Akademia Górniczo-Hutnicza im Stanisława Staszica
Wydział Automatyki, Elektroniki, Informatyki i Inżynierii Biomedycznej

ROZPOZNANIE I KLASYFIKACJA ZMIAN CHOROBOWYCH NA OBRAZACH RENTGENOWSKICH KLATKI PIERSIOWEJ

Projekt realizowany w ramach przedmiotu
Techniki Obrazowania Medycznego

Paulina Armatys
Karolina Gocyk
Gabriela Markowicz
Anna Marszałek

Inżynieria Biomedyczna
Rok 3
WEAIiIB

Kraków, 9 czerwca 2021

Spis treści

1. Wprowadzenie

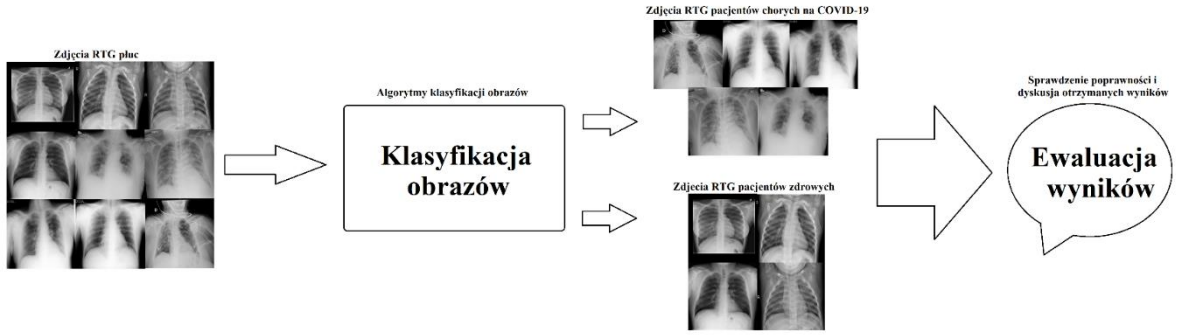
1.1 Temat projektu

Rozpoznanie i klasyfikacja zmian chorobowych na obrazach rentgenowskich klatki piersiowej.

1.2 Cel i założenia projektu

Celem projektu jest zapoznanie się z metodami ekstrakcji cech i klasyfikacji, a następnie wybranie kilku z nich i użycie ich do rozpoznania rentgenowskich obrazów klatki piersiowej charakterystycznych dla osób zdrowych oraz pacjentów zakażonych COVID-19.

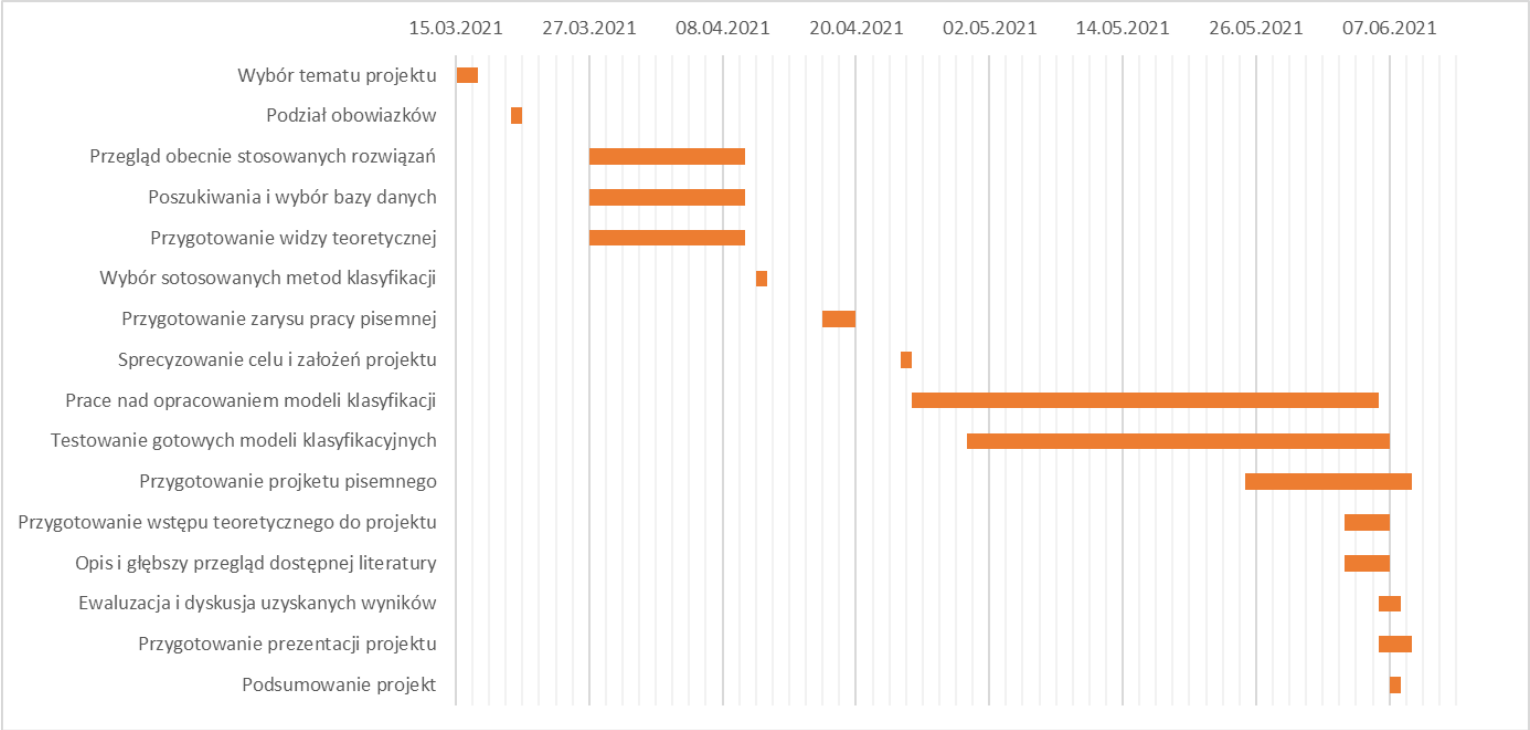
1.3 Abstrakt graficzny



1.4 Streszczenie

COVID-19 jest aktualnym problemem globalnym. Szczególnie niebezpieczne są ciężki przebieg choroby oraz występujące powikłania po przebytej chorobie, które dotykają nawet osoby z łagodnym przebiegiem choroby. Powikłania najczęściej dotyczą układu oddechowego i przyjmują postać zwłóknienia płuc widocznego jako charakterystyczne zmatowienia na obrazach RTG. Z uwagi na dużą liczbę osób i potrzebę szybkiego wykrywania degradacji tkanki płucnej ważne jest opracowanie algorytmu pozwalającego na szybką klasyfikację osób chorych na COVID-19. W naszej pracy dokonano przeglądu metod klasyfikacji dostępnych w programie Matlab. Najwyższym parametrem *accuracy*, wartością *recall*, najbardziej zbliżony do jedynki *F1 Score* spośród sprawdzonych przez nas metod charakteryzuje się metoda KNN z filtrem Sobela. Sprawdzono również działanie gotowych modeli dostępnych w programie Matlab. Potencjalnie najlepszymi wyborami okazało się SVM bez PCA lub LDA bez PCA.

1.5 Etapy pracy nad projektem



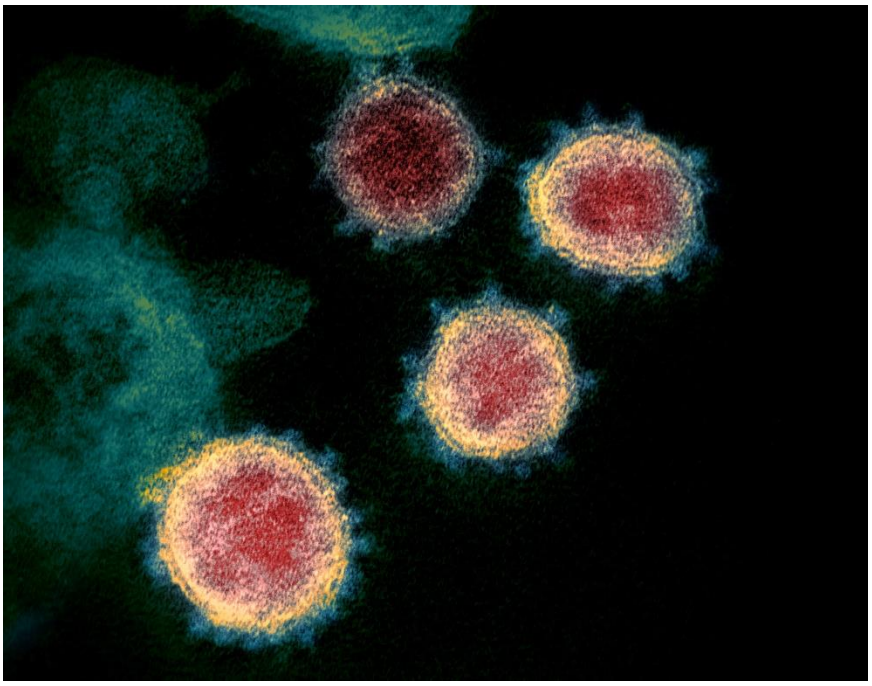
2. Wstęp teoretyczny

1.1 Spis skrótów

COVID-19–(ang. coronavirus disease 2019) choroba wywoływana przez wirusa SARS-Cov-2
SARS-Cov-2-(ang. Severe acute respiratory syndrome coronavirus 2) wirus wywołujący chorobę COVID-19
RTG- Zdjęcie rentgenowskie, rentgenogram, potocznie rentgen, prześwietlenie
PCA- metoda analizy głównych składowych (ang. principal component analysis, PCA)
LDA- Liniowa analiza dyskryminacyjna (ang. Linear discriminant analysis)
KNN- algorytm k najbliższych sąsiadów (ang. K nearest neighbours)
SVM-(ang. Support Vector Machine, maszyna wektorów wspierających)
WHO- Światowa Organizacja Zdrowia, WHO (ŚOZ; ang. World Health Organization)
TP (true positive) – liczba poprawnie sklasyfikowanych przykładów.
FN (false negative) - decyzja negatywna, podczas gdy przykład w rzeczywistości jest pozytywny.
TN (true negative) – liczba przykładów poprawnie odrzuconych.
FP (false positive) – liczba przykładów błędnie przydzielonych do wybranej klasy, podczas gdy w rzeczywistości do niej nie należą.

1.2 Czym jest COVID-19?

COVID-19 (ang. coronavirus disease 2019) jest chorobą zakaźną o podłożu wirusowym, wywoływaną przez koronawirusa SARS-Cov-2 (ang. Severe acute respiratory syndrome coronavirus 2), która atakuje drogi oddechowe. Może objawiać się podobnie jak przeziębienie (najczęściej występującymi objawami są gorączka, kaszel, duszności, bóle mięśni, zmęczenie, utrata węchu i smaku) ale w cięższych przypadkach wywołuje powikłania takie jak zapalenie płuc czy niewydolność oddechowa [1][2].

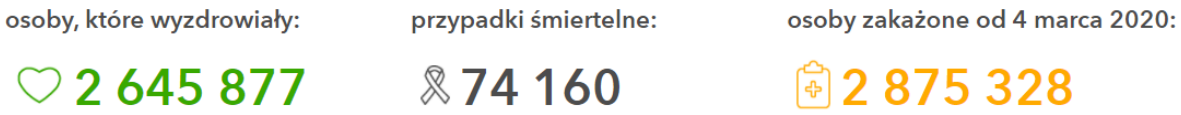


Rys.1. Wirus SARS-CoV-2 - obraz z mikroskopu elektronowego [20]

Ciężki przebieg choroby obserwuje się u ok.15-20% osób a do zgonów dochodzi u 2-3% osób chorych. Prawdopodobnie dane te są zawyżone, z uwagi na fakt, że u wielu osób z lekkim przebiegiem zakażenia nie dokonano potwierdzenia laboratoryjnego[1]. Według danych znajdujących się na stronie Światowej Organizacji Zdrowia, WHO (ŚOZ; ang. World Health Organization), od początku pandemii do dnia 06.06.2021r. potwierdzone zostały 172 630 637 przypadku zarażenia wirusem SARS-Cov-2 na całym świecie oraz odnotowano 3 718 683 przypadki śmiertelne [3].

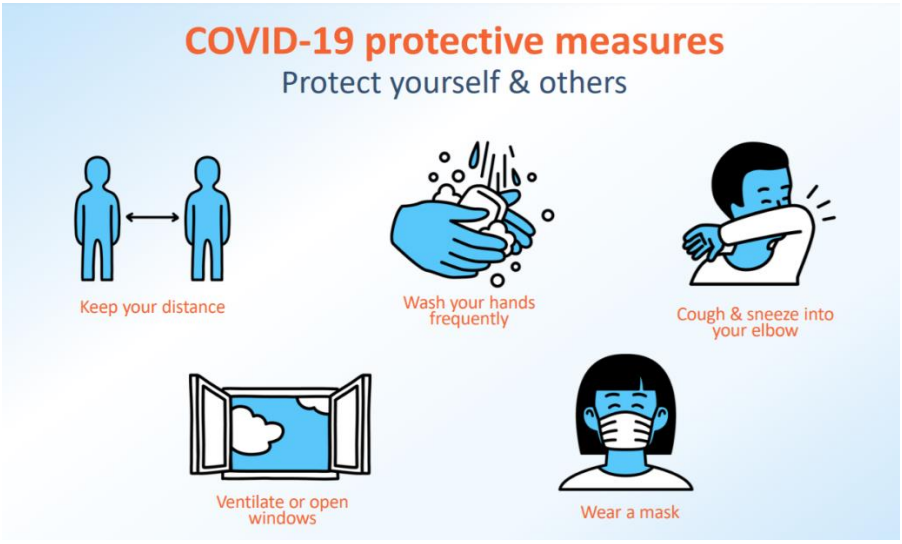
Rys.2. Stan epidemii na świecie na dzień 06.06.2021 [3]

Według strony państwowej gov.pl stan epidemii w Polsce na dzień 07.06.2021 prezentuje się zgodnie z informacjami przedstawionymi na rysunku 3.



Rys.3. Stan epidemii w Polsce na dzień 07.06.2021 [21]

Wirus przenosi się drogą kropelkową - transmisja odbywa się poprzez drobne kropelki pochodzące z dróg oddechowych, które są obecne w ustach czy nosie. Gdy osoba zakażona COVID-19 kaszle, kicha, mówi i wydycha powietrze, patogeny są wdychane przez osoby zdrowe znajdujące się w pobliżu. Wirusy mogą też osiadać na różnych powierzchniach i obiektach znajdujących się w pobliżu, skąd też wynika drugi możliwy sposób przeniesienia się infekcji, czyli poprzez dotknięcie tych powierzchni, a następnie dotknięcie okolic nosa, ust lub oczu [2]. Najbardziej narażone na rozwinięcie ciężkiej postaci choroby i zgon są osoby starsze, z obniżoną odpornością, którym towarzyszą inne choroby, w szczególności przewlekłe [1]. Aktualnie nie istnieje skuteczny lek przeciwko COVID-19 a w przypadku choroby stosuje się leczenie objawowe. Do najskuteczniejszych metod ochronnych przeciwko zakażeniu zalecanych przez WHO zaliczyć możemy szczepienia ochronne, dystans społeczny , częste mycie rąk, częsta wentylacja pomieszczeń, noszenie masek ochronnych, zasłanianie ust i nosa podczas kichania i kaszlu, w przypadku wystąpienia objawów chorobowych pozostanie w domu [3].



Rys.4. Metody ochrony przed zakażeniem wirusem SARS-Cov-2, [4]

1.3 Powikłania w układzie oddechowym po COVID-19

Według danych zawartych na stronie WHO osoby chore na COVID-19 osiągają stan zdrowia zazwyczaj po 2 do 6 tygodniach. Mimo, że większość osób wraca do zdrowia i normalnego trybu życia, u niektórych osób mogą wytępić objawy utrzymujące się przez tygodnie lub nawet miesiące po wyzdrowieniu z ostrej postaci choroby. Ten uporczywy stan złego stanu zdrowia jest znany jako „stan po COVID" (do opisu stanu używa się również innych nazw, jednak jak dotąd nie ma uzgodnionej na szczeblu międzynarodowym definicji stanu po COVID). Trwałych lub późnych objawów doświadczyć mogą nawet osoby, które nie były hospitalizowane a przebieg ich choroby był łagodny. U niektórych pacjentów pojawiają mogą się również komplikacje medyczne mające trwałe skutki zdrowotne. Wydają się one być mniej powszechne i występują głównie u pacjentów z ciężkim COVID-19, którzy byli hospitalizowani. Powikłania po przebytej chorobie COVID-1 wpływają na różne układy narządów w ciele w tym na układ oddechowy, układ sercowo-naczyniowy, układ nerwowy, układ wydalniczy [4].

W naszym projekcie szczególny nacisk położony został na powikłania występujące w układzie oddechowym będących jednocześnie najczęściej występującymi objawiamy. Badania wykazały degradację tkanki płucnej u 88 % ozdrowieńców po 6 tygodniach od ich wyjścia ze szpitala. Po 12 tygodniach odsetek ten zmalał do 56 %, co dowodzi, że płuca są w stanie samoistnie się regenerować, jednakże regeneracja ta może zająć wiele lat. W skrajnych przypadkach regeneracja nie występuje a występowanie trwałych zmian grozi koniecznością przeprowadzenia transplantacji płuc [5].

Degradacja tkanki płucnej wynika z faktu działania wirusa, który podczas namnażania niszczy komórki płus, wywołuje ostry stan zapalny i przyczynia się do tworzenia zatorów w naczyniach krwionośnych. Miejscowe obumarcie tkanek tudzież ich nieodwracalne zwłóknienie (będące następstwem bliznowacenia uszkodzeń) nie zawsze jednak wynika tylko z jego działania. Często jest to również efekt uboczny intensywnego leczenia, np. intubacji. Do zwłóknienia potrafi czasem dojść nawet wówczas, gdy infekcja nie daje wyraźnych objawów i pacjent nie trafia do szpitala. Zdarza się bowiem, że tzw. śródmiąższowe zapalenie płuc obejmuje tylko niewielką część tego narządu i jeśli nie jest on zmuszony do intensywnej pracy, dana osoba nie zauważa, iż braknie jej tchu. Może jednak zacząć doświadczać objawów takich jak przewlekł zmęczenie, obniżenie nastroju (wskutek niedotlenienia mózgu), ucisk lub ból w klatce piersiowej przy głębszych oddechach i w trakcie aktywności fizycznej,
duszności podczas wysiłku, nagły „spadek kondycji" [5].

W przypadku zaobserwowania powyższych objawów warto zdecydować się na przeprowadzanie badania RTG płuc. Na uzyskiwanym w toku owych badań obrazie płuc ukazują się charakterystyczne dla procesu zapalnego zmętnienia, przypominające mleczne szkło czy matową szybę. Czasami jest to jedyny sposób na wykrycie niepokojących zmian, jako że przy stosunkowo niewielkim nasileniu objawów badanie osłuchowe może niestety nic nie wykazać. Wcześnie wykryty wysięk do pęcherzyków płucnych, uwidaczniający się na prześwietleniu właśnie jako „matowa szyba", można leczyć, podając chociażby leki sterydowe, które przyspieszają wchłanianie płynu z płuc [5].

3. Przegląd literaturowy

W lutym 2020 WHO zadeklarowało COVID-19 jako globalną pandemię. Już w styczniu tego roku jej efekty odczuło do 12,3 miliona ludzi. Duże nadzieje w diagnozie tej choroby oraz przewidywaniu jej negatywnych skutków daje Machine Learning i Deep Learning.

Techniki te pozwalają na zasymulowanie sztucznej inteligencji i przerobienie olbrzymich ilości danych pozwalających na wyznaczenie trendu i zależności w danych pozyskanych z populacji osób chorych.

Deep Learning pozwala na konstruowanie sieci, które uczą się przekazując sobie dane między warstwami. Taka sieć jest w stanie zasymulować sztuczną inteligencję uczącą się i podejmującą decyzje dotyczące diagnozy objawów chorobowych na przykład na zdjęciach CT lub rentgenowskich.

Rozpoznawanie objawów chorobowych z obrazów jest szczególnie istotne w erze globalnej pandemii, ponieważ w większości krajów przez cały czas brakuje miejsc w szpitalach, zestawów badawczych, laboratoriów, a nawet samego personelu medycznego.
Wykorzystywanie Deep Learningu i Machine Learningu do wyręczenia ludzi w żmudnym procesie diagnozy zapewnia możliwość szybszego, sprawniejszego i przede wszystkim tańszego rozdzielania populacji zdrowej i chorej w ramach badań.

Aktualnie testowane są rozwiązania takie jak sieci konwolucyjne. Są to sieci konwolucyjne, które zostały przetrenowane i przetestowane na zbiorach danych pozyskanych przez naukowców z klinik diagnostycznych. Zbiór zawierał 5 000 zdjęć CT, pacjentów z COVID i nie. Sieć konwolucyjna wykazała się skutecznością wynoszącą 90%.

