenowania i walidacji przedstawiono na rysunku 7.21. Wynika z nich, że najlepsze dopasowanie do danych osiągnął model Xception. W przypadku MobileNetV2 oraz InceptionResNetV2 obserwujemy pewne niedopasowanie, natomiast dla VGG16, VGG19 i ResNet50 ma miejsce przetrenowanie. W tych ostatnich przypadkach dokonano ponownego uczenia modeli, przyjmując liczbę epok trenowania, równą 125.

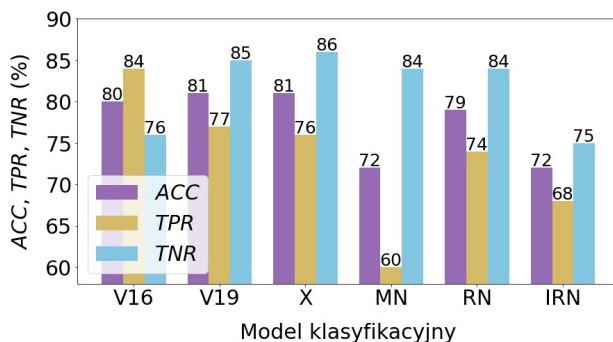
Wyniki ewaluacji przedstawione na rysunku 7.22 pokazują, że największą dokładność testowania uzyskały modele: Xception ($ACC = 81\%$), VGG19 ($ACC = 81\%$), VGG16 ($ACC = 80\%$) oraz ResNet50 ($ACC = 79\%$). Z kolei, skuteczność modeli MobileNetV2 oraz InceptionResNetV2 okazała się najgorsza ($ACC = 72\%$), co wynikało z najsłabszego dopasowania do danych. Przebieg krzywych ROC oraz wielkość pól pod tymi krzywymi (rys. 7.23) potwierdzają wcześniejszą ocenę.



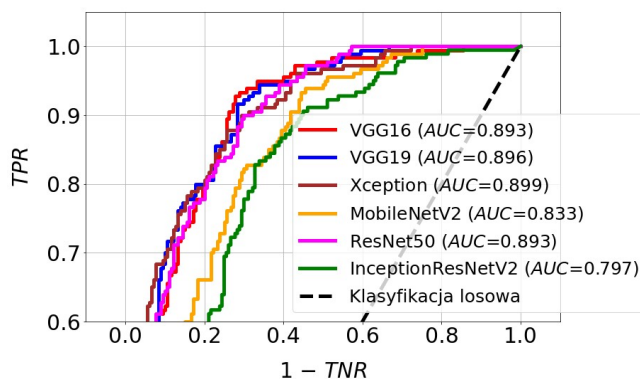
Rys. 7.20. Przykładowe obrazy wykorzystane w badaniach: a) znamiona; b) czerniaki. Górne wiersze prezentują wydzielone obszary zainteresowania (100x100 pikseli). Wiersze środkowe prezentują obrazy monochromatyczne, natomiast w dolnych wierszach pokazano wynik progowania lokalnego metodą Bradleya



Rys. 7.21. Wykresy straty trenowania i walidacji modeli dla dużego zbioru danych i obrazów binarnych: a) VGG16; b) VGG19; c) Xception; d) MobileNetV2; e) ResNet50; f) InceptionResNetV2



Rys. 7.22. Wyniki testowania modeli dla dużego zbioru danych i obrazów binarnych. Zastosowane symbole: V16 – VGG16, V19 – VGG19, X – Xception, MN – MobileNetV2, RN – ResNet50, IRN – InceptionResNetV2



Rys. 7.23. Krzywe ROC wraz z parametrami AUC dla modeli zbudowanych z wykorzystaniem dużego zbioru danych i obrazów binarnych

7.7. Zastosowanie metod uczenia głębokiego oraz małego zbioru danych

Podczas badań wykorzystujących mały zbiór danych zastosowano tylko obrazy kolorowe (podobnie, jak dla obrazów płomienia). Losowo podzielono zbiór treningowy, wykorzystany wcześniej przez metody uczenia tradycyjnego, na część treningową i uczącą, natomiast bez żadnych zmian zastosowano ten sam zbiór testowy. Informację dotyczącą liczebności podzbiorów zamieszczono w tabeli 7.8.

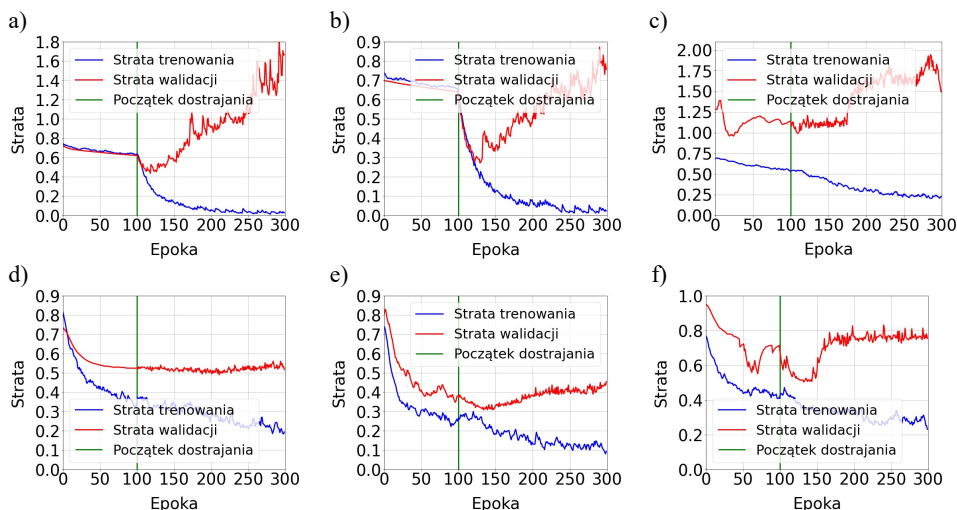
Tabela 7.8. Liczebność podzbiorów otrzymanych w wyniku podziału małego zbioru danych

Zbiór danych	Liczba obserwacji	Część zbioru pełnego	Znamię	Czerniak
Pełny	240	1,0	120	120
Treningowy	96	0,4	48	48
Walidacyjny	72	0,3	36	36
Testowy	72	0,3	36	36

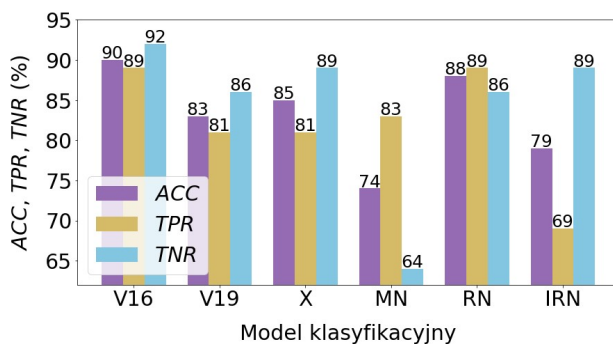
Wykresy krzywych straty trenowania i walidacji przedstawiono na rysunku 7.24. Objawy przetrenowania występują w przypadku modeli VGG16, VGG19 i ResNet50. W tych przypadkach trenowanie wykonano ponownie, ograniczając ten proces do 125 epok. Pozostałe modele wykazują objawy niedopasowania do danych.

Wyniki ewaluacji zaprezentowano na rysunku 7.25. Zdecydowanie najsukcesowniej okazały się modele: VGG16 ($ACC = 90\%$) i ResNet50 ($ACC = 88\%$). Wyróżniły się one także wysoką czułością (TPR), wynoszącą w obu przypadkach 89%. Krzywe ROC przedstawione na rysunku 7.26 potwierdzają wcześniejszą ocenę. Wartości parametru AUC były, dla wymienionych

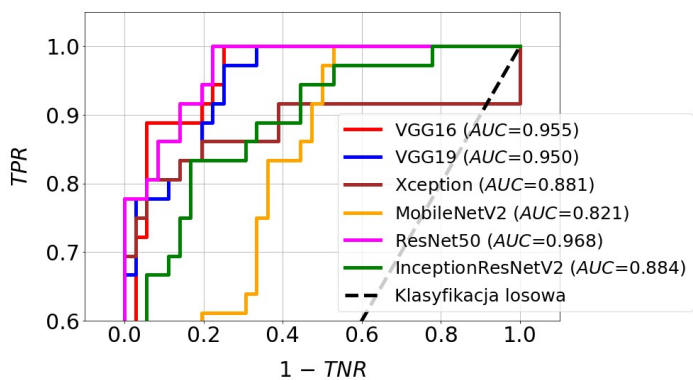
wcześniej modeli, najwyższe i wynosiły – 0,955 dla VGG16 oraz 0,968 w przypadku ResNet50. Bardzo słabe dopasowanie do danych modelu MobileNetV2 znalazło odzwierciedlenie w jego niskiej dokładności dla zbioru testowego ($ACC = 74\%$) oraz najmniejszej powierzchni pod krzywą ROC ($AUC = 0,821$).



Rys. 7.24. Wykresy straty trenowania i walidacji modeli dla małego zbioru danych i obrazów kolorowych: a) VGG16; b) VGG19; c) Xception; d) MobileNetV2; e) ResNet50; f) InceptionResNetV2



Rys. 7.25. Wyniki testowania modeli dla małego zbioru danych i obrazów kolorowych. Zastosowane symbole: V16 – VGG16, V19 – VGG19, X – Xception, MN – MobileNetV2, RN – ResNet50, IRN – InceptionResNetV2

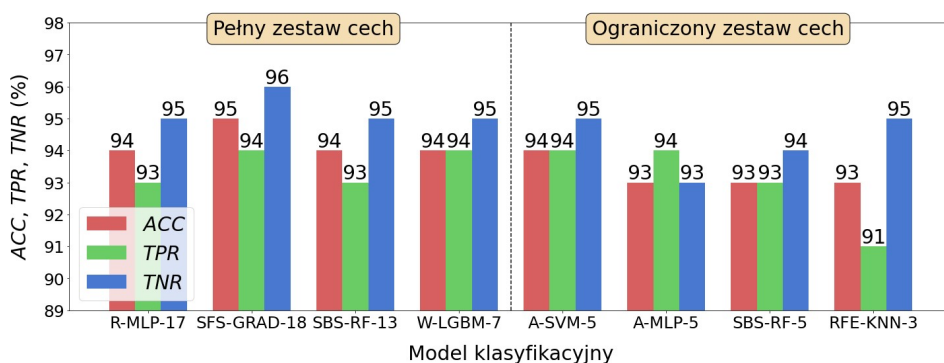


Rys. 7.26. Krzywe ROC wraz z parametrami AUC dla modeli zbudowanych z wykorzystaniem małego zbioru danych i obrazów kolorowych

8. Analiza wyników badań

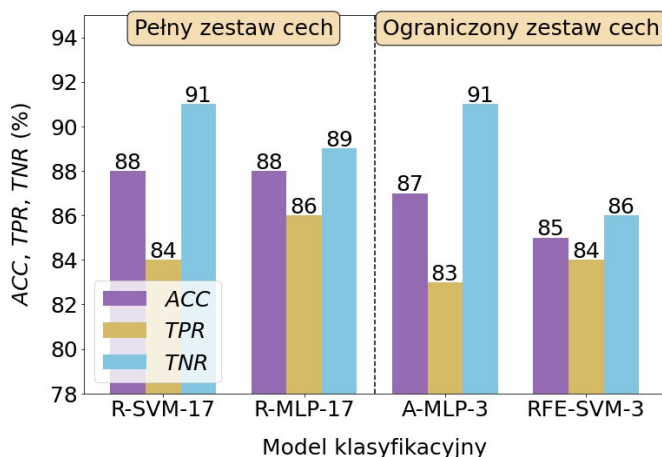
8.1. Wpływ liczby cech na dokładność klasyfikacji

Na rysunkach 8.1 i 8.2 porównano wyniki testowania najsłabszych modeli zbudowanych na podstawie dużego zbioru danych, przy wykorzystaniu pełnego i ograniczonego zestawu cech. Przypomnijmy, że pełna lista cech zawiera minimalną liczbę cech wybraną na podstawie rankingu, zapewniającą maksymalną dokładność walidacji. Natomiast, ograniczona lista cech, to lista obejmująca taką liczbę cech wybraną na podstawie rankingu, której zwiększanie nie powoduje już istotnego (większego, niż 0,05) wzrostu dokładności walidacji.



Rys. 8.1. Wyniki testowania najsłabszych modeli zbudowanych na podstawie dużego zbioru danych, przy wykorzystaniu pełnego i ograniczonego zestawu cech obrazów płomienia

W przypadku obrazów płomienia (rys. 8.1), dokładność (ACC) najsłabszych modeli zbudowanych na podstawie pełnego zestawu cech waha się w granicach 94÷95%. Z kolei, dla modeli zbudowanych na podstawie ograniczonego zestawu cech ACC należy do przedziału 93÷94%. Zauważmy, że najlepszy model dla ograniczonego zestawu cech (A-SVM-5) ma dokładność ($ACC = 94\%$) mniejszą zaledwie o 1% od dokładności najlepszego modelu dla pełnego zestawu cech (SFS-GRAD-18), wynoszącej 95%. Należy podkreślić, iż największą odporność na ograniczenie liczby cech wykazał model SBS-RF. W tym przypadku, zredukowanie tej liczby z 13 do 5 powoduje zmniejszenie dokładności z 94% do 93%, a więc zaledwie o 1%.



Rys. 8.2. Wyniki testowania najskuteczniejszych modeli zbudowanych na podstawie dużego zbioru danych, przy wykorzystaniu pełnego i ograniczonego zestawu cech obrazów zmian skórnych

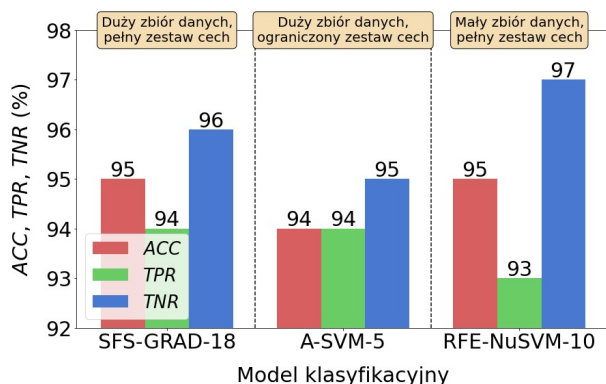
Analiza wyników testowania modeli zbudowanych dla obrazów zmian skórnych (rys. 8.2) dostarcza podobnych wniosków, jak w przypadku obrazów płomienia. Dokładność najskuteczniejszych modeli zbudowanych na podstawie pełnego zestawu cech jest równa 88%, natomiast dla ograniczonego zestawu cech *ACC* zmienia się od 85% do 87%. Dokładność najlepszego modelu dla ograniczonego zestawu cech (A-MLP-3) wynosi 87% i jest mniejsza tylko o 1% od wartości parametru *ACC* najlepszych modeli dla pełnego zestawu cech (R-SVM-17 i R-MLP-17), wynoszącej 88%.

