

Politechnika Warszawska

W Y D Z I A Ł M A T E M A T Y K I  
I N A U K I N F O R M A C Y J N Y C H



# Praca dyplomowa inżynierska

na kierunku Informatyka

Implementacja aplikacji wspomagającej diagnostykę nowotworu  
prostaty, wykorzystującej standard DICOM do integracji z realnym  
systemem szpitalnym

**Rafał Buzun**

Numer albumu 268744

**Łukasz Maciej Berwid**

Numer albumu 268740

**Jakub Dominik Karolak**

Numer albumu 268758

promotor

mgr inż. Piotr Sobecki

WARSZAWA 2018

.....

podpis promotora

.....

podpis autora

## **Streszczenie**

Implementacja aplikacji wspomagającej diagnostykę nowotworu prostaty, wykorzystującej standard DICOM do integracji z realnym systemem szpitalnym

Celem pracy dyplomowej jest stworzenie narzędzia wspomagającego diagnostykę nowotworu prostaty oraz jego integracja ze szpitalną bazą danych. Proponowane narzędzie będzie umożliwiało obliczanie objętości prostaty bazując na wysegmentowanym fragmencie obrazu pochodzącego z obrazowania metodą multiparametrycznego Rezonansu Magnetycznego. Dodatkowo stworzona zostanie baza badań zawierająca informacje o referencyjnych wartościach objętości gruczołu krokowego pochodzących z estymacji bazującej na obrazowaniu Ultrasonografem. Narzędzie zostanie przygotowane do wdrożenia w szpitalnym systemie informatycznym.

**Słowa kluczowe:** przetwarzanie obrazu, analiza danych, bioinformatyka



## **Abstract**

The aim of the diploma thesis is to create a tool that supports the diagnostics of prostate cancer and its integration with the hospital database. The proposed tool will be able to calculate prostate volume based on the segmented image from multiparameter Magnetic Resonance imaging. In addition, a database containing information on reference prostate volume values from the estimation based on ultrasonography imaging will be created. The tool will be prepared for implementation in the hospital IT system.

**Keywords:** image processing, data analysis, bioinformatics



Warszawa, dnia .....

### Oświadczenie

Oświadczam, że moją część pracy inżynierskiej (zgodnie z podziałem zadań opisanym w pkt. ...) pod tytułem; „Implementacja aplikacji wspomagającej diagnostykę nowotworu prostaty, wykorzystującej standard DICOM do integracji z realnym systemem szpitalnym”, której promotorem jest mgr. inż. Piotr Sobecki wykonałem samodzielnie, co poświadczam własnoręcznym podpisem.

.....





## Spis treści

<b>1. Wstęp</b>	<b>11</b>
1.1. Diagnostyka raka prostaty	11
1.2. Architektura szpitala	13
1.3. Metody wspomagania pracy specjalisty	15
1.4. Słownik pojęć	15
<b>2. Dane szpitalne</b>	<b>17</b>
2.1. Proces pozyskania danych	17
2.2. Dane szpitalne	18
2.3. Wstępna analiza	19
2.4. Statystyki	20
2.5. Ciekawe przypadki	24
<b>3. Specyfikacja</b>	<b>25</b>
3.1. Opis biznesowy	25
3.2. Wymagania funkcjonalne	26
3.3. Diagram przypadków użycia	26
3.4. Wymagania нефункционалне	27
3.5. Środowisko sprzętowe i programowe	28
<b>4. Opis aplikacji</b>	<b>33</b>
4.1. Architektura systemu	33
4.2. Komunikacja między modułami	34
<b>5. Opis modułów</b>	<b>35</b>
5.1. Moduł segmentacji	35
5.1.1. Opis danych	37
5.1.2. Opis rozwiązania	38
5.1.3. Wyniki	41
5.1.4. Alternatywne podejścia	42
5.2. Moduł obliczania objętości	42
5.2.1. Liczenie objętości	44
5.2.2. Liczenie pola powierzchni	45
5.2.3. Analiza wyników	45
5.2.4. Alternatywne podejścia	48
5.3. Moduł backedowy	49
5.3.1. Diagram zależności	50
5.3.2. Baza danych	50

5.4.	Moduł prezentacji . . . . .	52
5.4.1.	Widok startowy . . . . .	52
5.4.2.	Widok pacjenta . . . . .	53
5.4.3.	Edycja obrazu . . . . .	55
5.4.4.	Integracja z systemem PACS . . . . .	58
<b>6.</b>	<b>Podsumowanie rezultatów . . . . .</b>	<b>59</b>
6.1.	Integracja z systemem informatycznym . . . . .	59
6.2.	Trudności . . . . .	59
6.3.	Wnioski . . . . .	60
6.4.	Załącznik 1 - Opis API . . . . .	66

## 1. Wstęp

Rak prostaty to drugi po raku płuc najczęściej diagnozowany nowotwór u mężczyzn. W ciągu ostatnich lat w Polsce wzrasta zachorowalność na raka stercza, a także – choć w mniejszym stopniu – umieralność. Krajowy rejestr nowotworów zanotowało, że w naszym kraju w 2015 r. zostało wykrytych 14 tys. nowych przypadków raka prostaty, a 5 tys. chorych z wcześniej wykrytą chorobą zmarło. W 2012 roku, 7 lata temu, z powodu tego nowotworu zachorowało 11 tys. mężczyzn, a 4,1 tys. zmarło. Odwrotną tendencję obserwuje się w zachodniej Europie i Stanach Zjednoczonych, gdzie odnotowuje się spadek zachorowalności i umieralności na ten nowotwór. Wszystko dzięki zastosowaniu najnowszych metod leczenia i wczesnej diagnostyce. [1]

Dlatego też, w niniejszej pracy podjęto się zbadania możliwości automatyzacji wykrywania prostaty na zdjęciach rezonansu magnetycznego, poprzez automatyczną segmentację prostaty i obliczanie objętości. Algorytmy obliczania wielkości gruczołu opisane w tej pracy mogą być wskazówką dla lekarza podczas stawiania diagnozy.

System powstały jako produkt poniższej pracy dzięki zastosowaniu modularnej architektury może być dostosowany do zaawansowanych systemów szpitalnych oraz rozszerzony o alternatywne algorytmy, przykładowo oparte o głębokie uczenie maszynowe.

Za implementację oraz wdrożenie algorytmu segmentacji prostaty bazującej na danych konkursowych przy użyciu konwolucyjnych sieci neuronowych oraz moduł prezentacji aplikacji odpowiada Rafał Buzun.

Za stworzenie, opisanie zbioru testowego, przeprowadzenie analizy biznesowej oraz statystycznej wyników badań rezonansu magnetycznego oraz rozpoznanie architektury szpitalnego systemu informatycznego i integrację z systemem PACS odpowiada Jakub Karolak.

Za zaprojektowanie architektury oraz implementację serwera, w tym obsługę bazy danych i przetwarzanie plików DICOM oraz implementacja wybranych metod obliczania objętości prostaty odpowiada Łukasz Berwid

### 1.1. Diagnostyka raka prostaty

Objawami świadczącymi o problemach z gruczołem prostaty jest ból lub pieczenie przy oddawaniu moczu lub nieregularne oddawanie moczu. Odpowiada to również objawom łagodnym rozrostem gruczołu krokowego. Mężczyźni nie zwykli bagatelizować takich problemów, więc rak prostaty w 75% jest wykrywany w bezobjawowym stadium ograniczonym do narządu, dzięki czemu łatwo jest go wyleczyć.

Pierwszym z badań, które jest wykonywane w przypadku podejrzenia występowania nowotworu jest badanie poziomu PSA we krwi. PSA jest antygenem sterczowym, który daje nam podstawowe informacje na temat problemów z gruczołem pacjenta.

Zazwyczaj następnym badaniem jest badanie palpacyjne per rectum. W przypadku tego badania, lekarz szuka nieprawidłowości w kształcie i twardości stercza. Badanie obarczone jest dużą niedokładnością, ponieważ ok. 30% nowotworów znajduje się w rejonie prostaty nieosiągalnym przez badanie palpacyjne.

W przypadku podejrzeń nowotworu wykonuje się kolejne badania. Jest to biopsja stercza oraz badanie USG. Biopsja o wyniku pozytywnym jednoznacznie wskazuje na obecność posiadania nowotworu. Badanie USG cechuje się mniejszą dokładnością, jest robione w celu określenia wytycznych do badania MR, czyli rezonansu magnetycznego.

Badanie MR charakteryzuje się większą dokładnością, radiolog za pomocą oprogramowania jest w stanie znaleźć nawet najmniejsze zmiany w strukturze gruczołu. Często po wynikach rezonansu zleca się biopsję celowaną, która ma sięgnąć lokalizacji podejrzanych zmian określonych podczas badania za pomocą rezonansu magnetycznego. Pełny zestaw badań, czyli, badanie PSA, badanie palpacyjne, USG, MR oraz biopsja jest w stanie jednoznacznie postawić diagnozę.

W chwili obecnej, w szpitalu MSWiA w Warszawie, do opisu prawdopodobieństwa istnienia nowotworu w danym miejscu prostaty jest używana skala PI-RADS v2.

## PIRADS

Ocena PI-RADS v2 opiera się na wykorzystaniu pięciostopniowej skali określającej prawdopodobieństwo zmian nowotworowych, opracowanej w oparciu o zależność, jaka istnieje pomiędzy badanymi obszarami a prawdopodobieństwem, że uwidocznione zmiany mają charakter nowotworowy. Zaszeregowanie do określonej kategorii według PI-RADS v2 powinno opierać się na wynikach badań rezonansu magnetycznego stercza i nie powinno uwzględniać innych czynników, takich jak stężenie PSA, wynik badania per rectum, dane uzyskane z wywiadu lub przebytego leczenia. Przy wynikach rzędu 4-5 w skali PIRADS występuje wysokie prawdopodobieństwo posiadania istotnej klinicznie formy nowotworu. W takich przypadkach, jeśli jeszcze nie została wystawiona diagnoza, należy bezzwłocznie wykonać biopsję.

## PSA

PSA czyli swoisty antygen sterczowy. Jest to związek wytwarzany w gruczole krokowym (prostatie). Obecność tego enzymu we krwi jest wykorzystywana w diagnostyce, monitorowaniu i leczeniu raka prostaty od 1986 roku. Fakt posiadania podwyższonego poziomu PSA we krwi nie jest zawsze jednoznaczny z chorobą. Badanie to daje około 25% fałszywie dodatnich wyników. Niestety fakt nieposiadania podwyższonego PSA może być równie mylący. W 20% przypadków to badanie daje fałszywie ujemny wynik. Standardy Europejskiego Towarzystwa Urologicznego zalecają coroczne oznaczanie PSA u mężczyzn po 50 roku życia. Kompletnie zdrowy mężczyzna powinien mieć PSA na poziomie  $<1,5$  ng/ml. Przy tak niskiej wartości zaleca się kolejny pomiar w odstępie 5 lat.

## PSAD

W celu poprawienia wykrywalności raka stercza opracowano współczynnik gęstości PSA, tzw. PSAD, który oblicza się dzieląc stężenie PSA w ng/ml przez objętość stercza. Ten współczynnik pomaga w ustaleniu, czy mamy do czynienia z rakiem stercza, czy też łagodnym rozrostem

## 1.2. ARCHITEKTURA SZPITALA

prostaty, podczas którego także wydziela się antygen PSA. Badania potwierdzają, że istnieje korelacja pomiędzy wartością współczynnika PSAD a występowaniem nowotworu.

### Rezonans magnetyczny

Rezonans magnetyczny to technika obrazowania medycznego stosowana w radiologii do tworzenia obrazów anatomii i procesów fizjologicznych organizmu. Skanery wykorzystują silne pola magnetyczne, gradienty pola magnetycznego i fale radiowe do generowania obrazów narządów w ciele. Rezonans nie obejmuje promieni rentgenowskich ani promieniowania jonizującego, co odróżnia go od tomografii komputerowej lub skanów PET.

Pełna nazwa badania brzmi *Obrazowanie metodą rezonansu magnetycznego*. Warto jest zaznaczyć fakt obrazowania, ponieważ istnieje jeszcze drugie zastosowanie tego aparatu, mianowicie spektroskopia, wykorzystywana w chemii i fizyce molekularnej.

### Badanie objętości gruczołu

W szpitalach używane są trzy metody pomiaru objętości stercza. Pierwszym jest subiektywne badanie per rectum w którym lekarz ręcznie sprawdza objętość prostaty. Dwie pozostałe opierają się na tych samych obliczeniach, różnią się jedynie sposobem pozyskiwania danych. Pierwszym z nich jest badanie rezonansu magnetycznego, a drugim TRUS, czyli przezodbytnicze badanie USG. Lekarz ogląda gruczoł w różnych płaszczyznach, zaznaczając punkty skrajne. Po zaznaczeniu najbardziej skrajnych punktów wyliczane są trzy wymiary prostaty: szerokość, wysokość i długość. Aby uzyskać objętość gruczołu krokowego stosowany jest wzór:

$$V = h * w * l * 0.54$$

gdzie

h - wysokość

w - szerokość

l - długość

Stała 0.54 jest próbą przybliżenia kształtu prostaty figurą geometryczną. Gruczoł przypomina elipsoidę, więc zastosowany został wzór na objętość elipsy

$$V = \pi/6 * h * w * l$$

gdzie

h,w,l - to odpowiednio wysokość, długość, szerokość elipsoidy.

## 1.2. Architektura szpitala

Architektura szpitalna składa się z wielu komponentów w poniższej pracy wymieniamy komponenty, które będą niezbędne do integracji z opisywanym system. Podstawowym elementem w sieci IT szpitala jest system informacji szpitalnej (HIS), który koncentruje się głównie na potrzebach administracyjnych szpitali. W wielu wdrożeniach system HIS jest kompleksowym, zintegrowanym systemem informatycznym zaprojektowanym do

zarządzania wszystkimi aspektami funkcjonowania szpitala, takimi jak zarządzają danymi demograficznymi pacjentów, ubezpieczeniem informację, planowaniem leczenia klinicznego, działaniem laboratorium oraz zarządzaniem historią pacjenta i badaniami.

### **System archiwizacji obrazu**

PACS czyli system archiwizacji i komunikacji obrazu to technologia obrazowania medycznego stosowana głównie w szpitalach i jednostkach medycznych do bezpiecznego przechowywania oraz cyfrowej transmisji obrazów elektronicznych lub raportów o znaczeniu klinicznym. Korzystanie z PACS eliminuje potrzebę ręcznego przechowywania, wyszukiwania i wysyłania poufnych informacji, filmów i raportów. Daje to możliwość aby dokumentacja medyczna i obrazy były bezpiecznie przechowywane w zewnętrznych serwerach i dostępne z każdego miejsca na świecie przy użyciu oprogramowania PACS, stacji roboczych i urządzeń mobilnych. Ze względu na panujące prawo dotyczące ochrony danych osobowych i tego, że dane medyczne pacjentów są danymi wrażliwymi takie rozwiązania rzadko są stosowane w praktyce. Dane pacjentów są przechowywane na serwerach szpitalnych - tego wymaga prawo.