Innymi testowanymi sposobami wytworzenia rozpoznającej zdjęcia sieci była metoda kalsyfikacji SVM, czyli maszyny wektorów nośnych. Do uczenia sieci wykorzystano dane 800 pacjentów, pozyskane ze szpitali. Dane zebrane od pacjentów zawierały przypadki braku choroby, COVID-19 w stadium lekkim i COVID-19 w stadium ostrym i zaawansowanym. Ta metoda wykazała skuteczność około 75%.

Następną wykorzystywaną techniką był bazowany na sieci konwolucyjnej model Net. Ten model trenowany był na zbiorze zdjęć rentgenowskich 650 pacjentów hospitalizowanych, o różnych stadiach choroby. Skuteczność wyniosła 90%.

Metoda lasu losowego (random forest) jest metodą w Uczeniu Maszynowym i także jest stosowana do wykrywania zakażeń. Zbiór danych wykorzystany do tej metody składał się z danych 2 500 pacjentów. Ta metoda wykazała się skutecznością powyżej 93%.
Zbiorem testowym dla wszystkich, wyuczonych rodzajów sieci był pobrany z Kaggle zbiór zdjęć rentgenowskich osób chorych oraz zdrowych.

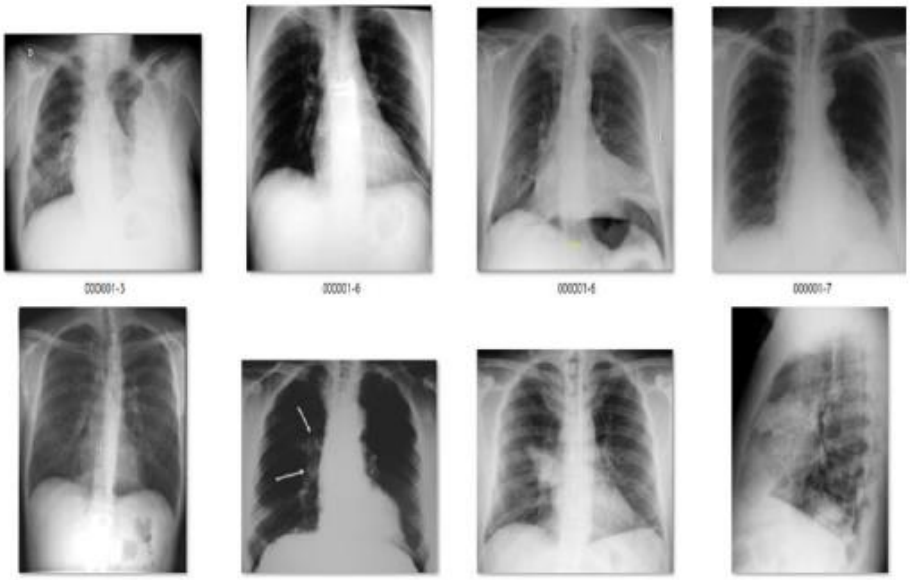


Figure 1: Covid-19 Patients Chest X-ray images.

Rys.5. Grafika przedstawia fragment testowego zbioru danych [21]

Podział zbioru:

Table 2: Covid-19 data Set description

Category	Normal images	Covid-19 Patient	Total Images
Training dataset	250	2000	2250
Testing dataset	350	3000	3500
All images	600	5000	5600

Rys.6. Podział zbioru testowego [21]

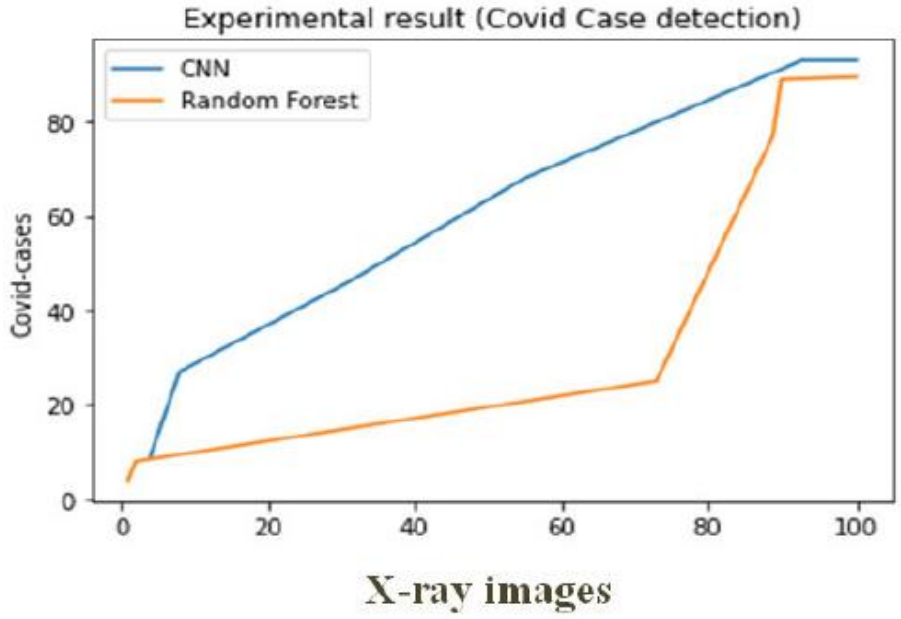
Po przeprowadzeniu testów najbardziej efektywne okazały się metody sieci konwolucyjnej oraz random forest.

Table 3: Performance of CNN & Random Forest

Scenario	X-Ray image Dataset		Method	Accuracy	Specificity	Sensitivity
Scenario-1	Training	Testing	Random Forest [2]	87.9	91.2	90.9
	2250	3500	CNN [1]	92.4	98.8	98.3
Scenario-2	Training	Testing	Random Forest [2]	86.5	93.5	91.7
	1125	1750	CNN [1]	91.74	96.28	95.13

Rys.7. Wyniki przeprowadzonego testu [21]

Zdecydowaną przewagę miała jednak metoda sieci konwolucyjnych na każdym testowanym zbiorze danych.



Rys.8. Wyniki porównawcze dla metody sieci konwolucyjnej i lasu losowego [21]

Z większości czytanej literatury wynika, że sieci konwolucyjne sprawdzają się najlepiej do rozpoznawania objawów na obrazach rentgenowskich. Dzieje się tak dzięki budowie sieci, gdzie każdy neuron wejściowej warstwy jest połączony z każdym neuronem wyjściowej warstwy. Te sieci nadają się najlepiej do analizy topografii o strukturze siatki, czyli właśnie takiej jaką mają zdjęcia z rentgena. W różnych badaniach stosuje się *kernele* albo określone odgórnie przez twórcę, albo wyuczone przez program.

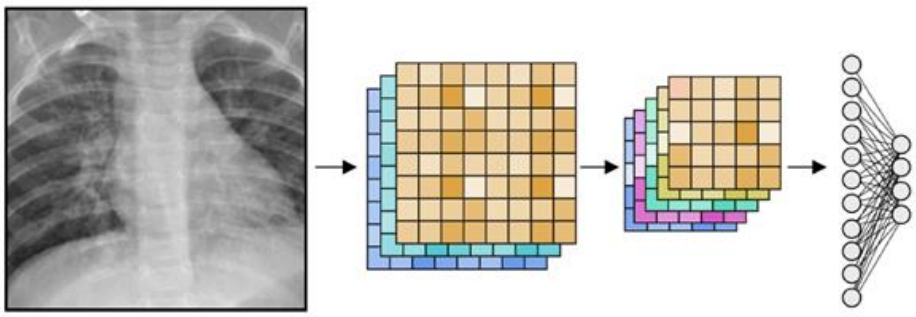


Figure 6.4: Visualization of typical CNN architecture.

Rys.9. Graficzna reprezentacja struktury sieci konwolucyjnej [21]

Wczesne wykrycie powikłań w układzie oddechowym po COVID-19, jest obecnie krytycznym zadaniem dla praktyków klinicznych, gdyż umożliwia szybkie rozpoczęcie leczenia i rehabilitacji a co za tym idzie pozwala na pełny powrót pacjentów do zdrowia. Ostatnie odkrycia uzyskane za pomocą technik obrazowania medycznego sugerują, że obrazy rentgenowskie zawierają istotne informacje o powikłaniach po COVID-19 i mogą stanowić cenne źródło wiedzy diagnostycznej. Z uwagi na duża ilość zachorowań, a co za tym idzie dużą

ilość pacjentów z powikłaniami po przebytej chorobie niezbędne wydaje się opracowanie szybkich automatycznych metod klasyfikacji pacjentów na podstawie zdjęć RTG. Działania te podjęte zostały przez wiele zespołów badawczych na całym świecie. Do aktualnych osiągnięć nauki i opracowanych metod możemy zaliczyć następujące przykłady:

- Badanie oparte na uczeniu głębokim wykorzystane do wykrywania i diagnozowania pacjentów z COVID-19 za pomocą zdjęć rentgenowskich płuc. Do diagnozy choroby użyte zostały dwa algorytmy obejmujące głęboką sieć neuronową (DNN) na fraktalnej cesze obrazów oraz metody konwolucyjnej sieci neuronowej (CNN) bezpośrednio z wykorzystaniem obrazów płuc. Klasyfikacja wyników pokazała, że przedstawiona architektura CNN z wyższą dokładnością (93,2%) i czułością (96,1%) przewyższa metodę DNN z dokładnością 83,4% i czułością 86%. [6]

- Badanie wykrywające pacjentów zakażonych COVID-19 za pomocą zdjęć rentgenowskich oparte na maszynie wektorów wspierających SVM pokazało, że technika ta jest przydatna podczas wczesnego wykrywania pacjentów zakażonych COVID-19. Zaproponowany został system wielopoziomowego progowania SVM, który wykazał dużą dokładność w klasyfikacji pacjentów cierpiących na Covid-19. Średnia czułość, swoistość i dokładność klasyfikacji płuc przy użyciu wyników zaproponowanego modelu wyniosły odpowiednio 95,76%, 99,7% i 97,48%. [7]

- Badanie stosujące zaawansowane techniki sztucznej inteligencji (AI) w połączeniu z obrazowaniem RTG w celu dokładnego wykrywania COVID-19. W badaniu przedstawiony został nowy model automatycznego wykrywania COVID-19 z wykorzystaniem surowych zdjęć rentgenowskich klatki piersiowej. Proponowany model został opracowany w celu zapewnienia dokładnej diagnostyki klasyfikacji binarnej (COVID vs. brak wyników) i klasyfikacji wieloklasowej (COVID vs. brak wyników vs. zapalenie płuc). Model wykazał dokładność klasyfikacji 98,08% dla klas binarnych i 87,02% dla przypadków wieloklasowych. Model DarkNet został wykorzystany w badaniu jako klasyfikator dla systemu wykrywania obiektów w czasie rzeczywistym. [8]

4. Użyte metody analizy

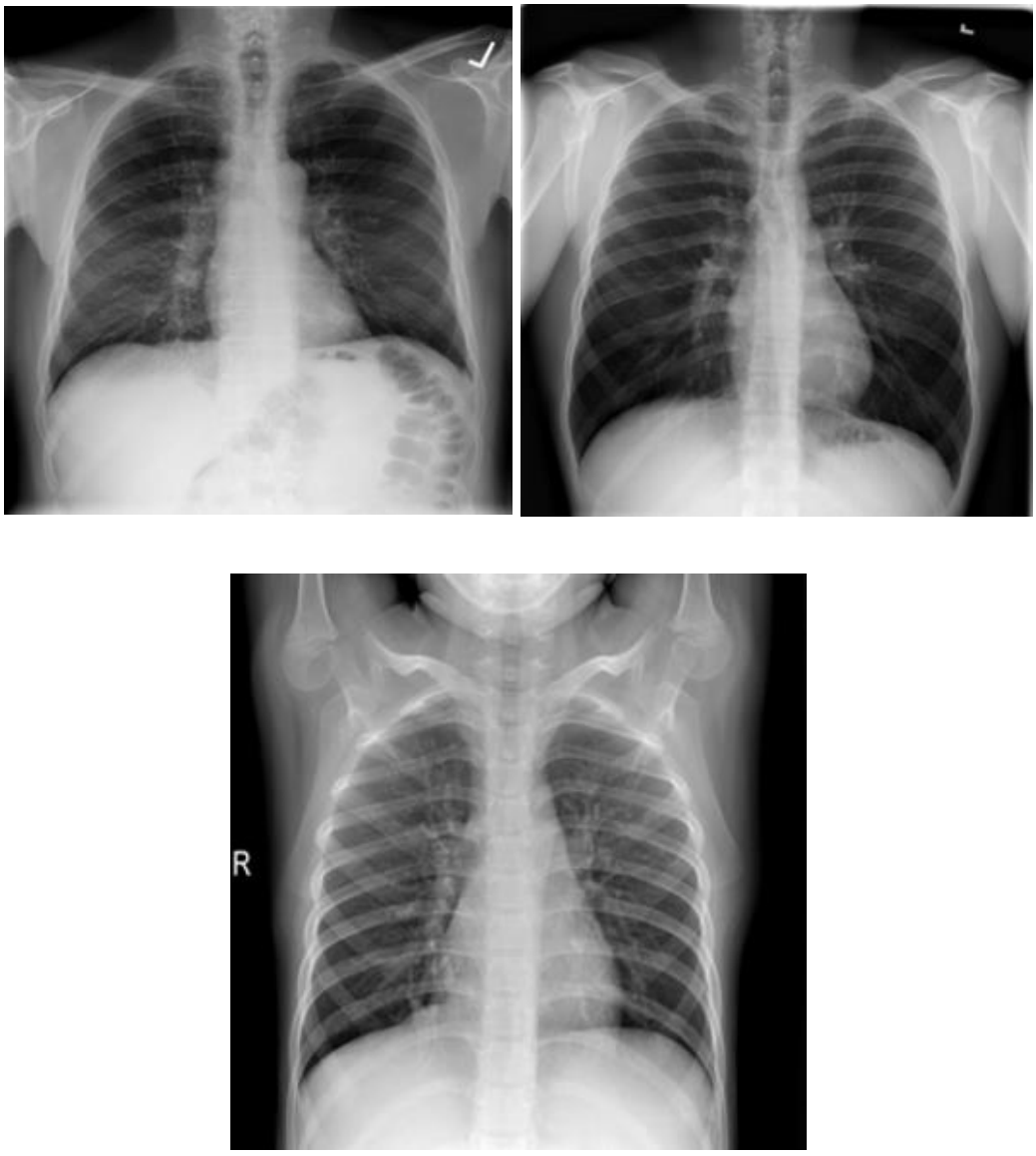
4.1 Zbiór danych

Użyty zbiór pochodzi z ,,COVID-19 radiography database'' i zawiera bazę danych zdjęć rentgenowskich klatki piersiowej pacjentów o pozytywnym wyniku na COVID-19 oraz pacjentów zdrowych. Zbiór ten został stworzony i opracowany przez zespół naukowców z uniwersytetu w Katarze, w Doha, w Dhace oraz w Bangladeszu wraz ze współpracownikami z Pakistanu i Malezji oraz we współpracy z lekarzami.

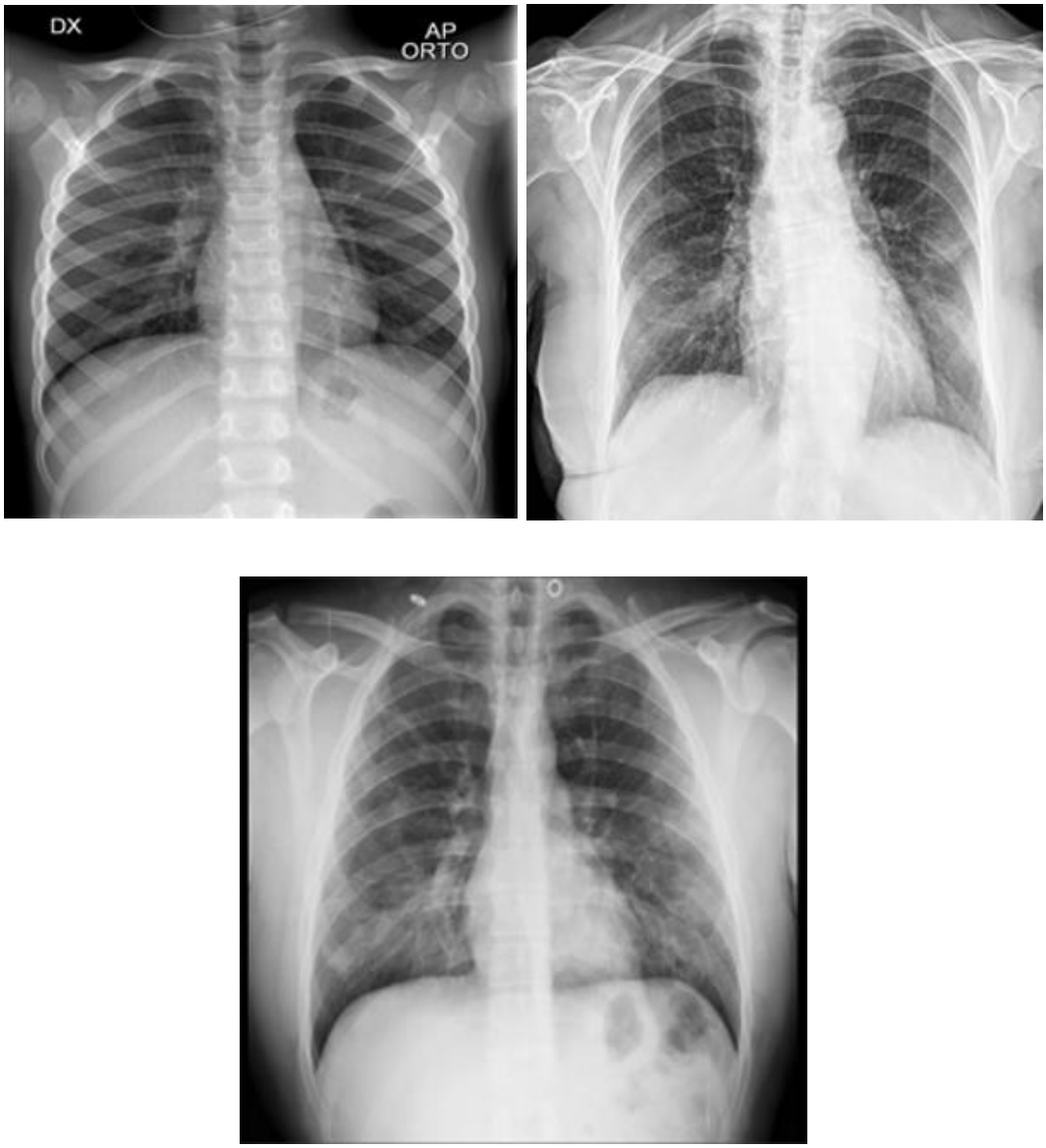
4.2 Metody

Projekt został zrealizowany w środowisku programistycznym Matlab. Do analizy obrazów płuc użyto sześciu zdjęć RTG, z których cztery posłużyły jako zbiory uczące, zaś dwa jako testowe. Na rys.1. oraz rys.2. zaprezentowano obrazy prawidłowych oraz chorych płuc, na których dokonywano analiz.

Do przeprowadzenie ekstrakcji cech z powyższych obrazów użyto metody PCA, filtrów: medianowego, Prewitta oraz Sobela, a także posłużono się detekcją masek „defektów" i laplasjanem morfologicznym. Z kolei do klasyfikacji zastosowano metody: LDA, KNN, klasyfikator Bayesa, SVM oraz drzewo decyzyjne. Klasyfikacja danych to proces dwuetapowy, kiedy to na początku należy stworzyć klasyfikator, który opisuje zbiór klas, a następnie otrzymany model zastosować do sklasyfikowania nowych danych.



Rys.10. Zdjęcia RTG płuc zdrowych [19].



Rys.11. Zdjęcia RTG płuc zmienionych chorobowo w wyniku wirusa COVID-19 [19].

Liniowa analiza dyskryminacyjna (ang. *Linear discriminant analysis*, LDA) jest używana w uczeniu maszynowym w celu znalezienia liniowej kombinacji cech, które najlepiej rozróżniają dwie lub więcej klas obiektów lub zdarzeń. Polega ona na znalezieniu hiperpłaszczyzny rzutowej, która minimalizuje wariancję międzyklasową i maksymalizuje odległość między rzutowanymi średnimi klas [9]. Szukany jest w niej klasyfikator, który pozwala na znalezienie tzw. granicy decyzyjnej, opisanej funkcją liniową, i podział całej przestrzeni na obszary odpowiadające klasom, a co za tym idzie przyporządkować jak najlepiej nowe obiekty *x* do tychże klas na podstawie podobieństwa ich cech. Spisuje się dobrze na prostych w analizie danych [10]

Jednym z kolejnych algorytmów używanych do klasyfikacji jest algorytm k najbliższych sąsiadów KNN (ang. *K nearest neighbours*). Jego działanie zależne jest od zbioru uczącego zawierającego obserwacje, z których każda ma przypisaną wartość zmiennej

objaśniającej, oraz z góry danej obserwacji O, dla której prognozowana jest wartość zmiennej objaśniającej. Algorytm ten porównuje wartości zmiennych objaśniających dla obserwacji O z wartościami tych zmiennych dla każdej obserwacji w zbiorze uczącym. Następnie dochodzi do wyboru założonej ilości *k* najbliższych do *C* obserwacji ze zbioru uczącego. Kolejno następuje uśrednienie i uzyskana zostaje żądana wartość.