8.2. Najlepsze metody selekcji i klasyfikacji

Wyniki testowania najskuteczniejszych modeli, zbudowanych na podstawie dużego i małego zbioru danych, zaprezentowano na rysunkach 8.3 i 8.4. W tab. 8.1 zamieszczono zestawienie metod selekcji, klasyfikacji oraz liczby cech użytych do budowy najlepszych modeli dla obrazów płomienia. W przypadku dużego zbioru danych i pełnej listy cech, najskuteczniejszy model zbudowano z wykorzystaniem metody selekcji SFS i metody klasyfikacji GRAD. Do budowy modelu użyto 18 cech. Dla dużego zbioru danych i ograniczonego zestawu cech zastosowano metodę ANOVA do selekcji oraz SVM do klasyfikacji. W tym przypadku, model wykorzystywał tylko 5 cech. W ostatnim przypadku, tj. dla małego zbioru danych i pełnej listy cech, zastosowano metodę RFE do selekcji i NuSVM do klasyfikacji. Do budowy modelu użyto 10 cech.

Powyższe obserwacje skłaniają do wniosku, iż dla obrazów płomienia nie można wskazać jednej metody selekcji i klasyfikacji, która byłaby najlepsza dla wszystkich wariantów dotyczących wielkości danych oraz zestawu cech. Inter-

sujący z praktycznego punktu widzenia jest także fakt, iż uzyskano niemal identyczną skuteczność modeli, niezależnie od postaci zbioru danych oraz zestawu cech (rys. 8.3).



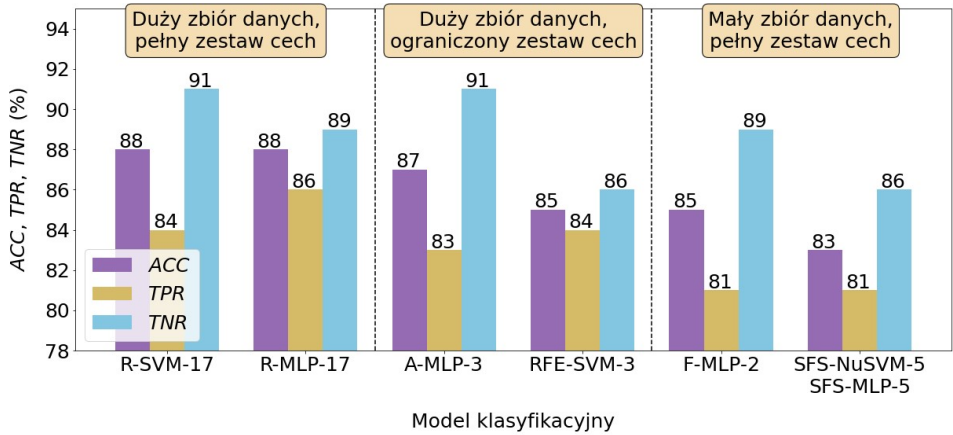
Rys. 8.3. Wyniki testowania najskuteczniejszych modeli zbudowanych na podstawie dużego i małego zbioru obrazów płomienia

Tabela 8.1. Zestawienie metod selekcji, klasyfikacji oraz liczby cech użytych do budowy najskuteczniejszych modeli wykorzystujących duży i mały zbiór obrazów płomienia

Zbiór danych	Zestaw cech	Metoda selekcji	Metoda klasyfikacji	Liczba cech
Duży	Pełny	SFS	GRAD	18
Duży	Ograniczony	ANOVA	SVM	5
Mały	Pełny	RFE	NuSVM	10

Informacje dotyczące najskuteczniejszych modeli dla obrazów zmian skórnych zamieszczono na rysunkach 8.4 i w tab. 8.2. Dla dużego zbioru danych i pełnego zestawu cech, najlepsza okazała się metoda selekcji Relief, która pozwoliła zbudować 2 modele z użyciem 17 cech. W pierwszym przypadku zastosowano metodę klasyfikacji SVM, a w drugim MLP. Dla dużego zbioru danych i ograniczonego zestawu cech, 2 najskuteczniejsze modele zbudowano z użyciem tych samych metod klasyfikacji (MLP i SVM), co dla pełnego zestawu cech. W przypadku MLP, wykorzystana była metoda selekcji ANOVA, a dla SVM metoda RFE. Należy podkreślić wysoką jakość tych modeli, a tym samym wysoką skuteczność metod selekcji i klasyfikacji zastosowanych do ich budowy. Zauważmy, że przy zaledwie trzech cechach wykorzystanych do budowy rozpatrywanych modeli, ich dokładność (rys. 8.4) jest tylko nieznacznie mniejsza od dokładności najlepszych modeli dla pełnego zestawu cech. W przypadku modelu A-MLP-3 ($ACC = 87\%$) różnica ta wynosi tylko 1%. Dla małego zbioru danych i pełnego zestawu cech, 3 modele można uznać za najlepsze. W dwóch przypad-

kach wykorzystano metodę SFS do selekcji cech, także 2 razy zastosowano metodę MLP do klasyfikacji (tab. 8.2). Należy podkreślić stosunkowo wysoką skuteczność modelu zbudowanego z wykorzystaniem metody Fishera do selekcji i MLP do klasyfikacji (F-MLP-2). Wykorzystując tylko 2 cechy, model ten osiągnął dokładność testowania na poziomie 85%, co jest wynikiem gorszym tylko o 3% od dokładności modeli opartych na 17 cechach, wynoszącej 88% (rys. 8.4).



Rys. 8.4. Wyniki testowania najskuteczniejszych modeli zbudowanych na podstawie dużego i małego zbioru obrazów zmian skórnych

Tabela 8.2. Zestawienie metod selekcji, klasyfikacji oraz liczby cech użytych do budowy najskuteczniejszych modeli wykorzystujących duży i mały zbiór obrazów zmian skórnych

Zbiór danych	Zestaw cech	Metoda selekcji	Metoda klasyfikacji	Liczba cech modelu
Duży	Pełny	Relief	SVM	17
			MLP	17
	Ograniczony	ANOVA	MLP	3
		RFE	SVM	3
Mały	Pełny	Fishera	MLP	2
		SFS	NuSVM	5
			MLP	5

Analiza zestawu metod użytych do budowy najlepszych modeli dla obrazów zmian skórnych pozwala stwierdzić, iż niektóre metody używane są częściej, niż inne. Przykładem jest metoda klasyfikacji MLP, która okazała się skuteczna, niezależnie od postaci zbioru danych i zestawu cech. Spośród sześciu modeli, których wyniki przedstawiono na rysunku 8.4, metoda ta wykorzystana była do budowy aż czterech z nich. Należy także podkreślić wysoką jakość metody

klasyfikacji SVM, którą wykorzystano do budowy najskuteczniejszych modeli dla dużego zbioru danych, zarówno dla pełnego, jak i dla ograniczonego zestawu cech. Jeżeli chodzi o metody selekcji, to dla pełnego zestawu cech wyróżniła się metoda Relief (dla dużego zbioru danych) oraz SFS (dla małego zbioru danych).

8.3. Najskuteczniejsze zestawy cech

W tabelach 8.3÷8.8 zamieszczono zestawy cech diagnostycznych, które wykorzystano do budowy modeli uznanych za najskuteczniejsze w rozwiązywaniu postawionych zadań klasyfikacji. Kolejność cech jest zgodna z ich rankingiem dokonany za pomocą metody selekcji wykorzystanej do budowy danego modelu. W pierwszej kolumnie każdej tabeli znajdują się nazwy cech, a w kolumnie drugiej – ich znaczenie. Dodatkowo, w nawiasach okrągłych podano symbole źródeł informacji, na podstawie których obliczono poszczególne cechy. Znaczenie tych symboli jest następujące: a – wektor histogramu obrazu; b – macierz gradientu obrazu; c – model autoregresji; d – macierz przejść tonalnych; e – macierz rozkładu długości pasm; f – dyskretna transformata falkowa.

Tabela 8.3. Pełny zestaw cech wykorzystany do budowy modelu SFS-GRAD-18 na podstawie dużego zbioru obrazów płomienia. Kolejność cech jest zgodna z ich rankingiem dokonany za pomocą metody SFS

Nr	Nazwa cechy	Znaczenie cechy (źródło informacji)
1	<i>Perc.01%</i>	Percentyl 1% (a)
2	<i>S(1,0)Contrast</i>	Kontrast (odległość – 1, kierunek – 90°) (d)
3	<i>WavEnHL_s-1</i>	Energia współczynnika falkowego dla podpasma HL w 1 skali (f)
4	<i>Mean</i>	Średnia (a)
5	<i>Variance</i>	Wariancja (a)
6	<i>45dgr_ShrtREmp</i>	Odwrotny moment uwydatnienia krótkich pasm dla kierunku 45° (e)
7	<i>S(1,0)AngScMom</i>	Drugi moment kątowy (odległość – 1, kierunek – 90°) (d)
8	<i>Horzl GLvNonU</i>	Niejednorodność szarości dla kierunku 0° (e)
9	<i>Horzl LngREmp</i>	Moment uwydatnienia długich pasm dla kierunku 0° (e)
10	<i>GrKurtosis</i>	Kurtoza (b)
11	<i>GrMean</i>	Średnia (b)
12	<i>GrSkewness</i>	Skośność (b)
13	<i>Kurtosis</i>	Kurtoza (a)
14	<i>WavEnHH_s-2</i>	Energia współczynnika falkowego dla podpasma HH w 2 skali (f)
15	<i>Teta3</i>	Parametr Q_3 (c)
16	<i>S(0,1)Contrast</i>	Kontrast (odległość – 1, kierunek – 0°) (d)
17	<i>Vertl_ShrtREmp</i>	Odwrotny moment uwydatnienia krótkich pasm dla kierunku 90° (e)
18	<i>Skewness</i>	Skośność (a)

Tabela 8.4. Ograniczony zestaw cech wykorzystany do budowy modelu A-SVM-5 na podstawie dużego zbioru obrazów płomienia. Kolejność cech jest zgodna z ich rankingiem dokonany za pomocą metody ANOVA

Nr	Nazwa cechy	Znaczenie cechy (źródło informacji)
1	<i>Teta3</i>	Parametr Q_3 (c)
2	<i>S(1,0)AngScMom</i>	Drugi moment kątowny (odległość – 1, kierunek – 90°) (d)
3	<i>S(1,0)Contrast</i>	Kontrast (odległość – 1, kierunek – 90°) (d)
4	<i>Variance</i>	Wariancja (a)
5	<i>Perc.01%</i>	Percentyl 1% (a)

Tabela 8.5. Pełny zestaw cech wykorzystany do budowy modelu RFE-NuSVM-10 na podstawie małego zbioru obrazów płomienia. Kolejność cech jest zgodna z ich rankingiem dokonany za pomocą metody RFE

Nr	Nazwa cechy	Znaczenie cechy (źródło informacji)
1	<i>Teta3</i>	Parametr Q_3 (c)
2	<i>Perc.01%</i>	Percentyl 1% (a)
3	<i>S(1,0)Contrast</i>	Kontrast (odległość – 1, kierunek – 90°) (d)
4	<i>WavEnLH_s-1</i>	Energia współczynnika falkowego dla podpasma LH w 1 skali (f)
5	<i>Vertl_ShrtREmp</i>	Odwrotny moment uwytatnienia krótkich pasm dla kierunku 90° (e)
6	<i>WavEnHH_s-2</i>	Energia współczynnika falkowego dla podpasma HH w 2 skali (f)
7	<i>Kurtosis</i>	Kurtoza (a)
8	<i>Teta4</i>	Parametr Q_4 (c)
9	<i>Mean</i>	Średnia (a)
10	<i>Horzl_GLevNonU</i>	Niejednorodność szarości dla kierunku 0° (e)

Tabela 8.6. Pełny zestaw cech wykorzystany do budowy modelu R-MLP-17 na podstawie dużego zbioru obrazów zmian skórnych. Kolejność cech jest zgodna z ich rankingiem dokonany za pomocą metody Relief

Nr	Nazwa cechy	Znaczenie cechy (źródło informacji)
1	<i>Kurtosis</i>	Kurtoza (a)
2	<i>Horzl_LngREmp</i>	Moment uwytatnienia długich pasm dla kierunku 0° (e)
3	<i>WavEnHL_s-3</i>	Energia współczynnika falkowego dla podpasma HL w 3 skali (f)
4	<i>WavEnLH_s-3</i>	Energia współczynnika falkowego dla podpasma LH w 3 skali (f)
5	<i>Teta4</i>	Parametr Q_4 (c)
6	<i>Teta3</i>	Parametr Q_3 (c)
7	<i>Teta1</i>	Parametr Q_1 (c)
8	<i>GrKurtosis</i>	Kurtoza (b)
9	<i>Variance</i>	Wariancja (a)
10	<i>GrSkewness</i>	Skośność (b)

c. d. Tabeli 8.6. Pełny zestaw cech wykorzystany do budowy modelu R-MLP-17 na podstawie dużego zbioru obrazów zmian skórnych. Kolejność cech jest zgodna z ich rankingiem dokonany za pomocą metody Relief

Nr	Nazwa cechy	Znaczenie cechy (źródło informacji)
11	<i>Horzl_GLevNonU</i>	Niejednorodność szarości dla kierunku 0° (e)
12	<i>Skewness</i>	Skośność (a)
13	<i>S(1,0)Entropy</i>	Entropia (odległość – 1, kierunek – 90°) (d)
14	<i>S(1,0)SumEntrp</i>	Sumaryczna entropia (odległość – 1, kierunek – 90°) (d)
15	<i>S(1,0)InvDfMom</i>	Odwrotny moment różnicowy (odległość – 1, kierunek – 90°) (d)
16	<i>S(1,0)Correlat</i>	Korelacja (odległość – 1, kierunek – 90°) (d)
17	<i>Mean</i>	Średnia (a)

Tabela 8.7. Ograniczony zestaw cech wykorzystany do budowy modelu A-MLP-3 na podstawie dużego zbioru obrazów zmian skórnych. Kolejność cech jest zgodna z ich rankingiem dokonany za pomocą metody ANOVA

Nr	Nazwa cechy	Znaczenie cechy (źródło informacji)
1	<i>Perc.01%</i>	Percentyl 1% (a)
2	<i>Mean</i>	Średnia (a)
3	<i>GrSkewness</i>	Skośność (b)

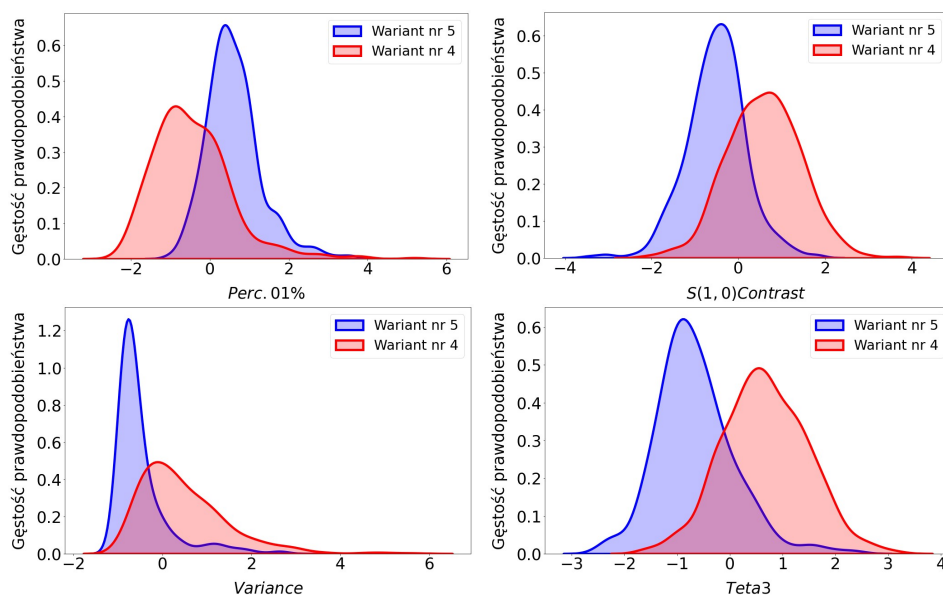
Tabela 8.8. Pełny zestaw cech wykorzystany do budowy modelu F-MLP-2 na podstawie małego zbioru obrazów zmian skórnych. Kolejność cech jest zgodna z ich rankingiem dokonany za pomocą metody Fishera

Nr	Nazwa cechy	Znaczenie cechy (źródło informacji)
1	<i>Mean</i>	Średnia (a)
2	<i>Horzl_LngREmph</i>	Moment uwydatnienia długich pasm dla kierunku 0° (e)

Na podstawie informacji zamieszczonych w tabelach 8.3÷8.8 określono zbiór cech, które były najczęściej wykorzystywane do budowy modeli. W zestawieniu zaprezentowanym w tabeli 8.9 uwzględniono tylko te cechy, które zajmowały 5 pierwszych pozycji na każdej liście rankingowej. W przypadku obrazów płomienia, za najskuteczniejsze należy uznać 4 cechy. Pierwszą jest percentyl 1% (*Perc.01%*), obliczony na podstawie wektora histogramu obrazu. Cecha ta wykorzystana była do budowy wszystkich modeli dla obrazów płomienia oraz jednego modelu dla obrazów zmian skórnych (tab. 8.9). Drugą cechą jest kontrast (*S(1,0)Contrast*), obliczony na podstawie macierzy przejść tonalnych dla odległości między pikselami równej 1, w kierunku 90°. Każda z kolejnych dwóch cech użyta była do budowy dwóch modeli dla obrazów płomienia. Wspomniane cechy to: wariancja (*Variance*), obliczona na podstawie wektora histogramu obrazu oraz parametr Q_3 (*Teta3*), do obliczenia którego wykorzystano model autoregresji. Na rysunku 8.5 przedstawiono rozkłady gęstości prawdopodobieństwa obserwacji dla czterech wyżej wymienionych cech. Wykresy dotyczą zbioru testowego wyłonięgo z dużego zbioru obrazów płomienia.