PACS ma cztery główne komponenty:

1. Sprzętowe urządzenia obrazujące.
2. Bezpieczna sieć do dystrybucji i wymiany obrazów pacjentów.
3. stacja robocza lub urządzenie mobilne do przeglądania, przetwarzania i interpretowania obrazów.
4. Elektroniczne archiwa do przechowywania i wyszukiwania obrazów oraz powiązanej dokumentacji i raportów.

### **Radiologiczny System Informatyczny**

System informacji radiologicznej (RIS) to sieciowy system oprogramowania do zarządzania obrazami medycznymi i powiązanymi danymi. System jest szczególnie przydatny do śledzenia zleceń obrazowania radiologicznego i informacji rozliczeniowych i jest często używany w połączeniu z PACS do zarządzania archiwami obrazów i rejestrowaniem.

Podstawowe funkcje systemu informacji radiologicznej:

1. Zarządzanie pacjentem - RIS może śledzić cały przebieg pracy pacjenta w dziale radiologii. Specjaliści mogą dodawać zdjęcia i raporty do systemów elektronicznego rejestru zdrowia, gdzie mogą być pozyskane i oglądane przez upoważniony personel radiologii.
2. Planowanie - System umożliwia personelowi umawianie się na wizyty zarówno w szpitalach, jak i ambulatoriach.
3. Monitorowanie pacjenta - Korzystając z systemu RIS, dostawcy mogą śledzić historię całej radiologii pacjenta od przyjęcia do wypisu i koordynować historię z przeszłymi, obecnymi i przyszłymi spotkaniami.
4. Raportowanie wyników - System może generować raporty statystyczne dla pojedynczego pacjenta, grupy pacjentów lub poszczególnych procedur.

#### **Format DICOM**

DICOM jest standardowym protokołem do zarządzania, przesyłania obrazów medycznych oraz powiązanych danych. Jest stosowany w większości szpitali. DICOM to międzynarodowy standard komunikacji, zarządzania obrazami oraz danymi medycznymi. Celem jest zapewnienie przechowywania, udostępniania, współdziałania systemów używanych do produkcji, wyświetlania, wysyłania, odpytywania, przetwarzania, pobierania i drukowania obrazów medycznych, a także do zarządzania powiązanymi przepływami pracy.

DICOM jest używany na całym świecie do przechowywania, wymiany i przesyłania obrazów medycznych, umożliwiając integrację medycznych urządzeń obrazujących wielu producentów. Dane pacjenta i powiązane obrazy są wymieniane, przechowywane w znormalizowanym formacie. Standaryzacja ułatwia dzielenie i interpretację informacji między różnymi urządzeniami do przetwarzania obrazu.

### **1.3. Metody wspomagania pracy specjalisty**

W chwili obecnej praca radiologa jest ściśle powiązana z pracą przy użyciu dostarczanego oprogramowania zintegrowanego z infrastrukturą szpitalną. Na rynku istnieje wiele różnych narzędzi, które mogą wspomóc pracę lekarza jak na przykład oprogramowanie Future Processing [10], które zapewnia możliwość automatycznej segmentacji obrazów narządów takich jak mózg oraz płuca, jednak obecnie nie jest dostępny moduł wykonujący to dla gruczołu prostaty. Sprzęt dostarczany przez dostawców oprogramowania w Polsce nie zapewnia takiej funkcjonalności.

W szpitalu MSWiA z którym współpracowaliśmy w trakcie pisania tej pracy podstawowym narzędziem było oprogramowanie dostarczane przez firmę Philips [9]. Radiolog odczytuje, opisuje i analizuje obrazy na stacji wyposażonej w monitor diagnostyczny, następnie przepisuje je na komputer osobisty. Jednakże wiele badań dostarczanych jest przez pacjenta w formie papierowej. Problemy wynikające z używania jednocześnie dokumentacji papierowej oraz elektronicznej jednocześnie są opisane w dalszej części pracy.

Pomimo tego, że na rynku istnieją rozwiązania pozwalające na automatyzację pracy lekarza poprzez na przykład automatyczne wykrywanie i oznaczanie zmian w organach. Narzędziach do przeglądania obrazów DICOM wykorzystywanych obecnie w szpitalach przez radiologów brakuje modułów automatyzujących pracę. Są to, w większości, przeglądarki zdjęć, które radiolog musi później sam ręcznie analizować.

Proponujemy stworzenie systemu zapewniającego automatyczną segmentację oraz liczenie objętości prostaty na podstawie obrazowania stercza metodą rezonansu magnetycznego. Dzięki temu będziemy mogli znacząco skrócić czas jaki lekarza poświęca na analizę obrazów oraz zapewnić szybką informację mającą wpływ na diagnozę.

### **1.4. Słownik pojęć**

Pojęcia których będziemy używać w poniższej pracy:

1. DICOM - (Digital Imaging and Communications in Medicine) Międzynarodowy standard służący do przesyłania, przechowywania, pobierania, przetwarzania oraz wyświetlania

obrazów medycznych. Darmowy standard umożliwiający unifikację zdjęć medycznych pochodzących z maszyn wielu producentów.

2. Segmentacja – Proces podziału obrazu na części określane jako obszary (regiony), które są jednorodne pod względem pewnych wybranych własności.
3. Slice - Warstwa obrazu DICOM zawierająca jeden obraz
4. Maska - Obraz przedstawiający wysegmentowaną prostatę
5. Komponent - Usługa sieciowa lub element jednego z serwisów spełniający jedno określone zadanie
6. Endpoint - Adres pozwalający na dostęp do API aplikacji z zewnątrz
7. API - Interfejs pozwalający na komunikowanie się aplikacji między sobą
8. Backend - Część serwerowa aplikacji
9. Frontend - Warstwa prezentacji aplikacji, w przypadku naszej pracy to strona internetowa
10. HTTP - Internetowy protokół komunikacyjny
11. Web serwis - Usługa sieciowa, do której można się odwołać za pomocą wywołań funkcji określonych przez jej API za pomocą protokołu komunikacyjnego
12. JSON - Format wymiany danych komputerowych
13. Base64 - Rodzaj kodowania
14. RIS – (Radiological Information System) System zarządzania pacjentami
15. PACS - (Picture Archiving And Communication System) System archiwizacji zdjęć medycznych
16. Użytkownik – osoba mająca dostęp do aplikacji oraz bazy danych
17. Baza danych – baza z badaniami pacjentów, zawiera obraz w DICOM oraz dodatkowe informacje z badania
18. Anonimizacja – proces usuwający dane wrażliwe pacjenta



## 2. Dane szpitalne

Analiza problemu została rozpoczęta przez spotkanie z doświadczonym radiologiem posiadającym ponad dwunastoletnie doświadczenie w zawodzie. Radiolog przedstawił problemy z którymi zmagają się w codziennej pracy i które uważa, że mogłyby zostać rozwiązane za pomocą dodatkowego oprogramowania. Ze względu na prowadzone prace analityczne w trakcie wytwarzania oprogramowania przyjęta została zwinna metoda wytwarzania oprogramowania. Po finalnym ustaleniu wymagań, przeprowadzona została analiza szpitalnej architektury informatycznej szpitala MSWiA w Warszawie oraz pozyskaliśmy dane dotyczące pacjentów.

Pierwszym krokiem było zgromadzenia danych pacjentów. Następnie poczynione zostały kroki w kierunku integracji docelowego programu wspomagającego pracę radiologa ze szpitalną bazą danych HIS. Z powodu umowy zawartej między szpitalem a dostawcą oprogramowania nie byliśmy w stanie zintegrować się z systemem informatycznym w standardowy sposób, jakim jest utworzenie bezpiecznego kanału komunikacji poprzez na przykład VPN między naszym serwisem backednowym a szpitalną infrastrukturą, ponieważ umowa zawiera zapis, iż wszelkie zmiany w kodzie klienta bazy danych mogą być wprowadzane jedynie przez pracowników firmy dostarczającej oprogramowanie.

Kolejnym kierunkiem jaki został przez nas obrany było wyszukiwanie przeglądarki zdjęć DICOM. Najistotniejszą kwestią w wyborze było czy narzędzie jest w łatwy sposób rozszerzalne. Po znalezieniu odpowiedniego rozwiązania nastąpiła próba instalacji tej przeglądarki na szpitalnym komputerze. Nie byliśmy w stanie zainstalować oprogramowania z powodu prawnych regulacji dotyczących bezpieczeństwa danych pacjentów znajdujących się na szpitalnych serwerach.

Finalnie zdecydowaliśmy na pozyskanie kopii zanonimizowanych danych pacjentów na dysk twardy i przenoszenie ich na komputer, który nie jest podłączony do sieci szpitalnej.

### 2.1. Proces pozyskania danych

Kompleksowe dane pacjentów w szpitalu MSWiA są przechowywane w formie papierowej. Każdy pacjent ma indywidualną dokumentację, w której znajdują się wyniki wszystkich badań wykonanych poza szpitalem MSWiA, wywiady z wizyt u lekarzy specjalistów oraz wyniki badań krwi zrobionych w szpitalu. W szpitalnej bazie danych przechowywane są wywiady środowiskowe, wyniki oraz opisy badań MR.

Nie została wyrażona zgoda na instalację zewnętrznego oprogramowania na komputerach podłączonych do sieci szpitalnej. Jest to spowodowane regulacjami prawnymi dotyczącymi ochrony wrażliwych danych pacjentów. Otwarte oprogramowanie używane przez nas mogłoby naruszyć regulacje bezpieczeństwa szpitala. Po podjęciu decyzji o przenoszeniu danych nośnikiem

fizycznym należało się skupić na module anonimizującym. Szpitalna baza danych posiada taką funkcję, więc dane pacjentów kopiowane na dysk twardy są już anonimowe. Został użyty dysk twardy, ponieważ plik zawierający kompleksowe badanie jednego pacjenta zajmuje średnio ponad Gigabajt danych.

Na komputerze niepodłączonym do sieci szpitalnej doktor może dokonać analizy i predykcji za pomocą naszego oprogramowania. Następnie musi sprawdzić identyfikator pacjenta w arkuszu kalkulacyjnym i odnieść się z wynikami wygenerowanymi przez nas do prawdziwego przypadku. Rozwiązanie takie jest bezpieczne, co jest kluczowe w kontekście działania szpitala jednak jest pracochłonne.

Procedura pozyskiwania danych polegała na otworzeniu karty pacjenta leczonego przez współpracującego z nami lekarza, znalezieniu jego wywiadu środowiskowego oraz opisu badania w bazie danych RIS. Następnie należało ręcznie wypełnić wszystkie pola w arkuszu kalkulacyjnym, które dotyczyły badanego pacjenta. Dane z wywiadu środowiskowego zostały zakodowane, w celu przyspieszenia wypełniania arkusza oraz zachowania ogólnej estetyki. Legendę opisu tych danych przedstawimy w dalszej części pracy.

Dokumentacja medyczna pacjentów zajmowała od 1 do 30 stron w postaci wyników opisowych. Zadaniem było wyznaczenie wartości wszystkich klinicznych deskryptorów wspólnych dla informacji znajdujących się w dokumentacji medycznej.

Po uzupełnieniu arkusza kalkulacyjnego należało jeszcze wykonać niezbędne obliczenia objętości oraz PSAD.

W początkowych fazach projektu rozważane było użycie systemu OCR na zeskanowanych kartach pacjentów. System OCR (z ang. Optical Character Recognition) to oprogramowanie pozwalające na znajdowanie i rozpoznawanie fragmentów tekstu w plikach graficznych. Jednak z uwagi na niejednorodny format kart oraz zmiany nanoszone ręcznie przez lekarza takie rozwiązanie okazało się nieefektywne.

Procedura pozyskiwania danych jest wyjątkowo nieoptymalna oraz podatna na błędy. Przy obecnej infrastrukturze szpitalnej ręczne przepisywanie danych jest jedynym sposobem na digitalizację zasobów pacjentów. Wypełnienie danych jednego pacjenta w badaniu było procesem trwającym około 15 minut. W szpitalu MSWiA lekarz radiolog dziennie wykonuje około 4 badań rezonansu magnetycznego, więc można założyć, że około godziny dziennie poświęca na przeglądanie papierowej dokumentacji. Po konsultacjach z doktor podjęta została decyzja aby stworzyć program, który pozwoli przyspieszenie tego procesu. Lekarze którzy dostarczyli swojej ekspertyzy przyznali, że w codziennych obowiązkach pomógłby im program wyliczający objętość prostaty na podstawie plików DICOM. Jest to jedna z płaszczyzn, na których informatyka jest w stanie pomóc oszczędzać czas lekarzy, co przekłada się na komfort oraz zdrowie pacjentów.

## 2.2. Dane szpitalne

W tej pracy badania prowadzono na danych od 222 pacjentów. Kompletowanie zbioru danych trwało przez 12 tygodni, od października 2017 końca grudnia 2017. Średni wiek pacjenta wynosi 67 lat, pacjenci byli badani przed średnio 4 miesiącami. W badaniach szczególną uwagę należało zwrócić poszczególne aspekty:

## 2.3. WSTĘPNA ANALIZA

**wiek pacjenta** - mężczyźni od 50 do 75 roku życia znajdują się w grupie ryzyka

**wywiad środowiskowy** - ważnym aspektem podczas wywiadu jest obciążenie rodzinne, czyli fakt występowania raka prostaty w najbliższej rodzinie pacjenta. Warto oczywiście zwrócić uwagę w jakim momencie diagnozy znajduje się pacjent. Czy ma stwierdzony nowotwór prostaty? Kolejną ważną informacją jest historia choroby pacjenta. Warto odnotować, iż niektórzy pacjenci byli badani po zakończeniu leczenia radioterapią bądź chemioterapią. Wtedy lekarz sprawdza czy udało się wyeliminować nowotwór, czy może pacjent jest w fazie nawrotu choroby.

**biopsja** - jeśli była przeprowadzona to jaki był jej wynik? Kiedy była przeprowadzona? W której części prostaty znaleziono zmiany nowotworowe?

**wynik w skali Gleasona** - w przypadku pacjenta ze stwierdzonym nowotworem kluczową rolę odgrywa skala Gleasona. Jest to skala opracowana z myślą o klasyfikacji nowotworów prostaty. Składa się ona z dwóch części, opisujących różne charakterystyki nowotworu. Po określeniu części składowych sumuje się je, aby otrzymać całościowy wynik Gleasona. Im wyższa punktacja tym bardziej histopatologicznie złośliwy jest nowotwór.

**PSA** - wynik ostatniego badania PSA we krwi. Specyfika antygeny PSA była przytoczona we wcześniejszej części tej pracy.

**Objętość prostaty** - wyliczona na podstawie wymiarów określonych w opisie badania USG oraz MR. Objętość prostaty potrzebna jest to wyliczenia współczynnika PSAD, oraz do ogólnej diagnozy.