Innym wykorzystanym klasyfikatorem jest naiwny klasyfikator Bayesa. To prosty probabilistyczny klasyfikator, który zakłada wzajemną niezależność zmiennych zależnych. Od tego jak dokładny jest model prawdopodobieństwa zależy skuteczność uczenia się tego klasyfikatora w trybie pod nadzorem. W analizie Bayesowskiej przyjęte z góry prawdopodobieństwo przynależności nowego obiektu do jakieś klasy nazywane są prawdopodobieństwami a priori. Prawdopodobieństwa a priori wynikają z posiadanych, wcześniejszych (a priori) obserwacji. W tym wypadku, chodzi o procent zielonych względem czerwonych. Prawdopodobieństwa a priori często służą do przewidywania klasy nieznanych przypadków, zanim one się pojawią. Pomimo swojej „naiwności" i prostoty działania w wielu sytuacjach klasyfikator Bayesa działa lepiej niż można było się tego spodziewać [12].

Użyty przez nas w analizie klasyfikator SVM (ang. *Support Vector Machine*, maszyna wektorów wspierających) ma na celu wyznaczenie płaszczyzny rozdzielającej dane należące do dwóch (lub więcej) klas z maksymalnym marginesem. W przypadku dwóch klas proces ten polega na znalezieniu najbliżej leżących siebie wektorów poprowadzonych od początku przyjętego układu współrzędnych do dwóch punktów należących do różnych klas [12].

Ostatnią stosowaną w projekcie metodą klasyfikacji jest drzewo decyzyjne. To graficzna metoda wspomagania procesu decyzyjnego. Konstrukcja takich drzew przebiega w kilku krokach. Na początku bada się, czy zbiór obiektów jest jednorodny, a następnie rozpatruje wszystkie możliwe podziały zbioru obiektów na podzbiory w jak najlepszy sposób ze względu na przyjęte kryterium. W metodzie tej następuje również tzw. kategoryzacja drzewa, czyli likwidacja tych fragmentów, które mają małe znaczenie dla jakości i wyników klasyfikacji. Po tak podzielonych podzbiorach drzewo można zastosować do sklasyfikowania nowych obiektów [13].

W projekcie do ekstrakcji cech stosowane były przede wszystkim filtry. Odpowiadają one za wyodrębnienie konkretnych cech obrazów tak, aby stworzyć odpowiednie wzorce do rozpoznawania. Maska Prewitta oraz Sobela są filtrami konturowymi, które służą do wykrywania krawędzi (przy czym filtr Sobela jest maską wagową). Z kolei używany filtr medianowy jest filtrem nieliniowym i ma tą właściwość, że poprzez usuwanie zakłóceń, nie niszczy on obrazu źródłowego, gdyż wybiera jedną z wartości (w tym przypadku medianę - wartość środkową z uporządkowanych rosnąco wartości pikseli) z otoczenia danego punktu z obrazu źródłowego [14].

Zadanie podobne do wcześniej wspomnianego filtra Prewitta czy Sobela ma detekcja masek „defektów". Wykorzystuje się ją do wyodrębniania elementów posiadających kształt i wielkość określoną za pomocą elementu strukturalnego SE [15]. Wspomnianą zależność można zapisać następująco:

$$Det\ m(L, SE) = (L - min[O(C(L, SE), SE), L])$$

gdzie *O* oznacza otwarcie, *C* zamknięcie, zaś *min* to minimum.

Zbliżoną rolę spełnia także laplasjan morfologiczny, którego działanie określone jest następującą zależnością:

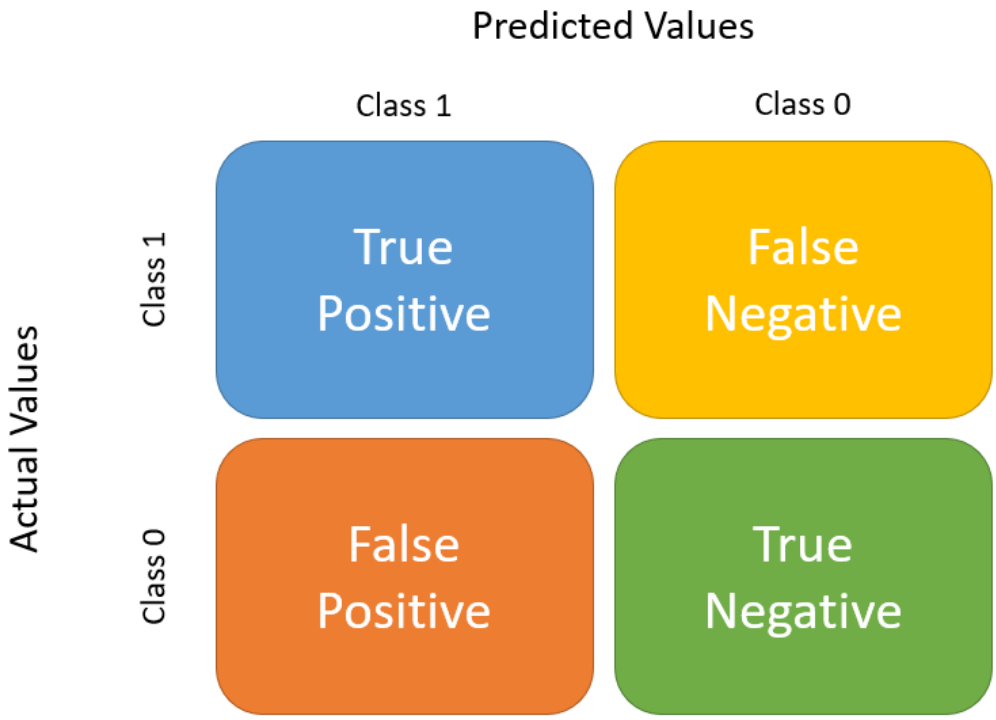
$$Laplace(L, SE) = (1/2)(\max_{SE}(L) + \min_{SE}(L) - 2L)$$

gdzie $\max_{SE}(L)$ oznacza dylatację [16].

Aby poprawnie przyporządkowane zdjęcia rentgenowskie płuc zdrowych oraz po COVID-19 do odpowiednich grup, zastosowana również została metoda analizy głównych składowych (ang. *principal component analysis*, PCA), która z kolei służy m.in. do redukcji liczby zmiennych opisujących zbiór czy do odkrycia prawidłowości między zmiennymi. Polega ona na wyznaczeniu składowych, które stanowią kombinację liniową badanych zmiennych. Analiza składowych głównych umożliwia wskazanie tych zmiennych, które mają duży wpływ na wygląd poszczególnych składowych głównych (tworzących grupę jednorodną). Taka składowa główna, u której wariancja jest zmaksymalizowana, staje się wówczas reprezentantem danej grupy [15].

5. Rozpoznawanie obrazów

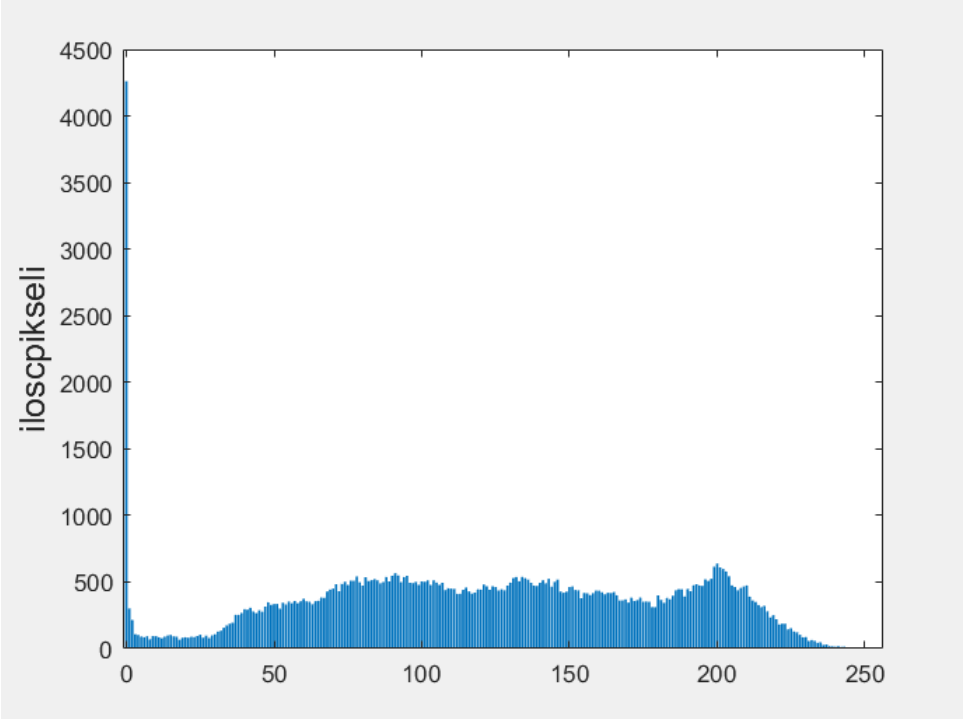
Rozpoznawania obrazów dokonywano powyżej opisanymi metodami przy pomocy odpowiednich dla nich funkcji w programie Matlab. Poniżej pokazano kody programów oraz wyniki ich działania, a także macierze pomyłek i krzywe ROC (*receiver operating characteristic curve*) uzyskane w teście wykonanym na 300 losowo wybranych obrazach.



Rys.12. Postać, w jakiej będą przedstawiane macierze pomyłek [19]

5.1 Klasyfikacja przy użyciu LDA

```
clear all
close all

%Wykreślenie histogramu w celu znalezienia progu binaryzacji (przykład dla obrazu COVID1.png)

L1=imread('COVID1.png');
figure, imshow(L1);
[y, x]=imhist(L1);
figure, bar(x,y);
ylabel('iloscpikseli', 'FontSize', 15, 'FontName', 'Arial CE');
```



Rys.13. Histogram uzyskany dla obrazu COVID1.png.

```
%Tworzenie wzorców
[L1]= imread('COVID1.png');
L1=L1>40;
figure, imshow(L1);
COVID1=sum(sum(L1));

[L2]= imread('COVID2.png');
L2=L2>40;
figure, imshow(L2);
COVID2=sum(sum(L2));

[L3]= imread('Normal1.png');
```

```
L3=L3>40;
figure, imshow(L3);
normal1=sum(sum(L3));

[L4]= imread('Normal2.png');
L4=L4>40;
figure, imshow(L4);
normal2=sum(sum(L4));

%Identyfikacja
[L5]= imread('Normal3.png');
L5=L5>40;
figure, imshow(L5);
Normal3=sum(sum(L5));

[L6]= imread('COVID3.png');
L6=L6>40;
figure, imshow(L6);
COVID3=sum(sum(L6));

%LDA
SL=[COVID1;COVID2;Normal1;Normal2];
group = ["COVID_";"COVID_";"zdrowy";"zdrowy"];
X=[COVID3; Normal3];
[C,err,P,logp,coeff] = classify([X],[SL], group,'linear');
C
```

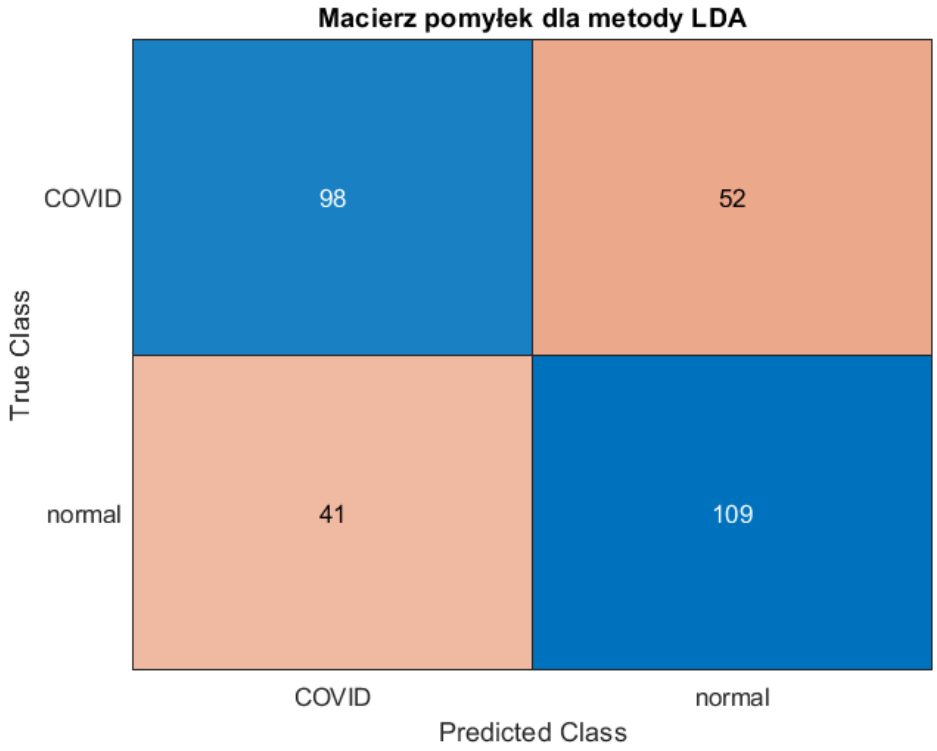
Wynik działania programu:

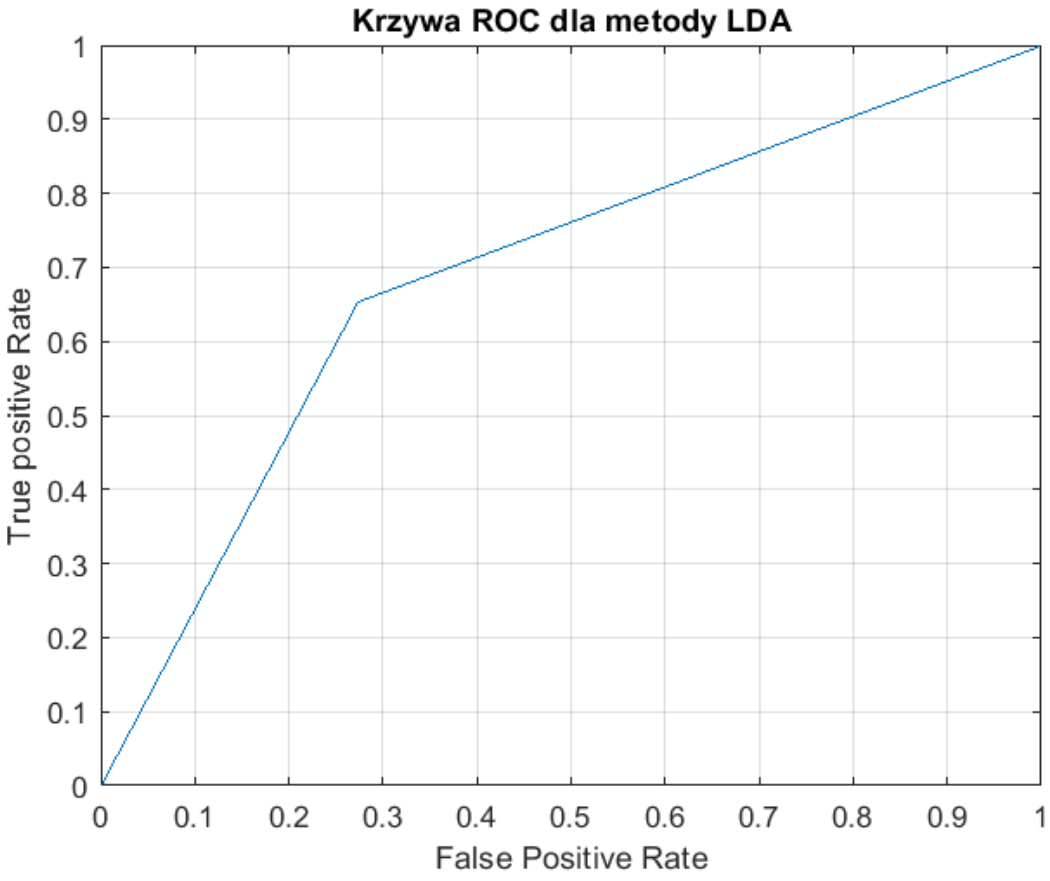
```
>> G_LDA_proj

C =

  2×6 char array

    'COVID_'
    'zdrowy'
```

Dla zbioru treningowego 150 COVID + 150 zdrowe (dla grupy testowej taka sama liczba danych)

```
clear all
close all

%%
imds_t = imageDatastore('Data_training', 'IncludeSubfolders',true,
'LabelSource','foldernames');
```

```
imds_s = imageDatastore('Data_sample', 'IncludeSubfolders',true,
'LabelSource','foldernames');

TRAINING = [];
SAMPLE   = [];
GROUP    = [];

%% stworzenie zbiorow treningowych i probki do identyfikacji

for i = 1:300
   path = imds_t.Files(i);
   path = string(path);
   [L1]= imread(path);
   L1=L1>110;
   TRAINING = [TRAINING; sum(sum(L1))];
   GROUP = [GROUP; imds_t.Labels(i)];
end

%% Identyfikacja

for i = 1:300
   path = imds_s.Files(i);
   path = string(path);
   [L1]= imread(path);
   L1=L1>110;
   SAMPLE = [SAMPLE; sum(sum(L1))];
end

%% LDA

[C,err,P,logp,coeff] = classify([TRAINING],[SAMPLE], GROUP,'linear');
```


Macierz pomyłek dla metody LDA
COVID
normal
True Class
98
52
41
109
COVID
normal
Predicted Class


Krzywa ROC dla metody LDA
True positive Rate
False Positive Rate
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

5.2 Klasyfikacja przy użyciu LDA z ekstrakcją cech metodą PCA

```
clear all
close all

%Tworzenie wzorców
[L1]= imread('COVID1.png');
L1=double(L1)/255;
figure, imshow(L1);
[pc1, zscores1, pcvars1] = pca(L1);
COVID1=sum(sum(pcvars1));

[L2]= imread('COVID2.png');
L2=double(L2)/255;
figure, imshow(L2);
[pc1, zscores1, pcvars1] = pca(L2);
COVID2=sum(sum(pcvars1));

[L3]= imread('Normal1.png');
L3=double(L3)/255;
figure, imshow(L3);
[pc1, zscores1, pcvars1] = pca(L3);
Normal1=sum(sum(pcvars1));

[L4]= imread('Normal2.png');
L4=double(L4)/255;
figure, imshow(L4);
[pc1, zscores1, pcvars1] = pca(L4);
Normal2=sum(sum(pcvars1));

% Identyfikacja
[L5]= imread('Normal3.png');
L5=double(L5)/255;
figure, imshow(L5);
[pc1, zscores1, pcvars1] = pca(L5);
Normal3=sum(sum(pcvars1));

[L6]= imread('COVID3.png');
L6=double(L6)/255;
figure, imshow(L6);
[pc1, zscores1, pcvars1] = pca(L6);
COVID3=sum(sum(pcvars1));

%LDA
```

```
SL=[COVID1;COVID2;Normal1;Normal2];
group = ['COVID_';'COVID_';'zdrowy';'zdrowy'];
X=[COVID3; Normal3];
[C,err,P,logp,coeff]= classify([X],[SL], group,'linear');
```

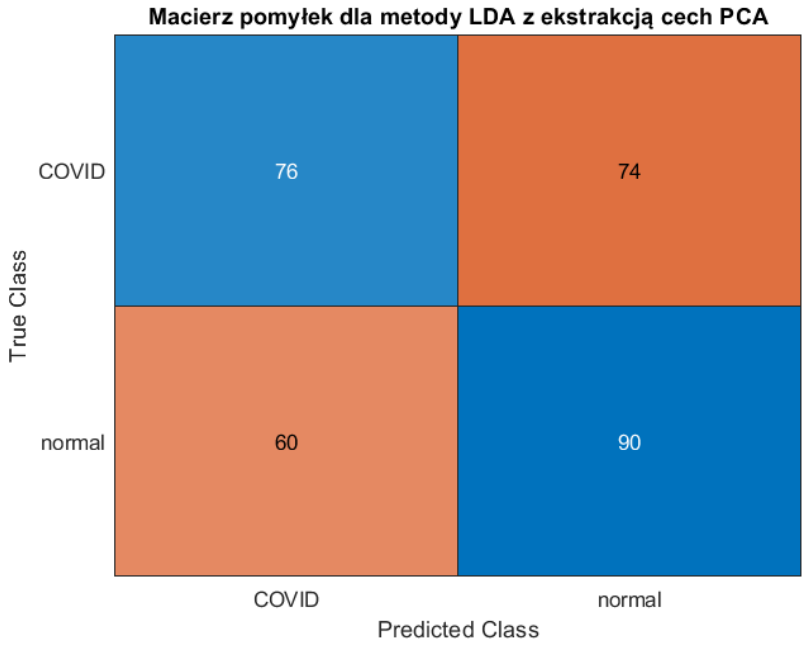
Wynik programu:

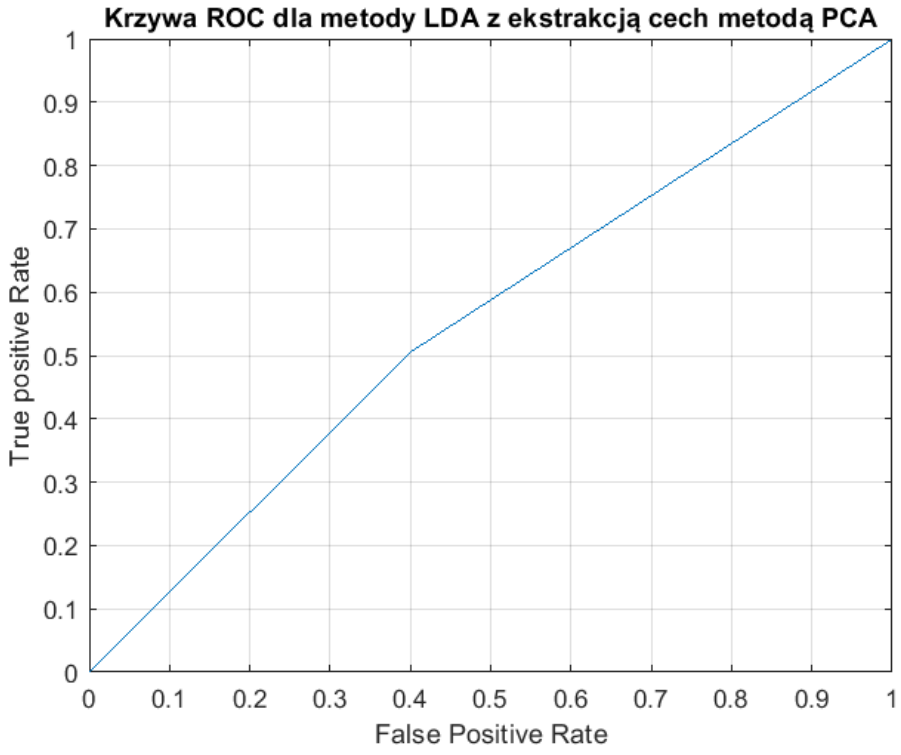
```
>> G_LDA_PCA

C =

  2×6 char array

    'COVID_'
    'COVID_'
```

Dla większego zbioru danych:

```
clear all
close all

%% Przygotowanie danych
imds_t = imageDatastore('Data_training', 'IncludeSubfolders',true,
'LabelSource','foldernames');
imds_s = imageDatastore('Data_sample', 'IncludeSubfolders',true,
'LabelSource','foldernames');

TRAINING = [];
SAMPLE   = [];
GROUP    = [];

%% Tworzenie wzorców

for i = 1:300
   path = imds_t.Files(i);
   path = string(path);
   [L1]= imread(path);
   L1=double(L1)/255;
   [pc1, zscores1, pcvars1] = pca(L1);
   TRAINING = [TRAINING; sum(sum(pcvars1))];
   GROUP = [GROUP; imds_t.Labels(i)];
end

% Identyfikacja

for i = 1:300
```

```
    path = imds_s.Files(i);
    path = string(path);
    [L5]= imread(path);
    L5=double(L5)/255;
    [pc1, zscores1, pcvars1] = pca(L5);
    SAMPLE = [SAMPLE; sum(sum(pcvars1))];
end

%% LDA

[C,err,P,logp,coeff] = classify([TRAINING],[SAMPLE], GROUP,'linear');
```





5.3 Klasyfikacja przy użyciu LDA z filtrem medianowym

```
clear all
close all

%Tworzenie wzorców
[L1]= imread('COVID1.png');
L1=medfilt2(L1, [3, 3]);
L1(L1<0)=0;
figure, imshow(L1);
COVID1=sum(sum(L1));

[L2]= imread('COVID2.png');
L2=medfilt2(L2, [3, 3]);
L2(L2<0)=0;
figure, imshow(L2);
COVID2=sum(sum(L2));

[L3]= imread('Normal1.png');
L3=medfilt2(L3, [3, 3]);
L3(L3<0)=0;
figure, imshow(L3);
Normal1=sum(sum(L3));

[L4]= imread('Normal2.png');
L4=medfilt2(L4, [3, 3]);
L4(L4<0)=0;
figure, imshow(L4);
Normal2=sum(sum(L4));

%Identyfikacja
[L5]= imread('Normal3.png');
L5=medfilt2(L5, [3, 3]);
L5(L5<0)=0;
figure, imshow(L5);
Normal3=sum(sum(L5));

[L6]= imread('COVID3.png');
L6=medfilt2(L6, [3, 3]);
L6(L6<0)=0;
figure, imshow(L6);
COVID3=sum(sum(L6));

SL=[COVID1;COVID2;Normal1;Normal2];
group = ['COVID_';'COVID_';'zdrowy';'zdrowy'];
X=[COVID3; Normal3];
[C,err,P,logp,coeff] = classify([X],[SL], group,'linear');
```

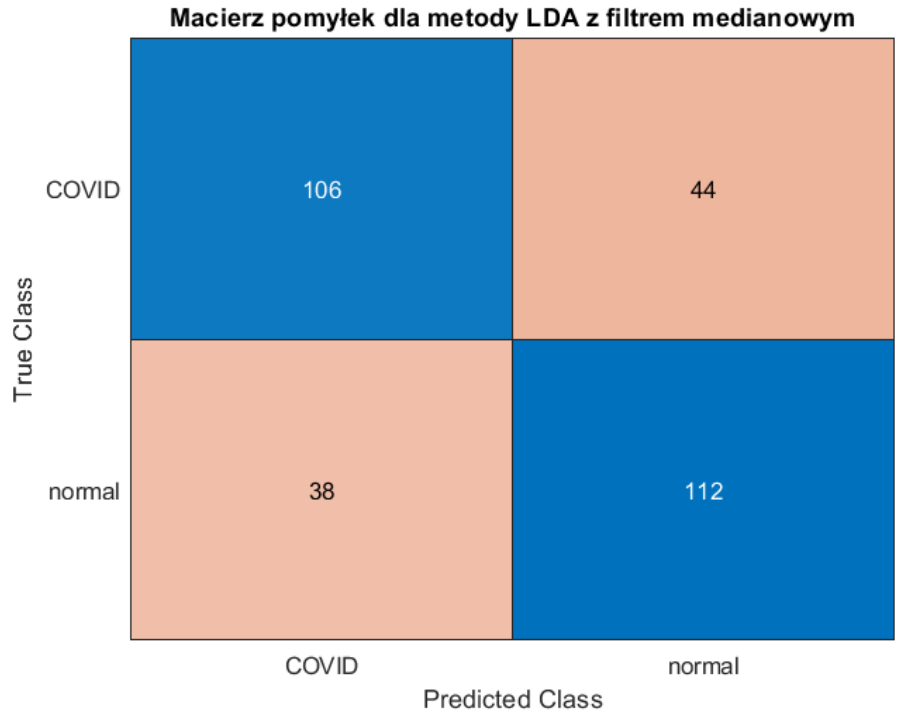
Wynik działania programu:

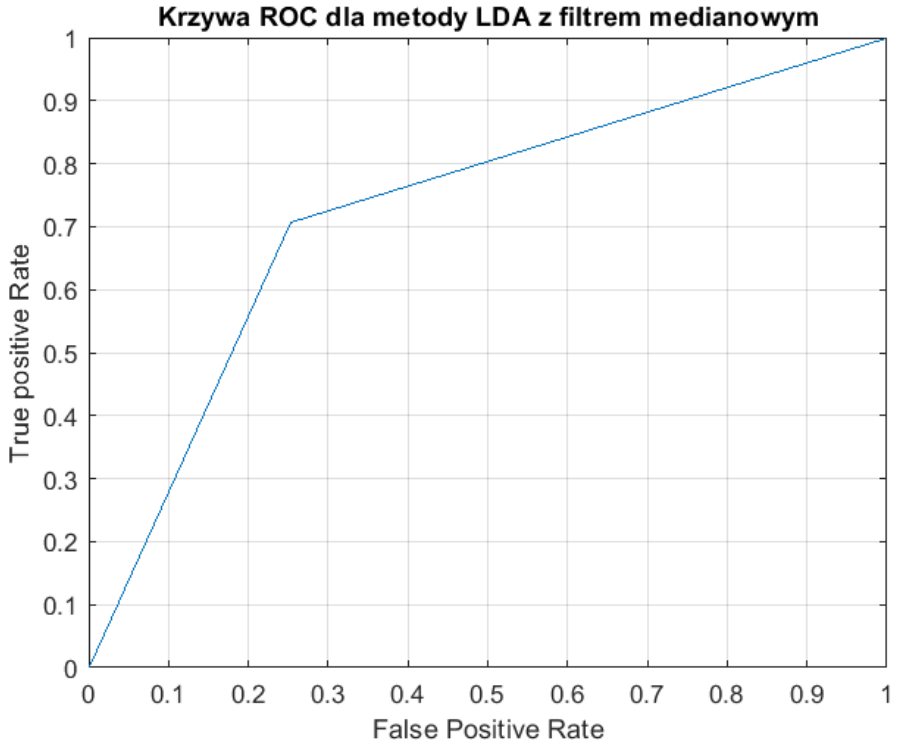
```
>> G_LDA_median

C =

  2×6 char array

    'COVID_'
    'zdrowy'
```

Dla większego zbioru danych:

```
clear all
close all

%% Przygotowanie danych
imds_t = imageDatastore('Data_training', 'IncludeSubfolders',true,
'LabelSource','foldernames');
imds_s = imageDatastore('Data_sample', 'IncludeSubfolders',true,
'LabelSource','foldernames');

TRAINING = [];
SAMPLE   = [];
GROUP    = [];

%% Tworzenie wzorców

for i = 1:300
   path = imds_t.Files(i);
   path = string(path);
   [L1]= imread(path);
   L1=medfilt2(L1, [3, 3]);
   L1(L1<0)=0;
   TRAINING = [TRAINING; sum(sum(L1))];
   GROUP = [GROUP; imds_t.Labels(i)];
end

% Identyfikacja

for i = 1:300
   path = imds_s.Files(i);
   path = string(path);
   [L5]= imread(path);
   L5=medfilt2(L5, [3, 3]);
   L5(L5<0)=0;
   SAMPLE = [SAMPLE; sum(sum(L5))];
end
```

```
%% LDA

[C,err,P,logp,coeff] = classify([TRAINING],[SAMPLE], GROUP,'linear');
```





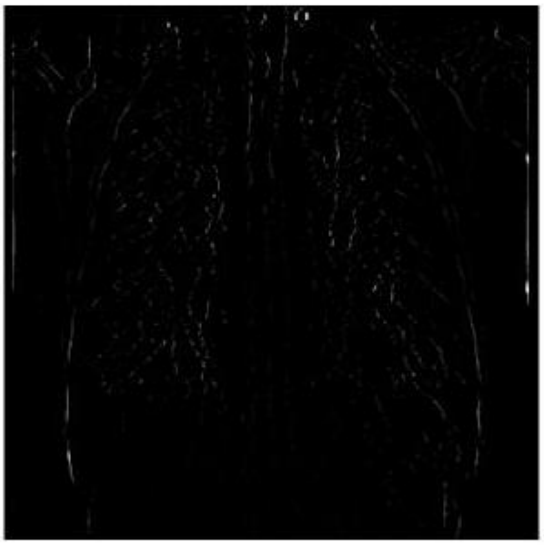
5.4 Klasyfikacja przy użyciu LDA z detekcją masek „defektów”

```
clear all
close all

% Wzorce uzyte do rozpoznawania zdjec płuc zdrowych oraz covidowych
% Uzyta metoda: detekcja masek "defektow"


SE=ones([1 3]);
[L1]= imread('COVID1.png');
L1a=(L1-min(imopen(imclose(L1, SE),SE),L1));
figure, imshow(L1a, []);
COVID1=sum(sum(L1a));

[L2]= imread('COVID2.png');
L2a=(L2-min(imopen(imclose(L2, SE),SE),L2));
figure, imshow(L2a, []);
COVID2=sum(sum(L2a));

[L3]= imread('Normal1.png');
L3a=(L3-min(imopen(imclose(L3, SE),SE),L3));
figure, imshow(L3a, []);
Normal1=sum(sum(L3a));

[L4]= imread('Normal2.png');
L4a=(L4-min(imopen(imclose(L4, SE),SE),L4));
figure, imshow(L4a, []);
Normal2=sum(sum(L4a));

% Identyfikacja zdjec pod katem covid
% metoda: LDA


[L5]= imread('Normal3.png');
L5a=(L5-min(imopen(imclose(L5, SE),SE),L5));
figure, imshow(L5a, []);
Normal3=sum(sum(L5a));

[L6]= imread('COVID3.png');
L6a=(L6-min(imopen(imclose(L6, SE),SE),L6));
figure, imshow(L6a, []);
COVID3=sum(sum(L6a));

% LDA

SL=[COVID1; COVID2; Normal1; Normal2];
group = ['COVID_';'COVID_'; 'zdrowy'; 'zdrowy'];
X=[COVID3; Normal3];
[C,err,P,logp,coeff] = classify([X],[SL], group,'linear');
```

C

Wynik działania programu:

```
>> G_detmas_LDA

C =

  2×6 char array

    'zdrowy'
    'zdrowy'
```

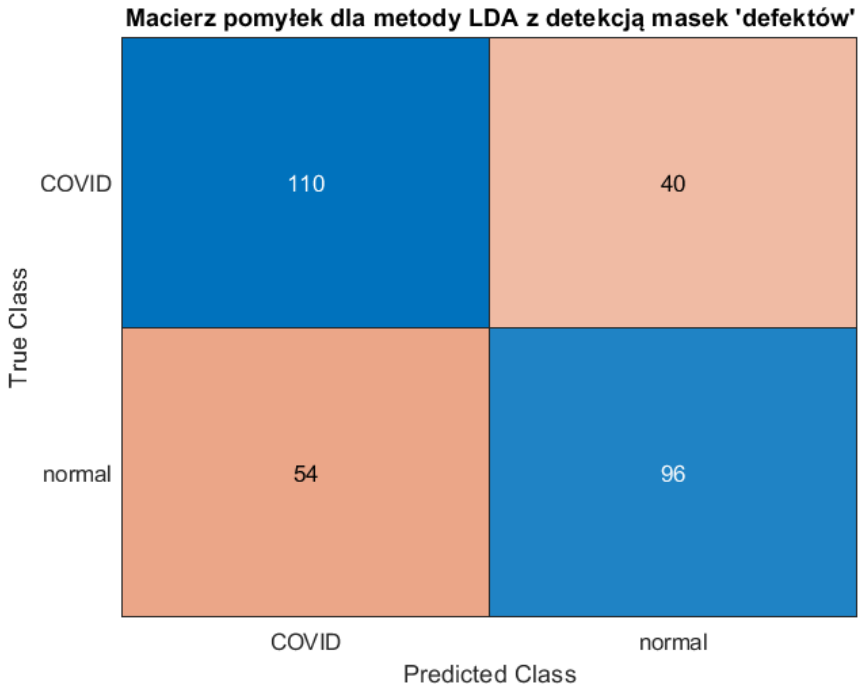


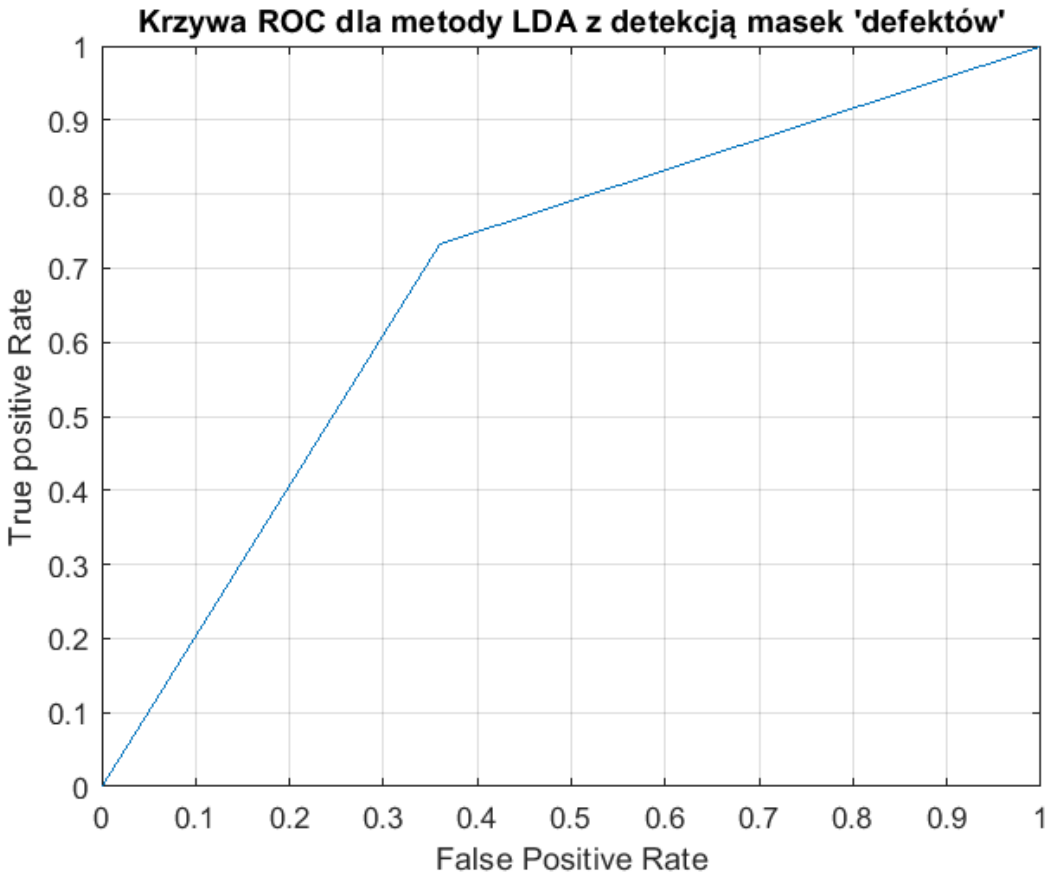
Rys.14. Obraz płuc zdrowych po zastosowaniu detekcji masek „defektów”.

Rys.15. Obraz płuc chorych na COVID detekcji masek „defektów”.

Dla większego zbioru danych:

```
clear all
close all

%% Przygotowanie danych
imds_t = imageDatastore('Data_training', 'IncludeSubfolders',true,
'LabelSource','foldernames');
imds_s = imageDatastore('Data_sample', 'IncludeSubfolders',true,
'LabelSource','foldernames');

SE       = ones([1 3]);
TRAINING = [];
SAMPLE   = [];
GROUP    = [];

%% Tworzenie wzorców

for i = 1:300
   path = imds_t.Files(i);
   path = string(path);
   [L1] = imread(path);
   L1a = (L1-min(imopen(imclose(L1, SE),SE),L1));
   TRAINING = [TRAINING; sum(sum(L1a))];
   GROUP = [GROUP; imds_t.Labels(i)];
end

% Identyfikacja

for i = 1:300
   path = imds_s.Files(i);
   path = string(path);
   [L5] = imread(path);
   L5a = (L5-min(imopen(imclose(L5, SE),SE),L5));
   SAMPLE = [SAMPLE; sum(sum(L5a))];
end

%% LDA

[C,err,P,logp,coeff] = classify([TRAINING],[SAMPLE], GROUP,'linear');
```


Macierz pomyłek dla metody LDA z detekcją masek 'defektów'
COVID
normal
True Class
110
40
54
96
COVID
normal
Predicted Class


Krzywa ROC dla metody LDA z detekcją masek 'defektów'
True positive Rate
False Positive Rate
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

5.5 Klasyfikacja przy użyciu klasyfikatora *k* najbliższych sąsiadów (KNN) z ekstrakcją cech metodą PCA

```
close all
clear all

% Wzorce uzyte do rozpoznawania zdjec chorych i zdrowych
% Uzyta metoda: PCA


[L1]= imread('COVID1.png');
L1=double(L1)/255;
[pc1, zscores1, pcvars1] = pca(L1); % analiza glownych skladowych
COVID1=sum(sum(pcvars1));

[L2]= imread('COVID2.png');
L2=double(L2)/255;
[pc2, zscores2, pcvars2] = pca(L2);
COVID2=sum(sum(pcvars2));

[L3]= imread('Normal1.png');
L3=double(L3)/255;
[pc3, zscores3, pcvars3] = pca(L3);
Normal1=sum(sum(pcvars3));

[L4]= imread('Normal2.png');
L4=double(L4)/255;
[pc4, zscores4, pcvars4] = pca(L4);
Normal2=sum(sum(pcvars4));

% Identyfikacja zdjec pod katem COVID
% Metoda k najblizszych sasiadow - KNN


[L5]= imread('COVID3.png');
L5=double(L5)/255;
[pc5, zscores5, pcvars5] = pca(L5);
COVID3=sum(sum(pcvars5));

[L6]= imread('Normal3.png');
L6=double(L6)/255;
[pc6, zscores6, pcvars6] = pca(L6);
Normal3=sum(sum(pcvars6));

% KNN

SL=[Normal1; Normal2; COVID1; COVID2];
group = ['n' ; 'n' ; 'C' ; 'C'];
```