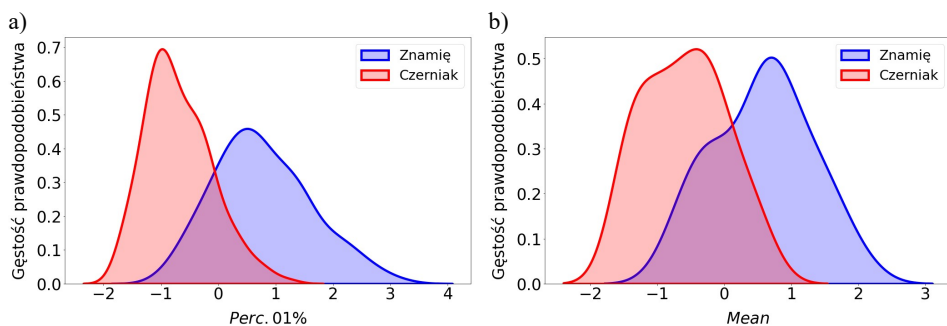
Tabela 8.9. Cechy obrazu najczęściej wykorzystywane do budowy modeli uznanych za najskuteczniejsze dla poszczególnych wariantów zbioru danych oraz zestawu cech

Nazwa cechy	Znaczenie cechy (źródło informacji)	Modele zbudowane na podstawie obrazów płomienia	Modele zbudowane na podstawie obrazów zmian skórnych
<i>Perc.01%</i>	Percentyl 1% (a)	SFS-GRAD-18, A-SVM-5, RFE-NuSVM-10	A-MLP-3
<i>S(1,0)Contrast</i>	Kontrast (odległość – 1, kierunek – 90°) (d)	SFS-GRAD-18, A-SVM-5, RFE-NuSVM-10	
<i>Variance</i>	Wariancja (a)	SFS-GRAD-18, A-SVM-5	
<i>Teta3</i>	Parametr Q_3 (c)	A-SVM-5, RFE-NuSVM-10	
<i>Mean</i>	Średnia (a)	SFS-GRAD-18	A-MLP-3, F-MLP-2
<i>Horzl_LngREmph</i>	Moment uwydatnienia długich pasm dla kierunku 0° (e)		R-MLP-17, F-MLP-2



Rys. 8.5. Rozkłady gęstości prawdopodobieństwa obserwacji dla dużego zbioru obrazów płomienia (zbiór testowy), uwzględniające 4 cechy najczęściej wykorzystywane do budowy najskuteczniejszych modeli

Dla obrazów zmian skórnych, szczególnie istotne okazały się 2 cechy. Pierwszą była średnia intensywność pikseli (*Mean*), obliczona na podstawie wektora histogramu obrazu. Została ona wykorzystana do budowy dwóch modeli dla obrazów zmian skórnych oraz jednego modelu dla obrazów płomienia (tab. 8.9). Cecha druga to moment uwydatnienia długich pasm dla kierunku 0° (*Horz_LngREmph*), obliczony z wykorzystaniem macierzy rozkładu długości pasm. Tę cechę zastosowano do budowy dwóch modeli dla obrazów zmian skórnych. Na podkreślenie zasługuje także duża skuteczność percentyla 1% (*Perc.01%*), który został wykorzystany do budowy wszystkich modeli dla obrazów płomienia, ale także był najistotniejszą cechą użytą do budowy modelu A-MLP-3 dla obrazów zmian skórnych. Na rysunku 8.6 przedstawiono rozkład gęstości prawdopodobieństwa obserwacji dla dwóch cech, które okazały się najistotniejsze w budowie dwóch modeli dla obrazów zmian skórnych. Wspomniany wykres został wykonany na podstawie zbiorów testowych.

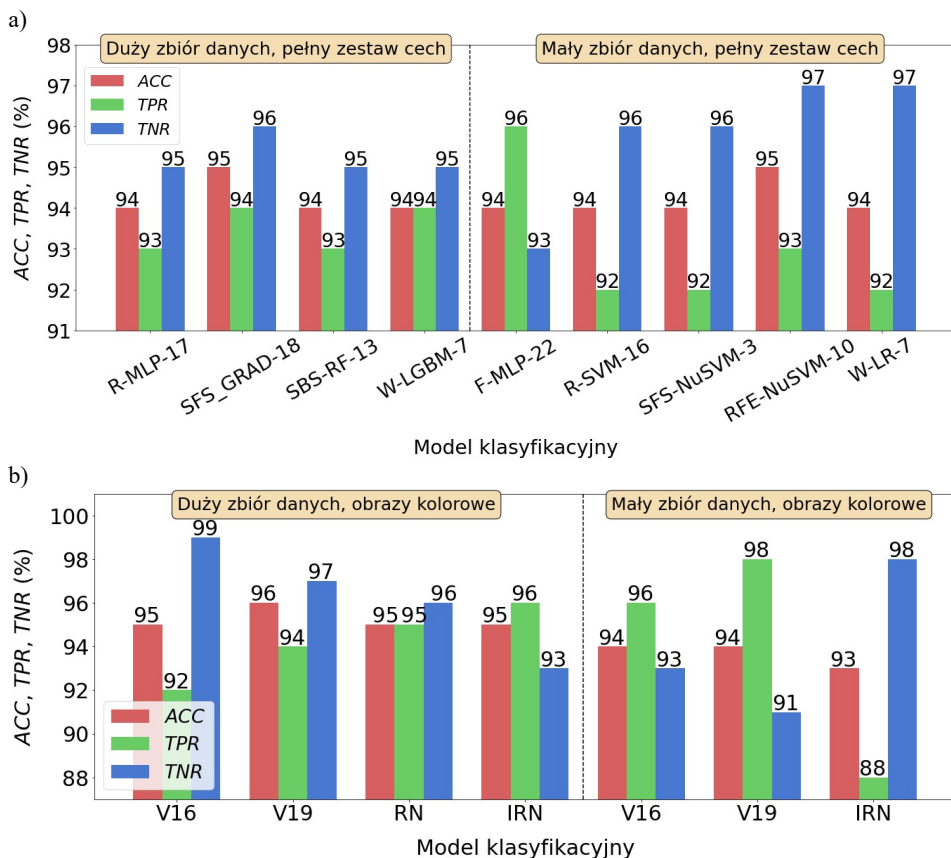


Rys. 8.6. Rozkłady gęstości prawdopodobieństwa obserwacji dla cech, które były najistotniejsze w budowie modeli: a) A-MLP-3 na podstawie dużego zbioru danych i ograniczonego zestawu cech (tab. 8.7); b) F-MLP-2 na podstawie małego zbioru danych i pełnego zestawu cech (tab. 8.8)

8.4. Wrażliwość modeli na wielkość zbioru danych

Duży zbiór obrazów płomienia zawierał 3000 obserwacji należących do dwóch klas (wariant nr 5 i wariant nr 4). Dokładność najskuteczniejszych modeli uczenia tradycyjnego była dla tych danych na poziomie 94÷95% (rys. 8.7a). Zmniejszenie liczby obserwacji do 600 (5 razy) nie miało wpływu na pogorszenie ogólnej dokładności klasyfikacji. Oczywiście, zestaw najskuteczniejszych modeli różnił się od tego dla dużego zbioru danych, ale ich dokładność była także na poziomie 94÷95%. Zmniejszenie wielkości zbioru danych miało natomiast drobny wpływ na skuteczność wykrywania wariantu nr 4 (nadmiar powietrza) i wariantu nr 5 (normalne warunki spalania). W przypadku wariantu nr 4, dokładność (*TPR*) nieco zmalała (wyjątkiem był model F-MLP-22), a dla wariantu nr 5 (*TNR*) – parametr ten wzrósł.

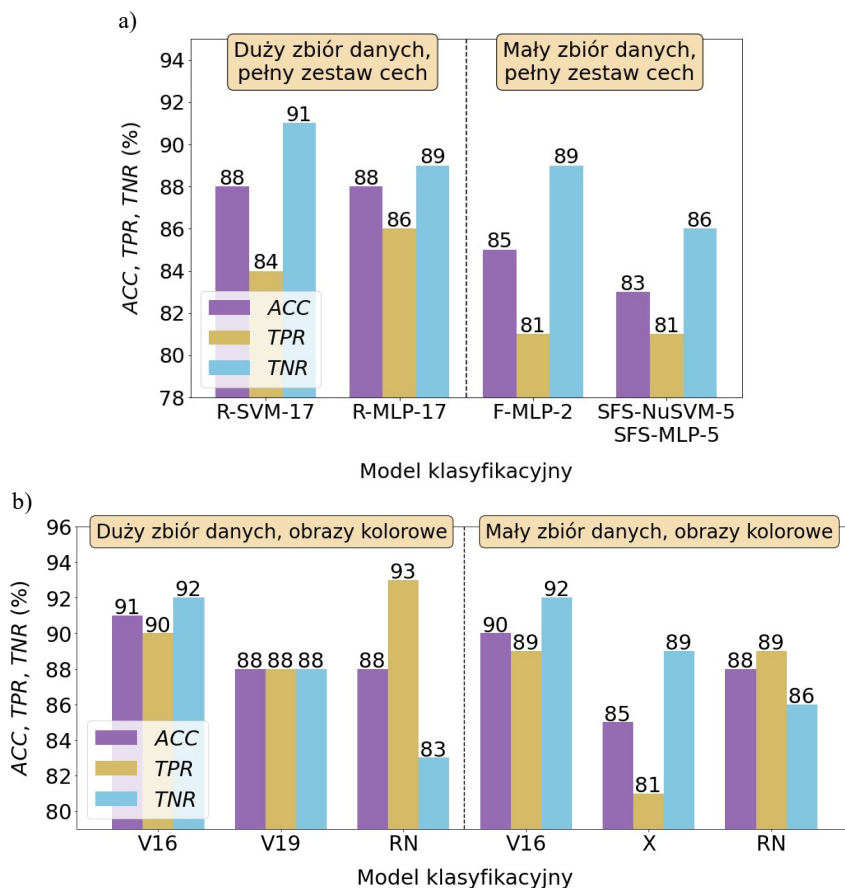
Rozważając modele uczenia głębokiego, dla dużego zbioru danych, parametr *ACC* był na poziomie 95÷96%. Pięciokrotne zmniejszenie wielkości zbioru spowodowało nieznaczny spadek możliwości do uzyskania dokładności, do zakresu 93÷94% (rys. 8.7b). Dla modelu VGG19⁸, który okazał się najskuteczniejszy dla obu postaci zbioru danych, różnica ta wyniosła 2%. Najmniej wrażliwy na wielkość zbioru danych okazał się model VGG16. Parametr *ACC* zmniejszył się w jego przypadku z 95% do 94%, a więc tylko o 1%.



Rys. 8.7. Wyniki testowania najskuteczniejszych modeli zbudowanych na podstawie dużego i małego zbioru obrazów płomienia: a) uczenie tradycyjne; b) uczenie głębokie

⁸ Dla uproszczenia rozważań, stosujemy tutaj skrót myślowy. W rzeczywistości, mamy do czynienia z dwoma różnymi modelami (dla dużego i małego zbioru danych), zbudowanymi w oparciu o architekturę VGG19. Podobna sytuacja występuje w przypadku pozostałych modeli.

Pełny zbiór obrazów zmian skórnych składał się z 1200 zdjęć należących do znamion oraz czerniaków skóry. Podobnie, jak dla obrazów płomienia, także i w tym przypadku, mały zbiór danych powstał w wyniku pięciokrotnej redukcji liczebności zbioru pełnego. Najlepsze modele uczenia tradycyjnego osiągnęły dokładność dla zbioru pełnego na poziomie 88%. Po zmniejszeniu wielkości zbioru danych do 240 obserwacji, dokładność oferowana przez najskuteczniejsze modele była na poziomie 83÷85% (rys. 8.8a). Zatem, nastąpiło pogorszenie o 3% najlepszej, możliwej do uzyskania dokładności modeli.



Rys. 8.8. Wyniki testowania najskuteczniejszych modeli zbudowanych na podstawie dużego i małego zbioru obrazów zmian skórnych: a) uczenie tradycyjne; b) uczenie głębokie

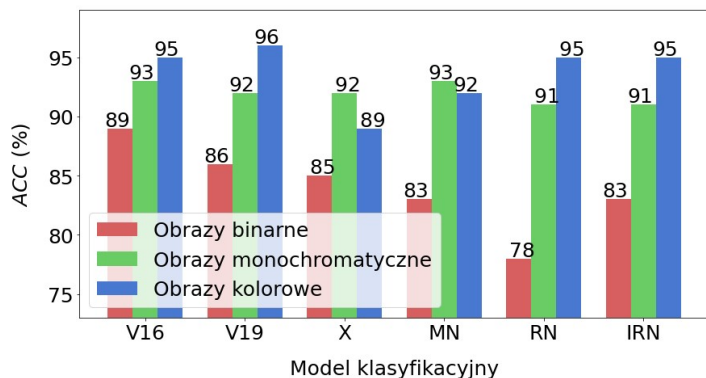
Mniej wrażliwe na zmniejszenie liczebności obserwacji okazały się modele uczenia głębokiego. Dla zbioru dużego, osiągnęły one dokładność na poziomie 88÷91% (rys. 8.8b). Zmniejszenie wielkości zbioru danych spowodowało tylko

nieznaczny spadek dokładności najskuteczniejszych modeli. Parametr *ACC* wahał się tym przypadku w zakresie 85÷90%. Na uwagę zasługuje model VGG16. W wyniku zmniejszenia liczebności obserwacji, podobnie, jak dla obrazów płomienia, dokładność tego modelu zmniejszyła się tylko o 1% (spadek z 91% do 90%). Poza tym, wspomniany model okazał się najskuteczniejszy dla obu postaci zbioru danych. Należy także podkreślić odporność modelu ResNet50 na redukcję liczebności obserwacji. Dla obu wariantów zbioru, dokładność tego modelu miała wartość 88%.