**PIRADS** - najbardziej znaczące jest ognisko o najwyższej ocenie w skali PIRADS. Gdy pacjent ma ocenę 5, co zdarzało się w 25% zbadanych przypadków, to istnieje bardzo wysokie prawdopodobieństwo posiadania nowotworu prostaty w obserwowanym przez radiologa miejscu.

2.1 przedstawia wybrane kolumny uzyskane z danych szpitalnych

## 2.3. Wstępna analiza

U wszystkich pacjentów z nowotworem poziom PSA jest zdecydowanie podwyższony (norma to około 4 ng/ml [11]), jednakże u pacjentów, u których nie stwierdzono zmian nowotworowych PSA może być podwyższone. Skala PI-RADS V2 zapewnia dobrą dokładność, aczkolwiek, jest obarczona błędem wynikającym z ograniczonej dokładności i charakterystyki metody obrazowania, niejasności w interpretacji skali PIRADSV2 oraz dokładności standardu. Jedynym badaniem determinującym istnienie raka stercza jest bezpośrednia biopsja przezodbytnicza.

Wstępna analiza wykazała rozbieżność między objętościami wyliczonymi z badania USG oraz MR. Średnia różnica pomiędzy objętościami otrzymanymi w tych dwóch badaniach wynosi 14 cm<sup>3</sup>. Lekarz nie ma w momencie badania dokładnych wymiarów prostaty. Sposób wyliczania jej objętości jest przybliżeniem jej kształtu przez elipsoidę, co jest źródłem utraty dokładności pomiaru.

Tablica 2.1: Dane szpitalne

Nazwa kolumny	Opis
PW	unikalne ID nadawane pacjentowi w naszym badaniu
Data	Data badania MR
Dane kliniczne	Informacje na temat pacjenta klinicznego
Hist-pat	dane z badania histopatologicznego, czyli biopsji prostaty
Data hist-pat	Data wykonania biopsji
Gleason	klasyfikacja patomorfologiczna raka gruczołu krokowego
Płat	płat prostaty, w którym znajduje się zmiana nowotworowa
Liczba biopsji	
PSA	poziom swoistego antygenu stercza w ng/ml
Liczba PSA	liczba wykonanych badań PSA, które zostały wykonane
Objętość z usg	szacowana na podstawie przybliżenia elipsoidą
PSAD USG	zależność między objętością prostaty, a PSA
Objętość z MR	szacowana na podstawie przybliżenia elipsoidą
PSAD MR	zależność między objętością prostaty, a PSA
PI-RADS V2	radiologiczna skala wykorzystywana do wyznaczania prawdopodobieństwa wystąpienia zmian nowotworowych
Płat	oznaczenie płatu prostaty z największym oznaczeniem PI-RADS V2
Oznaczenie strefy	dokładna część prostaty, w której znajduje się zmiana

Najniższe wyniki PSA mają pacjenci, którzy przeszli resekcję prostaty, co oznacza, że Ci pacjenci nie wykazują cech wznowy. Najwyższe wyniki, natomiast, mają pacjenci ze stwierdzoną, przez biopsję, co jest zgodne z udowodnioną korelacją pomiędzy poziomem PSA a stopniem złośliwości histopatologicznej wykrytego nowotworu.

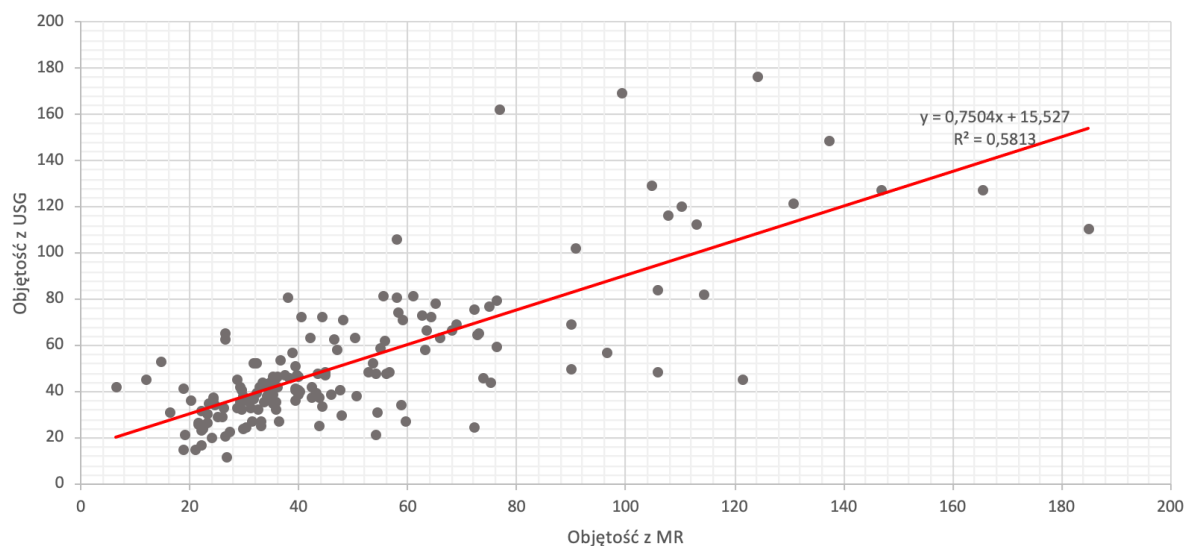
## 2.4. Statystyki

Tabela 2.2 przedstawia statystyki zebrane z utworzonego zbioru danych

## 2.4. STATYSTYKI

Tablica 2.2: Podstawowe statystyki

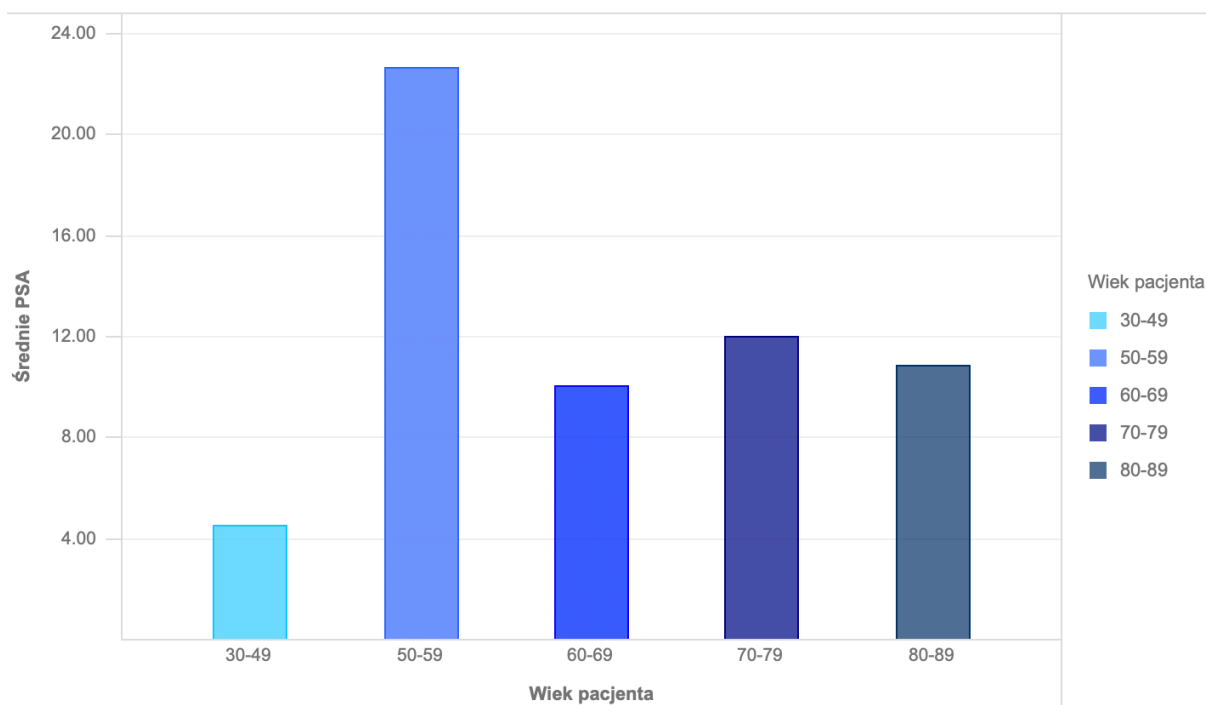
Nazwa	Wartość
Liczba pacjentów	222
Średni wiek pacjenta	67
Liczba pacjentów u których zdiagnozowano raka	69
Średnia objętość z USG	52,02
Odchylenie standardowe objętości z USG	30,43
Średnia objętość z MR	50,28
Odchylenie standardowe objętości z MR	32,63
Średnie PSA	12,05
Odchylenie standardowe PSA	36,46



Rysunek 2.1: Korelacja między objętością z MR i objętością z USG

Z przeprowadzonych obliczeń wynika wniosek, że te dwie objętości są ze sobą w silnej korelacji. Współczynnik korelacji dla tych dwóch zmiennych wynosi 0.76.

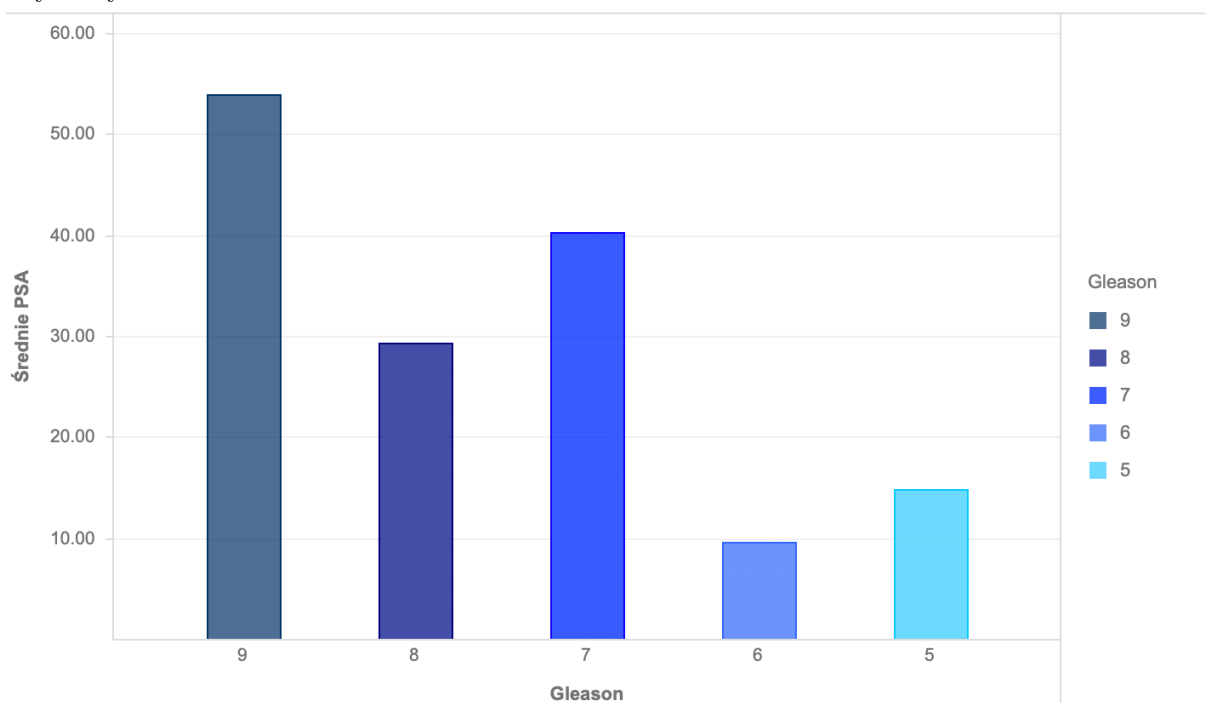
Kolejna ze statystyk umieszczonych na wykresie to rozkład PSA względem wieku. Chcieliśmy uzyskać informację jak wiek pacjentów przekłada się na potencjalna zachorowalność na raka prostaty.



Rysunek 2.2: PSA względem wieku

Wiek pacjentów został pogrupowany. Widać, że największe średnie PSA mają mężczyźni między 50 a 60 rokiem życia. Jest to obserwacja, którą dało się przewidzieć, ponieważ mężczyźni często dopiero zaczynają się badać po 50. roku życia.

Następnie chcieliśmy sprawdzić czy istnieje korelacja pomiędzy poziomem PSA i wynikiem uzyskanym w skali Gleasona.

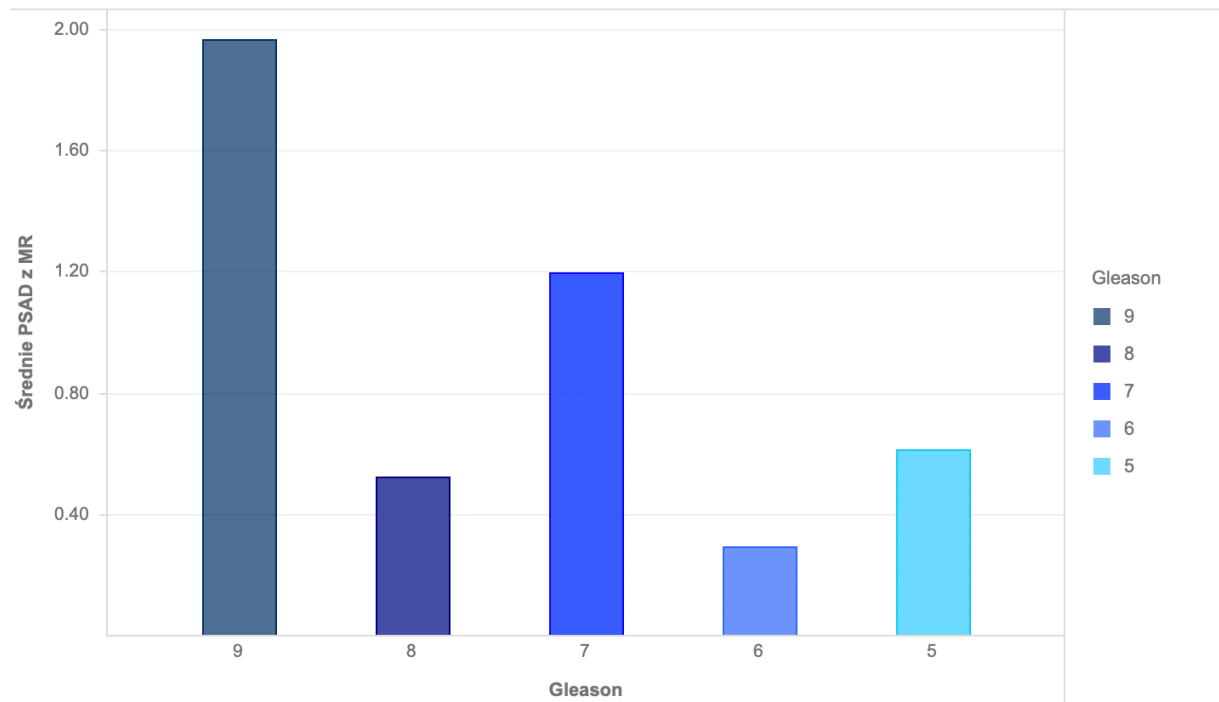


Rysunek 2.3: PSA względem Gleasona

U dwóch badanych pacjentów zdiagnozowano raka z oceną Gleasona na poziomie 9. Średnie PSA tych pacjentów wynosiło ponad 50 ng/ml co jest pięćdziesięciokrotnym przekroczeniem normy.