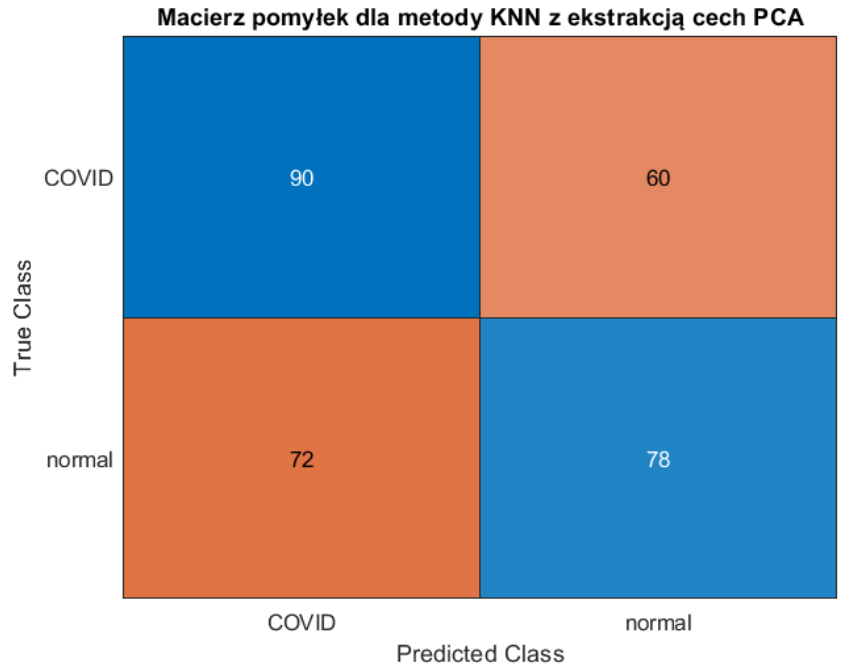
```
X=[Normal3];
Y=[COVID3];

[n,d] = knnsearch(SL , X ,'k',20);
[n1,d1] = knnsearch(SL , Y ,'k',20);

SL(n,:);
SL(n1,:);
S1=tabulate(group(n));
S2=tabulate(group(n1));

S3=S1{1};
S4=S2{1};
 if(S3=='C')
    S5='COVID'
 end
 if(S3=='n')
    S5='zdrowy'
 end

 if(S4=='C')
    S6='COVID'
 end
 if(S4=='n')
    S6='zdrowy'
 end
```

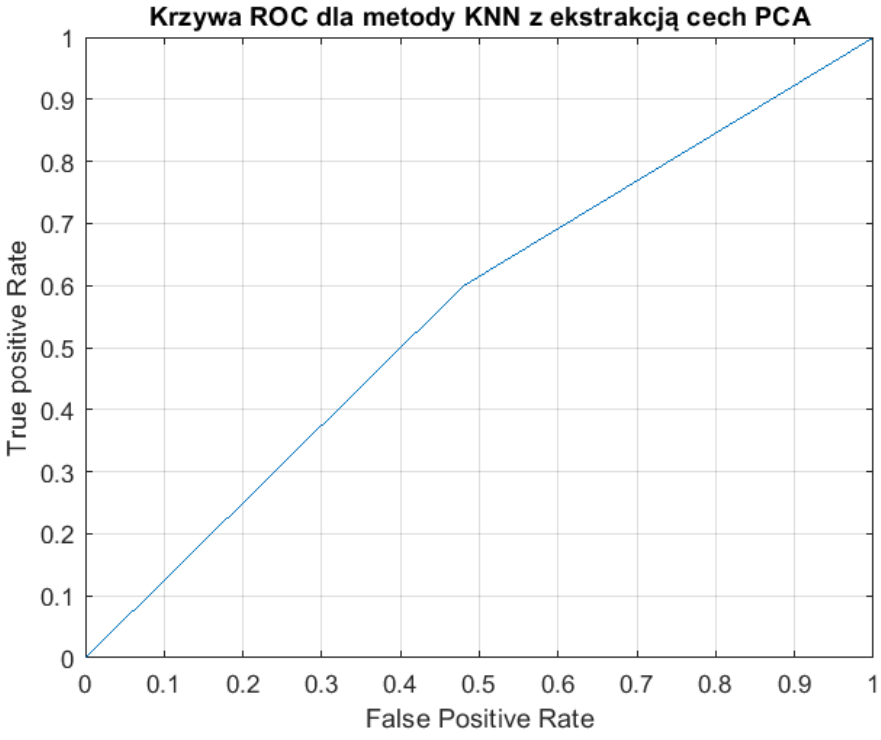
Wynik działania programu:

```
>> G_PCA_KNN

S5 =

     'COVID'


S6 =

     'COVID'
```

```
clear all
close all

%% Przygotowanie danych
imds_t = imageDatastore('Data_training', 'IncludeSubfolders',true,
'LabelSource','foldernames');
```

```
imds_s = imageDatastore('Data_sample', 'IncludeSubfolders',true,
'LabelSource','foldernames');

TRAINING = [];
SAMPLE   = [];
GROUP    = [];

%% Tworzenie wzorców

for i = 1:300
   path = imds_t.Files(i);
   path = string(path);
   [L1]= imread(path);
   L1=double(L1)/255;
   [pc1, zscores1, pcvars1] = pca(L1);
   TRAINING = [TRAINING; sum(sum(pcvars1))];
   GROUP = [GROUP; imds_t.Labels(i)];
end

% Identyfikacja

for i = 1:300
   path = imds_s.Files(i);
   path = string(path);
   [L5]= imread(path);
   L5=double(L5)/255;
   [pc1, zscores1, pcvars1] = pca(L5);
   SAMPLE = [SAMPLE; sum(sum(pcvars1))];
end

%% KNN

model = fitcknn(TRAINING,GROUP,'NumNeighbors',5);
```


Macierz pomyłek dla metody KNN z ekstrakcją cech PCA
COVID
normal
True Class
90
60
72
78
COVID
normal
Predicted Class


Krzywa ROC dla metody KNN z ekstrakcją cech PCA
True positive Rate
False Positive Rate
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

5.6 Klasyfikacja przy użyciu klasyfikatora k najbliższych sąsiadów (KNN) z filtrem Sobela

```
clear all
close all

% Wzorce uzyte do rozpoznawania zdjec chorych i zdrowych
% Uzyta metoda: filtr Sobela

[L1, map]= imread('COVID1.png');
h=fspecial('sobel');
L1a=filter2(h,L1);
figure, imshow(L1a);
L1a(L1a<0)=0;
COVID1=sum(sum(L1a));

[L2, map]= imread('COVID2.png');
h=fspecial('sobel');
L2a=filter2(h,L2);
figure, imshow(L2a);
L2a(L2a<0)=0;
COVID2=sum(sum(L2a));

[L3, map]= imread('Normal1.png');
h=fspecial('sobel');
L3a=filter2(h,L3);
figure, imshow(L3a);
L3a(L3a<0)=0;
Normal1=sum(sum(L3a));

[L4, map]= imread('Normal2.png');
%L4=rgb2gray(L4);
h=fspecial('sobel');
L4a=filter2(h,L4);
figure, imshow(L4a);
L4a(L4a<0)=0;
Normal2=sum(sum(L4a));


% Identyfikacja zdjec pod katem COVID
% Metoda: KNN

[L5, map]= imread('COVID3.png');
h=fspecial('sobel');
L5a=filter2(h,L5);
figure, imshow(L5a);
L5a(L5a<0)=0;
COVID3=sum(sum(L5a));
```

```
[L6, map]= imread('Normal3.png');
h=fspecial('sobel');
L6a=filter2(h,L6);
figure, imshow(L6a);
L6a(L6a<0)=0;
Normal3=sum(sum(L6a));

%KNN

SL=[Normal1; Normal2; COVID1; COVID2];
group = ['n' ; 'n' ; 'C' ; 'C'];
X=[Normal3];
Y=[COVID3];

[n,d] = knnsearch(SL , X ,'k',20);
[n1,d1] = knnsearch(SL , Y ,'k',20);

SL(n,:);
SL(n1,:);
S1=tabulate(group(n));
S2=tabulate(group(n1));

S3=S1{1};
S4=S2{1};
 if(S3=='C')
    S5='COVID'
 end
 if(S3=='n')
    S5='zdrowy'
 end

 if(S4=='C')
    S6='COVID'
 end
 if(S4=='n')
    S6='zdrowy'
 end
```

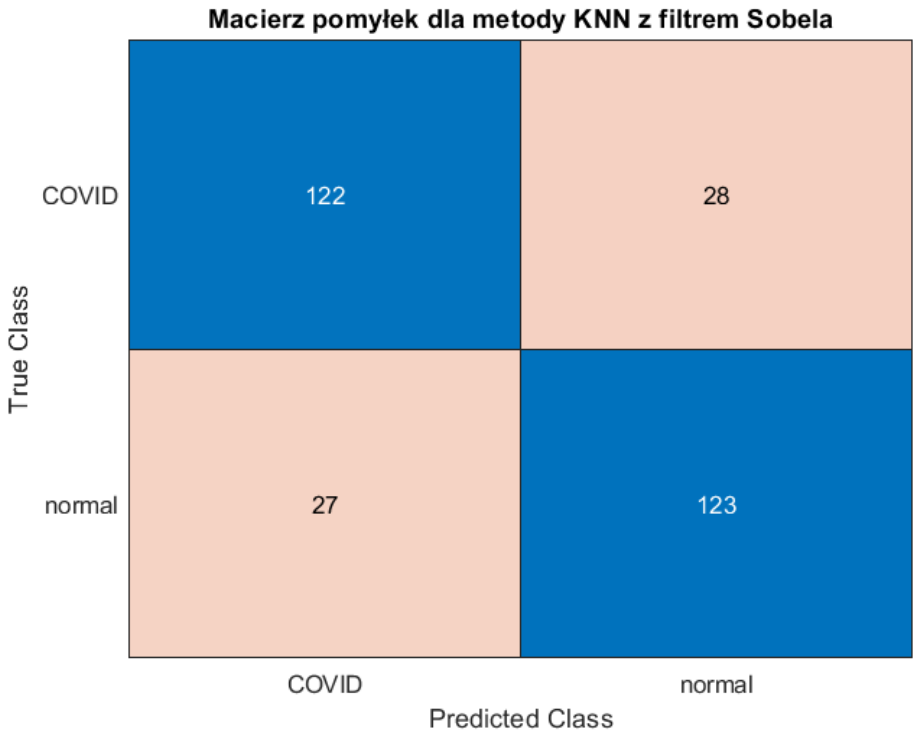
Wynik działania programu:

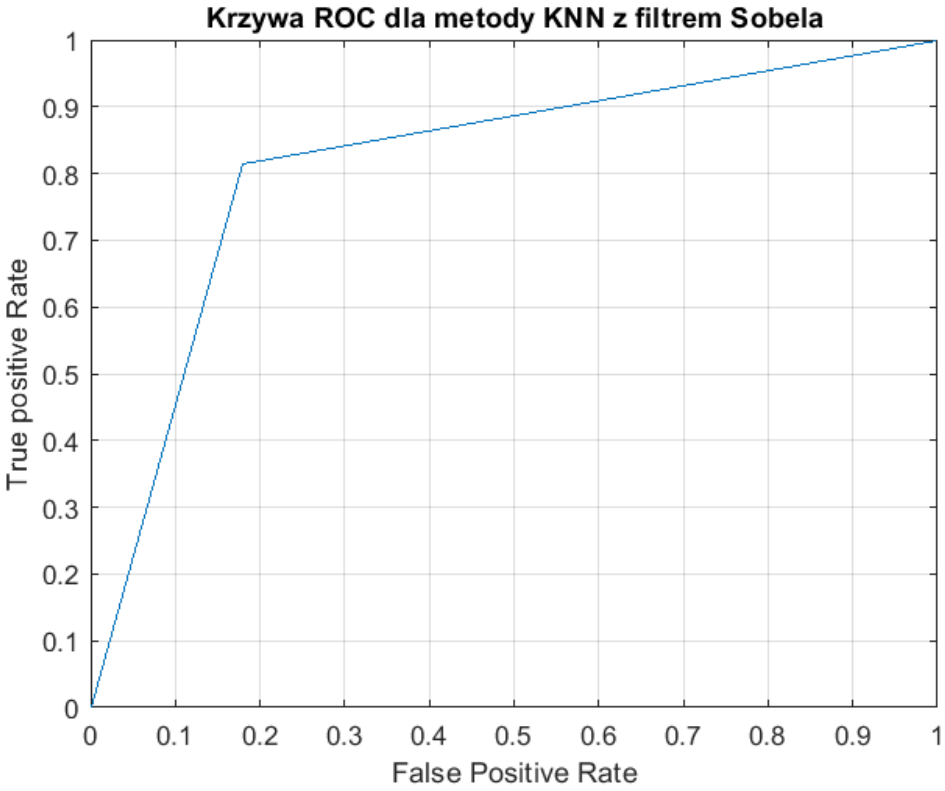
```
>> G_Sobel_KNN

S5 =

    'zdrowy'


S6 =

    'zdrowy'
```

Rys.16. Obraz chorych płuc przefiltrowany filtrem Sobela.

Rys.17. Obraz zdrowych płuc przefiltrowany filtrem Sobela.

```
clear all
close all

%% Przygotowanie danych
imds_t = imageDatastore('Data_training', 'IncludeSubfolders',true,
'LabelSource','foldernames');
imds_s = imageDatastore('Data_sample', 'IncludeSubfolders',true,
'LabelSource','foldernames');

TRAINING = [];
SAMPLE   = [];
GROUP    = [];

%% Tworzenie wzorców

for i = 1:300
   path = imds_t.Files(i);
   path = string(path);
   [L1,map]= imread(path);
   h=fspecial('sobel');
   L1a=filter2(h,L1);
   L1a(L1a<0)=0;
   TRAINING = [TRAINING; sum(sum(L1a))];
   GROUP = [GROUP; imds_t.Labels(i)];
end

% Identyfikacja

for i = 1:300
   path = imds_s.Files(i);
   path = string(path);
   [L5,map]= imread(path);
   h=fspecial('sobel');
   L5a=filter2(h,L5);
   L5a(L5a<0)=0;
   SAMPLE = [SAMPLE; sum(sum(L5a))];
end

%% KNN

model = fitcknn(TRAINING,GROUP,'NumNeighbors',5);
```


Macierz pomyłek dla metody KNN z filtrem Sobela
COVID
normal
True Class
122
28
27
123
COVID
normal
Predicted Class


Krzywa ROC dla metody KNN z filtrem Sobela
True positive Rate
False Positive Rate
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

5.7 Klasyfikacja przy użyciu klasyfikatora k najbliższych sąsiadów (KNN) z laplasjanem morfologicznym

```
clear all
close all

% Wzorce uzyte do rozpoznawania zdjec chorych i zdrowych
% Uzyta metoda: laplasjan morfologiczny

SE=ones([1 2]);
[L1]= imread('COVID1.png');
L1a=0.5*(imdilate(L1,SE)+imerode(L1, SE)-2*(L1));
figure, imshow(L1a, []);
COVID1=sum(sum(L1a));

[L2]= imread('COVID2.png');
L2a=0.5*(imdilate(L2,SE)+imerode(L2, SE)-2*(L2));
figure, imshow(L2a, []);
COVID2=sum(sum(L2a));

[L3]= imread('Normal1.png');
L3a=0.5*(imdilate(L3,SE)+imerode(L3, SE)-2*(L3));
figure, imshow(L3a, []);
Normal1=sum(sum(L3a));

[L4]= imread('Normal2.png');
L4a=0.5*(imdilate(L4,SE)+imerode(L4, SE)-2*(L4));
figure, imshow(L4a, []);
Normal2=sum(sum(L4a));

% Identyfikacja zdjec pod katem covid
% Metoda: KNN

[L5]= imread('Normal3.png');
L5a=0.5*(imdilate(L5,SE)+imerode(L5, SE)-2*(L5));
figure, imshow(L5a, []);
Normal3=sum(sum(L5a));

[L6]= imread('COVID3.png');
L6a=0.5*(imdilate(L6,SE)+imerode(L6, SE)-2*(L6));
figure, imshow(L6a, []);
COVID3=sum(sum(L6a));
```

```
% KNN

SL=[Normal1; Normal2; COVID1; COVID2];
group = ['n' ; 'n' ; 'C' ; 'C'];
X=[Normal3];
Y=[COVID3];

[n,d] = knnsearch(SL , X ,'k',20);
[n1,d1] = knnsearch(SL , Y ,'k',20);

SL(n,:);
SL(n1,:);
S1=tabulate(group(n));
S2=tabulate(group(n1));

S3=S1{1};
S4=S2{1};
 if(S3=='C')
    S5='COVID'
 end
 if(S3=='n')
    S5='zdrowy'
 end

 if(S4=='C')
    S6='COVID3'
 end
 if(S4=='n')
    S6='zdrowy'
 end
```

Wynik działania programu:

```
>> G_lap_KNN

S5 =

    'zdrowy'


S6 =

    'zdrowy'
```

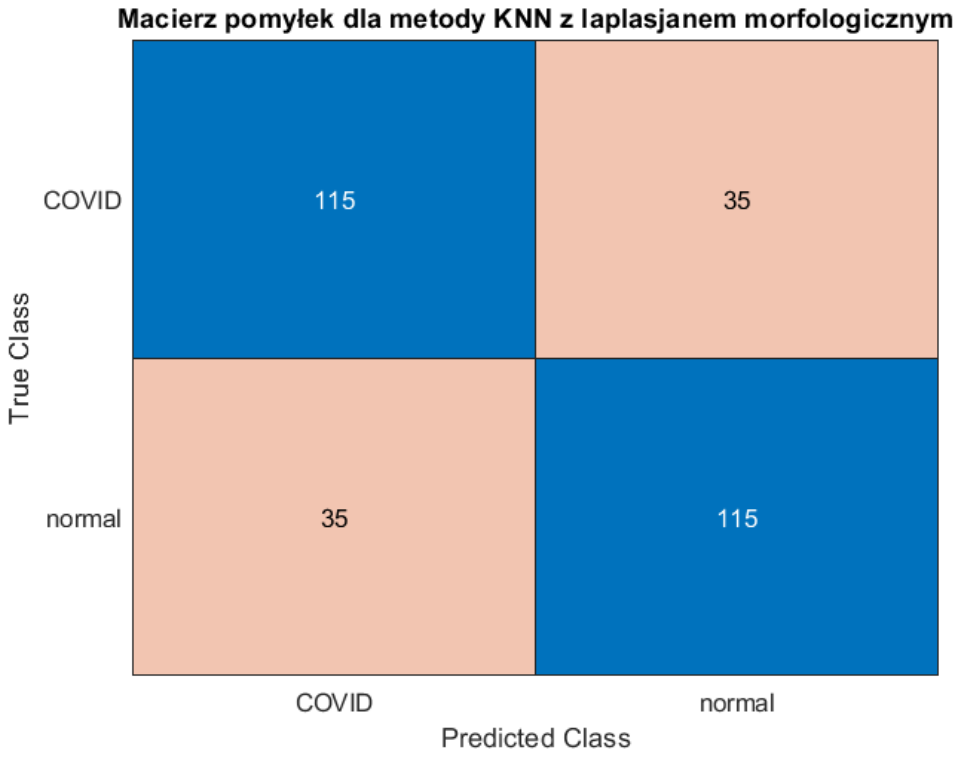
Rys.18. Obraz chorych płuc po zastosowaniu laplasjanu morfologicznego.

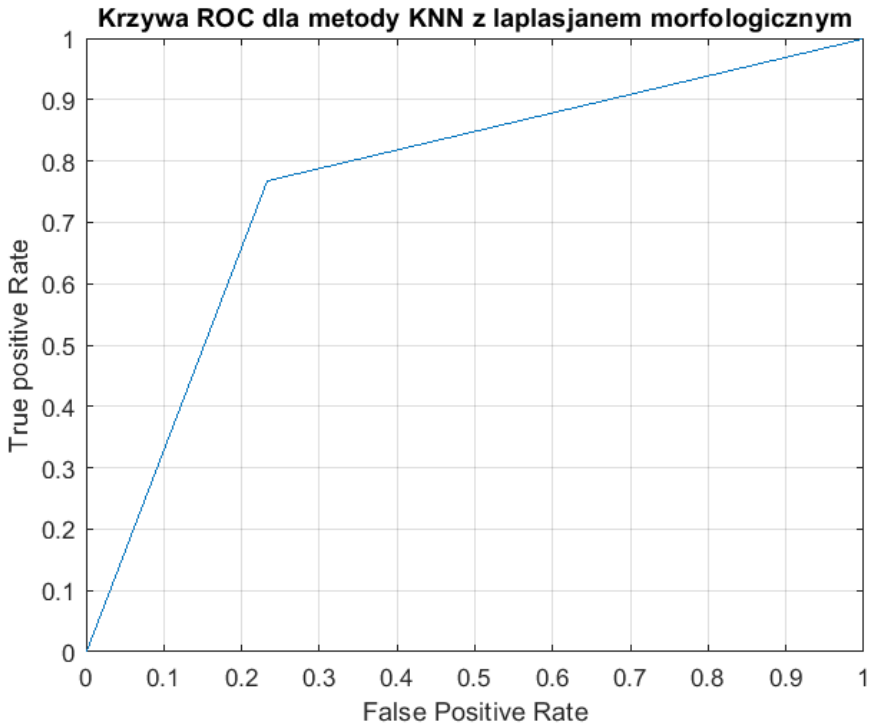
Rys.19. Obraz zdrowych płuc po zastosowaniu laplasjanu morfologicznego.