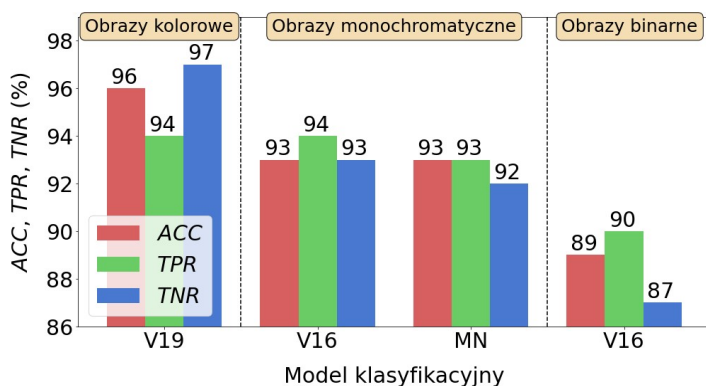
8.5. Wpływ trybu koloru na dokładność klasyfikacji

W przypadku kolorowych obrazów płomienia, najskuteczniejszy model (VGG19) osiągnął dokładność równą 96% (rys. 8.9). Z kolei, najlepsze modele dla obrazów monochromatycznych (VGG16 i MobileNetV2) uzyskały wartość *ACC* na poziomie 93%. Zatem, nastąpiło pogorszenie rozważanego parametru o 3%. Taka jest też ogólna tendencja, w wyniku zmiany postaci obrazów z kolorowych na monochromatyczne. Dla obrazów szarych, dokładność czterech z sześciu testowanych modeli pogorszyła się od 2% do 4%, w porównaniu z obrazami kolorowymi. Wyjątkiem są modele Xception i MobileNetV2, dla których wartość *ACC* wzrosła, odpowiednio o 3% oraz 1%. Po zastosowaniu obrazów binarnych, nastąpiło znacznie większe pogorszenie dokładności. W porównaniu z obrazami kolorowymi, różnica wartości tego parametru dla poszczególnych modeli wyniosła od 4% do 17%.

Na rysunku 8.10 porównano wyniki testowania najskuteczniejszych modeli, dla różnych trybów koloru. Ogólna dokładność klasyfikacji (*ACC*) modeli zbudowanych na podstawie obrazów szarych (VGG16 i MobileNetV2) jest mniejsza tylko o 3% od dokładności najlepszego modelu dla obrazów kolorowych (VGG19). Natomiast, nie nastąpiło pogorszenie dokładności wykrywania wariantu spalania nr 4 (parametr *TPR*). Zarówno dla obrazów kolorowych, jak i monochromatycznych, dla najskuteczniejszych modeli, wartość wspomnianego parametru była na poziomie 94%. Na uwagę zasługuje fakt, iż dla obrazów szarych, wszystkie rozważane indeksy jakości (*ACC*, *TPR*, *TNR*) modeli VGG19 i MobileNetV2 osiągnęły wyrównany i jednocześnie wysoki poziom. Świadczy to o wysokiej jakości wspomnianych modeli. Z kolei, dla obrazów binarnych, najskuteczniejszy okazał się model VGG16. Niestety, w jego przypadku nastąpiło dość istotne pogorszenie jakości, w porównaniu z najlepszym modelem dla obrazów kolorowych (VGG19). Wartości poszczególnych indeksów zmniejszyły się następująco: *ACC* – 7%, *TPR* – 4%, *TNR* – 10%.



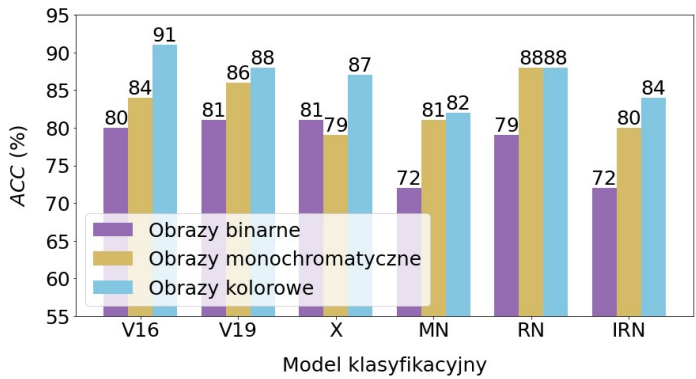
Rys. 8.9. Wpływ trybu koloru na dokładność testowania modeli uczenia głębokiego, zbudowanych na podstawie dużego zbioru obrazów płomienia



Rys. 8.10. Wyniki testowania najskuteczniejszych modeli uczenia głębokiego, zbudowanych na podstawie dużego zbioru obrazów płomienia, dla różnych trybów koloru

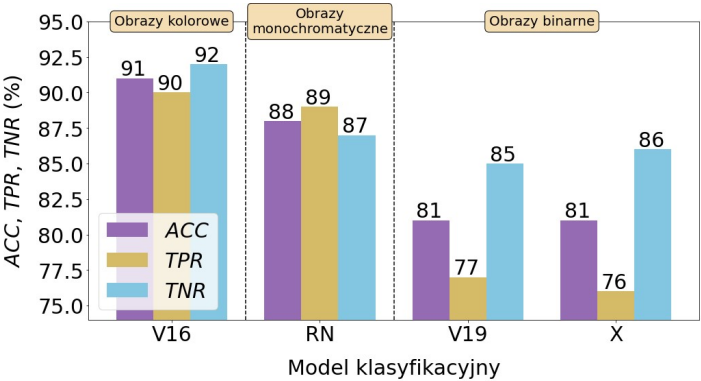
Na rysunku 8.11. przedstawiono wyniki testowania modeli uczenia głębokiego, zbudowanych na podstawie kolorowych obrazów zmian skórnych. Najskuteczniejszy okazał się model VGG16 ($ACC = 91\%$). Nieco gorszą dokładność osiągnęły modele VGG19 ($ACC = 88\%$), ResNet50 ($ACC = 88\%$) oraz Xception ($ACC = 87\%$). Zastosowanie obrazów monochromatycznych, generalnie spowodowało zmniejszenie dokładności modeli, ale w większości przypadków, spadek ten nie był zbyt duży. Na uwagę zasługuje model ResNet50, który okazał się niewrażliwy na zmianę trybu koloru. Zarówno dla obrazów kolorowych, jak i szarych, jego dokładność wyniosła 88% . W przypadku dwóch innych modeli, wystąpił bardzo mały spadek dokładności. Mianowicie, dla MobileNetV2 wartość ACC zmalała z 82% dla obrazów kolorowych do 81% dla monochromatycznych, a w przypadku VGG19 nastąpił spadek dokładności o 2% . Zastoso-

wanie obrazów binarnych spowodowało, że skuteczność modeli spadła bardziej, niż miało to miejsce dla obrazów szarych. Różnica dokładności, w porównaniu z obrazami kolorowymi, wyniosła od 6% do 12%.



Rys. 8.11. Wpływ trybu koloru na dokładność testowania modeli uczenia głębokiego, zbudowanych na podstawie dużego zbioru obrazów zmian skórnych

Wyniki ewaluacji najskuteczniejszych modeli, dla różnych trybów koloru, przedstawiono na rysunku 8.12. Największą dokładnością ($ACC = 91\%$) charakteryzuje się model VGG16, zbudowany na podstawie obrazów kolorowych. Także pozostałe indeksy jakości są na wysokim poziomie ($TPR = 90\%$, $TNR = 92\%$).



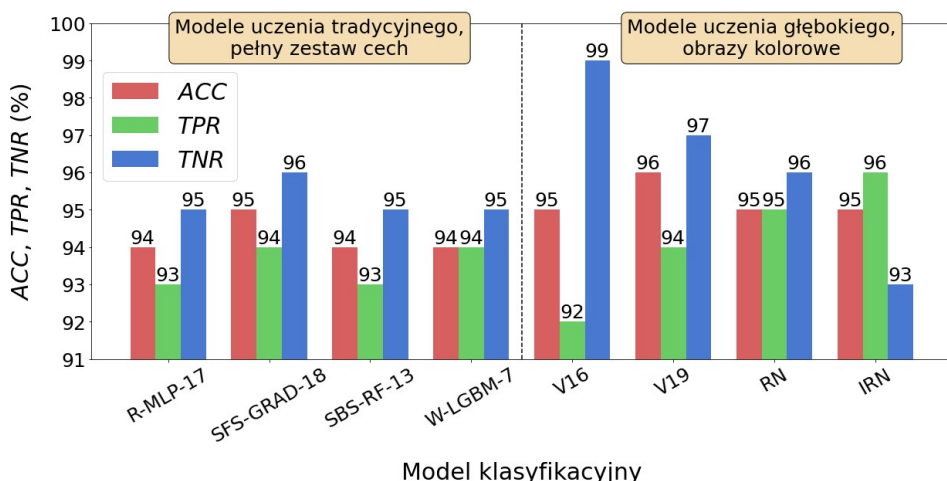
Rys. 8.12. Wyniki testowania najskuteczniejszych modeli uczenia głębokiego, zbudowanych na podstawie dużego zbioru obrazów zmian skórnych, dla różnych trybów koloru

Najlepszy model, zbudowany na podstawie obrazów szarych, tj. ResNet50, osiągnął dokładność na poziomie 88% oraz specyficzność (TNR) równą 87%. Wyni-

ki te są gorsze, odpowiednio, o 3% i 5% w porównaniu z modelem VGG16 dla obrazów kolorowych. Porównując oba powyższe modele, należy jednak zauważyć, że czułość (*TPR*) zmniejszyła się z 90% dla VGG16 do 89% dla ResNet50, a więc tylko o 1%. Jest to sytuacja korzystna dla pacjentów, z uwagi na wysoką skuteczność wykrywania przypadków dotkniętych czerniakiem skóry. Po zastosowaniu obrazów binarnych, najwyższa dokładność testowania wystąpiła dla modeli VGG19 oraz Xception ($ACC = 81\%$). W porównaniu z modelem VGG16 dla obrazów kolorowych ($ACC = 91\%$), pogorszenie dokładności klasyfikacji było znaczne, gdyż wynosiło aż 10%.

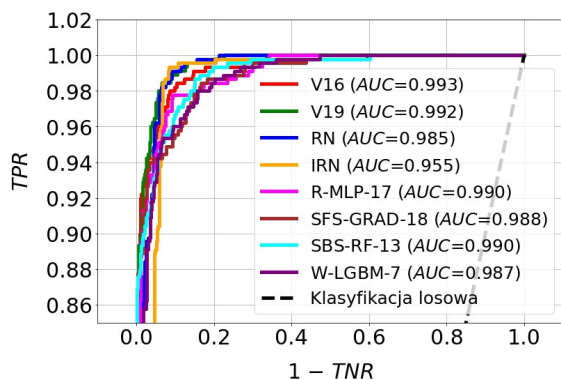
8.6. Porównanie skuteczności modeli predykcyjnych

Na rysunku 8.13 porównano wyniki ewaluacji najskuteczniejszych modeli uczenia tradycyjnego i głębokiego, zbudowanych na podstawie dużego zbioru obrazów płomienia. Ogólna dokładność klasyfikacji (ACC) modeli uczenia tradycyjnego jest na poziomie 94÷95%. Najskuteczniejszy model w tej grupie to SFS-GRAD-18 ($ACC = 95\%$, $TPR = 94\%$, $TNR = 96\%$). W przypadku modeli uczenia głębokiego, parametr ACC osiąga wartość w zakresie 95÷96%. Największą dokładność uzyskał tutaj model VGG19 ($ACC = 96\%$, $TPR = 94\%$, $TNR = 97\%$). Widać, iż przy tej samej wartości TPR , pozostałe 2 wskaźniki jakości wymienionych modeli (ACC , TNR) różnią się zaledwie o 1%. Można więc uznać, iż generalnie, skuteczność obu kategorii modeli w klasyfikacji obrazów płomienia jest porównywalna.



Rys. 8.13. Wyniki testowania najskuteczniejszych modeli uczenia tradycyjnego i głębokiego, zbudowanych na podstawie dużego zbioru obrazów płomienia

Obserwując wykresy ROC przedstawione na rysunku 8.14 widać, iż jakość wszystkich modeli uczenia tradycyjnego, mierzona wielkością pola powierzchni pod krzywą ROC, jest bardzo zbliżona. Parametr AUC jest w tym przypadku na poziomie $0,987 \div 0,990$. Biorąc pod uwagę modele uczenia głębokiego, należy wyróżnić VGG16 i VGG19, dla których AUC osiągnął wartość, odpowiednio, 0,993 oraz 0,992. Zdecydowanie najgorzej na wykresie ROC wypadł model InceptionResnetV2, dla którego parametr AUC był równy 0,955.

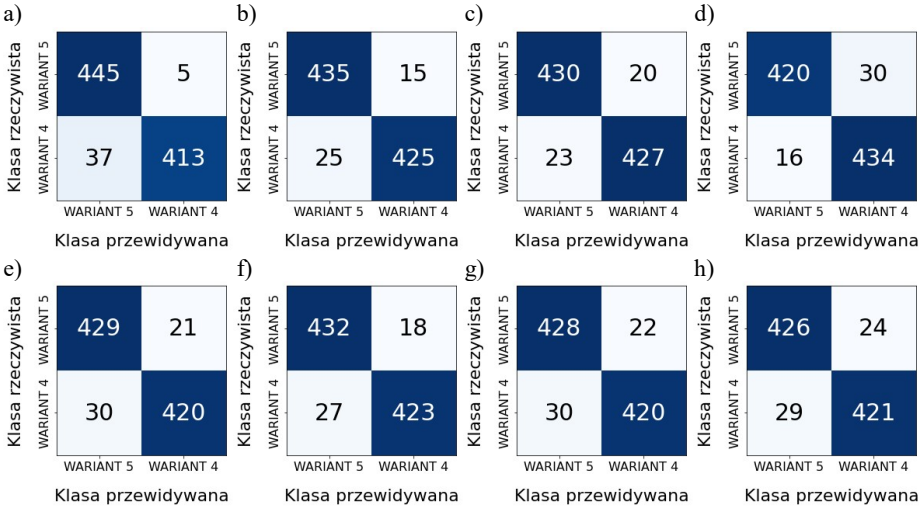


Rys. 8.14. Krzywe ROC wraz z parametrami AUC dla najskuteczniejszych modeli zbudowanych z wykorzystaniem dużego zbioru obrazów płomienia

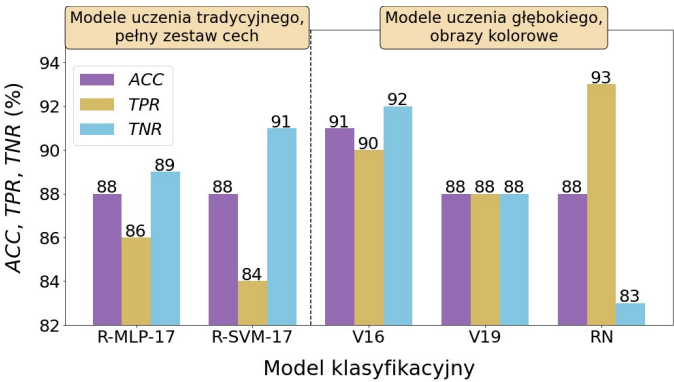
Macierze pomyłek poszczególnych modeli, podczas klasyfikacji obrazów należących do zbioru testowego, zaprezentowano na rysunku 8.15. Najlepszy model uczenia tradycyjnego, tj. SFS-GRAD-18, pomylił się w 18-tu przypadkach, rozpoznając wariant nr 5 jako wariant nr 4. Z kolei, podczas rozpoznawania wariantu nr 4, wystąpiło 27 przypadków mylnie wykrytych jako wariant nr 5. Wyniki te są porównywalne z indeksami jakości najlepszego modelu uczenia głębokiego, tj. VGG19. W tym ostatnim przypadku, wariant nr 5 został 15 razy mylnie rozpoznany jako wariant nr 4 (3 błędy mniej, niż SFS-GRAD-18), a liczba pomyłek przy rozpoznawaniu wariantu nr 4 wyniosła 25 (2 błędy mniej, niż SFS-GRAD-18). Tak małe różnice pomiędzy błędami obu modeli świadczą o ich porównywalnej skuteczności.