## 2.4. STATYSTYKI

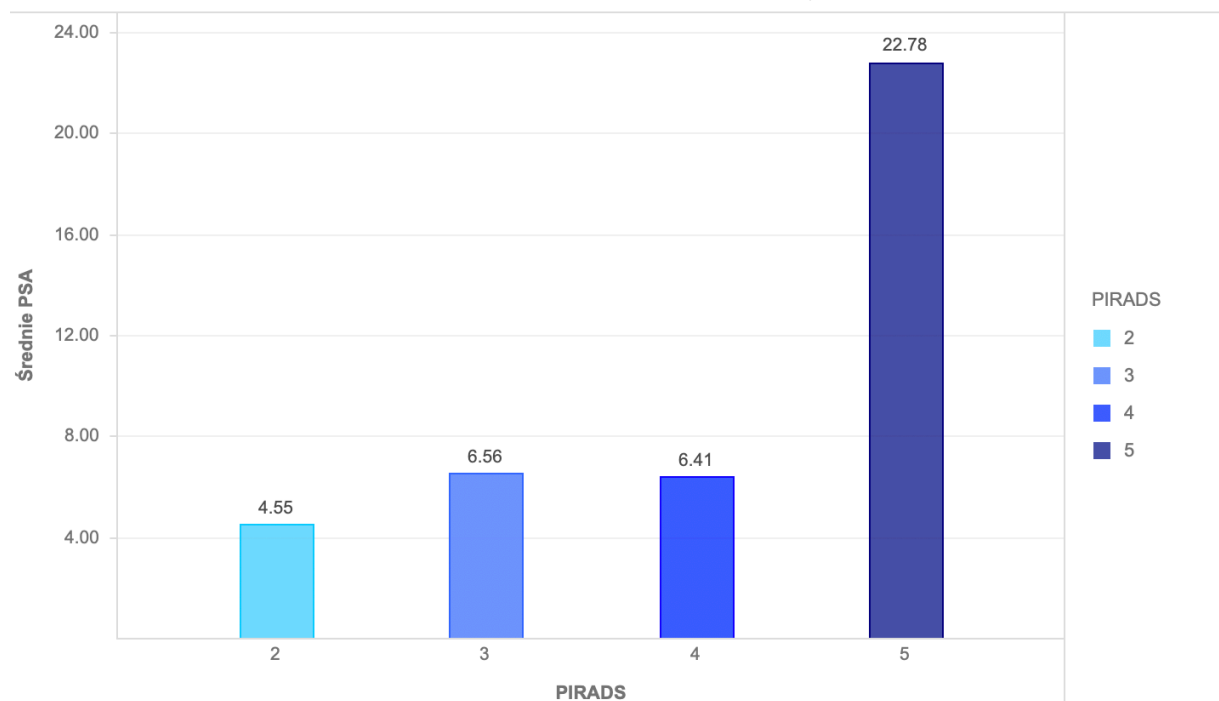
Kolejnym badaniem jakie przeprowadziliśmy było sprawdzenie powiązania czy PSAD rośnie wraz z punktacją Gleasona



Rysunek 2.4: PSAD względem Gleasona

Na tym wykresie można zobaczyć Jak skala Gleasona wpływa na PSAD, czyli połączenie PSA oraz objętości prostaty.

Ostatnim testem było sprawdzenie czy istnieje zależność pomiędzy PSA oraz PIRADS.



Rysunek 2.5: PSA względem PIRADS

Na wykresie poziomy PIRADS 2,3 i 4 mają stosunkowo podobne wyniki średniego PSA, natomiast poziom PIRADS 5 ma zdecydowanie zawyżony poziom PSA.

## 2.5. Ciekawe przypadki

Podczas analizy danych pacjentów udało się zidentyfikować trzy najbardziej interesujące przypadki. Kryterium wyboru ich ciekawości było czysto subiektywne. Przypadki opisane są za pomocą identyfikatora PW, który został nadany pacjentom w celu anonimizacji ich danych.

PW86 - PSA na poziomie 483,3 ng/ml, najwyższe dotychczas spotkane. Podwyższone PSA, oczywiście jest następstwem raka znalezione w obu płatach prostaty.

PW35 - ognisko PI-RADS V2 o mocy 5 znalezione w prawym płacie prostaty, natomiast po biopsji stwierdzono raka w lewym płacie.

PW143 - 65 wykonanych badań PSA. Na każdym z nich wynik jest podwyższony, natomiast ani badanie rektalne, ani MR ani biopsja nie wykazały obecności raka.



### 3. Specyfikacja

Celem pracy dyplomowej jest stworzenie narzędzia wspomagającego diagnostykę nowotworu prostaty. Powinno ono wspierać pracę lekarza w diagnozie oraz zapewnić automatyczne wyznaczanie objętości prostaty na podstawie zdjęć rezonansu magnetycznego. Priorytetem aplikacji jest to aby posiadała modułową strukturę, która zapewni łatwą integrację ze szpitalną bazą danych oraz szybki rozwój lub zmianę istniejących komponentów. Aplikacja aby zapewnić łatwy dostęp do szpitalnych baz danych musi posiadać komponent odpowiadający za integrację z systemem PACS. Na potrzeby poniższej pracy utworzyliśmy naszą instancję bazy danych w oparciu o otwarte rozwiązanie ORTHANC [8] w celach deweloperskich.

Narzędzie będzie w stanie przyjąć nowe dane pacjenta w formacie DICOM, przetworzyć zdjęcia zawarte w pliku, wykonać segmentację gruczołu prostaty oraz wyświetlić użytkownikowi przewidywaną maskę. Ponadto użytkownik jest w stanie samodzielnie modyfikować predykcję maski używając do tego opcji edycji. W takim wypadku jako segmentacja zostaje zapisana najnowsza wersja maski stworzonej przez lekarza. Dzięki temu w ramach prowadzonej diagnostyki przeprowadzanej przez specjalistów powstają wysokiej jakości zbiory danych potrzebne do wprowadzenia metod wspomagania opartych o narzędzia sztucznej inteligencji. Kolejny moduł to moduł liczenia objętości wysegmentowanego gruczołu. Aplikacja powinna zapewniać kilka alternatywnych algorytmów obliczania, dzięki czemu będziemy mogli porównywać otrzymane wyniki.

Ponieważ chcemy dostosować aplikację pod jak największą ilość środowisk, serwer backendowy powinien być możliwy do instalacji na każdym systemie operacyjnym.

#### 3.1. Opis biznesowy

Aktualnie jest dostępnych wiele przeglądarek formatu DICOM, lecz żadna z nich nie zawiera modułu pozwalającego na segmentację gruczołu prostaty. Jedną z ważniejszych informacji jakie wykorzystuje się przy diagnostyce raka gruczołu prostaty jest objętość. Na tej podstawie szacuje się parametr PSAD, który silniej niż w przypadku PSA koreluje z obecnością nowotworu w prostatie. [12]. Ponadto, żadne z narzędzi używanych obecnie w szpitalu nie posiada modułu pozwalającego na liczenie objętości gruczołu.

Nasze rozwiązanie ułatwia pracę lekarza przedstawiając aproksymowaną objętość gruczołu prostaty. Dodatkowo użytkownik jest w stanie samodzielnie poprawić predykcję używając opcji edycji maski. Dzięki takiej informacji wraz z rozwojem systemu mamy dostęp do coraz dokładniejszych danych dotyczących segmentacji pacjenta. Aplikacja ułatwi pracę lekarza oraz zgromadzi dane zawierające dokładne segmentacje utworzone przy użyciu algorytmu, które następnie podlegają kontroli lekarza. Narzędzie dostarcza również informacje na temat objętości

gruczołu krokowego biorąc pod uwagę precyzję urządzenia na którym zostały wykonane zdjęcia.

Objętość jest kluczową informacją dla lekarza przy wystawianiu diagnozy. Wizualizacja wysegmentowanego gruczołu prostaty oraz obliczona wartość objętości pomoże lekarzowi w diagnostyce, a dzięki łatwemu rozszerzaniu aplikacji rozwój i modyfikacje modułu obliczania objętości lub segmentacji można łatwo zmieniać algorytmy, za pomocą, których je wyznaczamy.

### 3.2. Wymagania funkcjonalne

Podstawowym wymaganiem aplikacji jest możliwość kalkulacji segmentacji, objętości oraz integracja ze szpitalnym systemem PACS. Metody realizacji każdego z nich zostały opisane w dalszej części pracy.

Podstawą interakcji użytkownika z systemem jest strona WWW - użytkownik powinien być w stanie otworzyć ją na dowolnym komputerze z dostępem do internetu, wyposażonym w przeglądarkę:

1. Google Chrome w wersji 49 lub wyższej
2. Mozilla Firefox w wersji 52 lub wyższej
3. Safari w wersji 10.1 lub wyższej

Posiadanie przeglądarki innej niż wymienione lub w starszej wersji nie oznacza że strona nie będzie działać, jednak nie da się zagwarantować że będzie to działanie w pełni poprawne. Każdy z użytkowników ma dostęp do tych samych danych oraz takie same możliwości Funkcjonalność aplikacji WWW rozbita jest na trzy podstrony:

1. Wyświetlenie listy dostępnych pacjentów z aplikacyjnej bazy danych lub ze szpitalnego systemu PACS po wpisaniu danych logowania lekarza.
2. Dodanie nowego pacjenta z pliku lub ze szpitalnej bazy danych. Dane szpitalne przed wysłaniem na stronę serwera powinny być zanonimizowane tak aby serwer nie przechowywał żadnych danych wrażliwych.
3. Wyświetlenie danych pacjenta w tym obrazu z rezonansu magnetycznego, maski uzyskanej poprzez segmentację obrazu, danych z pliku DICOM oraz wartości objętości gruczołu na podstawie aktualnej maski.

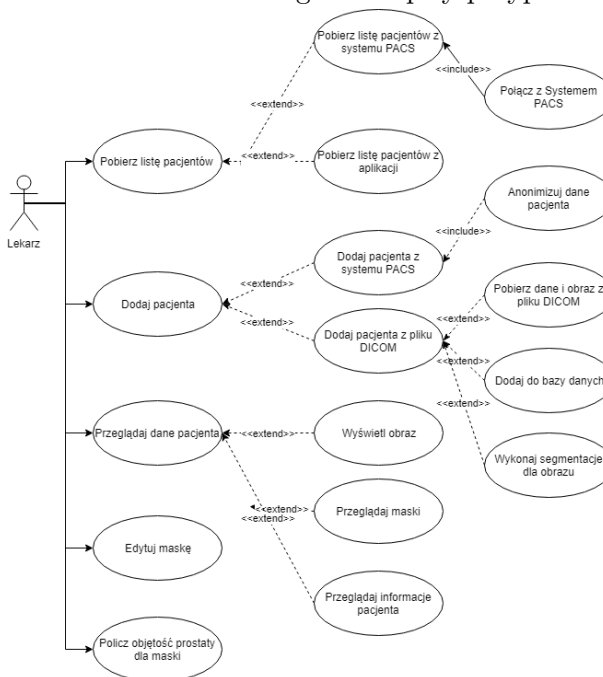
### 3.3. Diagram przypadków użycia

Po wejściu na stronę internetową wyświetlana jest lista pacjentów znajdujących się obecnie w bazie danych. Dodatkowo po wcześniejszym połączeniu się z systemem szpitalnym PACS ma możliwość wyświetlenia pacjentów ze szpitalnej bazy danych. W ramach środowiska deweloperskiego dane, którymi inicjalnie zasilona jest aplikacyjna baza danych pochodzą z konkursu. [2] Lekarz ma możliwość dodania nowego pacjenta z pliku lub ze szpitalnej bazy danych. Aby dodać pacjenta z pliku wystarczy w przeglądarce wybrać pacjenta, obraz DICOM zostanie pobrany z bazy PACS aplikacja, zanonimizowany po stronie przeglądarki, a następnie

### 3.4. WYMAGANIA NIEFUNKCJONALNE

wysłany do serwera. Dodatkowo użytkownik ma możliwość dodania pacjenta z pliku DICOM poprzez przeciągnięcie zbioru zdjęć pacjenta na okno przeglądarki. W przypadku gdy wybrany pacjent nie znajduje się jeszcze w bazie danych aplikacja tworzy nową encję opisującą tego pacjenta, w przeciwnym przypadku, jeżeli obraz nie znajduje się jeszcze w bazie danych dla pacjenta o takim identyfikatorze, aplikacja dodaje kolejne zdjęcia dla już istniejącego pacjenta. Następnie lekarz ma możliwość oglądania zdjęć pacjenta, oraz danych jakie o tym pacjencie aktualnie znajdują się w bazie danych. Strona wyświetla kolejne obrazy, które zostały dodane dotyczące wybranego przypadku. Dodatkowo wyświetlane są informacje z pliku DICOM dotyczące informacji o pacjencie i wyników badań jeżeli taki znajdowały się w pliku, który został wysłany do bazy danych. Ponadto aplikacja wyświetla maskę przedstawiającą wynik procesu segmentacji prostaty. Użytkownik ma możliwość ukrycia maski oraz jej edycji za pomocą dostarczonych narzędzi. Aplikacja dodatkowo liczy objętość wysegmentowanej prostaty na podstawie masek znajdujących się w bazie danych. W aplikacji dostępne jest kilka alternatywnych algorytmów liczenia objętości. Wartość nie jest przeliczana automatycznie w przypadku edycji. Lekarz po edycji wszystkich masek, naciska przycisk na stronie, który uruchamia proces liczenia objętości

Na poniższym rysunku przedstawiono w postaci diagramu UML zbiór przypadków użycia platformy dla lekarza. W tabeli 3.1 zebrano szczegółowe opisy przypadków użycia.



Rysunek 3.1: Diagram przypadków użycia

### 3.4. Wymagania niefunkcjonalne

Opiszemy tutaj wymagania niefunkcjonalne systemu. Z punktu widzenia użytkownika jest aby miał dostęp do stabilnego systemu, który zapewnia komfortową pracę. Z punktu widzenia osoby utrzymującej system istotne jest aby architektura była zrozumiała, a modyfikacje mogły być łatwo wprowadzane przy tym bez potrzeby zmian całego systemu. Tabla 3.2 przedstawia wymagania niefunkcjonalne w naszym systemie.