Dla większego zbioru danych:

```
clear all
close all

%% Przygotowanie danych
imds_t = imageDatastore('Data_training', 'IncludeSubfolders',true,
'LabelSource','foldernames');
imds_s = imageDatastore('Data_sample', 'IncludeSubfolders',true,
'LabelSource','foldernames');

SE       = ones([1 2]);
TRAINING = [];
SAMPLE   = [];
```

```
GROUP      = [];

%% Tworzenie wzorców

for i = 1:300
   path = imds_t.Files(i);
   path = string(path);
   [L1]= imread(path);
   L1a=0.5*(imdilate(L1,SE)+imerode(L1, SE)-2*(L1));
   TRAINING = [TRAINING; sum(sum(L1a))];
   GROUP = [GROUP; imds_t.Labels(i)];
end

% Identyfikacja

for i = 1:300
   path = imds_s.Files(i);
   path = string(path);
   [L5]= imread(path);
   L5a=0.5*(imdilate(L5,SE)+imerode(L5, SE)-2*(L5));
   SAMPLE = [SAMPLE; sum(sum(L5a))];
end

%% KNN

model = fitcknn(TRAINING,GROUP,'NumNeighbors',5);
```





5.8 Klasyfikacja przy użyciu klasyfikatora Bayesa z ekstrakcją cech metodą PCA

```
clear all
close all

% Wzorce uzyte do rozpoznawania zdjec płuc zdrowych i po COVID
% Uzyta metoda: PCA

[L1]= imread('COVID1.png');
L1=double(L1)/255;
[pc1, zscores1, pcvars1] = pca(L1); % analiza glownych skladowych
COVID1=sum(sum(pcvars1));

[L2]= imread('COVID2.png');
L2=double(L2)/255;
[pc2, zscores2, pcvars2] = pca(L2);
COVID2=sum(sum(pcvars2));

[L3]= imread('Normal1.png');
```

```
L3=double(L3)/255;
[pc3, zscores3, pcvars3] = pca(L3);
Normal1=sum(sum(pcvars3));

[L4]= imread('Normal2.png');
L4=double(L4)/255;
[pc4, zscores4, pcvars4] = pca(L4);
Normal2=sum(sum(pcvars4));

% Identyfikacja zdjec pod katem COVID
% Metoda: klasyfikator Bayesa

[L5]= imread('COVID3.png');
L5=double(L5)/255;
[pc5, zscores5, pcvars5] = pca(L5);
COVID3=sum(sum(pcvars5));

[L6]= imread('Normal3.png');
L6=double(L6)/255;
[pc6, zscores6, pcvars6] = pca(L6);
Normal3=sum(sum(pcvars6));

% Bayes

SL=[Normal1; Normal2; COVID1; COVID2];
group = ['zdrowy' ; 'zdrowy' ; 'COVID_' ; 'COVID_'];
X=[Normal3];
Y=[COVID3];

nb = fitcnb([SL], group);

x = predict(nb,X)
y = predict(nb,Y)
```

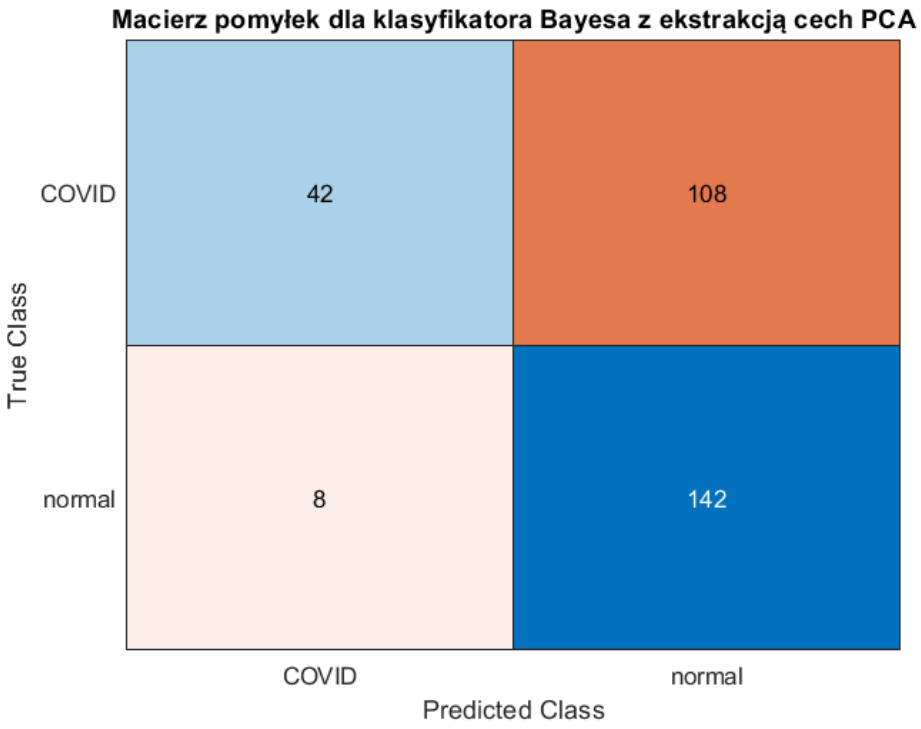
Wynik działania programu:

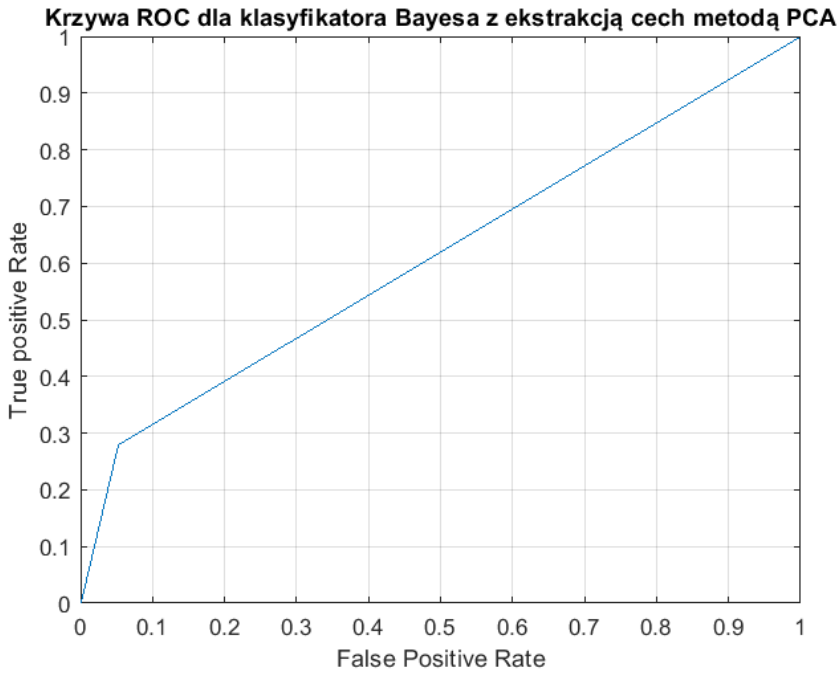
```
>> G_PCA_Bayes

x =

    'COVID_'


y =

    'COVID_'
```

Dla większej ilości danych:

```
clear all
close all

%% Przygotowanie danych
imds_t = imageDatastore('Data_training', 'IncludeSubfolders',true,
'LabelSource','foldernames');
imds_s = imageDatastore('Data_sample', 'IncludeSubfolders',true,
'LabelSource','foldernames');

TRAINING = [];
SAMPLE   = [];
GROUP    = [];

%% Tworzenie wzorców

for i = 1:300
   path = imds_t.Files(i);
   path = string(path);
   [L1]= imread(path);
   L1=double(L1)/255;
   [pc1, zscores1, pcvars1] = pca(L1); % analiza glownych skladowych
   TRAINING = [TRAINING; sum(sum(pcvars1))];
   GROUP = [GROUP; imds_t.Labels(i)];
end

% Identyfikacja

for i = 1:300
   path = imds_s.Files(i);
   path = string(path);
   [L5]= imread(path);
   L5=double(L5)/255;
```

```
    [pc5, zscores5, pcvars5] = pca(L5);
    SAMPLE = [SAMPLE; sum(sum(pcvars5))];
end

%% Klasyfikator Bayesa

nb = fitcnb(TRAINING, GROUP);
```





5.9 Klasyfikacja przy użyciu SVM z filtrem Prewitta

```
clear all
close all

%Tworzenie wzorców
[L1]= imread('COVID1.png');
h=fspecial('prewitt');
L1=filter2(h,L1);
L1(L1<0)=0;
figure, imshow(L1);
COVID1=sum(sum(L1));

[L2]= imread('COVID2.png');
h=fspecial('prewitt');
L2=filter2(h,L2);
L2(L2<0)=0;
figure, imshow(L2);
COVID2=sum(sum(L2));

[L3]= imread('Normal1.png');
h=fspecial('prewitt');
L3=filter2(h,L3);
L3(L3<0)=0;
figure, imshow(L3);
Normal1=sum(sum(L3));

[L4]= imread('Normal2.png');
h=fspecial('prewitt');
L4=filter2(h,L4);
L4(L4<0)=0;
figure, imshow(L4);
Normal2=sum(sum(L4));

%Identyfikacja
[L5]= imread('Normal3.png');
h=fspecial('prewitt');
L5=filter2(h,L5);
L5(L5<0)=0;
figure, imshow(L5);
Normal3=sum(sum(L5));
```

```
[L6]= imread('COVID3.png');
h=fspecial('prewitt');
L6=filter2(h,L6);
L6(L6<0)=0;
figure, imshow(L6);
COVID3=sum(sum(L6));

SL=[COVID1;COVID2;Normal1;Normal2];
group = ['COVID_';'COVID_';'zdrowy';'zdrowy'];
X=[COVID3; Normal3];
SVMStruct=fitcsvm([SL], group);
C=predict(SVMStruct, [X]);
C
```

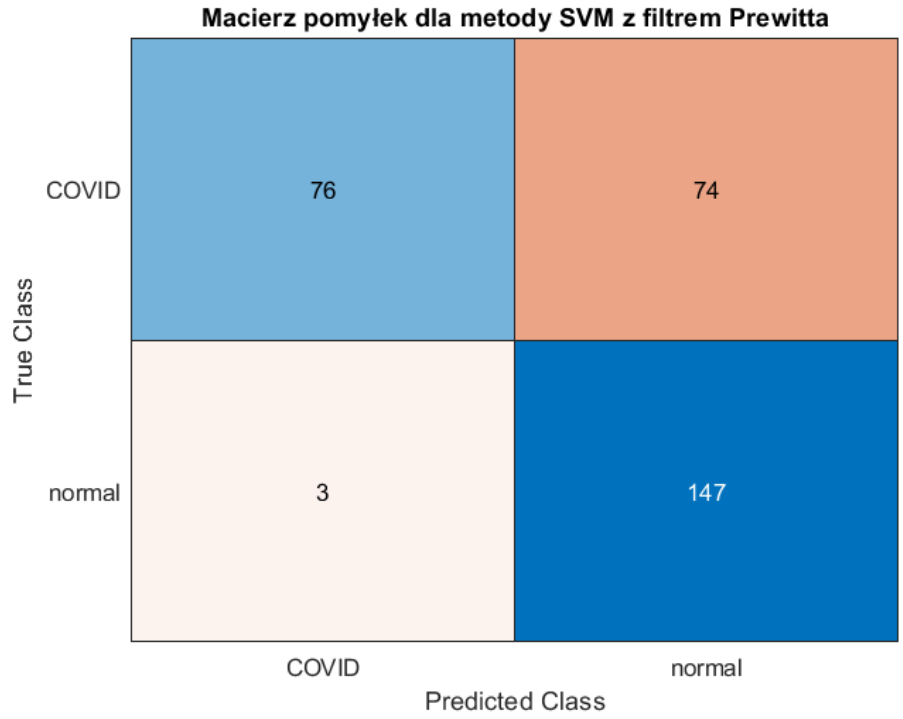
Wynik działania programu:

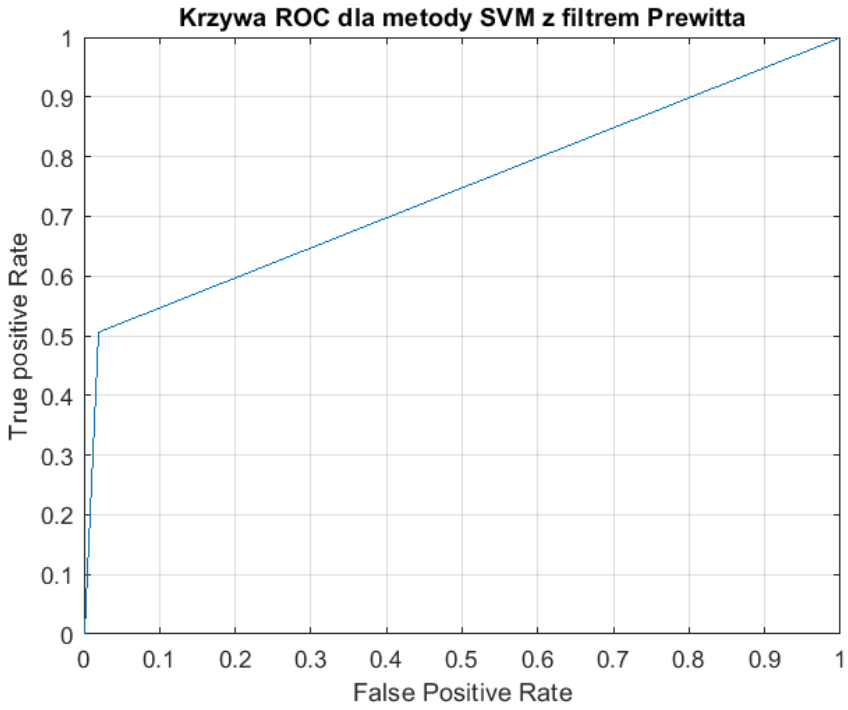
```
>> G_SVM_prewitt

C =

  2×6 char array

    'COVID_'
    'COVID_'
```

Dla większej ilości danych:

```
clear all
close all

%% Przygotowanie danych
imds_t = imageDatastore('Data_training', 'IncludeSubfolders',true,
'LabelSource','foldernames');
imds_s = imageDatastore('Data_sample', 'IncludeSubfolders',true,
'LabelSource','foldernames');

TRAINING = [];
SAMPLE   = [];
GROUP    = [];

%% Tworzenie wzorców

for i = 1:300
   path = imds_t.Files(i);
   path = string(path);
```

```
    [L1]= imread(path);
    h=fspecial('prewitt');
    L1a=filter2(h,L1);
    L1a(L1a<0)=0;
    TRAINING = [TRAINING; sum(sum(L1a))];
    GROUP = [GROUP; imds_t.Labels(i)];
end

% Identyfikacja

for i = 1:300
    path = imds_s.Files(i);
    path = string(path);
    [L5]= imread(path);
    h=fspecial('prewitt');
    L5a=filter2(h,L5);
    L5a(L5a<0)=0;
    SAMPLE = [SAMPLE; sum(sum(L5a))];
end

%% SVM

SVMStruct = fitcsvm(TRAINING, GROUP);
```





5.10 Klasyfikacja przy użyciu drzewa decyzyjnego z ekstrakcją cech metodą PCA

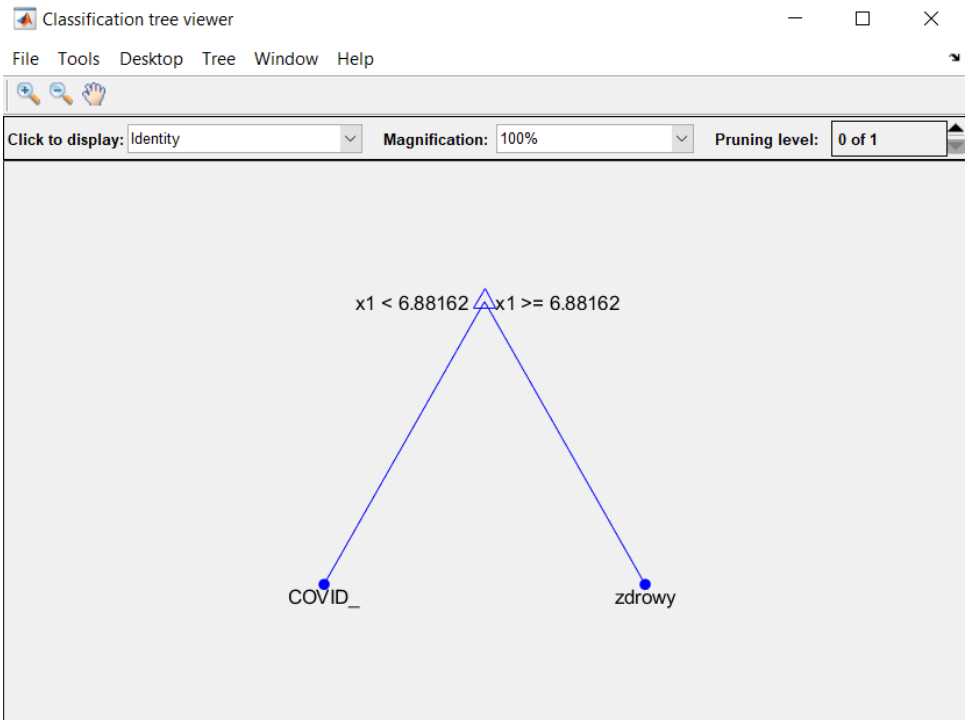
```
clear all
close all

%Tworzenie wzorców
[L1]= imread('COVID1.png');
L1=double(L1)/255;
figure, imshow(L1);
[pc1, zscores1, pcvars1] = pca(L1);
COVID1=sum(sum(pcvars1));

[L2]= imread('COVID2.png');
L2=double(L2)/255;
figure, imshow(L2);
[pc1, zscores1, pcvars1] = pca(L2);
COVID2=sum(sum(pcvars1));
```

```
[L1a]= imread('COVID4.png');
L1a=double(L1a)/255;
figure, imshow(L1a);
[pc1, zscores1, pcvars1] = pca(L1a);
COVID4=sum(sum(pcvars1));

[L1b]= imread('COVID5.png');
L1b=double(L1b)/255;
figure, imshow(L1b);
[pc1, zscores1, pcvars1] = pca(L1b);
COVID5=sum(sum(pcvars1));

[L1c]= imread('COVID6.png');
L1c=double(L1c)/255;
figure, imshow(L1c);
[pc1, zscores1, pcvars1] = pca(L1c);
COVID6=sum(sum(pcvars1));

[L3]= imread('Normal1.png');
L3=double(L3)/255;
figure, imshow(L3);
[pc1, zscores1, pcvars1] = pca(L3);
Normal1=sum(sum(pcvars1));

[L4]= imread('Normal2.png');
L4=double(L4)/255;
figure, imshow(L4);
[pc1, zscores1, pcvars1] = pca(L4);
Normal2=sum(sum(pcvars1));

[L3a]= imread('Normal4.png');
L3a=double(L3a)/255;
figure, imshow(L3a);
[pc1, zscores1, pcvars1] = pca(L3a);
Normal4=sum(sum(pcvars1));

[L3b]= imread('Normal5.png');
L3b=double(L3b)/255;
figure, imshow(L3b);
[pc1, zscores1, pcvars1] = pca(L3b);
Normal5=sum(sum(pcvars1));

[L3c]= imread('Normal6.png');
L3c=double(L3c)/255;
```

```
figure, imshow(L3c);
[pc1, zscores1, pcvars1] = pca(L3c);
Normal6=sum(sum(pcvars1));

%Identyfikacja
[L5]= imread('Normal3.png');
L5=double(L5)/255;
figure, imshow(L5);
[pc1, zscores1, pcvars1] = pca(L5);
Normal3=sum(sum(pcvars1));

[L6]= imread('COVID3.PNG');
L6=double(L6)/255;
figure, imshow(L6);
[pc1, zscores1, pcvars1] = pca(L6);
COVID3=sum(sum(pcvars1));

%Drzewo decyzyjne

SL=[COVID1;COVID2;COVID4;COVID5;COVID6;Normal1;Normal2;Normal4;Normal5;
Normal6];
group =
['COVID_';'COVID_';'COVID_';'COVID_';'COVID_';'zdrowy';'zdrowy';'zdrowy';'zdrowy';'zd
rowy'];
X=[COVID3; Normal3];

tree = ClassificationTree.fit([SL], group)
view(tree, 'mode', 'graph');
C = predict(tree,[X]);
C
```

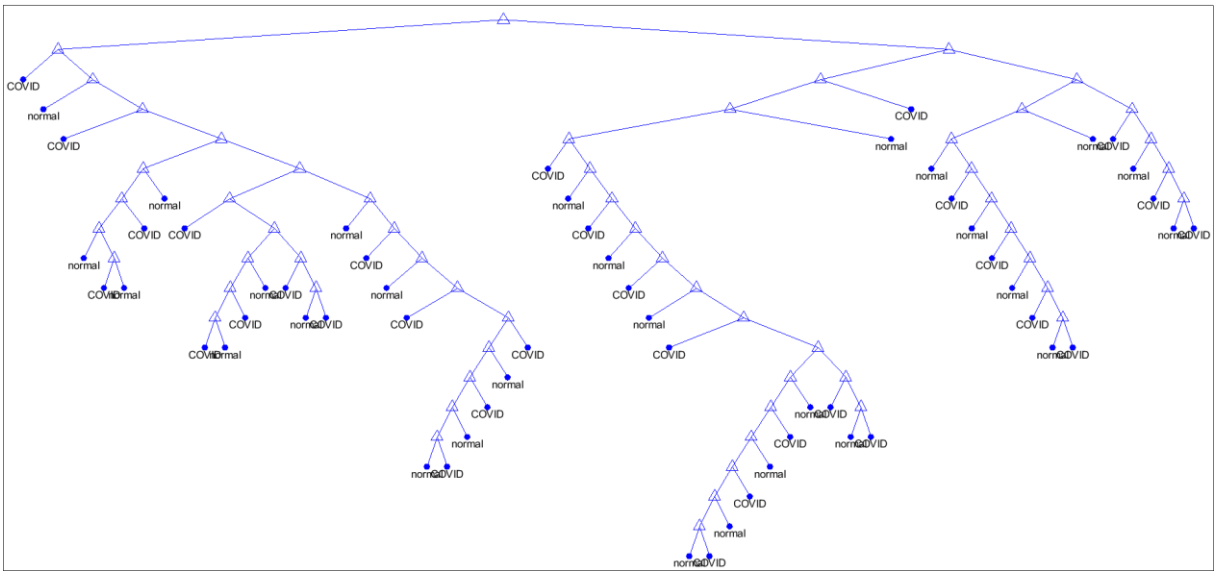
Wynik działania programu:

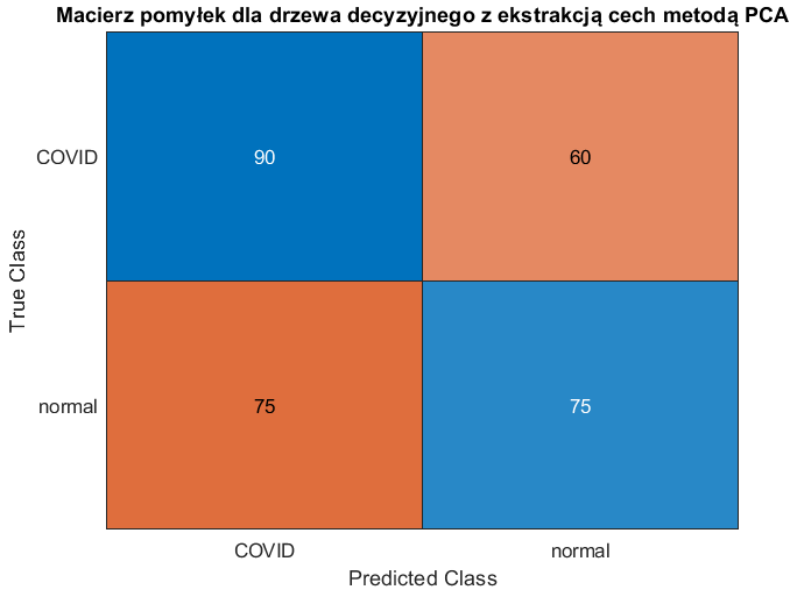
```
>> G_drzewo_PCA

tree =

  ClassificationTree
             ResponseName: 'Y'
    CategoricalPredictors: []
               ClassNames: [2×6 char]
           ScoreTransform: 'none'
          NumObservations: 10


  Properties, Methods


C =

  2×6 char array

    'zdrowy'
    'zdrowy'
```

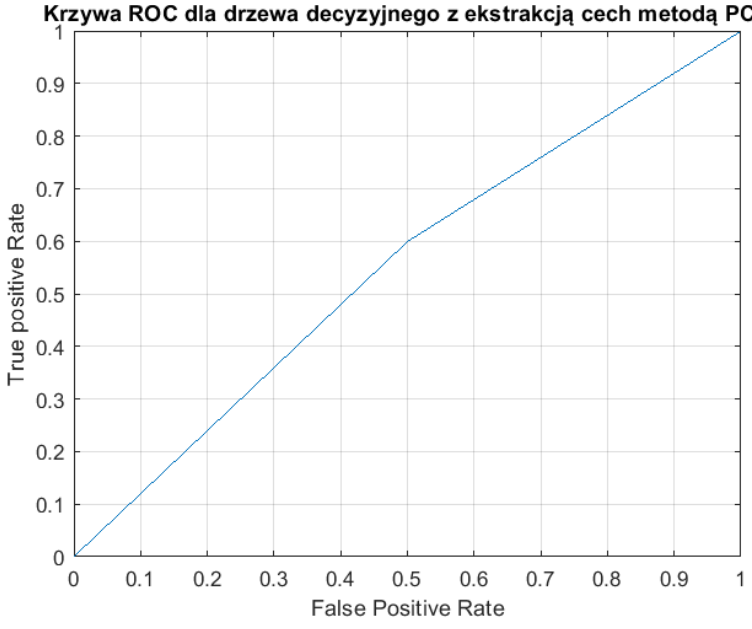


Dla większej ilości danych:

```
clear all
close all

%% Przygotowanie danych
imds_t = imageDatastore('Data_training', 'IncludeSubfolders',true,
'LabelSource','foldernames');
imds_s = imageDatastore('Data_sample', 'IncludeSubfolders',true,
'LabelSource','foldernames');

TRAINING = [];
SAMPLE   = [];
GROUP    = [];

%% Tworzenie wzorców

for i = 1:300
   path = imds_t.Files(i);
   path = string(path);
   [L1]= imread(path);
   L1=double(L1)/255;
   [pc1, zscores1, pcvars1] = pca(L1);
   TRAINING = [TRAINING; sum(sum(pcvars1))];
   GROUP = [GROUP; imds_t.Labels(i)];
end

% Identyfikacja

for i = 1:300
   path = imds_s.Files(i);
   path = string(path);
   [L5]= imread(path);
   L5=double(L5)/255;
   [pc1, zscores1, pcvars1] = pca(L5);
   SAMPLE = [SAMPLE; sum(sum(pcvars1))];
end

%% Drzewo decycyjne

tree = ClassificationTree.fit(TRAINING, GROUP);
view(tree, 'mode', 'graph');
```




Macierz pomyłek dla drzewa decyzyjnego z ekstrakcją cech metodą PCA
COVID
normal
90
60
75
75
True Class
COVID
normal
Predicted Class


Krzywa ROC dla drzewa decyzyjnego z ekstrakcją cech metodą PCA
True positive Rate
False Positive Rate
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

5.11 Ewaluacja wyników

Dla każdej metody obliczono następujące parametry:

- *Accuracy* – określa, ile przypadków zostało prawidłowo zaklasyfikowanych, zarówno jako pozytywne, jak i negatywne.

$$\frac{TP + TN}{TP + TN + FN + FP}$$

- *Recall* – mówi, ile pozytywnych przypadków zostało wykrytych ze wszystkich naprawdę pozytywnych.

$$\frac{TP}{TP + FN}$$

- *Precision* – określa, ile przypadków z uznanych przez klasyfikator za pozytywne jest w rzeczywistości pozytywnych.

$$\frac{TP}{TP + FP}$$

- *F1 Score* – średnia harmoniczna parametrów *precision* i *recall.* Im bliższy 1, tym lepiej radzi sobie klasyfikator.

$$2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}$$

- *False-Positive Rate* – związane jest z uzyskaniem wyniku pozytywnego, podczas gdy w rzeczywistości przypadek jest negatywny.

$$\frac{FP}{FP + TN}$$

- *AUC (area under curve)* – powierzchnia pod krzywą ROC, została wyliczona w programie Matlab z wykorzystaniem funkcji *perfcurve( )* oraz *trapz( )*. Im wartość ta jest bliższa 1, tym lepszy jest model – bardziej zbliżony do idealnego.

Gdzie:
TP (*true positive*) – liczba poprawnie sklasyfikowanych przykładów.
FN (*false negative*) - decyzja negatywna, podczas gdy przykład w rzeczywistości jest pozytywny.
TN (*true negative*) – liczba przykładów poprawnie odrzuconych.
FP (*false positive*) – liczba przykładów błędnie przydzielonych do wybranej klasy, podczas gdy w rzeczywistości do niej nie należą.

Tabela 1. Wyniki obliczeń dla własnych metod

Metoda	Accuracy	Recall	Precision	F1 Score	False-Positive Rate	AUC
LDA	0,69	0,65	0,71	0,68	0,27	0,69
LDA + PCA	0,55	0,51	0,56	0,53	0,40	0,53
LDA + filtr medianowy	0,73	0,71	0,74	0,72	0,25	0,72
LDA + maski defektów	0,69	0,73	0,67	0,70	0,36	0,69
KNN + PCA	0,56	0,60	0,56	0,58	0,48	0,56
KNN + filtr Sobela	0,82	0,81	0,82	0,81	0,18	0,82
KNN + laplasjan	0,77	0,77	0,77	0,77	0,23	0,77
Bayes + PCA	0,61	0,28	0,84	0,42	0,05	0,61
SVM + filtr Prewitta	0,74	0,51	0,96	0,67	0,02	0,74
Drzewo decyzyjne + PCA	0,55	0,60	0,55	0,57	0,50	0,55

Najwyższym parametrem *accuracy* (dokładnością) charakteryzuje się metoda KNN z filtrem Sobela. Również ona ma najwyższą wartość *recall*, czyli dobrze wykrywa pozytywne przypadki. Ma również najbardziej zbliżony do jedynki *F1 Score*, co świadczy o tym, że dobrze radzi sobie z klasyfikacją zdjęć, jak również wysoka wartość pola pod krzywą ROC (AUC) wskazuje, że spośród wszystkich sprawdzanych modeli ten jest najbliższy ideałowi.
Jednak większym *precision* charakteryzuje się metoda SVM z filtrem Prewitta (a także nieznacznie klasyfikator Bayesa z ekstrakcją cech metodą PCA). Posiada także najmniejsze ze wszystkich klasyfikatorów *False-positive rate*, ponieważ zaledwie 3 przypadki ze 150 zdjęć płuc zdrowych zostały błędnie zaklasyfikowane jako chore. Jednak mimo to lepszym wyborem jest metoda KNN z filtrem Sobela, ponieważ ważne jest wykrycie jak największej liczby przypadków pozytywnych, a więc zmaksymalizowanie wartości *recall*.
Natomiast najgorzej z klasyfikacją poradziły sobie modele oparte na drzewie decyzyjnym, liniowej analizie dyskryminacyjnej oraz KNN, wszystkie z ekstrakcją cech metodą PCA. Klasyfikują poprawnie niewiele ponad połowę przypadków, a o niezbyt dobrym działaniu świadczy również kształt krzywej ROC, dla wszystkich trzech zbliżony do linii prostej, co oznacza, że klasyfikują zdjęcia niemal losowo.
Bardzo złym wyborem byłby również klasyfikator Bayesa, cechujący się małą wartością *recall* – dużo przypadków chorych klasyfikuje jako zdrowe, co jest o wiele groźniejsze dla pacjentów, niż błędne zakwalifikowanie ich jako chorych jeśli są zdrowi.

6. Użycie gotowych modeli

Sprawdzono również działanie gotowych modeli dostępnych w programie, w module Classification Learner: SVM, klasyfikator Bayesa, LDA, drzewo decyzyjne oraz KNN, każdy w wersji wykorzystującej PCA oraz niekorzystającej z niego.

W pierwszym cyklu testów wykorzystano po 1000 zdjęć płuc zdrowych i zarażonych, zaś w drugim wykorzystano wszystkie obrazy płuc zdrowych (10192) i dotkniętych COVID (3616).

Ekstrakcji 200 cech dokonano metodą *bag of features*. Modele używające włączonej opcji PCA korzystały ze 159 cech w teście dla 1000 obrazów oraz 164 cech w przypadku testu na wszystkich obrazach. Walidacja została wykonana metodą *cross validation* (walidacja krzyżowa), polegającą na podzieleniu oryginalnej próby na *k* podzbiorów (w tym przypadku 5). Następnie kolejno każdy z nich został wzięty jako zbiór testowy, a pozostałe razem jako zbiór uczący, po czym wykonywana jest analiza. Uzyskane *k* rezultaty łączy się w celu uzyskania jednego wyniku, na przykład poprzez uśrednienie.

```
%% Wczytanie danych
imset = imageSet('Data', 'recursive');

%% Ekstrakcja cech
bag = bagOfFeatures(imset,'VocabularySize',200,...
    'PointSelection','Detector');
imagefeatures = encode(bag,imset);

%% Tworzenie tabeli z cechami
Data          = array2table(imagefeatures);
Data.health = getImageLabels(imset);

%% Uruchomienie classification learner
classificationLearner
```

Efekt działania programu dla wszystkich obrazów:

```
>> class_learn2

Creating Bag-Of-Features.
-------------------------
* Image category 1: COVID_all
* Image category 2: Normal_all
* Selecting feature point locations using the Detector method.
* Extracting SURF features from the selected feature point locations.
** detectSURFFeatures is used to detect key points for feature extraction.

* Extracting features from 3616 images in image set 1...done. Extracted 108473 features.
* Extracting features from 10192 images in image set 2...done. Extracted 579123 features.

* Keeping 80 percent of the strongest features from each category.

* Balancing the number of features across all image categories to improve clustering.
** Image category 1 has the least number of strongest features: 86778.
** Using the strongest 86778 features from each of the other image categories.

* Using K-Means clustering to create a 200 word visual vocabulary.
* Number of features          : 173556
* Number of clusters (K)      : 200

* Initializing cluster centers...100.00%.
* Clustering...completed 23/100 iterations (~1.14 seconds/iteration)...converged in 23 iterations.

* Finished creating Bag-Of-Features


Encoding images using Bag-Of-Features.
--------------------------------------
* Image category 1: COVID_all
* Image category 2: Normal_all
* Encoding 3616 images from image set 1...done.
* Encoding 10192 images from image set 2...done.

* Finished encoding images.
```

Po czym uruchomiony został Classification Learner i sprawdzono dokładność klasyfikacji dla 10 modeli. Zestawienie otrzymanych dokładności dla obu testów znajduje się poniżej.

Po 1000 obrazów

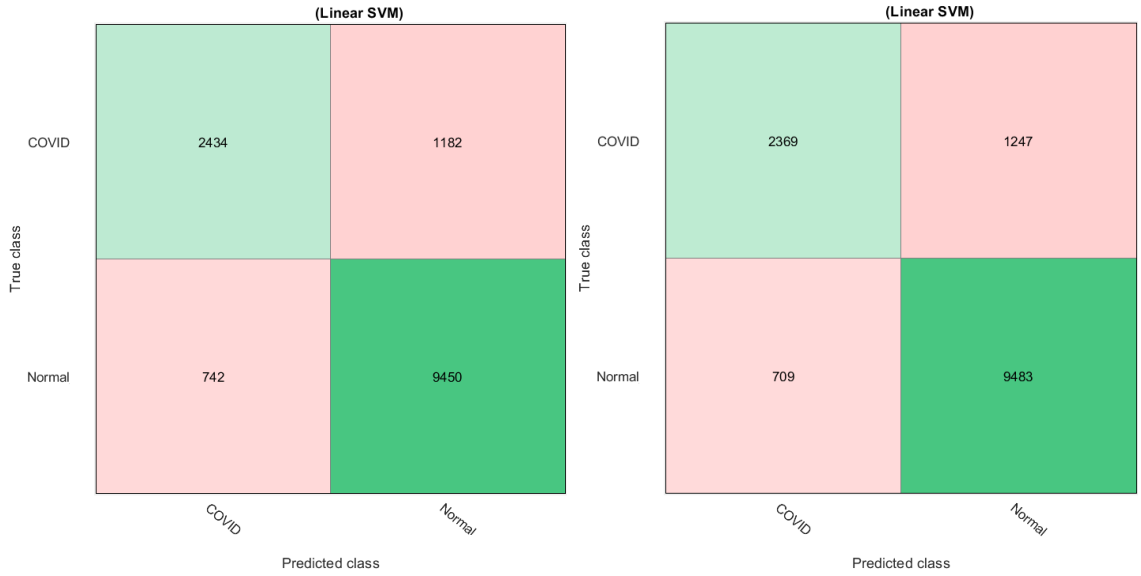
1 ☆ SVM Last change: Linear SVM	Accuracy: 96.2% 200/200 features	
2 ☆ SVM Last change: PCA explaining 95% variance	Accuracy: 96.1% 159/200 features (PCA on)	
3 ☆ Naive Bayes Last change: Disabled PCA	Accuracy: 95.5% 200/200 features	
4 ☆ Naive Bayes Last change: PCA explaining 95% variance	Accuracy: 94.3% 159/200 features (PCA on)	
5 ☆ Linear Discriminant Last change: Linear Discriminant	Accuracy: 96.2% 200/200 features	
6 ☆ Linear Discriminant Last change: PCA explaining 95% variance	Accuracy: **96.3%** 159/200 features (PCA on)	
7 ☆ Tree Last change: Fine Tree	Accuracy: 93.4% 200/200 features	
8 ☆ Tree Last change: PCA explaining 95% variance	Accuracy: 93.5% 159/200 features (PCA on)	
9 ☆ KNN Last change: Fine KNN	Accuracy: 63.8% 159/200 features (PCA on)	
10 ☆ KNN Last change: Disabled PCA	Accuracy: 95.9% 200/200 features	

Wszystkie obrazy

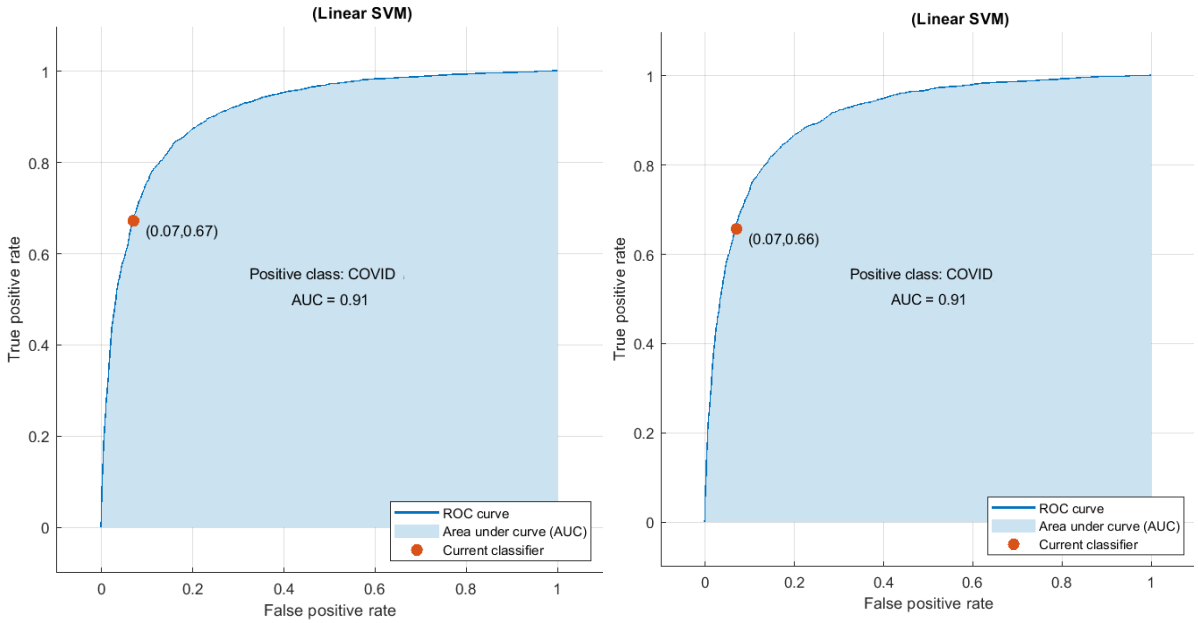
1 ☆ SVM Last change: Linear SVM	Accuracy: **86.1%** 200/200 features	
2 ☆ SVM Last change: PCA explaining 95% variance	Accuracy: 85.8% 164/200 features (PCA on)	
3 ☆ Naive Bayes Last change: Disabled PCA	Accuracy: 77.3% 200/200 features	
4 ☆ Naive Bayes Last change: PCA explaining 95% variance	Accuracy: 76.7% 164/200 features (PCA on)	
5 ☆ Linear Discriminant Last change: Disabled PCA	Accuracy: 85.9% 200/200 features	
6 ☆ Linear Discriminant Last change: PCA explaining 95% variance	Accuracy: 85.5% 164/200 features (PCA on)	
7 ☆ Tree Last change: Disabled PCA	Accuracy: 77.8% 200/200 features	
8 ☆ Tree Last change: PCA explaining 95% variance	Accuracy: 78.4% 164/200 features (PCA on)	
9 ☆ KNN Last change: Fine KNN	Accuracy: 58.4% 164/200 features (PCA on)	
10 ☆ KNN Last change: Disabled PCA	Accuracy: 78.1% 200/200 features	

Rys.20. Zestawienie otrzymanych dla każdego modelu dokładności klasyfikacji w obu próbach