Wyniki ewaluacji najskuteczniejszych modeli uczenia tradycyjnego i głębokiego, zbudowanych na podstawie dużego zbioru obrazów zmian skórnych, zaprezentowano na rysunku 8.16. W grupie modeli uczenia tradycyjnego, najskuteczniejsze okazały się R-MLP-17 i R-SVM-17. W obu przypadkach, dokładność klasyfikacji wyniosła 88%. W grupie modeli uczenia głębokiego, najdokładniejszy był VGG16. W porównaniu z dwoma poprzednimi modelami, jego dokładność była o 3% wyższa ($ACC = 91\%$). Także 2 pozostałe indeksy jakości były o kilka procent większe, niż adekwatne parametry modeli uczenia tradycyj-

nego. Przedstawione wyniki skłaniają do dwóch wniosków. Po pierwsze, do klasyfikacji obrazów zmian skórnych należy rekomendować modele uczenia głębokiego, ponieważ ich przedstawiciel (VGG16) uzyskał dokładność wyższą o 3% od dokładności najlepszych modeli uczenia tradycyjnego. Po drugie, dokładność na poziomie 88% uzyskana przez modele R-MLP-17 i R-SVM-17 jest także bardzo dobrym wynikiem. Świadczy on o tym, że modele uczenia tradycyjnego mogą skutecznie konkurować z modelami uczenia głębokiego w obszarze klasyfikacji zmian skórnych.

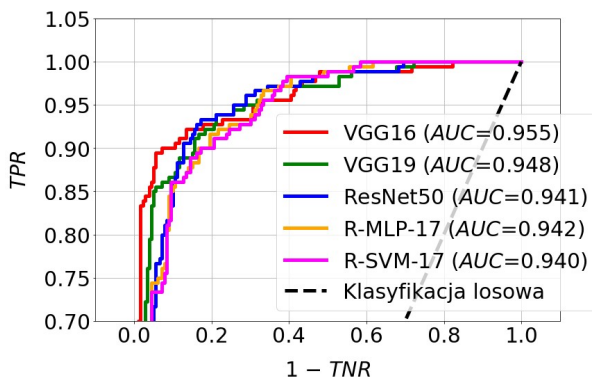


Rys. 8.15. Macierze pomyłek: a) VGG16; b) VGG19; c) ResNet50; d) InceptionResnetV2; e) R-MLP-17; f) SFS-GRAD-18; g) SBS-RF-13; h) W-LGBM-7



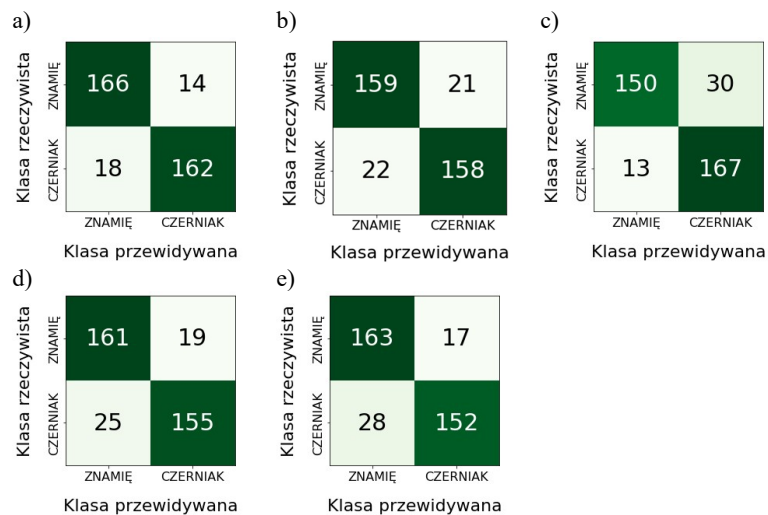
Rys. 8.16. Wyniki testowania najskuteczniejszych modeli uczenia tradycyjnego i głębokiego, zbudowanych na podstawie dużego zbioru obrazów zmian skórnych

Wcześniejsze wnioski na temat jakości modeli znajdują odzwierciedlenie w przebiegu krzywych ROC przedstawionych na rysunku 8.17. Najmniejszy obszar pod krzywą ROC należy do modeli R-SVM-17 ($AUC = 0,940$) i R-MLP-17 ($AUC = 0,942$). W przypadku modeli należących do drugiej grupy, wspomniany obszar jest większy. Dla najlepszego modelu VGG16, parametr AUC jest równy 0,955 (wartość najwyższa). W przypadku dwóch pozostałych modeli, tj. VGG19 i ResNet50, AUC przyjmuje wartość, odpowiednio, 0,948 i 0,941.



Rys. 8.17. Krzywe ROC wraz z parametrami AUC dla najskuteczniejszych modeli zbudowanych z wykorzystaniem dużego zbioru obrazów zmian skórnych

Na rysunku 8.18 zaprezentowano macierze pomyłek, zbudowane w wyniku klasyfikacji obrazów zmian skórnych należących do zbioru testowego. Model VGG16, który okazał się najskuteczniejszy, 14 razy błędnie rozpoznał znamiona jako czerniaki skóry. Z kolei, podczas rozpoznawania czerniaków, liczba pomyłek wyniosła 18. Biorąc pod uwagę modele R-MLP-17 i R-SVM-17, pomyliły się one przy rozpoznawaniu znamion, odpowiednio, 19 i 17 razy oraz 25 i 28 razy podczas rozpoznawania czerniaków. Ten drugi rodzaj pomyłek jest bardzo istotny z medycznego punktu widzenia, ponieważ oznacza on, że nie są wykrywane przypadki choroby. W tym kontekście, model VGG16 jest korzystniejszy dla pacjenta, gdyż w jego przypadku, tylko 18 obrazów czerniaka zostało rozpoznanych jako znamiona.



Rys. 8.18. Macierze pomyłek: a) VGG16; b) VGG19; c) ResNet50; d) R-MLP-17; e) R-SVM-17

9. Podsumowanie

Analiza wyników badań, przeprowadzona w rozdziale nr 8, daje odpowiedź na ważne pytania postawione w początkowej części monografii. Pierwszą poruszoną kwestią był wpływ ograniczenia zestawu cech diagnostycznych na dokładność modeli uczenia tradycyjnego. Uzyskane wyniki pokazały, iż ograniczenie liczby cech nie powoduje zmniejszenia skuteczności modeli. Obserwowany spadek dokładności o 1% jest praktycznie pomijalny, ponieważ często może on wynikać z zaokrąglenia liczb rzeczywistych, różniących się w istocie ułamkiem procenta. Należy zatem z pewną rezerwą podchodzić do wyników zwracanych przez procedury numeryczne, które wspomagają wybór cech diagnostycznych. Dodatkowa analiza tych wyników i podjęcie świadomych decyzji przez osobę prowadzącą badania może poprawić jakość wyników końcowych, co w kontekście ograniczenia liczby cech przekłada się bezpośrednio na prostszy model i łatwiejszą implementację.

Wyniki zamieszczone w rozdziale 8.2 dają przekrój metod selekcji i klasyfikacji, najlepszych do rozwiązywania postawionych zadań diagnostycznych. Przypomnijmy, że zadaniami tymi była klasyfikacja obrazów płomienia oraz obrazów zmian skórnych. W tym pierwszym przypadku, nie można wskazać jednej metody selekcji i klasyfikacji, która byłaby najlepsza dla wszystkich wariantów dotyczących wielkości danych oraz zestawu cech. W przypadku obrazów zmian skórnych, niezależnie od postaci zbioru danych i zestawu cech, najskuteczniejszą metodą klasyfikacji okazała się metoda MLP. Nie można natomiast wskazać metody selekcji o podobnych własnościach.

Pamiętajmy, że w praktyce, zbiór danych może się znacząco różnić od tego wykorzystanego w monografii. Zależnie od specyfiki rozwiązywanego problemu, różnice mogą dotyczyć zarówno liczebności obserwacji, jak i wielkości oraz struktury zestawu cech. W takiej sytuacji, wyboru metod selekcji i klasyfikacji nie należy dokonywać na podstawie wąskiej grupy metod. Często prowadzi to do pominięcia metod, które mogą okazać się skuteczniejsze od tych wybranych. Sugerowanym podejściem jest porównanie wyników uzyskanych dla szerszego spektrum metod selekcji i klasyfikacji. Mamy wówczas większą pewność wyboru metody najwłaściwszej dla danej sytuacji. Jest to istotne także dlatego, że, jak wykazano w monografii, dla danych różniących się wielkością zbioru oraz zestawem cech, metody selekcji i klasyfikacji zastosowane do budowy najskuteczniejszych modeli – różnią się.

Uzyskane wyniki pozwoliły wybrać najskuteczniejsze zestawy cech do budowy modeli klasyfikacyjnych. W przypadku obrazów płomienia, są to: percentyl 1% (*Perc.01%*), obliczony na podstawie wektora histogramu obrazu; kontrast (*S(I,0)Contrast*), obliczony na podstawie macierzy przejść tonalnych dla odległości między pikselami równej 1, w kierunku 90°; wariancja (*Variance*), obliczona na podstawie wektora histogramu obrazu oraz parametr Q_3 (*Teta3*), do

obliczenia którego wykorzystano model autoregresji. Podczas budowy modeli dla obrazów zmian skórnych, skuteczne okazały się 2 cechy: średnia intensywność pikseli (*Mean*), obliczona na podstawie wektora histogramu obrazu oraz moment uwydatnienia długich pasm dla kierunku 0° (*Horzl_LngREmph*), obliczony z wykorzystaniem macierzy rozkładu długości pasm.

Obserwując rozkłady gęstości prawdopodobieństwa obserwacji przedstawione na rysunkach 8.5 i 8.6 widzimy, iż w większym lub mniejszym stopniu zachodzą one na siebie. Idealne warunki do dyskryminacji obserwacji należących do dwóch różnych klas występowałyby wówczas, gdyby wspomniane rozkłady były rozłączne. Niestety, w rzeczywistości rzadko dysponujemy danymi o takich własnościach. Najczęściej, rozkłady gęstości dla poszczególnych cech częściowo pokrywają się. Taka sytuacja występuje w przypadku danych będących przedmiotem badań opisanych w niniejszej monografii. Należy zaznaczyć, iż wykresy przedstawione na rysunkach 8.5 i 8.6 dotyczą cech najistotniejszych, a więc tych, które mają najlepsze własności dyskryminacyjne. Fakt, iż dla tego rodzaju cech rozkłady gęstości nie są rozłączne, lecz w pewnym zakresie nakładają się, świadczy o wysokiej złożoności danych oraz dużym stopniu trudności zadania klasyfikacji, jakie było rozwiązywane.

Analiza wyników zawarta w rozdziale 8.4 pozwala ocenić wpływ wielkości zbioru danych na dokładność modeli uczenia tradycyjnego i głębokiego. W przypadku obrazów płomienia, pięciokrotne zmniejszenie liczebności obserwacji nie miało wpływu na pogorszenie ogólnej dokładności klasyfikacji modeli uczenia tradycyjnego. Dla obu wariantów zbioru danych, najwyższa dokładność wyniosła 95%. Rozważając modele uczenia głębokiego, największa dokładność, która była możliwa do uzyskania, zmniejszyła się z 96% do 94% (pogorszenie o 2%). Model VGG16 okazał się najmniej wrażliwy na wielkość zbioru danych. Parametr *ACC* zmniejszył się w jego przypadku z 95% do 94%, a więc tylko o 1%. Dla obrazów zmian skórnych, wrażliwość modeli uczenia tradycyjnego na wielkość zbioru danych okazała się nieco większa, niż miało to miejsce dla obrazów płomienia. Zmniejszenie liczebności obserwacji spowodowało w tym przypadku pogorszenie o 3% (spadek z 88% do 85%) najlepszej, możliwej do uzyskania dokładności. Mniej wrażliwe na redukcję wielkości zbioru okazały się modele uczenia głębokiego. W tej grupie, najskuteczniejszy dla obu postaci zbioru danych był model VGG16. Jego dokładność zmniejszyła się tylko o 1% (spadek z 91% do 90%). Podsumowując można stwierdzić, że ograniczenie wielkości zbioru danych nie wpływa w sposób istotny na skuteczność modeli. Ta własność jest szczególnie cenna dla modeli uczenia głębokiego, ponieważ pozwala je stosować także dla zbiorów danych o stosunkowo małej liczebności. Przypomnijmy, że jest to możliwe dzięki wykorzystaniu modeli, które zostały wcześniej wytrenowane na dużym zbiorze danych (*transfer learning*).

Na podstawie przeprowadzonych badań można ocenić wpływ trybu koloru obrazów na dokładność modeli uczenia głębokiego. Wyniki testowania przed-

stawione na rysunkach 8.9÷8.12 pokazują, iż dla obu rodzajów danych (obrazy płomienia oraz zmian skórnych), modele zbudowane z wykorzystaniem obrazów kolorowych oferują najwyższą skuteczność klasyfikacji. Taka sytuacja jest zgodna z naszym oczekiwaniem, gdyż jak wiadomo, obrazy kolorowe mają największą nośność informacyjną. Na uwagę zasługuje jednak jakość modeli zbudowanych w oparciu o obrazy monochromatyczne. Okazuje się, że generalnie, ich dokładność jest tylko nieznacznie gorsza od dokładności modeli dla obrazów kolorowych. Zależnie od modelu, różnica ta wynosi przeważnie od 1% do 4%. Zdarzają się też przypadki, że po zastosowaniu obrazów szarych, dokładność modelu jest taka sama lub nieznacznie większa, w porównaniu z obrazami kolorowymi.

W świetle zaprezentowanych wyników, można powiedzieć, że modele uczenia głębokiego są mało wrażliwe na zmianę trybu koloru obrazu. Oczywiście, mamy tutaj na myśli obrazy kolorowe i monochromatyczne. To spostrzeżenie jest bardzo istotne w takich obszarach badań, gdzie z natury mamy do czynienia z obrazami szarymi. Przykładem mogą być obrazy biomedyczne pozyskiwane za pomocą różnych technik obrazowania (USG, RTG, CT i inne). W takich sytuacjach, zastosowanie modeli uczenia głębokiego wydaje się być w pełni uzasadnione. Jak wskazują uzyskane wyniki, nie zaobserwowano istotnego zmniejszenia skuteczności tych modeli, w porównaniu z wykorzystaniem obrazów kolorowych. Z kolei, zastosowanie obrazów binarnych sprawia, że jakość modeli istotnie spada, w porównaniu z modelami zbudowanymi w oparciu o obrazy kolorowe. Ma to miejsce zarówno dla obrazów płomienia, jak i obrazów zmian skórnych. Nośność informacyjna takich obrazów jest zbyt mała, aby zbudować skuteczne modele do klasyfikacji danych, które były przedmiotem badań.