Tablica 3.1: Przypadki użycia

Nazwa	Opis	Odpowiedź systemu
Połącz z systemem PACS	Użytkownik po podaniu informacji dostępu do szpitalnego systemu PACS zostaje z nim połączony	Lista pacjentów znajdujących się w wybranym systemie
Pobierz pacjenta z systemu PACS	Pobranie obrazu lub obrazów w formacie DICOM i wysłanie ich po uprzedniej anonimizacji do serwera aplikacyjnego	Informacja o sukcesie lub błędzie przy dodawaniu użytkownika
Dodaj pacjenta z pliku	Wysłanie pliku lub plików DICOM do serwera aplikacyjnego	Informacja o sukcesie lub błędzie przy dodawaniu użytkownika
Wyświetl listę pacjentów dostępnych w systemie	Pobiera listę pacjentów dostępnych w systemowej bazie danych, inicjalnie baza danych jest zasilona danymi pochodzącymi z archiwum danych pacjentów ze zdiagnozowanym rakiem prostaty	Lista identyfikatorów dostępnych pacjentów
Zobacz pacjenta	Wyświetla dane dotyczące danego pacjenta aktualnie znajdujące się w bazie danych, w szczególności listę obrazów rezonansu magnetycznego oraz maskę z wysegmentowaną prostatą	Dane pacjenta, dla czytelności dane powinny być podzielone na części takie jak na przykład informacje o pacjencie takie jak identyfikator czy data urodzenia, osobno informacje na temat obrazów.
Segmentuj	Oblicza segmentację dla wybranego obrazu	Maska opisująca prostatę na wybranym obrazie
Policz objętość	Liczy objętość na podstawie listy masek na podstawie wybranego algorytmu	Wartość objętości prostaty

### 3.5. Środowisko sprzętowe i programowe

Opiszemy w dużym skrócie technologie wykorzystane w poniższej pracy, decyzje jakie stały za poszczególnymi wyborami, oraz alternatywy jakie były rozważane w procesie projektowania.

#### Języki i technologie

##### Docker

Podstawą całej architektury backendowej aplikacji jest wykorzystanie kontenerów czyli środowiska, w którym każdy serwis uruchamiany jest w odizolowanym środowisku zawierającym wszystkie zależności wymagane do jego działania. Ponieważ każdy z nas miał inne środowisko deweloperskie, Docker zapewnia nam, że serwisy będą działały w dokładnie ten sam sposób na każdej maszynie. Ma to również zalety dla środowisk produkcyjnych, ponieważ niezależnie czy udostępnimy naszą aplikację jako program on-premise czy zdecydujemy się na umieszczenie go w chmurze, konfiguracja będzie tak samo nieskomplikowana. Zapewnia to niezależność poszczególnych komponentów oraz odizolowanie aplikacji od systemu operacyjnego. Zdecydowaną przewagą Docker'a nad

### 3.5. ŚRODOWISKO SPRZĘTOWE I PROGRAMOWE

Tablica 3.2: Wymagania niefunkcjonalne

Obszar wymagań	Nr wymagania	Opis
Użyteczność	1	Aplikacja będzie aplikacją internetową dostępną na systemach Linux lub Windows. Aplikacja będzie zawierać przyjazne użytkownikowi karty z opcją segmentacji oraz obliczania objętości prostaty. Dodatkowo lekarz będzie miał możliwość dodawania kolejnych pacjentów
Niezawodność	2	Aplikacja będzie w stanie wykonywać funkcjonalności opisane wyżej. Ponadto Aplikacja będzie zapisywać logi z błędami oraz przyczyną awarii
Wydajność	3	Aplikacja WWW powinna działać płynnie na każdym komputerze z dowolnym systemem operacyjnym wyposażonym w odpowiednią przeglądarkę. Obliczenia zależnie od rozmiarów danych wejściowych aplikacja powinna wykonywać takim czasie aby praca użytkownika nie była przez nią spowolniona, przy tym nie blokując interfejsu użytkownika
Utrzymanie	4	Aplikacja będzie miała architekturę modułową, która będzie łatwo rozszerzalna. Cały kod aplikacji zostanie udokumentowany. W ramach środowiska developerskiego powstanie obraz Docker.

wirtualizacją jest możliwość uruchomienia aplikacji w wydzielonym kontenerze, ale bez konieczności emulowania całej warstwy sprzętowej i systemu operacyjnego. Do zarządzania kontenerami użyliśmy udostępnianego przez Docker narzędzia docker-compose. W ten sposób w przypadku potrzeby modyfikacji jednego z serwisów, aplikacja dalej pozostaje aktywna a zmianie ulega tylko jeden moduł.

#### Środowisko .NET Core

Serwis backendowy, który odpowiada za integrację z bazą danych oraz komunikację z warstwą prezentacji został napisany przy użyciu frameworku .NET Core. Główną zaletą tego wyboru jest wieloplatformowość rozwiązania. Dzięki nowemu środowisku uruchomieniowemu aplikacja może działać na dowolnym systemie operacyjnym. Korzystamy z najnowszej na chwilę pisanie pracy wersji (2.1), która wspiera większość pakietów dostępnych dla .NET Framework. Dodatkowo głównym założeniem przy powstawaniu systemu było aby był oparty o architekturę mikroservisów przy użyciu kontenerów, które również są wspierane przez framework.

#### Język programowania C#

System został wykonany z wykorzystaniem języka programowania C# oraz platformy programistycznej .NET Framework. Wybór tej technologii był podyktowany doświadczeniem zespołu w jej wykorzystaniu. .NET Framework został wykorzystany do napisania modułu backendowego odpowiedzialnego za obsługę akcji użytkownika oraz komunikację z bazą danych. Ponieważ serwer powinien działać na wszystkich systemach operacyjnych użyliśmy .NET core oraz ASP.NET MVC Core.

### Entity Framework Core

Entity Framework Core jest dedykowanym framework’iem dla aplikacji działających w środowisku .NET Core zapewniający dostęp i zarządzanie bazą danych. Jest to narzędzie typu ORM (ang. Object Relational Mapping). Dzięki użyciu tego narzędzia jako warstwy dostępu do bazy danych mogliśmy w łatwy sposób stosować podejście code-first przy projektowaniu aplikacji. Dodatkowo w przeciwieństwie do wersji z .NET Framework Entity Framework zapewnia wydajność porównywalną do natywnych wywołań zapytań bazodanowych za pomocą dedykowanego sterownika dla .NET czyli ADO.NET.

### SQL Server 2016

Microsoft SQL Server jest systemem służącym do zarządzania bazą danych stworzonym przez firmę Microsoft. Platforma spełnia wszystkie podstawowe funkcjonalności jakich oczekujemy od bazy danych ale dodatkowo ponieważ w przypadku operowania na dużych wolumenach danych, co jest bardzo prawdopodobne w przypadku danych szpitalnych, oferuje wiele możliwości optymalizacji oraz zaawansowane narzędzia do utrzymania oraz analityki. Jest to platforma preferowana do projektów działających w .NET Framework. Pomimo stosowania w projekcie podejścia *Code first* nie wykluczamy, że w przypadku rozwoju platformy będzie trzeba je zmienić.

### Python

W aplikacji wykorzystaliśmy język Python do modułu odpowiadającego za segmentację prostaty na obrazie. Wybór ten podyktowany jest dużą ilością dostępnych pakietów do uczenia maszynowego oraz głębokiego uczenia. Python jest wspierany na wszystkich większych systemach operacyjnych oraz zapewnia duże wsparcie dla uruchamiania w środowiskach kontenerowych. Ponadto brak procesu kompilacji wspomagał proces prototypowania algorytmu, co w przypadku uczenia maszynowego ma duże znaczenie.

### Framework Flask

Aby zapewnić komunikację za pomocą protokołu http między serwisem segmentacji, a pozostałymi elementami systemu potrzebowaliśmy biblioteki, która udostępnia taką funkcjonalność. Framework zapewnia elementarną implementację serwera http, który pozwala na komunikację z pozostałymi komponentami. Ponieważ segmentacja wspiera tylko i wyłącznie jeden punkt dostępowy uznaliśmy, że wybór rozwiązania bardziej rozbudowanego jest nieuzasadniony. Django czy Pyramid, to dwa bardzo popularne rozwiązania dla serwerów http zaimplementowane w języku Python jednak o wiele bardziej rozbudowanej konfiguracji nie przynosząc zarazem odczuwalnych korzyści.

### Biblioteka Keras

Do implementacji algorytmu segmentacji gruczołu krokowego na zdjęciach potrzebowaliśmy biblioteki pozwalającej na tworzenie sieci neuronowych. Podczas poszukiwań najbardziej odpowiedniego rozwiązania kierowaliśmy się kilkoma kryteriami. Przede wszystkim chcieliśmy aby rozwiązanie było napisane w języku Python oraz wspierało użycie biblioteki TensorFlow. Dodatkowo chcieliśmy aby biblioteka umożliwiała szybkie prototypownie oraz była przenośna między systemami operacyjnymi. Keras jest obecnie jednym z najpopularniejszych bibliotek do takiego zastosowania. Zapewnia wysoko poziomowy interfejs ułatwiający tworzenie rozwiązań opartych o uczenie maszynowe.

### 3.5. ŚRODOWISKO SPRZĘTOWE I PROGRAMOWE

Decyzja o użyciu frameworku zapadła na podstawie spełniania wszystkich powyższych założeń oraz doświadczeniem zespołu w jego wykorzystaniu.

#### **Tensorflow**

TensorFlow jest biblioteką typu otwarty kod do obliczeń numerycznych i uczenia maszynowego. Biblioteka łączy ze sobą szereg modeli i algorytmów głębokiego uczenia. Korzysta z Pythona, aby zapewnić wygodne API do budowania aplikacji, a obliczenia wykonywane są w C++. Biblioteki nie wykorzystaliśmy bezpośrednio tylko przez nakładkę Keras, która udostępnia wysokopoziomowe API.

#### **OpenCV, OpenCvSharpCore**

OpenCV [4] to biblioteka oprogramowania do nauki i uczenia maszynowego typu open source. Jest to chyba najpopularniejsza biblioteka do przetwarzania obrazów. Do integracji z .NETem użyliśmy wrappera działającego w środowisku .NET Core OpenCvSharpCore. Zapewnia dostęp do metod biblioteki poprzez kod C#. W naszej pracy została wykorzystana w module liczącym objętość do znajdowania otoczki maski oraz liczenia pola powierzchni poligonów.

#### **React**

React okazał się idealnym rozwiązaniem, ponieważ jest jedynie biblioteką dostarczającą komponenty, która może być rozszerzalna przez kolejne biblioteki. Projekt ten dojrzałym framework’iem stworzonym i rozwijanym przez Facebook. Przygotowując aplikację SPA postanowiliśmy znaleźć rozwiązanie, które najbardziej dopasuje się do naszych wymagań.

Ponadto wszystkie wymienione technologie zostały wybrane do realizacji pracy, ponieważ zespół, który nad nią pracował, miał doświadczenie w pracy z nimi wyniesione z poprzednich projektów realizowanych w ramach studiów i pracy zawodowej.

#### **Rozważane alternatywne technologie**

W momencie pisania pracy ustaliliśmy, że najpopularniejszymi aktualnie rozwiązaniami są następujące projekty:

#### **Java**

Jeden z najpopularniejszych obecnie języków programowania, teoretycznie spełniający wszystkie nasze wymagania. Zapewnia modularność, jest wspierany przez wszystkie języki programowania, ma wiele dostępnych bibliotek gotowych do użycia w aplikacji. Odrzuciliśmy ten wybór z powodu dużej ilości konfiguracji potrzebnej do utworzenia systemu jaki tutaj opisujemy. Dodatkowo ponieważ przetwarzamy pliki zajmujące często nawet powyżej 100MB obawialiśmy się problemów z wydajnością zwalniania pamięci w Javie. W przypadku przetwarzania dużych wolumenów danych Java wymaga dużej ingerencji w mechanizm zwalniania pamięci.

#### **Angular**

Projekt Angular jest jednym z najpopularniejszych obecnie rozwiązań do tworzenia warstwy prezentacji aplikacji. Początkowo wydawał się być doskonałym wyborem dla naszego zastosowania (dojrzały projekt, wspierany przez Google), aczkolwiek wymuszał on używanie konkretnych rozwiązań stworzonych na potrzeby tego konkretnego frameworka

do wykonywania zapytań. Z tego powodu zdecydowaliśmy się zrezygnować z tego projektu, ponieważ naruszał nasze kryterium dotyczące modularności.

#### **Vue**

Vue jest coraz bardziej popularnym frameworkiem, który wyróżnia się modularnością, przenośnością oraz niskim progiem wejścia. Nie dopasował się do naszych wymagań ze względu na niedojrzałość projektu, przez co dostępne komponenty nie spełniały naszych oczekiwań. Wciąż jest to rozwiązanie nie wspierane przez dużych klientów biznesowych.

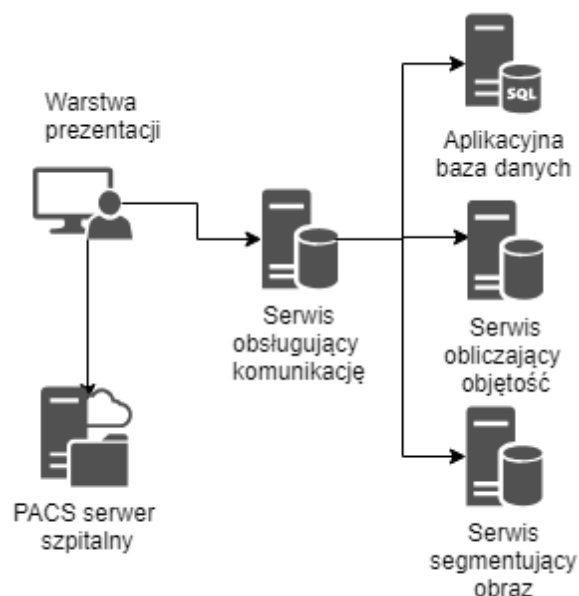


## 4. Opis aplikacji

W tym rozdziale przedstawione zostaną efekty pracy. Na wstępie przedstawimy wysoko poziomową architekturę całego rozwiązania. Następnie opiszemy poszczególne moduły zawarte w aplikacji oraz zasadę ich działania, przedstawiając szczegóły techniczne implementacji. Dzięki podziałowi aplikacji na osobne serwisy działające niezależnie, zawarte w dalszej części opisu modułów również opiszemy niezależnie. Kontenery oddzielają aplikacje od siebie oraz infrastruktury, na której działają zapewniając dodatkową warstwę ochronną dla aplikacji.