6.1 Klasyfikacja przy użyciu SVM



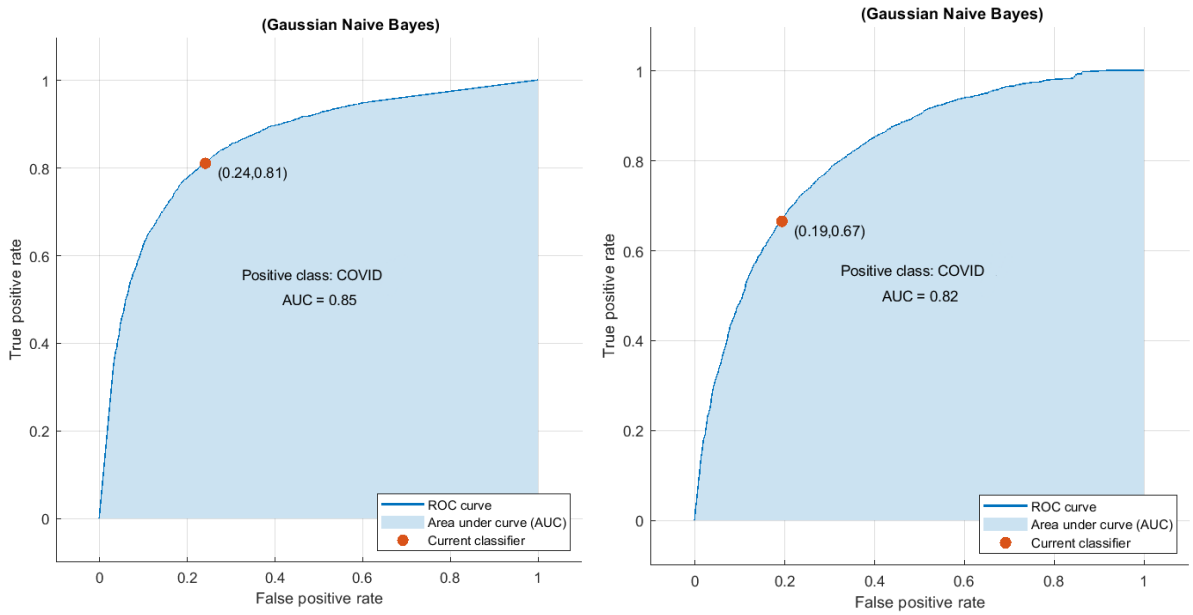
Rys. 21. Macierze pomyłek dla metody SVM z wyłączoną (po lewej) i włączoną (po prawej) opcją PCA



Rys. 22. Krzywe ROC dla metody SVM z wyłączoną (po lewej) i włączoną (po prawej) opcją PCA

6.2 Klasyfikacja przy użyciu klasyfikatora Bayesa

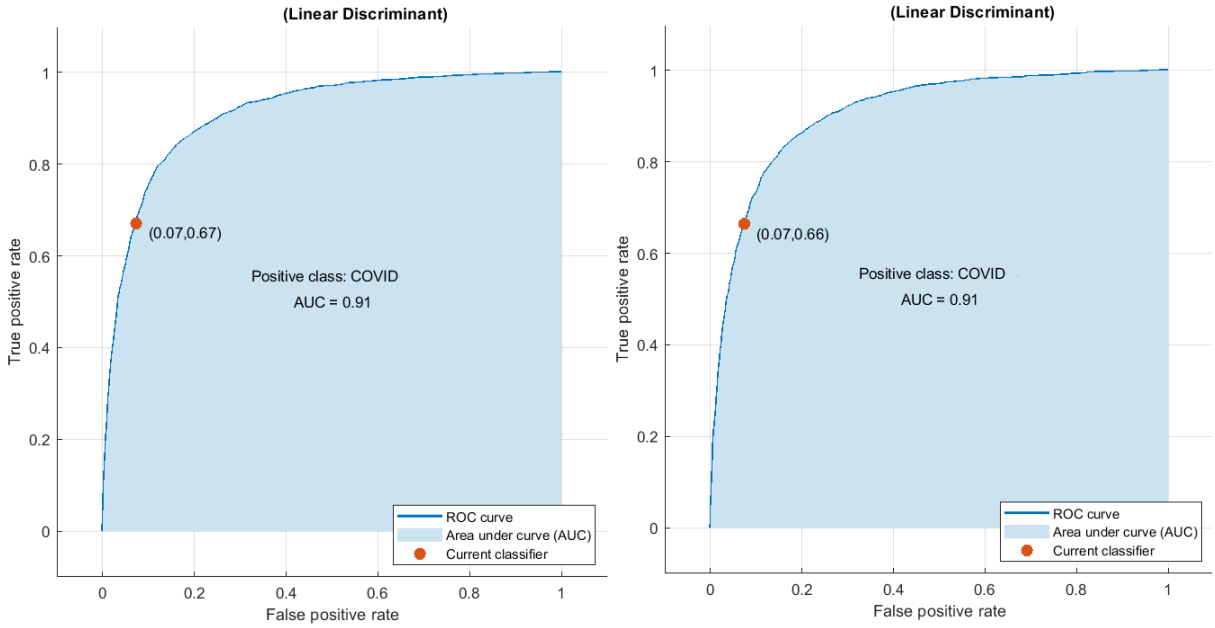
Rys. 23. Macierze pomyłek dla metody klasyfikatora Bayesa z wyłączoną (po lewej) lub włączoną (po prawej) opcją PCA



Rys. 24. Krzywe ROC dla metody klasyfikatora Bayesa z wyłączoną (po lewej) lub włączoną (po prawej) opcją PCA

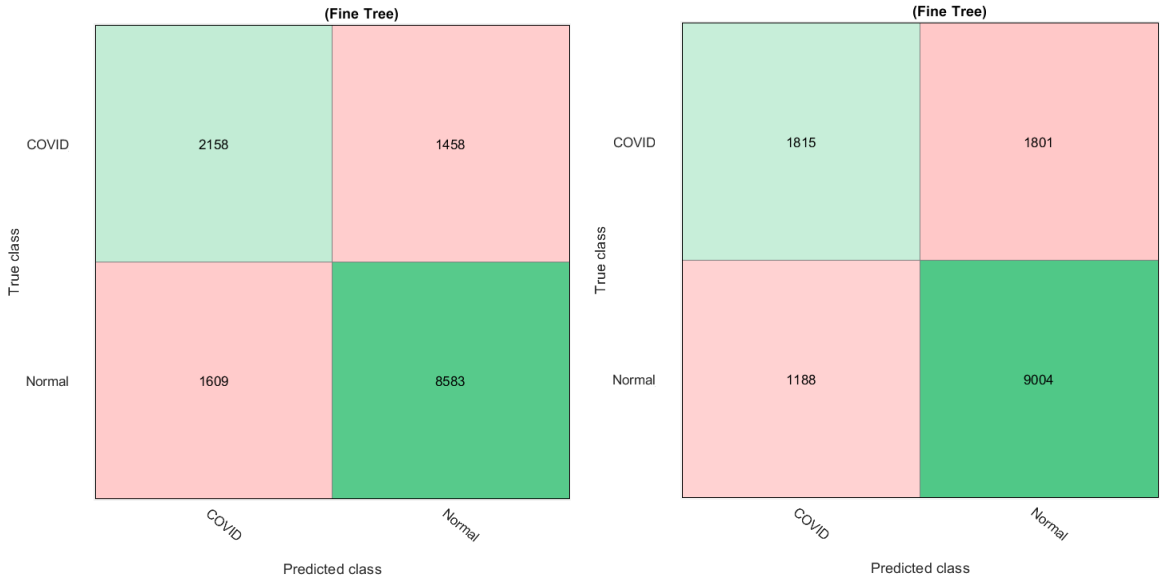
6.3 Klasyfikacja przy użyciu LDA

Rys. 25. Macierze pomyłek dla metody LDA z wyłączoną (po lewej) lub włączoną (po prawej) opcją PCA

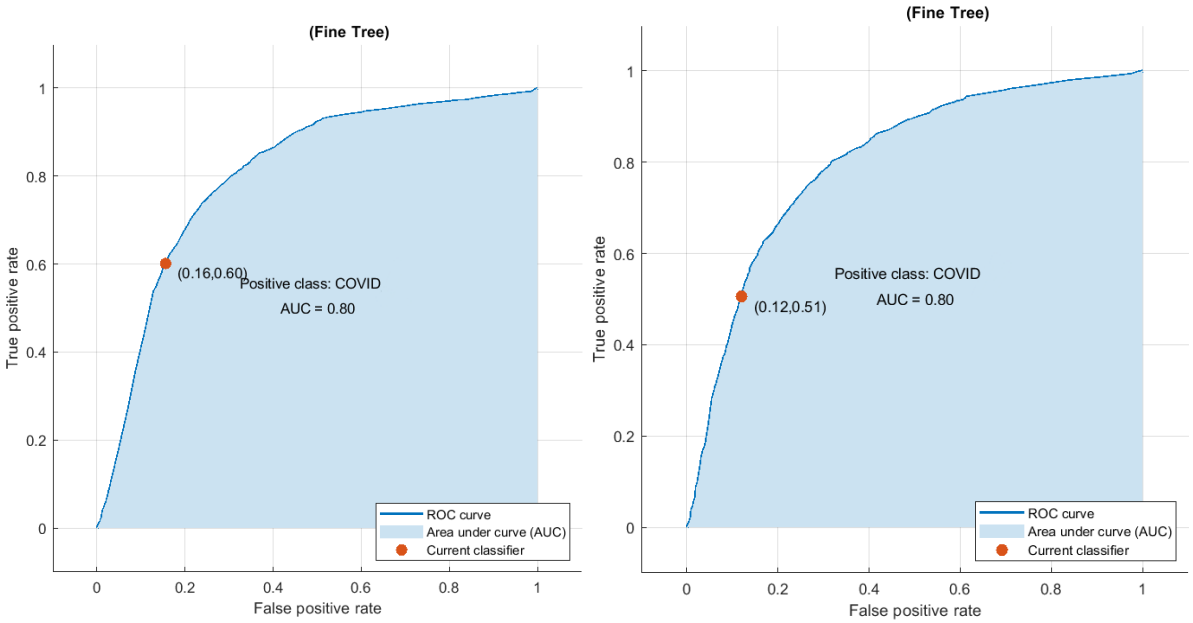


Rys. 26. Krzywe ROC dla metody LDA z wyłączoną (po lewej) lub włączoną (po prawej) opcją PCA

6.4 Klasyfikacja przy użyciu drzewa decyzyjnego

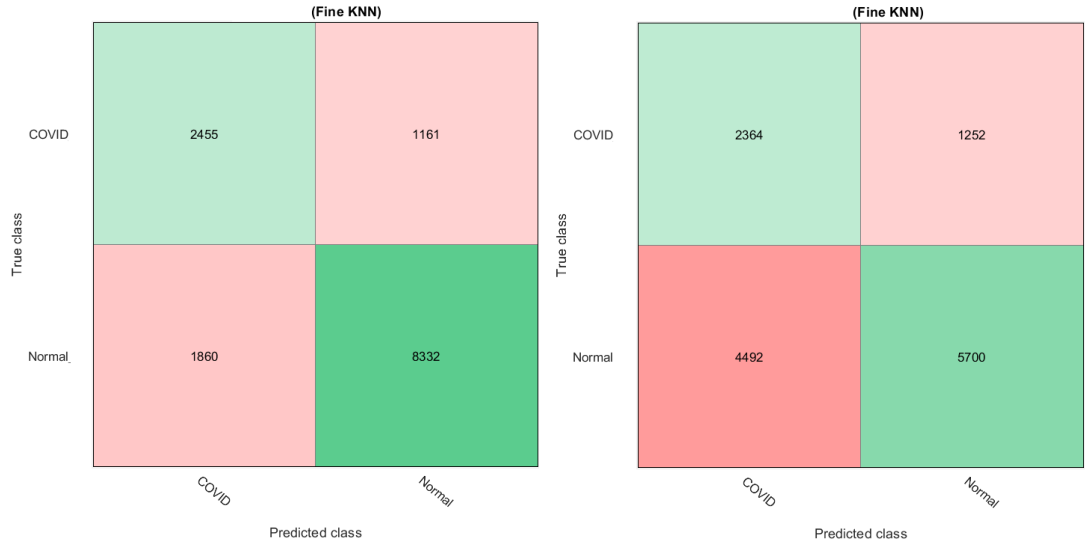


Rys. 27. Macierze pomyłek dla drzewa decyzyjnego z wyłączoną (po lewej) lub włączoną (po prawej) opcją PCA

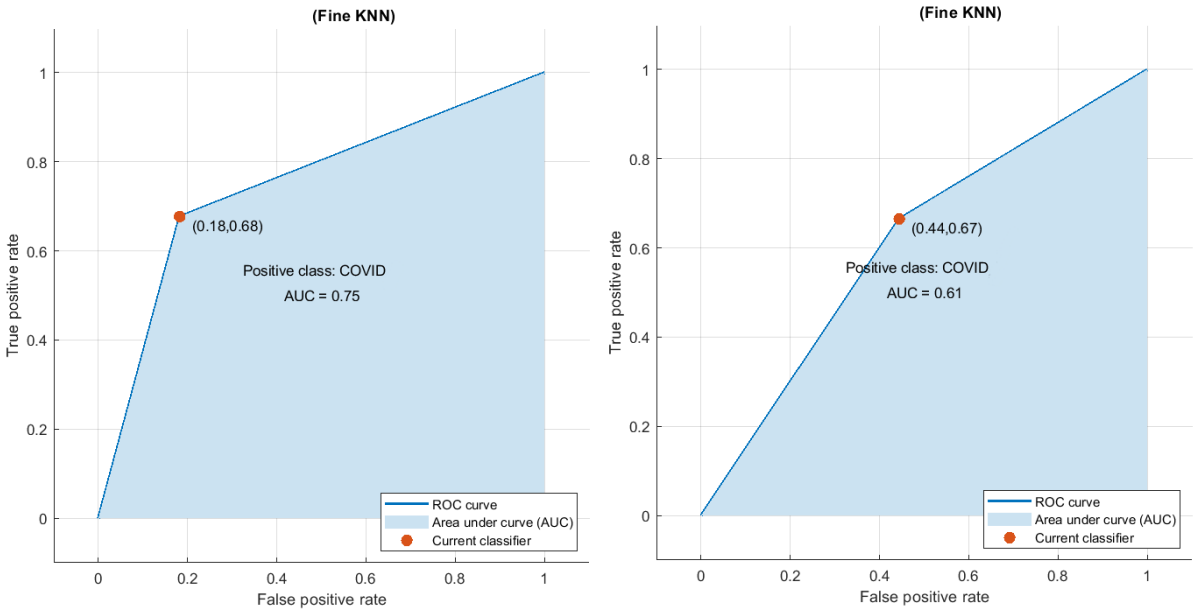


Rys.28. Krzywe ROC dla drzewa decyzyjnego z wyłączoną (po lewej) lub włączoną (po prawej) opcją PCA

6.5 Klasyfikacja przy użyciu klasyfikatora k najbliższych sąsiadów



Rys. 29. Macierze pomyłek dla metody KNN z wyłączoną (po lewej) lub włączoną (po prawej) opcją PCA



Rys. 30. Krzywe ROC dla metody KNN z wyłączoną (po lewej) lub włączoną (po prawej) opcją PCA

6.6 Ewaluacja wyników

Nie można sensownie porównywać rezultatów dla modeli gotowych i samodzielnie opracowanych w punkcie 5., przede wszystkim ze względu na wykorzystanie innego zbioru danych, jak i różne metody przygotowania ich do klasyfikacji oraz modyfikacje metod własnych poprzez poddaniu obrazów filtracji przed klasyfikacją. Możliwe jest jednak porównanie między sobą gotowych modeli – w tym celu obliczono te same parametry co dla metod przedstawionych w punkcie 5., korzystając z tych samych wzorów. Wyniki przedstawiono w poniższej tabeli.

Tabela 2. Wyniki obliczeń dla gotowych modeli

Model		Accuracy	Recall	Precision	F1 Score	False-Positive Rate	AUC
SVM	-	0,86	0,67	0,77	0,72	0,07	0,91
	PCA	0,86	0,66	0,77	0,71	0,07	0,91
Bayes	-	0,77	0,81	0,54	0,65	0,24	0,85
	PCA	0,77	0,66	0,55	0,60	0,20	0,82
LDA	-	0,86	0,67	0,76	0,71	0,07	0,91
	PCA	0,86	0,66	0,75	0,70	0,08	0,91
Drzewo	-	0,78	0,60	0,57	0,58	0,16	0,80
	PCA	0,78	0,50	0,61	0,55	0,12	0,80
KNN	-	0,78	0,68	0,57	0,62	0,18	0,75
	PCA	0,58	0,65	0,34	0,44	0,44	0,61

Sądząc po parametrze AUC, gotowe modele są lepsze niż metody przedstawione w punkcie 5., ale jak już wspomniano, nie można ich wiarygodnie porównać. Wszystkie wartości są bardzo zbliżone dla modeli SVM oraz LDA, a użycie PCA nieznacznie pogarsza ich wyniki. Z kolei dla pozostałych skorzystanie z PCA silniej pogarsza działanie klasyfikatora k najbliższych sąsiadów, natomiast w przypadku klasyfikatora Bayesa bardzo obniża wartość *recall*, co jest bardzo niebezpiecznym zjawiskiem, ponieważ świadczy to o obniżeniu zdolności prawidłowego wykrywania przypadków pozytywnych. Ogółem zastosowanie PCA w celu redukcji liczby cech nie jest korzystne.

Patrząc na kształt krzywej ROC i wartości AUC, najkorzystniej wypadają klasyfikatory SVM oraz LDA, zaś najbliższa losowej klasyfikacji jest metoda KNN z wykorzystaniem PCA. Dowolny z nich będzie dobrym wyborem do klasyfikacji badanego zbioru danych; decyzję można podjąć na przykład w oparciu o czas trwania obliczeń, znacznie dłuższy dla metody SVM.

7. Podsumowanie i wnioski

COVID-19 jest chorobą stanowiącą wciąż aktualny problem i szeroko badaną. Szczególnie duże zainteresowanie budzi perspektywa diagnostyki osób zarażonych za pomocą Uczenia Maszynowego oraz wykorzystywania zdjęć RTG z uwagi na problemy z dostępnością i czasochłonność przeprowadzania testów laboratoryjnych.

Aktualnie trwają poszukiwania najskuteczniejszej metody klasyfikacji zmian chorobowych na zdjęciach RTG. W badaniach wykorzystuje się obecnie wiele metod takich jak sieci konwolucyjne czy metoda lasu losowego.

Zastosowane przez nasz zespół metody wykazały się dużym potencjałem w wykrywaniu pożądanych elementów na zdjęciach oraz klasyfikowaniu chorych

Uzyskane przez nasz zespół wyniki wykazały się wysoką jakością przede wszystkim na LDA i KNN z filtrami Sobela.

Metody opierające się na PLC, KNN, SVM i LDA, najprawdopodobniej mają szansę znaleźć szerokie zastosowanie w przyszłości diagnostyki.

Dalsze badania wybranych przez nas metod mogłyby doprowadzić do zmniejszenia ilości zapotrzebowania na testy i ograniczenia czasu koniecznego na laboratoryjną ewaluację wyników.

8. Bibliografia

[1] https://www.gov.pl/web/zdrowie/co-musisz-wiedziec-o-koronawirusie

[2] https://www.medonet.pl/choroby-od-a-do-z/choroby-ukladu-oddechowego-i-alergie,covid-19---przyczyny--objawy--przebieg--leczenie,artykul,61168587.html

[3] https://www.who.int/emergencies/diseases/novel-coronavirus-2019

[4] https://www.who.int/docs/default-source/coronaviruse/risk-comms-updates/update54_clinical_long_term_effects.pdf?sfvrsn=3e63eee5_8&fbclid=IwAR3BYd2PjeUw_yZeyqYlDqO6BNxL5DZwD-iQJmf7Ss2mvQErL-UiF8sbYGo

[5] https://badania.znanylekarz.pl/blog/badania-obrazowe-w-kierunku-powiklan-po-zakazeniu-koronawirusem-covid-19/

[6] Hassantabar, S., Ahmadi, M., & Sharifi, A. (2020). Diagnosis and detection of infected tissue of COVID-19 patients based on lung x-ray image using convolutional neural network approaches. Chaos, Solitons & Fractals, 140, 110170. doi:10.1016/j.chaos.2020.110170

[7] Mahdy i in., „Automatic X-ray COVID-19 Lung Image Classification System based on Multi-Level Thresholding and Support Vector Machine”.

[8] Ozturk, T., Talo, M., Yildirim, E. A., Baloglu, U. B., Yildirim, O., & Rajendra Acharya, U. (2020). Automated detection of COVID-19 cases using deep neural networks with X-ray images. Computers in Biology and Medicine’’

[9] Xanthopoulos P., Pardalos P. M., Trafalis T. B., „Linear Discriminant Analysis’’

[10] Balakrishnama S., Ganapathiraju A., „Linear discriminant analysis - a brief tutorial’’

[11]https://www.statsoft.pl/textbook/stathome_stat.htmlhttps%3A%2F%2Fwww.statsoft.pl%2Ftextbook%2Fstnaiveb.html

[12] Suthaharan S., „Support Vector Machine’’

[13] Priyam A., Abhijeet, Gupta R., Rathee A., ,,Comparative Analysis of Decision Tree Classification Algorithms''

[14] Stoliński, S. , Grabowski, S., ,,Eksperymentalne porównanie filtrów medianowych do usuwania szumów impulsowych z obrazów barwnych''

[15] Graur I., Vengertsev D., Stobert I., ,,New method of detection and classification of yield-impacting EUV mask defects''

[16] Anand A., Tripathy S. S., Kumar R. S., ,,An improved edge detection using morphological Laplacian of Gaussian operator''

[17] Ringer M., ,,What is principal component analysis?''

[18] https://ailearnerhub.com/2020/05/10/what-is-the-confusion-matrix/ [dostęp: 07.06.2021]

[19] https://www.kaggle.com/tawsifurrahman/covid19-radiography-database

[20] https://www.flickr.com/photos/niaid/49534865371/

[21] https://www.gov.pl/web/koronawirus/wykaz-zarazen-koronawirusem-sars-cov-2