Wydaje się, iż dobrym podsumowaniem wcześniejszych rozważań jest porównanie skuteczności modeli uczenia tradycyjnego i głębokiego w rozwiązywaniu postawionych zadań diagnostycznych. Dla obrazów płomienia, skuteczność obu kategorii modeli jest porównywalna. Najlepszym modelem uczenia głębokiego okazał się VGG19 ($ACC = 96\%$, $TPR = 94\%$, $TNR = 97\%$). W przypadku modeli uczenia tradycyjnego był to SFS-GRAD-18 ($ACC = 95\%$, $TPR = 94\%$, $TNR = 96\%$). Zatem, przy tej samej wartości TPR , pozostałe 2 wskaźniki jakości wymienionych modeli (ACC , TNR) różnią się zaledwie o 1%. W przypadku obrazów zmian skórnych, najskuteczniejszymi modelami uczenia tradycyjnego były R-MLP-17 i R-SVM-17. W obu przypadkach, dokładność testowania wyniosła 88%. Największą skuteczność osiągnął jednak model uczenia głębokiego VGG16. Jego dokładność ($ACC = 91\%$) była o 3% wyższa, w porównaniu z modelami R-MLP-17 i R-SVM-17. Pozostałe indeksy jakości były także o kilka procent większe od adekwatnych parametrów wcześniej wymienionych modeli. Uzyskane wyniki świadczą o tym, że model VGG16 powinien być brany pod uwagę podczas rozwiązywania zadania klasyfikacji obrazów zmian skórnych.

Analiza skuteczności modeli prowadzi do spostrzeżenia, że ich własności zależą od rodzaju danych wejściowych. Przypomnijmy, że dla obrazów płomienia, modele uczenia tradycyjnego oferują wysoką dokładność, równą 95%. Natomiast, parametr ten jest znacznie niższy dla obrazów zmian skórnych ($ACC = 88\%$). Podobna sytuacja występuje dla modeli uczenia głębokiego. Ich dokładność dla obrazów płomienia jest na poziomie 96%. Z kolei, dla obrazów zmian skórnych, obserwujemy spadek ACC do wartości 91%. Taka sytuacja oznacza, że dane odpowiadające obrazom zmian skórnych są bardziej złożone i trudniejsze do klasyfikacji dla obu grup modeli, niż ma to miejsce w przypadku obrazów płomienia.

Wnioski zawarte w niniejszym podsumowaniu sformułowano na podstawie badań obrazów płomienia oraz obrazów zmian skórnych, jednak większość z nich ma charakter bardziej ogólny. Dlatego też, zawarte w monografii uwagi i spostrzeżenia mogą być pomocne w rozwiązywaniu również innych zadań klasyfikacji, niż te będące przedmiotem niniejszego opracowania.

Literatura

- [1] Kopaliński, W., *Słownik wyrazów obcych i zwrotów obcojęzycznych z almanachem*, Rytm, Warszawa 2014.
- [2] Żółtowski, B., Ćwik, Z., *Leksykon diagnostyki technicznej*, Wydawnictwo Akademii Techniczno-Rolniczej, Bydgoszcz 1996.
- [3] Żółtowski, B. *Podstawy diagnostyki maszyn*, ATR, Bydgoszcz 1996.
- [4] Cholewa, W., *Diagnostyka techniczna maszyn*, Wyd. Pol. Śl., Gliwice 1992.
- [5] Cempel, C., *Diagnostyka wibroakustyczna maszyn*, PWN, Warszawa 1989.
- [6] Wiśniewski, W., *Diagnostyka techniczna wytwórczych urządzeń energetycznych w Elektrowniach*, PWN, Warszawa 1991.
- [7] Morel, J., *Drgania maszyn i diagnostyka ich stanu technicznego*, PTDT, 1992.
- [8] Eisenmann, Sr., R.C., Eisenmann, Jr., R.C., *Machinery Malfunction Diagnosis and Correction*, Prentice Hall, 1997.
- [9] Kościelny, J.M., *Diagnostyka zautomatyzowanych procesów przemysłowych*, Akademicka Oficyna Wydawnicza EXIT, Warszawa 2001.
- [10] Korbicz, J., Kościelny, J.M., Kowalczyk, Z., Cholewa, W., *Diagnostyka procesów. Modele, metody sztucznej inteligencji, zastosowania*, Wydawnictwo Naukowo-Techniczne, Warszawa 2002.
- [11] Pieczyński, A., *Komputerowe systemy diagnostyczne procesów przemysłowych*, Wydawnictwo Politechniki Zielonogórskiej, Zielona Góra 1999.
- [12] Kościelny, J.M., Syfert, M., Wnuk, P., System zaawansowanego monitorowania i diagnostyki procesów przemysłowych 'AMandD'. *Diagnostyka* 2006, 3(39), 229–238.
- [13] Smolarz, A., *Diagnostyka procesów spalania paliw gazowych, pyłu węglowego oraz mieszaniny pyłu węglowego i biomasy z wykorzystaniem metod optycznych*, Politechnika Lubelska, Lublin 2013.
- [14] Xu, L., Tan, C., Li, X., i in., Fuel-Type Identification Using Joint Probability Density Arbiter and Soft-Computing Techniques. *IEEE Transactions on Instrumentation and Measurement* 2012, 61(2), 286–296.
- [15] Xu, L., Tan, C., Li, X., i in., On-line Identification of Fuel Type Using Joint Probability Density Arbiter and Support Vector Machine Techniques. *Proceedings of Instrumentation and Measurement Technology Conference*, IEEE, Austin, USA, 2010.
- [16] Kittler, H., Pehamberger, H., Wolff, K., Binder, M., Diagnostic accuracy of dermoscopy. *The Lancet Oncology* 2002, 3(3), 159–165.
- [17] Carli P., Quercioli, E., Sestini, S., i in., Pattern analysis, not simplified algorithms, is the most reliable method for teaching dermoscopy for

- melanoma diagnosis to residents in dermatology. *Br J Dermatol.* 2003, 148(5), 981–984.
- [18] Vestergaard, M.E., Macaskill, P., Holt, P.E., i in., Dermoscopy compared with naked eye examination for the diagnosis of primary melanoma: a meta-analysis of studies performed in a clinical setting. *Br J Dermatol.* 2008, 159, 669–676.
 - [19] Stolz, W., Riemann, A., Cagnetta, A., Pillet, L., Abmayr, W., Holz, 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.
 - [20] Menzies, S.W., Ingvar, C., Crotty, K.A., McCarthy, W.H., Frequency and morphologic characteristics of invasive melanomas lacking specific surface microscopic features. *Archives of Dermatology* 1996, 132(10), 1178–1182.
 - [21] Saez, A., Acha, B., Serrano, C., Pattern analysis in dermoscopic images. *Computer Vision Techniques for the Diagnosis of Skin Cancer.* Springer 2014, 23–48.
 - [22] Maglogiannis, I., Doukas, C.N., Overview of advanced computer vision systems for skin lesions characterization. *IEEE transactions on information technology in biomedicine* 2009, 13(5), 721–733.
 - [23] Argenziano, G., Fabbrocini, G., Carli, P., De Giorgi, V., Sammarco, E., Delfino, M., Epiluminescence microscopy for the diagnosis of doubtful melanocytic skin lesions: comparison of the abcd rule of dermatoscopy and a new 7-point checklist based on pattern analysis. *Archives of dermatology* 1998, 134(12), 1563–1570.
 - [24] Argenziano, G., i in., Dermoscopy of pigmented skin lesions: Results of a consensus meeting via the Internet. *J. American Academy of Dermatology* 2003, 48(5).
 - [25] Gachon, J., i in., First Prospective Study of the Recognition Process of Melanoma in Dermatological Practice. *Arch Dermatol.* 2005, 141(4), 434–438.
 - [26] Mishra, N.K., Celebi, M.E., An Overview of Melanoma Detection in Dermoscopy Images Using Image Processing and Machine Learning. *arxiv.org: 1601.07843.*
 - [27] Ali, A.A., Deserno, T.M., A Systematic Review of Automated Melanoma Detection in Dermatoscopic Images and its Ground Truth Data. *Proc. SPIE* 2012, 8318.
 - [28] Codella, N.C.F., 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).

- [29] Barata, C., Ruela, M., i in., Two Systems for the Detection of Melanomas in Dermoscopy Images using Texture and Color Features. *IEEE Systems Journal* 2014, 8(3), 965–979.
- [30] Lee, T., Ng, V., Gallagher, R., Coldman, A., McLean, D., Dullrazor R: A software approach to hair removal from images. *Computers in biology and medicine* 1997, 27(6), 533–543.
- [31] Celebi, M.E., Kingravi, H.A., Uddin, B., Iyatomi, H., Aslandogan, Y.A., Stoecker, W.V., Moss, R.H., A methodological approach to the classification of dermoscopy images. *Computerized Medical Imaging and Graphics* 2007, 31(6), 362–373.
- [32] Barata, C., Marques, J.S., Rozeira, J., A system for the detection of pigment network in dermoscopy images using directional filters. *IEEE transactions on biomedical engineering* 2012, 59(10), 2744–2754.
- [33] Marques, J.S., Barata, C., Mendonca, T., On the role of texture and color in the classification of dermoscopy images. W: *Engineering in Medicine and Biology Society (EMBC), 2012 Annual International Conference of the IEEE*, IEEE, 2012, 4402–4405.
- [34] Barata, C., Ruela, M., Francisco, M., Mendonca, T., Marques, J.S., Two systems for the detection of melanomas in dermoscopy images using texture and color features. *IEEE Systems Journal* 2014, 8(3), 965–979.
- [35] Situ, N., Wadhawan, T., Yuan, X., Zouridakis, G., Modeling spatial relation in skin lesion images by the graph walk kernel. W: *Engineering in Medicine and Biology Society (EMBC), 2010 Annual International Conference of the IEEE*, IEEE, 2010, 6130–6133.
- [36] Barata, C., Figueiredo, M.A., Celebi, M.E., Marques, J.S., Local features applied to dermoscopy images: Bag-of-features versus sparse coding. W: *Iberian Conference on Pattern Recognition and Image Analysis*, Springer, 2017, 528–536.
- [37] Jafari, M.H., Samavi, S., Karimi, N., Soroushmehr, S.M.R., Ward, K., Najarian, K., Automatic detection of melanoma using broad extraction of features from digital images. W: *Engineering in Medicine and Biology Society (EMBC), 2016 IEEE 38th Annual International Conference of the*, IEEE, 2016, 1357–1360.
- [38] Ballerini, L., Fisher, R.B., Aldridge, B., Rees, J., A color and texture based hierarchical k-nn approach to the classification of non-melanoma skin lesions. W: *Color Medical Image Analysis*, Springer, 2013, 63–86.
- [39] Blum, A., Luedtke, H., Ellwanger, U., Schwabe, R., Rassner, G., Garbe, C., Digital image analysis for diagnosis of cutaneous melanoma. Development of a highly effective computer algorithm based on analysis of 837 melanocytic lesions. *British Journal of Dermatology* 2004, 151(5), 1029–1038.

- [40] Mishra, N.K., Celebi, M.E., An overview of melanoma detection in dermoscopy images using image processing and machine learning. *arXiv:1601.07843* 2016.
- [41] Korotkov, K., Garcia, R., Computerized analysis of pigmented skin lesions: a review. *Artificial intelligence in medicine* 2012, 56(2), 69–90.
- [42] Yu, Z., Ni, D., Chen, S., Qin, J., Li, S., Wang, T., Lei, B., Hybrid dermoscopy image classification framework based on deep convolutional neural network and fisher vector. W: *Biomedical Imaging (ISBI 2017), 2017 IEEE 14th International Symposium on*, IEEE, 2017, 301–304.
- [43] Giotis, I., Molders, N., Land, S., i in., MED-NODE: A computer-assisted melanoma diagnosis system using non-dermoscopic images. *Expert Systems with Applications* 2015, 6578–6585.
- [44] Amelard, R., Glaister, J., Wong, A., Clausi, D., High-Level Intuitive Features (HLIFs) for Intuitive Skin Lesion Description, *IEEE Transactions on Biomedical Engineering* 2015, 62(3), 820–831.
- [45] Alcon, J.F., Ciuhu, C., Ten Kate, W., i in., Automatic Imaging System With Decision Support for Inspection of Pigmented Skin Lesions and Melanoma Diagnosis. *IEEE Journal of Selected Topics in Signal Processing* 2009, 1, 14–25.
- [46] Alfed, N., Khelifi, F., Bouridane, A., Seker, H., Pigment networkbased skin cancer detection. W: *Engineering in Medicine and Biology Society (EMBC), 2015 37th Annual International Conference of the IEEE*, 2015, 7214–7217.
- [47] Cavalcant, P.G., Scharcanski, J., Baranoski, G.V., A two-stage approach for discriminating melanocytic skin lesions using standard cameras. *Expert Systems with Applications* 2013, 40(10), 4054–4064.
- [48] Curiel-Lewandowski, C., Novoa, R.A., Berry, E., i in., Artificial Intelligence Approach in Melanoma. W: Fisher, D., Bastian, B., *Melanoma* 2019, Springer, New York, NY, 599–628.
- [49] Langman, J., Slipek, Z., Diagnostyka techniczna i diagnostyka medyczna – podobieństwa i zapożyczenia. *Inżynieria Rolnicza* 2008, 6(104), 121–126.
- [50] Géron, A., *Uczenie maszynowe z użyciem Scikit-Learn i TensorFlow*. Helion, Gliwice 2018.
- [51] Rudnicki, Z., *Metody komputerowej analizy obrazów w badaniach tribologicznych*. Wydawnictwa AGH, Kraków 2010.
- [52] Tadeusiewicz, R., Korohoda, P., *Komputerowa analiza i przetwarzanie obrazów*. Wydawnictwo Fundacji Postępu Telekomunikacji, Kraków 1997.
- [53] Strzelecki, M., Materka, A., *Tekstura obrazów biomedycznych*, Wydawnictwo naukowe PWN, Warszawa 2017.