### 4.1. Architektura systemu

System składa się z modułów realizujących ściśle określone zadania, niezależne zadania, dlatego też łatwo wydzielić je jako osobne serwisy działające niezależnie od siebie. Każdy z serwisów działa jako osobny kontener. Konteneryzacja polega na tym, że umożliwia się uruchomienie wskazanych procesów aplikacji w wydzielonych obiektach, które z punktu widzenia aplikacji są odrębnymi instancjami środowiska uruchomieniowego. Każdy kontener posiada wydzielony obszar pamięci, odrębny interface sieciowy z własnym prywatnym adresem IP oraz wydzielony obszar na dysku.



Rysunek 4.1: Schemat architektury aplikacji

Komunikacja serwera backendowego z serwerem PACS odbywa się za pośrednictwem aplikacji internetowej, z której korzysta klient. Jest to zrobione specjalnie i daje nam możliwość łatwej integracji z systemem szpitalny. W zasadzie jest to jedyna opcja jaką znaleźliśmy

w trakcie naszego badania architektury szpitalnej nie wymagająca ustawiania specjalnego połączenia VPN między naszym serwisem a serwerami szpitalnymi. Dzięki temu zyskujemy niezależność. Aplikacja może działać na dowolnym środowisku przy tym zapewniając klientowi dostęp i wysyłanie danych pacjenta na nasze serwisy do przetwarzania. Rozwiązanie w którym nasz serwer komunikuje się bezpośrednio z bazą danych szpitalnych ma kilka dużych wad. Przede wszystkim przesyłamy poufne dane pacjentów przez sieć. Aby to robić bezpiecznie potrzebowalibyśmy dedykowanego bezpiecznego połączenia, którego nie udało nam się uzyskać przez cały okres pisania pracy.

Reszta architektury jest standardowa w przypadku aplikacji o architekturze mikroserwisowej. Mamy główny serwer odpowiadający za komunikację z użytkownikiem i obsługę bazy danych. Serwisy do liczenia objętości i segmentacji są niezależne i nie mają bezpośredniego dostępu do bazy danych. Jedyna modyfikacja jaką można by wykonać w przypadku intensywnego użytkowania aplikacji to skonfigurować osobne bazy dla każdego serwisu aby uniknąć niepotrzebnego przesyłania danych przez sieć. Jednak w naszym przypadku proponowane rozwiązanie jest bezpieczniejsze, łatwiejsze w utrzymaniu i w zupełności wystarczające.

## 4.2. Komunikacja między modułami

Komunikacja między komponentami odbywa się za pomocą protokołu *http*. Każdy z serwisów udostępnia swoje punkty dostępu. W przypadku serwisu kalkulującego objętość oraz wykonującego segmentację dostęp jest jedynie z poziomu kontenera.

Serwisy są lokalizowane dzięki mechanizmom Dockera, który tworzy podsieć dla wszystkich działających w obrębie jednego Docker-compose. Dostęp do serwisów odbywa się przez adres postaci `http://nazwa-serwisu:port` (jeżeli nie zmieniony to serwis działa na porcie 80). Dostęp z zewnątrz zapewnia jedynie serwis obsługujący komunikację, a pozostałe serwisy są niedostępne.

Każdy z naszych serwisów udostępnia REST API. Najwięcej metod ma moduł do komunikacji z klientem, którego pełna dokumentacja jest opisana w Załączniku 1. Może być przydatne w przypadku chęci użycia naszego rozwiązania z inną warstwą prezentacji np. aplikacją okienkową.

## 5. Opis modułów

W poniższym rozdziale opiszemy zostaną dokładnie opisane poszczególne moduły. Przedstawiona zostanie droga dojścia do rozwiązania oraz ewentualne modyfikacje jakie można zastosować przy dalszym rozwoju. Skupiliśmy się na unikaniu zależności pomiędzy modułami. Wszystkie moduły zostały zaprojektowane w sposób modularny, tak aby mogły być dowolnie wymienialne.

### 5.1. Moduł segmentacji

Moduł odpowiada za znajdowanie na obrazie wejściowym prostaty. Algorytm działa na zdjęciach w formacie PNG i zwraca również obraz w tym samym formacie. Do obliczeń użyto konwolucyjnych sieci neuronowych, na podstawie danych z konkursu [2], które składały się z obrazów badań MRI oraz wysegmentowanych masek przez specjalistów. Problem segmentacji gruczołu prostaty jest niezwykle istotny z perspektywy lekarzy, którzy na podstawie wysegmentowanej maski gruczołu są w stanie łatwiej postawić diagnozę dotyczącą raka prostaty.

#### Segmentacja obrazu

Segmentacja obrazu jest procesem dzielenia obrazu cyfrowego na wiele segmentów). Celem segmentacji jest uproszczenie lub zmiana reprezentacji obrazu na obszary, które są bardziej znaczące i łatwiejsze do analizy. Segmentacja obrazu jest używana do lokalizowania obiektów i granic (linii, krzywych itp.) w obrazach. Jest to proces przypisywania etykiety do każdego piksela na obrazie, tak że piksele z tą samą etykietą mają pewne cechy. Proces polega na podziale obrazu na obszary, które są jednorodne pod względem pewnych wybranych cech. Przez obszar najczęściej rozumiemy zbiór pikseli obrazu. Zadanie segmentacji nie tylko wymaga zrozumienia co się znajduje na obrazie, lecz również ustalenia gdzie się ten obszar znajduje. Celem segmentacji jest wyeksponowanie poszukiwanych własności obrazu, które są istotniejsze od pozostałych. Istnieje wiele algorytmów segmentacji, najpopularniejszymi są: progowanie obrazu, wykrywanie krawędzi, rozszerzanie obszaru. Każda z wyżej wymienionych metod jest zestawem reguł które są przygotowane przez człowieka podczas tworzenia algorytmu. Algorytmy te nie zależą w żaden sposób od danych na których będą wykonywane obliczenia. W ostatnich latach głębokie sieci neuronowe oraz konwolucyjne sieci neuronowe wykazały, że radzą sobie z problemem klasyfikacji lepiej niż statyczne rozwiązania.[5]. Sieci na podstawie dużej ilości danych dobierają zestaw filtrów, który daje najlepsze wyniki segmentacji dla danych treningowych. W naszej pracy zostały zastosowane sieci neuronowe, ponieważ rozwiązania oparte o głębokie uczenie od lat wygrywają konkursy na najlepsze wyniki klasyfikacji oraz segmentacji.

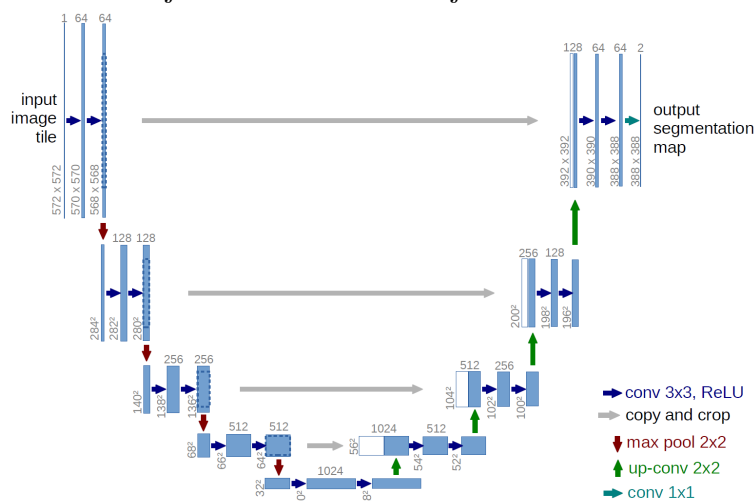
## Sieci neuronowe

Głęboka sieć neuronowa (DNN) to sztuczna sieć neuronowa z wieloma warstwami między warstwami wejściową i wyjściową. DNN znajduje matematyczne przekształcenie, aby zmienić dane wejściowe na dane wyjściowe, niezależnie od tego, czy jest to relacja liniowa czy nieliniowa. Sieć przechodzi przez warstwy obliczające prawdopodobieństwo każdego wyjścia. Każda manipulacja matematyczna jest uważana za warstwę. Złożone głębokie sieci neuronowe mają wiele warstw. Celem wielu warstw jest, aby ostatecznie sieć została nauczona w zakresie rozkładania obrazu na funkcje, identyfikowania trendów istniejących we wszystkich próbkach i klasyfikowania nowych obrazów według ich podobieństw bez konieczności wprowadzania danych przez człowieka.

Konwolucyjna sieć neuronowa (CNN) to rodzaj sztucznej sieci neuronowej wykorzystywanej w rozpoznawaniu i przetwarzaniu obrazów, która jest specjalnie zaprojektowana do przetwarzania obrazów. W przeciwieństwie do sieci użytej do głębokiego uczenia maszynowego, każda ukryta warstwa aktywacyjna jest obliczana jako iloczyn kartezjański wyników kolejnych warstw. Jednak w sieciach konwolucyjnych każda ukryta aktywacja jest obliczana przez pomnożenie małego lokalnego sygnału wejściowego względem wag. Wagi są następnie dzielone na całej przestrzeni wejściowej. Po obliczeniu ukrytych jednostek, maksimum warstwa pomaga usunąć zmienność ukrytych jednostek. W szczególności każda jednostka gromadząca maksimum otrzymuje aktywacje ze znalezionej ilości konwolucyjnych warstw, i wyprowadza maksymalną aktywację z tych konwolucji. Większość CNN użytych do rozpoznawania obrazu ma niższe warstwy sieciowe, które są konwolucyjne, a wyższe warstwy sieciowe są w pełni połączone.

## U-Net

Sieć U-Net jest zbudowana na podstawie architektury pełnych sieci konwolucyjnych(ang. Fully Convolutional Network) zmodyfikowaną w taki sposób, aby uzyskiwać lepsze rezultaty segmentacji na obrazach medycznych. Lepsze rezultaty pochodzą z kształtu architektury U, który skutkuje większą ilością cech w ścieżce wstępującej co przekłada się na lepszy przepływ informacji o kontekście i lokalizacji na obrazach o niskiej rozdzielczości.



Rysunek 5.1: Architektura sieci U-Net

W procesie uczenia sieci neuronowych najbardziej czasochłonnym elementem jest proces trenowania. Uczenie sieci neuronowych odbywa się na kartach graficznych ze względu na ich

## 5.1. MODUŁ SEGMENTACJI

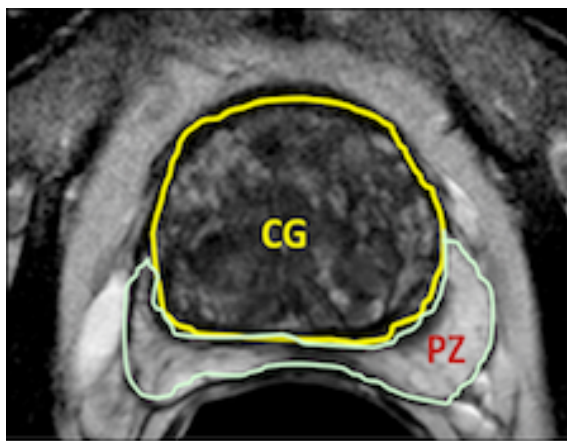
lepsze przystosowanie do przetwarzania dużej ilości operacji w sposób równoległy. Procesor ma za zadanie przełączanie kontekstów, ale nie zapewnia wysokiej przepustowości. Musi przełączać się między wieloma zadaniami przechowując konteksty podczas przełączania i dbać o pamięć współdzieloną. Procesor może być wykorzystywany do zadań o niskiej przepustowości i wysokiej latencji. Z drugiej strony procesor graficzny został zaprojektowany z myślą o dużym obciążeniu pracą i wysokiej przepustowości.

### 5.1.1. Opis danych

Najlepszym źródłem pozyskania danych dla naszych celów badawczych są strony konkursowe ponieważ zawierają obrazy z już odnalezioną prostatą na zdjęciu. Organizowanych jest wiele konkursów [2] w celu opracowania stabilnego algorytmu wyznaczania gruczołu prostaty, jednak po analizie zdecydowaliśmy się na dane z konkursu *Automated Segmentation of Prostate Structures Challenge* [6].

#### Dane

Dostarczone dane składały się z plików DICOM zawierających dane pacjentów oraz z maski zawierających obszary gdzie znajduje się prostata. Segmentacja wykonana została na danych pochodzących z obrazowania rezonansem magnetycznym mpMRI. Do segmentacji użyta jest modalność T2WI ponieważ niedokładniej odwzorowuje struktury anatomiczne. Dane były przygotowane przez lekarzy, oraz podlegały wcześniejszemu procesowi anonimizacji, stąd w plikach nie ma żadnych danych wrażliwych na temat pacjentów. Osoby poddano badaniom zarówno pod maszynami o mocy 1.5T oraz 3T z użyciem cewki endorektalnej oraz cewki powierzchniowej. Maski zostały przygotowane przez zespół lekarzy z uniwersytetów Boston University School of Medicine oraz Case Western University. Zdjęcia zostały przygotowane w formacie nrrd, gdzie zaznaczone są dwa regiony: część obwodową gruczołu krokowego (PZ) oraz centralny gruczoł(CG).



Rysunek 5.2: Zaznaczone regiony prostaty na zdjęciu z badania MR

**Zbiór treningowy** 60 pacjentów, łącznie 1073 obrazów

**Zbiór testowy** 10 pacjentów, 147 obrazów

**Zbiór walidacyjny** 10 pacjentów, 146 obrazów

#### Obrazy wejściowe

Dane zostały przetransformowane z formatu DICOM w taki sposób, aby wyeksportować

tag `pixelData` i przekonwertować do postaci tablicy zawierającą tylko intensywność pixeli.



Rysunek 5.3: Przykładowy obraz z badania MRI

### Oznaczone maski

Przygotowane maski podzielone zostały na dwa regiony gruczołu PZ oraz CG. Obszar CG został oznaczony wartościami 2, obszar PZ wartościami 1 natomiast pozostały obszar wartością 0. Ponieważ nasz problem był bardziej ogólny nie było potrzeby wprowadzenia rozróżnienia między strefami prostaty. Dlatego dane zostały poddane procesowi normalizacji maski do wartości z przedziału  $[0;1]$ .



Rysunek 5.4: Przykładowy obraz z badania MRI

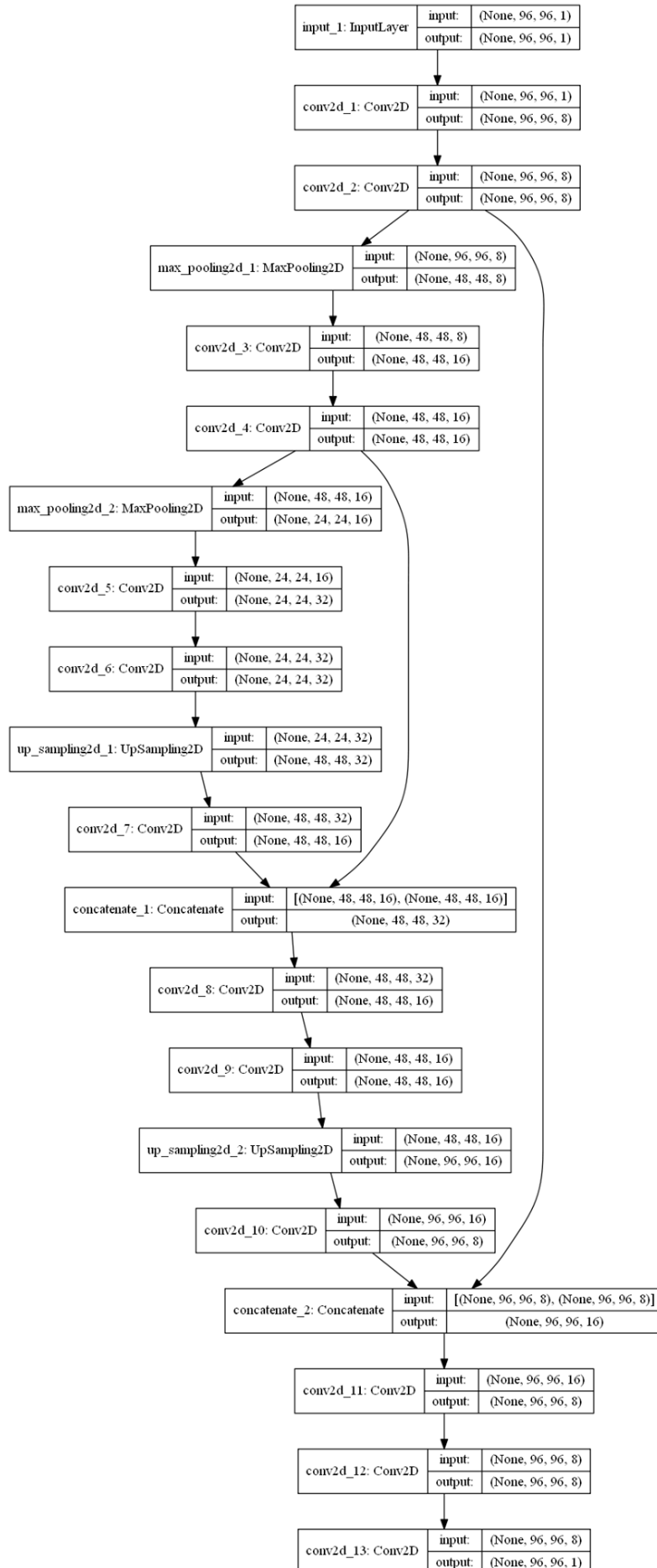
#### 5.1.2. Opis rozwiązania

W przypadku analizy obrazów medycznych problemem jest mały zbiór danych testowych dobrej jakości. Zostały do tego celu wykorzystane dane pochodzące z konkursu opisane powyżej. Eksplorację danych rozpoczęliśmy od przeglądania obrazów DICOM. Na wstępie zauważyliśmy, że granice pomiędzy tkankami miękkimi nie są dobrze podkreślone. Co więcej dane jakie

## 5.1. MODUŁ SEGMENTACJI

otrzymano są niskiej rozdzielczości. Segmentacja gruczołu krokowego nie jest zadaniem trywialnym z powodu podobnych intensywności obrazu między gruczołem prostaty a sąsiadującą tkanką oraz z powodu niejednorodności intensywności obrazu w okolicy prostaty. Co więcej, gruczoł krokowy ma dużą różnorodność wyglądu i kształtów u różnych pacjentów. Z uwagi na bardzo ograniczony zbiór testowy zastosowaliśmy mechanizmy *Data augmentation*, dzięki czemu w efektywny sposób został zwiększony zbiór testowy. Rozwiązanie jest procesem stosowanym przy przetwarzaniu obrazów medycznych, polega na zwielokrotnieniu zbioru testowego poprzez zastosowanie na obrazach inicjalnych podstawowych operacji przetwarzających obraz takich jak np. rotowanie obrazu, stosowanie filtru Gaussowskiego lub progowania.

Przed wyborem rozwiązania przeprowadziliśmy analizę aktualnie wykorzystywanych algorytmów do segmentacji obrazów. Jednym z założeń było znalezienie rozwiązania, które pomimo niewielkiego zbioru danych treningowych poradzi sobie z obrazami o niskiej rozdzielczości. Najbardziej dopasowanym rozwiązaniem okazała się uproszczona sieć U-Net, która powstała jako zgłoszenie do konkursu Automated Segmentation of Prostate Structures [?]. Wybraliśmy te rozwiązanie ponieważ sieć jest symetryczna, dzięki czemu zyskujemy większą dokładność znajdowania regionów obrazu oraz pomija kilka warstw co skutkuje lepszymi rezultatami przy małej ilości danych. Rozwiązanie nauczaliśmy ponownie na danych opisanych w rozdziale powyżej. Architektura sieci przedstawia poniższy diagram.





## 5.1. MODUŁ SEGMENTACJI

Architektura ta dzieli się na trzy części:

**Warstwa zstępująca** Zbiera informacje na temat kontekstu i znaczenia regionu. Interpretuje znaczenie obszaru poprzez zmniejszanie wielkości przestrzennej w każdym bloku. Wszystkie bloki w tej ścieżce składają się z następujących warstw:

**Conv2D 3x3** Warstwa konwolucyjna + funkcja aktywacji ReLu

**Conv2D 3x3** Warstwa konwolucyjna + funkcja aktywacji ReLu

**MaxPooling2D 2x2** Max Pooling

Każdy blok podwaja liczbę konwolucji oraz zmniejsza rozmiar przestrzenny o połowę.

**Wąskie gardło** Znajduje się pomiędzy dwoma ścieżkami. Składa się z dwóch warstw konwolucyjnych

**Conv2D 3x3** Warstwa konwolucyjna + funkcja aktywacji ReLu

**Conv2D 3x3** Warstwa konwolucyjna + funkcja aktywacji ReLu

**Warstwa wchodząca** Składa się z warstw powiększających dane przestrzenne regionu oraz jest łączona z symetrycznie odbitymi blokami ścieżki zstępującej co pozwala ustalić obszar oraz jego kontekst. Wszystkie bloki w tej ścieżce składają się z następujących warstw:

**UpSampling2D 2x2** Warstwa powiększająca która jest ztransponowaną warstwą konwolucyjną

**Conv2D 3x3** Warstwa konwolucyjna + funkcja aktywacji ReLu

**Concatenate** Połączenie informacji o lokalizacji przestrzennej z informacjami o kontekście obszaru

**Conv2d 3x3** Warstwa konwolucyjna + funkcja aktywacji ReLu

**Conv2d 3x3** Warstwa konwolucyjna + funkcja aktywacji ReLu

Rozwiązanie oparte na prostej architekturze U-Net rozszerzyliśmy o odwrócenie wartości pixeli. Pomysł ten pojawił się po obserwacji jak lekarz diagnozuje obrazy z badania MRI oraz na co zwraca uwagę. Popularną praktyką jest inwersja pixeli w celu uzyskania lepszego kontrastu.

Po augmentacji obrazów została wykonana *cross validacja* parametrów uczenia podając liczbę epok równą 30 oraz *batch\_size* wynoszący 32.

### 5.1.3. Wyniki

Do ustalenia skuteczności użyto miary wykorzystywanej w zadaniach segmentacji tj. współczynnika podobieństwa Sørensen. Miara ta mówi jak podobne są dwa obiekty.

$$dice\_coefficient = \frac{\sum(2 * TRUE\_SEG \cap PRED\_SEG)}{\sum TRUE\_SEG + \sum PRED\_SEG}$$

gdzie:

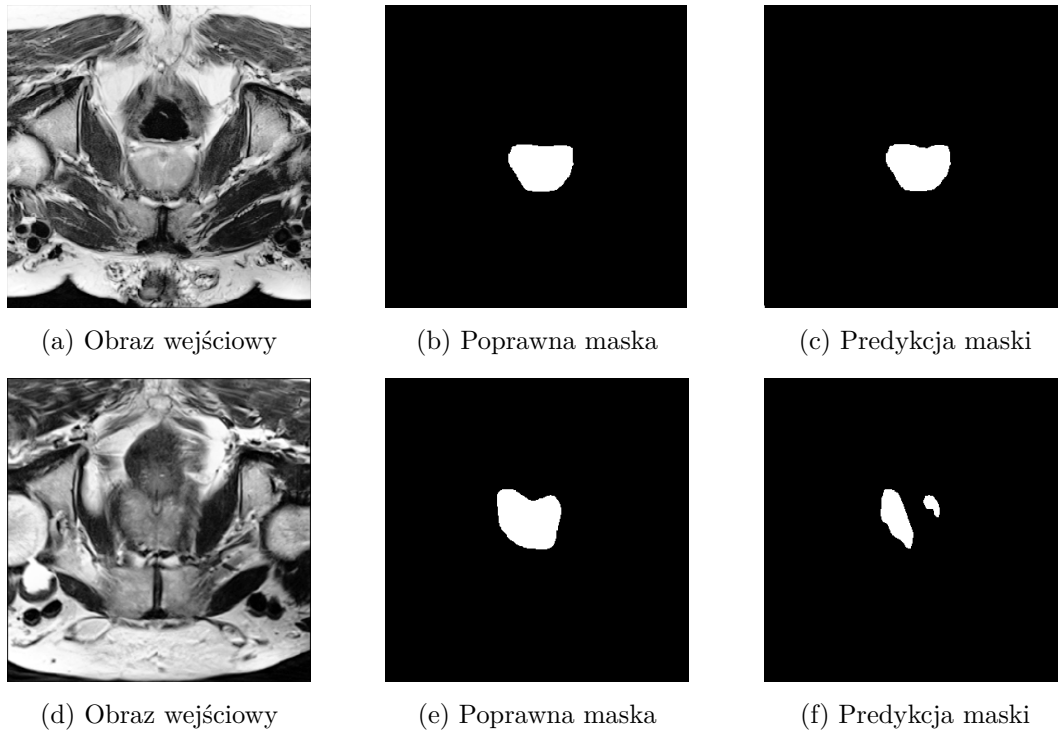
TRUE\_SEG - Binarny obraz zawierający prawidłową maskę segmentacji

PRED\_SEG - Binarny obraz zawierający predykcję maski segmentacji

Jej wartość to wspólny obszar regionu segmentacji podzielony przez całkowity rozmiar dwóch obrazów.

Po nauczaniu sieci osiągnęliśmy współczynnik podobieństwa równy około 0.70 na danych testowych. Wartość ta pokrywa się z wartością uzyskaną w zgłoszeniu do konkursu. Jest to wynik zadawalający, który pozwala na obliczenie objętości gruczołu alternatywnymi metodami.

Analizując wyniki segmentacji zauważyliśmy, że sieć radzi sobie bardzo dobrze w momentach, gdy granica pomiędzy organami jest mocno zarysowna. Problematiczne są zdjęcia rozmazane, na których gruczoł zlewa się z otaczającą go tkanką.



Rysunek 5.6: Przykłady wyników

#### 5.1.4. Alternatywne podejścia

1. Rozszerzenie zbioru treningowego poprzez augmentację danych zawierających obrazy przetworzone filtrem Gaussowskim może poprawić przypadki w których zdjęcie jest rozmyte.
2. Zwiększenie rozdzielczości obrazów wejściowych również powinno polepszyć wyniki segmentacji.

Każde z wyżej opisanych podejść jest warte rozważenia, jednak z uwagi na stopień skomplikowania nie podjęliśmy się badaniu ich w pracy inżynierskiej.

## 5.2. Moduł obliczania objętości

Jedynym z celów pracy było zaproponowanie alternatywnej metody do liczenia objętości. W szpitalu stosowany jest wzór na objętość gruczołu, jako aproksymację kształtem elipsoidy, przedstawiony wzorem:

## 5.2. MODUŁ OBLICZANIA OBJĘTOŚCI

$$V = (h_{max} - h_{min}) * V_{max} * 0.54$$

gdzie

$h_{max}$  - wysokość górnego końca gruczoły prostaty

$h_{min}$  - wysokość dolnego końca gruczoły prostaty

$S_{max}$  - Objętość warstwy o największym polu powierzchni, każde liczone jako suma objętości voxelów maski, gdzie objętość każdego voxelu liczona jest jako:

$$V_{d_{ij}} = PixelSpacing_0 * PixelSpacing_1 * SliceThickness$$

gdzie

$PixelSpacing_0$  - pierwsza wartość atrybutu PixelSpacing (0028, 0030) z pliku DICOM

$PixelSpacing_1$  - druga wartość atrybutu PixelSpacing (0028, 0030) z pliku DICOM

$SliceThickness$  - atrybut SliceThickness (0018, 0050) z pliku DICOM

Metoda polega na znalezieniu prostopadłościanu w który możemy wpisać gruczoł i obliczenie jego objętości. Następnie wartość jest przemnażana przez stałą aby przybliżyć kształt elipsoidy. Metoda jest niedokładna, co potwierdzają specjaliści diagnostyki obrazowej. Obecnie najdokładniejszą metodą jest zbadanie objętości po wycięciu gruczołu. Dowiedzieliśmy się, że wyniki uzyskane z aproksymacji objętości oraz wartości uzyskane po ekstrakcji są bardzo rozbieżne.

Postanowiliśmy zaproponować alternatywne rozwiązanie. Objętość w dalszych obliczeniach będziemy obliczać jako sumę voxelów znajdujących się w każdej z warstw obrazu.

$$V_{total} = \sum_{i=0}^n \sum_{d_i, j=0}^m V_{d_{ij}} \text{ gdzie } d_{ij} \text{ pixelem maski}$$

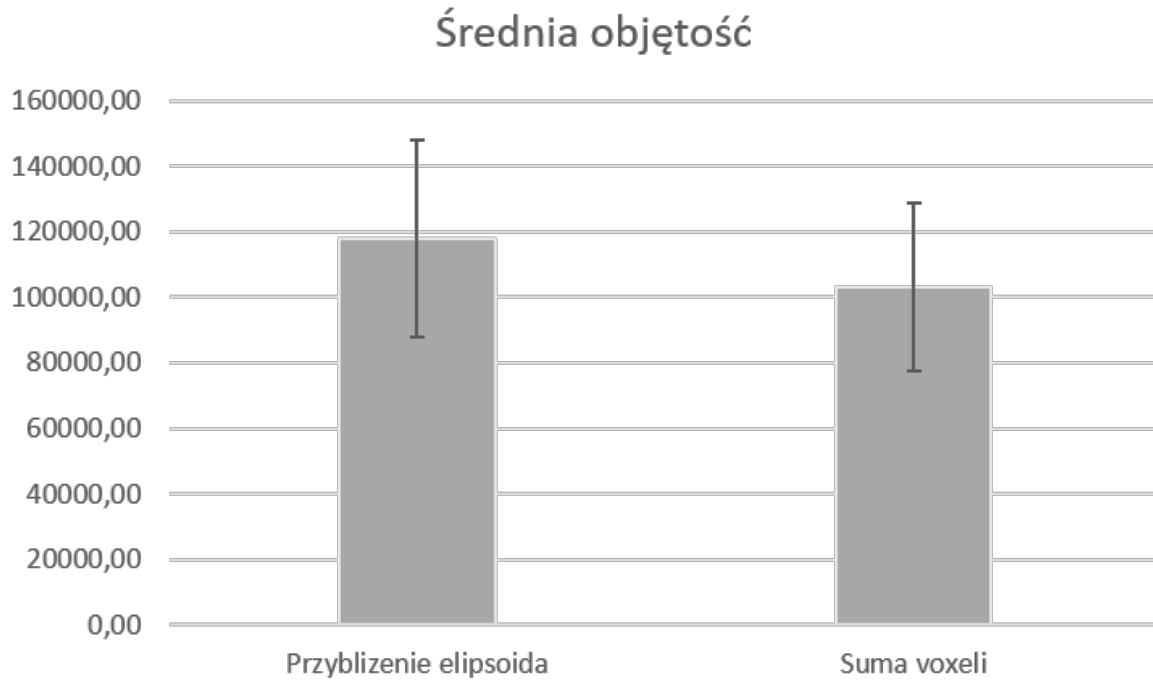
gdzie

n - liczba warstw pliku DICOM

m - liczba pixeli na warstwie

Ze wzoru wynika, że wartości atrybutów nie wpływają na poprawność wyniku, stąd w dalszej części opisu dla uproszczenia przyjmujemy parametry pochodzące z pliku DICOM jako stałe równe 1. Przy współczynnikach jednostkowych wartość objętości warstwy jest równa polu powierzchni maski dla danej warstwy. Stąd w dalszej części będziemy posługiwać się polem powierzchni jako czynnikiem determinującym objętość.