- [54] Omiotek, Z., Fractal analysis of the grey and binary images in diagnosis of Hashimoto's thyroiditis. *Biocybernetics and Biomedical Engineering* 2017, 37(4), 655–665.
- [55] Omiotek, Z., Improvement of the classification quality in detection of Hashimoto's disease with a combined classifier approach. *Proceedings of the Institution of Mechanical Engineers, Part H: Journal of Engineering in Medicine* 2017, 231(8), 774–782.
- [56] Omiotek, Z., Burda, A., Feature Selection Methods in Image-Based Screening for the Detection of Hashimoto's Thyroiditis in First-Contact Hospitals. *Barometr Regionalny* 2016, 14(2), 187–196.
- [57] Omiotek, Z., Wójcik, W., An efficient method for analyzing measurement results on the example of thyroid ultrasound images. *Przegląd Elektrotechniczny* 2016, 11, 15–18.
- [58] Omiotek, Z., Burda, A., Wójcik, W., Application of selected classification methods for detection of Hashimoto's thyroiditis on the basis of ultrasound images, W: Pancerz, K., Zaitseva, E., *Computational Intelligence, Medicine and Biology. Studies in Computational Intelligence* 2015, 600, 23–37.
- [59] Omiotek, Z., Burda, A., Wójcik, W., The use of decision tree induction and artificial neural networks for automatic diagnosis of Hashimoto's disease. *Expert Systems with Applications* 2013, 40(16), 6684–6689.
- [60] Omiotek, Z., Dzierżak, R., Uhlig, S., Fractal analysis of the computed tomography images of vertebrae on the thoraco-lumbar region in diagnosing osteoporotic bone damage. *Proceedings of the Institution of Mechanical Engineers, Part H: Journal of Engineering in Medicine* 2019, 233(12), 1269–1281.
- [61] Dzierżak, R., Omiotek, Z., Tkacz, E., Kępa, A., The Influence of the Normalisation of Spinal CT Images on the Significance of Textural Features in the Identification of Defects in the Spongy Tissue Structure, W: Tkacz, E., i in., *Innovations in Biomedical Engineering 2018. IBE 2018. Advances in Intelligent Systems and Computing* 2019, 925, 55–66.
- [62] Omiotek, Z., Prokop, P., The construction of the feature vector in the diagnosis of sarcoidosis based on the fractal analysis of CT chest images. *Informatyka, Automatyka, Pomiary w Gospodarce i Ochronie Środowiska* 2019, 9(2), 16–23.
- [63] Omiotek, Z., Stepanchenko, O., Wójcik, W., Legieć, W., Szatkowska, M., The use of the Hellwig's method for feature selection in the detection of myeloma bone destruction based on radiographic images. *Biocybernetics and Biomedical Engineering* 2019, 39(2), 328–338.
- [64] Galloway, M.M., Texture analysis using gray level run lengths. *Computer Graphics and Image Processing* 1975, 4, 172–179.
- [65] Haralick, R., Statistical and Structural Approaches to Texture. *Proceedings of the IEEE* 1979, 67(5), 786–804.

- [66] Haralick, R., Shanmugam, K., Dinstein, I., Textural Features for Image Classification. *IEEE Transactions on Systems, Man and Cybernetics* 1973, 3(6), 610–621.
- [67] Cross, G.R., Jain, A.K., Markov random fields texture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 1985, 5(1), 25–39.
- [68] Chen, C., Daponte, J.S., Fox, M.D., Fractal feature analysis and classification in medical imaging. *IEEE Transactions on Medical Imaging* 1989, 8, 133–142.
- [69] Clausi, D.A., Jernigan, M.E., Designing Gabor filters for optimal texture separability. *Pattern Recognition* 2000, 33, 1835–1849.
- [70] Mallat, S.G., A theory for multiresolution signal decomposition: The wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 1989, 11(7), 674–693.
- [71] Haralick, R., Sternberg, S.R., Zhuang, X., Image analysis using mathematical morphology. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 1989, 9(4), 532–550.
- [72] Program MaZda (dostęp: 01.09.2020), http://www.eletel.p.lodz.pl/programy/cost/progr_mazda.html
- [73] Hu, Y., Dennis, T., Textured Image Segmentation by Context Enhanced Clustering. *IEE Proc.-Visual Image and Signal Processing* 1994, 141(6), 413–421.
- [74] Lerski, R., Straughan, K., Shad, L., Boyce, D., Bluml, S., Zuna, I., MR Image Texture Analysis – An Approach to Tissue Characterization. *Magnetic Resonance Imaging* 1993, 11, 873–887.
- [75] Zieliński, K.W., Strzelecki, M., *Komputerowa analiza obrazu biomedycznego*, PWN, Warszawa-Łódź 2002.
- [76] Pavlidis, T., *Grafika i przetwarzanie obrazów*, WNT, Warszawa 1987.
- [77] Materka, A., Strzelecki, M., *Texture Analysis Methods – A Review*, Technical University of Lodz, COST B11 Report, 1998.
- [78] Magoulas, G.D., Karkanis, S.A., Karras, D.A., Vrohatis, M.V., Comparison study of Textural Descriptors for Training Neural Networks Classifiers. *IMACS/IEEE CSCC'99 Proc.* 1999, 6221–6226.
- [79] Białasiewicz, J.T., *Falki i aproksymacje. Wyd. II*, WNT, Warszawa 2004.
- [80] Zieliński, T.P., *Cyfrowe przetwarzanie sygnałów - od teorii do zastosowań*, WKŁ, Warszawa 2005.
- [81] Engell, S., Frehse, G., Schnieder, E., *Modeling, Analysis, and Design of Hybrid Systems*, Springer, Berlin 2002.
- [82] Kargupta, H., i in., *Next Generation of Data Mining*, CRC Press, 2008.
- [83] Shawkat Ali, A.B.M., Xiang, Y., *Dynamic and Advanced Data Mining for Progressing Technological Development: Innovations and Systemic Approaches*, Hershey, New York 2010.

- [84] Roszkowski, M., Modelowanie predykcyjne na przykładzie równoległej hybrydy regresyjnej w zastosowaniu do predykcji smaku wina. *Metody Informatyki Stosowanej* 2019, 2(23), 137–142.
- [85] Raschka, S., *Python. Uczenie maszynowe*, Helion, Gliwice 2016.
- [86] Albon, C., *Uczenie maszynowe w Pythonie. Receptury*, Helion, Gliwice 2019.
- [87] Swamynathan, M., *Mastering Machine Learning with Python in Six Steps. A Practical Implementation Guide to Predictive Data Analytics Using Python. Second Edition*, Apress, Berkeley, CA, 2019.
- [88] Boschetti, A., Massaron, L., *Python. Podstawy nauki o danych. Wydanie II*, Helion, Gliwice 2016.
- [89] James, G., Witten, D., Hastie, T., Tibshirani, R., *An Introduction to Statistical Learning with Applications in R*, Springer Science+Business Media, New York 2013.
- [90] Koronacki, J., Ćwik, J.: *Statystyczne systemy uczące się*, WNT, Warszawa 2005.
- [91] Krzyśko, M., Wołyński, W., Górecki, T., Skorzybut, M., *Systemy uczące się. Rozpoznawanie wzorców, analiza skupień i redukcja wymiarowości*, WNT, Warszawa 2008.
- [92] Cichosz, P., *Systemy uczące się*, WNT, Warszawa 2000.
- [93] Stapor, K., *Metody klasyfikacji obiektów w wizji komputerowej*, PWN, Warszawa 2011.
- [94] Guyon, I. *Feature extraction: foundations and applications*, Springer Science & Business Media 2006, 207.
- [95] Liu, H., Motoda, H., *Computational Methods of Feature Selection*, Chapman & Hall 2008.
- [96] Gu, Q., Li, Z., Han, J., Generalized fisher score for feature selection. *CoRR abs/1202.3725* 2012.
- [97] Biblioteka scikit-learn 0.20.2 (dostęp: 01.09.2020), <https://scikit-learn.org/stable/documentation.html>
- [98] Pakiet ‘caret’ (dostęp: 01.09.2020), <https://cran.r-project.org/web/packages/caret/caret.pdf>
- [99] Pakiet ‘clusterSim’ (dostęp: 01.09.2020), <https://cran.r-project.org/web/packages/clusterSim/clusterSim.pdf>
- [100] Feature Selection Library (dostęp: 01.09.2020), <https://www.mathworks.com/matlabcentral/fileexchange/56937-feature-selection-library>
- [101] Program Weka (dostęp: 01.09.2020), <https://www.cs.waikato.ac.nz/ml/weka/index.html>
- [102] Liao, S.-H., Chu, P.-H., Hsiao, P.-Y., Data mining techniques and applications – A decade review from 2000 to 2011. *Expert Systems with Applications* 2012, 39, 11303–11311.

- [103] Ligeza, A., *Logical Foundations for Rule-Based Systems*, Uczelniane Wydawnictwo Naukowo-Dydaktyczne Akademii Górniczo-Hutniczej w Krakowie, Kraków 2005.
- [104] Fisher, L.A., The use of multiple measurements in taxonomic problems. *Annals of Eugenics* 1936, 7, 179–188.
- [105] Enas, G.G., Chai, S.C., Choice of the smoothing parameter and efficiency of the k-nearest neighbor classification. *Computers and Mathematics with Applications* 1986, 12, 235–244.
- [106] Tadeusiewicz, R., *Sieci neuronowe*, Akademicka Oficyna Wydawnicza, Warszawa 1993.
- [107] Żurada, J., Barski, M., Jędruch, W., *Sztuczne sieci neuronowe*. PWN, Warszawa 1996.
- [108] Trzęsiok, M., Analiza wybranych własności metody dyskryminacji wykorzystującej wektory nośne, W: Barczak, A.S., *Postępy ekonometrii* 2004, 331–342.
- [109] Trzęsiok, M., Zastosowanie metody SVM w klasyfikacji danych. W: Jajuga, K., Walesiak, M., *Taksonomia 11, Klasyfikacja i analiza danych – teoria i zastosowania*. Wydawnictwo Akademii Ekonomicznej we Wrocławiu, Wrocław 2004.
- [110] Walesiak, M., Gatnar, E., *Statystyczna analiza danych z wykorzystaniem programu R*, PWN, Warszawa 2009.
- [111] Breiman, L., Friedman, J., Olshen, R., Stone, C., *Classification and Regression Trees*, CRC Press, London 1984.
- [112] Hill T., Lewicki P., *STATISTICS: Methods and Applications*. StatSoft, Tulsa, 2007.
- [113] Quinlan, J.R., Induction of decision trees. *Machine Learning* 1986, 1, 81–106.
- [114] Walesiak, M., Propozycja uogólnionej miary odległości w statystycznej analizie wielowymiarowej, W: Paradysz, J., *Statystyka Regionalna w służbie samorządu lokalnego i biznesu*, Internetowa Oficyna Wydawnicza Centrum Statystyki Regionalnej 2002, 115–121.
- [115] Breiman, L., Bagging Predictors. *Machine Learning* 1996, 24, 123–140.
- [116] Freund, Y., Schapire, R.E., A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences* 1996, 55, 119–139.
- [117] Hothorn, T., Lausen, B., Bundling classifiers by bagging trees. *Computational Statistics & Data Analysis* 2005, 49(4), 1068–1078.
- [118] Breiman, L., Random Forests. *Machine Learning* 2001, 45, 5–32.
- [119] Tumer, K., Ghosh, J., Analysis of decision boundaries in linearly combined neural classifiers. *Pattern Recognition* 1996, 29, 341–348.
- [120] Klimek, K.M., Analiza mocy testu i liczebności próby oraz ich znaczenie w badaniach empirycznych. *Wiadomości Lekarskie* 2008, LXI, 7–9.

- [121] Petrie, A., Sabin, C., *Statystyka medyczna w zarysie*, Wydawnictwo Lekarskie PZWL, Warszawa 2006.
- [122] Rumelhart, D.E., Hinton, G.E., Williams, R.J., Learning representations by back-propagating errors. *Nature* 1986, 323.
- [123] Zocca, V., Spacagna, G., Slater, D., Roelants, P., *Deep Learning. Uczenie głębokie z językiem Python. Sztuczna inteligencja i sieci neuronowe*, Helion, Gliwice 2018.
- [124] Nielsen, M. A., *Neural Networks and Deep Learning*, Determination Press, 2015.
- [125] Glorot, X., Bengio, Y., Understanding the difficulty of training deep feedforward neural networks. *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics* 2010, PMLR 9, 249–256.
- [126] He, K., Zhang, X., Ren, S., Sun, J., Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification, *arXiv:1502.01852 [cs.CV]* 2015.
- [127] Vapnik, V., Chervonenkis, A., A Note on One Class of Perceptrons. *Automation and Remote Control* 1964, 25.
- [128] Smolensky, P., Information processing in dynamical systems: Foundations of harmony theory. *Parallel distributed processing: explorations in the microstructure of cognition* 1986, 1, 194–281.
- [129] Hinton, G.E., A Practical Guide to Training Restricted Boltzmann Machines. W: Montavon, G., Orr, G.B., Müller, K.R., *Neural Networks: Tricks of the Trade. Lecture Notes in Computer Science* 2012, 7700.
- [130] Hinton, G.E., Osindero, S., Teh, Y.W., A fast learning algorithm for deep belief nets. *Neural Computation* 2006, 18(7), 1527–54.
- [131] Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., Manzagol, P., Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion. *Journal of Machine Learning Research* 2010, 11, 3371–3408.
- [132] Kingma, D.P., Welling, M., Auto-Encoding Variational Bayes. *arXiv:1312.6114v10 [stat.ML]* 2014.
- [133] Osinga, D., *Deep Learning. Receptury*, Helion, Gliwice 2019.
- [134] Krizhevsky, A., Sutskever, I., Hinton, G., Imagenet classification with deep convolutional neural networks. *Proceedings of the Advances in Neural Information Processing Systems* 2012, 1097–1105.
- [135] Simonyan, K., Zisserman, A., Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv:1409.1556 [cs.CV]* 2014.
- [136] Russakovsky, O., Deng, J., Su, H., i in., Imagenet large scale visual recognition challenge. *Int. J. Comput. Vis.* 2014, 115(3), 1–42.
- [137] Nahom, N., Teitel, J., Morris, M.R., i in., Convolutional Neural Network for Second Metacarpal Radiographic Osteoporosis Screening. *J. Hand. Surg. Am.* 2020, 45.

- [138] Mahdianpari, M., Salehi, B., Rezaee, M., i in., Very Deep Convolutional Neural Networks for Complex Land Cover Mapping Using Multispectral Remote Sensing Imagery. *Remote Sens.* 2018, 10(7).
- [139] Chauhan, S., Vig, L., Anomaly detection in ECG time signals via deep long short-term memory networks. W: *2015 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, Paris, France, 19–21 October 2015.
- [140] Nanduri, A., Sherry, L., Anomaly detection in aircraft data using Recurrent Neural Networks (RNN). W: *2016 Integrated Communications Navigation and Surveillance (ICNS)*, Herndon, VA, USA, 19-21 April 2016.
- [141] Jean, S., Cho, K., Memisevic, R., Bengio, Y., On Using Very Large Target Vocabulary for Neural Machine Translation. *arXiv:1412.2007v2 [cs.CL]* 2015.
- [142] Bahdanau, D., Cho, K.-H, Bengio, Y., Neural Machine Translation by Jointly Learning to Align and Translate. *arXiv:1409.0473v7 [cs.CL]* 2016.
- [143] Cheng, J., Dong, L., Lapata, M., Long Short-Term Memory-Networks for Machine Reading. *arXiv:1601.06733v7 [cs.CL]* 2016.
- [144] Xu, K., Ba, J.L., Kiros, R., i in., Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. *arXiv:1502.03044v2 [cs.LG]* 2015.
- [145] Goodfellow, I., Pouget-Abadie, J., Mirza, M., i in., Generative Adversarial Networks (PDF). *Proceedings of the International Conference on Neural Information Processing Systems (NIPS 2014)* 2014, 2672–2680.
- [146] Kingma, D.P., Ba, J., Adam: A Method for Stochastic Optimization. *arXiv:1412.6980 [cs.LG]* 2014.
- [147] Duchi, J., Hazan, E., Singer, Y., Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. *Journal of Machine Learning Research* 2011, 12.
- [148] Zeiler, M.D., ADADELTA: An Adaptive Learning Rate Method. *arXiv:1212.5701 [cs.LG]* 2012.
- [149] Bonaccorso, G., *Algorytmy uczenia maszynowego. Zaawansowane techniki implementacji*, Helion, Gliwice 2019.
- [150] Chollet, F., *Deep Learning. Praca z językiem Python i biblioteką Keras*, Helion, Gliwice 2019.
- [151] 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.
- [152] 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.
- [153] Pan, S.J., Yang, Q., A Survey on Transfer Learning. *IEEE Transactions on Knowledge and Data Engineering* 2010, 22(10).

- [154] Platforma TensorFlow (dostęp: 01.09.2020), <https://www.tensorflow.org/>.
- [155] Biblioteka Keras (dostęp: 01.09.2020), <https://keras.io/>.
- [156] Chollet, F., Xception: Deep Learning with Depthwise Separable Convolutions. *arXiv:1610.02357v3 [cs.CV]* 2017.
- [157] Simonyan, K., Zisserman, A., Very Deep Convolutional Networks for Large-Scale Image Recognition. W: *International Conference on Learning Representations (ICLR 2015)* May 7-9 2015, San Diego, CA, USA.
- [158] He, K., Zhang, X., Ren, S., Sun, J., Deep Residual Learning for Image Recognition. *arXiv:1512.03385v1 [cs.CV]* 2015.
- [159] He, K., Zhang, X., Ren, S., Sun, J., Identity Mappings in Deep Residual Networks. *arXiv:1603.05027v3 [cs.CV]* 2016.
- [160] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z., Rethinking the Inception Architecture for Computer Vision. *arXiv:1512.00567v3 [cs.CV]* 2015.
- [161] Szegedy, C., Ioffe, S., Vanhoucke, V., Alemi, A., Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning. *arXiv:1602.07261v2 [cs.CV]* 2016.
- [162] Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., i in., MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *arXiv:1704.04861v1 [cs.CV]* 2017.
- [163] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.-C., MobileNetV2: Inverted Residuals and Linear Bottlenecks. *arXiv:1801.04381v4 [cs.CV]* 2019.
- [164] Huang, G., Liu, Z., Maaten, L., Weinberger, K.Q., Densely Connected Convolutional Networks. *arXiv:1608.06993v5 [cs.CV]* 2018.
- [165] Zoph, B., Vasudevan, V., Shlens, J., Le, Q.V., Learning Transferable Architectures for Scalable Image Recognition. *arXiv:1707.07012v4 [cs.CV]* 2018.
- [166] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., Fei-Fei, L., ImageNet: A large-scale hierarchical image database. W: *2009 IEEE Conference on Computer Vision and Pattern Recognition* June 20-25 2009, Miami, FL, USA.
- [167] Hii, N.C, Tan, C.K., Chong, A.Z.S., Wilcox, S.J., Ward, J., The measurement of pulverised fuel flows by high frequency acoustic emission techniques. W: *Proceedings of the 7th European Conference on Industrial Furnaces and Boilers* April 18-21 2006, Porto, Portugal.
- [168] Ward, J., Tan, C.K., Chong, A.Z.S., Wilcox, S., The Application of Artificial Intelligence Techniques in Combustion Systems. W: Kalogirou, S., *Artificial Intelligence in Energy and Renewable Energy Systems* 2006, Higher Technical Institute, Nicosia, Cyprus: Nova Science Publishers, 363–394.

- [169] Xu, L., Tan, C., Li, X., Li, X., On-line Identification of Fuel Type Using Joint Probability Density Arbiter and Support Vector Machine Techniques. W: *2010 IEEE Instrumentation & Measurement Technology Conference Proceedings* May 3-6 2010, Austin, TX, USA.
- [170] Ross, T.J., *Fuzzy Logic with Engineering Applications*, John Wiley & Sons, Chichester, 2009.
- [171] Bruschi, R., Giacomazzi, E., Giulietti, E., Stringola, C., Nobili, M., Pagliaroli, T., Giammartini, S., An optical technique for the identification and tracking of combustion instabilities. W: *7th Mediterranean Combustion Symposium* 2011, Sardinia, Italy.
- [172] Smolarz, A., Kotyra, A., Wójcik, W., Ballester, J., Advanced diagnostics of industrial pulverized coal burner using optical methods and artificial intelligence. *Experimental Thermal and Fluid Science* 2012, 43, 82–89.
- [173] Smolarz, A., Wojcik, W., Gromaszek, K., Komada, P., Lytvynenko, V., Mussabekov, N., Yesmakhanova, L., Toigozhinova, A., Artificial intelligence methods in diagnostics of coal-biomass blends co-combustion in pulverised coal burners. W: Pawłowska, M., Pawłowski, L., *Environmental engineering V* 2017, CRC Press, Taylor & Francis Group, Boca Raton, 311–317.
- [174] Xu, L., Tan, C., Li, X., Cheng, Y., Li, X., Fuel-Type Identification Using Joint Probability Density Arbiter and Soft-Computing Techniques. *IEEE Transactions on Instrumentation and Measurement* 2012, 61(2), 286–296.
- [175] Hernández, R., Ballester, J., Flame imaging as a diagnostic tool for industrial combustion. *Combustion and Flame* 2008, 155, 509–528.
- [176] Sawicki, D., Omiotek, Z., Evaluation of the possibility of using fractal analysis to study the flame in the co-firing process. W: Romaniuk, R.S., Linczuk, M., *Proc. SPIE* 2018, 10808.
- [177] Sawicki, D., Omiotek, Z., Kotyra, A., Zastosowanie wymiaru fraktalnego do analizy obrazów płomienia w procesie współpalania. *Przegląd Elektrotechniczny* 2018, 7, 142–145.
- [178] Muhammad, K., Ahmad, J., Mehmood, I., Rho, S., Baik, S.W., Convolutional Neural Networks Based Fire Detection in Surveillance Videos. *IEEE Access* 2018, 6, 18174–18183.
- [179] Muhammad, K., Ahmad, J., & Baik, S.W., Early fire detection using convolutional neural networks during surveillance for effective disaster management. *Neurocomputing* 2018, 288, 30–42.
- [180] Abdurakipov, S.S., Gobyzov, O.A., Tokarev, M.P., Dulin, V.M., Combustion Regime Monitoring by Flame Imaging and Machine Learning. *Optoelectronics, Instrumentation and Data Processing* 2018, 54(5), 513–519.
- [181] Lyu, Y., Chen, J., Song, Z., Image-Based Process Monitoring Using Deep Belief Networks. *IFAC PapersOnLine* 2018, 51(18), 115–120.

- [182] Wang, Z., Song, C., Chen, T., Deep learning based monitoring of furnace combustion state and measurement of heat release rate. *Energy* 2017, 131, 106–112.
- [183] Gangopadhyay, T., Locurto, A., Boor, P., Michael, J. B., Sarkar, S., Characterizing combustion instability using deep convolutional neural network. *Proceedings of the ASME 2018 Dynamic Systems and Control Conference* 2018, Atlanta, Georgia, USA.
- [184] Qiu, T., Liu, M., Zhou, G., Wang, L., Gao, K., An Unsupervised Classification Method for Flame Image of Pulverized Coal Combustion Based on Convolutional Auto-Encoder and Hidden Markov Model. *Energies* 2019, 12(13), 2585–2602.
- [185] Liu, Y., Fan, Y., Chen, J., Flame Images for Oxygen Content Prediction of Combustion Systems Using DBN. *Energy & Fuels* 2017, 31, 8776–8783.
- [186] Tóth, P., Garami, A., & Csordás, B. (2017). Image-based deep neural network prediction of the heat output of a step-grate biomass boiler. *Applied Energy*, 200, 155–169.
- [187] Li, N., Lu, G., Li, X., Yan, Y., Prediction of NO_x Emissions from a Biomass Fired Combustion Process Based on Flame Radical Imaging and Deep Learning Techniques. *Combustion Science and Technology* 2016, 188(2), 233–246.
- [188] Akintayo, A., Lore, K.G., Sarkar, S., Sarkar, S., Prognostics of combustion instabilities from hi-speed flame video using a deep convolutional selective autoencoder. *International Journal of Prognostics and Health Management* 2016, 7(023), 1–14.
- [189] Sarkar, S., Lore, K. G., Sarkar, S., Early detection of combustion instability by neural-symbolic analysis on hi-speed video. *Proceedings of the 2015th International Conference on Cognitive Computation: Integrating Neural and Symbolic Approaches* 2015, 1583, 93–101.
- [190] Sarkar, S., Lore, K.G., Sarkar, S., Ramanan, V., Chakravarthy, S.R., Phoha, S., Ray, A., Early Detection of Combustion Instability from Hi-speed Flame Images via Deep Learning and Symbolic Time Series Analysis. *Proceedings of the Annual Conference of the Prognostics and Health Management Society* 2015, 6, 353–362.
- [191] Zhang, Y., He, Z., Video Object Segmentation Through Deep Convolutional Networks. *International Journal on Artificial Intelligence Tools* 2017, 26(4).
- [192] Baştan, M., Yılmaz, Ö., Multi-View Product Image Search with Deep ConvNets Representations. *International Journal on Artificial Intelligence Tools* 2018, 27(8).
- [193] Kotyra, A. *Diagnostyka procesu spalania pyłu węglowego z wykorzystaniem metod przetwarzania obrazu*. Politechnika Lubelska, Lublin 2010.

- [194] Mackiewicz A., Skowronek J., Żygulska-Mach H., *Czerniak złośliwy, podręcznik dla lekarzy i studentów*, Wydawnictwo medyczne TERME-DIA, Poznań 1998.
- [195] Brady, M.S., Oliveria, S.A., Christos, P.J., Berwick, M., Coit, D.G., Katz, J., Halpern, A.C., Patterns of detection in patients with cutaneous melanoma. *Cancer* 2000, 89(2), 342–7.
- [196] Kittler, H., Pehamberger, H., Wolff, K., Binder, M., Diagnostic accuracy of dermoscopy. *The Lancet Oncology* 2002, 3(3), 159–165.
- [197] Korotkov, K., Garcia, R., Computerized analysis of pigmented skin lesions: A review. *Artif Intell Med.* 2012, 56, 69–90.
- [198] Al-masni, M.A., Al-antari, M.A., Park, J.-M., i in., Simultaneous detection and classification of breast masses in digital mammograms via a deep learning YOLO-based CAD system. *Computer Methods and Programs in Biomedicine* 2018, 157, 85–94.
- [199] Al-antari, M.A., Al-masni, M.A., Choi, M.-T., Han, S.-M., Kim, T.-S., A fully integrated computer-aided diagnosis system for digital Xray mammograms via deep learning detection, segmentation, and classification. *International Journal of Medical Informatics* 2018, 117, 44–54.
- [200] Al-masni, M.A., Al-antari, M.A., Choi, M.-T., Han, S.-M., Kim, T.-S., Skin lesion segmentation in dermoscopy images via deep full resolution convolutional networks. *Computer Methods and Programs in Biomedicine* 2018, 162, 221–231.
- [201] Al-antari, M.A., Al-masni, M.A., Park, S.-U., Park, J., Metwally, M.K., Kadah, Y.M., Han, S.-M., Kim, T.-S., An Automatic Computer-Aided Diagnosis System for Breast Cancer in Digital Mammograms via Deep Belief Network. *Journal of Medical and Biological Engineering* 2018, 38, 443–456.
- [202] Barata, C., Marques, J.S., Rozeira, J., A system for the detection of pigment network in dermoscopy images using directional filters. *IEEE transactions on biomedical engineering* 2012, 59(10), 2744–2754.
- [203] Barata, C., Ruela, M., i in., Two Systems for the Detection of Melanomas in Dermoscopy Images using Texture and Color Features. *IEEE Systems Journal* 2014, 8(3), 965–979.
- [204] Mendonca, T., Ferreira, P.M., Marques, J.S., Marcal, A.R., Rozeira, J., PH2 - a dermoscopic image database for research and benchmarking. *Conf Proc IEEE Eng Med Biol Soc* 2013, 5437–5440.
- [205] Gutman, D., Codella, N., Celebi, E., Helba, B., Marchetti, M., Mishra, N., Halpern, A., Skin Lesion Analysis toward Melanoma Detection: A Challenge at the International Symposium on Biomedical Imaging (ISBI) 2016, hosted by the International Skin Imaging Collaboration (ISIC). *arXiv:1605.01397 [cs.CV]* 2016.

- [206] Marchetti, M., i in., Results of the 2016 International Skin Imaging Collaboration International Symposium on Biomedical Imaging challenge: Comparison of the accuracy of computer algorithms to dermatologists for the diagnosis of melanoma from dermoscopic images. *Journal of the American Academy of Dermatology* 2017.
- [207] Codella, N., Gutman, D., Celebi, M.E., Helba, B., Marchetti, M.A., i in., Skin Lesion Analysis Toward Melanoma Detection: A Challenge at the 2017 International Symposium on Biomedical Imaging (ISBI), Hosted by the International Skin Imaging Collaboration (ISIC). *arXiv: 1710.05006 [cs.CV]* 2017.
- [208] Codella, N., Rotemberg, V., Tschandl, P., Celebi, M.E., i in., Skin Lesion Analysis Toward Melanoma Detection 2018: A Challenge Hosted by the International Skin Imaging Collaboration (ISIC). *arXiv:1902.03368v2 [cs.CV]* 2019.
- [209] Tschandl, P., Rosendahl, C., Kittler, H., The HAM10000 dataset, a large collection of multi-source dermoscopic images of common pigmented skin lesions. *Sci. Data* 2018, 5(180161).
- [210] 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. W: *Proceedings of the International Workshop on Machine Learning in Medical Imaging* 2015, 118–126.
- [211] Demyanov, S., Chakravorty, R., Abedini, M., Halpern, A., Garnavi, R., Classification of dermoscopy patterns using deep convolutional neural networks. W: *Proceedings of the IEEE International Symposium on Biomedical Imaging* 2016, 364–368.
- [212] Kawahara, J., BenTaieb, A., Hamarneh, G., Deep features to classify skin lesions. W: *Proceedings of the IEEE International Symposium on Biomedical Imaging* 2016, 1397–1400.
- [213] Kawahara, J., Hamarneh, G., Multi-resolution-tract CNN with hybrid pretrained and skin-lesion trained layers. W: *Proceedings of the Machine Learning in Medical Imaging. Lecture Notes in Computer Science* 2016, 10019, 164–171.
- [214] 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.
- [215] Menegola, A., Fornaciali, M., Pires, R., Avila, S., Valle, E., Towards automated melanoma screening: exploring transfer learning schemes. *arxiv: 1609.01228* 2016.
- [216] Esteva, A., Kuprel, B., Novoa, R.A., Ko, J., Swetter, S.M., Blau, H.M., Thrun, S., Dermatologist-level classification of skin cancer with deep neural networks. *Nature* 2017, 542, 115–118.
- [217] Portsmouth, N.H., Dermnet. Skin Disease Atlas (dostęp: 01.09.2020), <http://www.dermnet.com/>.

- [218] Oakley, A., DermNet NZ. The Dermatology Resource. (dostęp: 01.09.2020), <http://www.dermnetnz.org/>.
- [219] Silva, S.F., Calheiros, D.B., Dermatology Atlas. (dostęp: 01.09.2020), <http://www.atlasdermatologico.com.br/>.
- [220] MED-NODE Dataset. (dostęp: 01.09.2020), http://www.cs.rug.nl/~imaging/databases/melanoma_naevi/.
- [221] PH2 Dataset. (dostęp: 01.09.2020), <https://www.fc.up.pt/addi/ph2%20database.html>.
- [222] ISIC Archive. (dostęp: 01.09.2020), <https://www.isic-archive.com/#!/topWithHeader/onlyHeaderTop/gallery>.