Aby zobrazować niedokładność aproksymacji elipsoidą postanowiliśmy porównać objętości za pomocą algorytmu aproksymującego oraz sumy objętości objętości kolejnych warstw voxelów na bazie masek określonych przez lekarzy.



Rysunek 5.7: Średnia objętość uzyskana przy obliczaniu objętości przybliżając elipsoidą oraz przy sumowaniu voxelu

Średnio różnica pomiędzy objętością obliczoną poprzez aproksymację elipsoidą daje wynik większy o  $14 \text{ cm}^3$  (około 15 punktowy wynik w skali procentowej względem sumy objętości voxelu), ochylenie standardowe różne o około  $3 \text{ cm}^3$ , dla tych samych masek. Ponieważ nie wykorzystujemy masek wyznaczonych przez lekarza tylko bazujemy na algorytmie segmentacji wymaga to dostosowania algorytmu do potrzeb tej pracy. W dalszej części przyjmiemy objętość wyznaczoną sumą jako wartość źródłową i będziemy oznaczać jako *źródło*.

### 5.2.1. Liczenie objętości

Po pierwsze proponujemy aby wszystkie obrazy nie były traktowane jako całość, a objętość całej prostaty jest liczona metodą przekrojów jako objętość kolejnych warstw nachodzących po sobie, a końcowa objętość jest sumą objętości.

$$V = \sum_{i=0}^{n-1} ((S_i + S_{i+1}) * (h_{d_{i+1}} - h_{d_i})) / 2$$

gdzie

$n$  - liczba warstw

$d_i$  - indeksy kolejnych warstw obrazu

$S_i$  - objętość kolejnych warstw obrazu

$h_{d_i}$  - lokalizacja warstwy o indeksie  $d_i$

Liczenie pola powierzchni maski została opisana w kolejnej sekcji.

W przypadku naszego systemu algorytm należy poprawić o niedokładność segmentacji prostaty. Zdarzają się sytuacje gdzie kolejna warstwa nie jest warstwą ostatnią na której znajduje się

## 5.2. MODUŁ OBLICZANIA OBJĘTOŚCI

prostata a algorytm nie znalazł na zdjęciu prostaty. W tym przypadku bierzemy kolejną następującą po niej niepustą warstwę.

### 5.2.2. Liczenie pola powierzchni

Z uwagi na niedokładności masek zwracanych przez algorytm segmentacji proponujemy 3 alternatywne funkcje poprawiające obraz:

1. Znalezienie otoczki wypukłej największego spójnego fragmentu maski. Alternatywnym podejściem do rozwiązania problemu jest algorytm znajdujący otoczkę wypukłą dla największego spójnego elementu. Jest rozwiązaniem gdzie wyszukujemy kształt o największym polu powierzchni, a następnie liczymy tylko jego pole pomijając miejsca gdzie również została znaleziona prostata. W tym przypadku pole powierzchni jest policzone dużo dokładniej, niż w pierwszym przypadku jednak gdy maska nie jest spójna mogą być mocno zaniżone.
2. Pole powierzchni prostokąta na którym można opisać maskę. Najprostszy algorytm, wzorujący się bezpośrednio na obecnym podejściu stosowanym w szpitalu. Jednak w tym przypadku każdą warstwę opisujemy innym prostokątem, przez co wyniki powinny być dokładniejsze
3. Znalezienie otoczki wypukłej dla wszystkich elementów maski. Jest to modyfikacja poprzedniego algorytmu, w której szukamy otoczki wypukłej dla wszystkich pixeli maski.

Następnie pole powierzchni liczone jest jako pole poligonu uzyskanego w procesie poprawiania obrazu. Poniżej przedstawiamy różnice wynikające z zastosowania każdego z algorytmów.

Przykład pierwszy, w którym wysegmentowana maska jest jednym spójnym elementem.

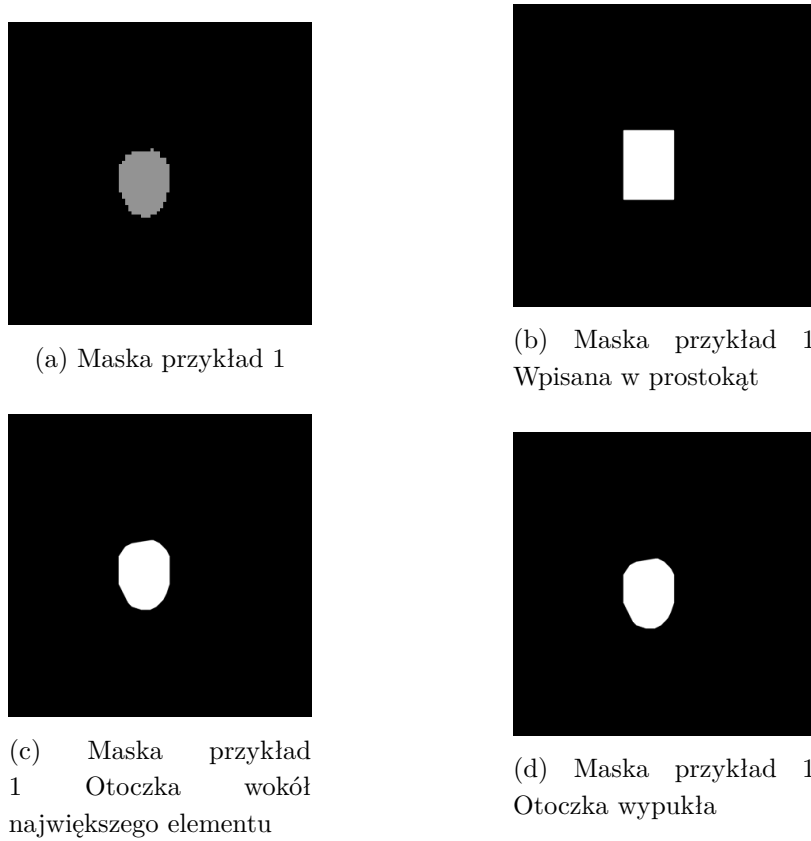
Algorytmy liczenia otoczek wypukłych dają ten sam rezultat w przypadku gdy maska jest spójna. Przykład obrazuje naddatek jaki pojawia się przy aproksymowaniu maski prostokątem. Prostata jest kształtem, który nie odpowiada, żadnemu kształtowi geometrycznemu, stąd przybliżanie go czworokątem jest nieefektywne.

Przykład drugi, w którym wysegmentowana maska nie jest spójna.

Widać dużą różnicę między algorytmami liczenia otoczek wypukłych. Ponieważ prostata jest znaleziona częściowo w kilku elementach obrazu, algorytm liczący otoczkę dla całości obrazu daje o wiele większy wynik niż liczący tylko dla największego spójnego fragmentu. Przykład obrazuje jak ważną rolę pełni algorytm segmentacji w procesie liczenia objętości.

### 5.2.3. Analiza wyników

Aby przeanalizować wyniki osiągnięte przez proponowany przez nas algorytm segmentacji oraz liczenia objętości postanowiliśmy porównać wyniki liczenia objętości algorytmem szpitalnym na danych konkursowych (segmentacja prostaty została wykonana przez lekarza) oraz proponowane przez nas metody liczenia objętości na maskach uzyskanych w przez aplikacyjny system segmentacji wraz z metodami poprawy obrazu. Zbiór testowy składał się z 13 pacjentów



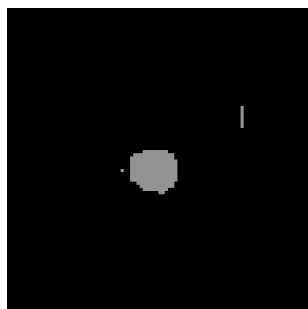
Rysunek 5.8: Przykłady wyników

pochodzących z danych konkursowych. Wartości uzyskane dla każdego z nich, oraz różnice pomiędzy wynikiem oraz wartością źródłową przedstawia tabela 5.1.

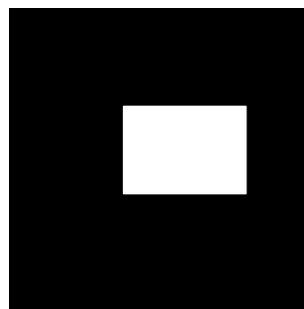
Tabela 5.1: Wartości objętości ( $cm^3$ )

Pacjent	Źródło	Otoczka element (1)	Wpisanie w prostokąt (2)	Otoczka cała (3)
Prostate3T-01-0001	47	21(26)	28(19)	23(24)
Prostate3T-01-0006	58	49(9)	68(-10)	55(3)
Prostate3T-01-0007	24	9(14)	24(-1)	18(6)
Prostate3T-01-0008	48	27(20)	44(3)	34(14)
Prostate3T-01-0009	129	81(48)	107(21)	86(42)
ProstateDx-02-0002	141	89(52)	638(-498)	248(-108)
ProstateDx-02-0003	185	145(39)	221(-37)	168(17)
ProstateDx-02-0004	176	91(84)	162(14)	116(60)
ProstateDx-02-0005	117	50(67)	75(42)	59(58)
ProstateDx-03-0002	67	98(-31)	137(-70)	111(-44)
ProstateDx-03-0003	133	66(67)	147(-14)	109(24)
ProstateDx-03-0004	81	124(-44)	265(-184)	192(-111)
ProstateDx-03-0005	125	113(12)	172(-47)	137(-13)

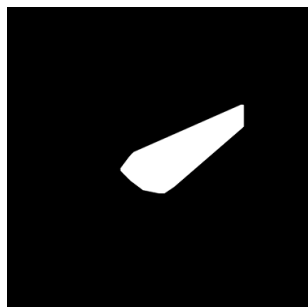
Rozbieżność pomiędzy wynikami przeprowadzonymi na maskach uzyskanych poprzez segmentację przez lekarza i przez algorytm różni się znacząco.



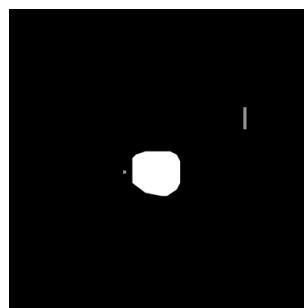
(a) Maska przykład 2



(b) Maska przykład 2  
Wpisana w prostokąt



(c) Maska przykład 2  
Otoczka wokół  
największego elementu

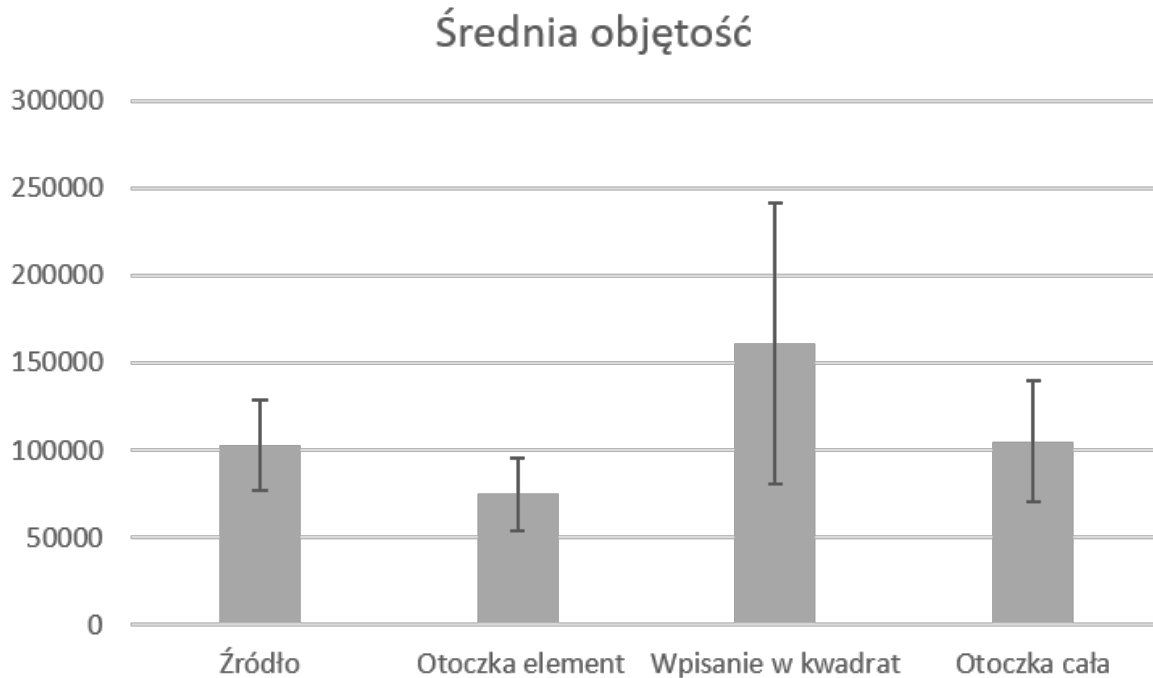


(d) Maska przykład 2  
Otoczka wypukła

Rysunek 5.9: Przykłady wyników

Największym problemem jaki zaobserwowaliśmy jest w przypadku gdy maska zostanie wykryta przez algorytm na warstwie, która znajduje się dużo przed faktycznym początkiem prostaty. Najbardziej jest to widoczne na przykładzie pacjenta *ProstateDx-02-0002*, dla którego wyniki uzyskany przez nasz algorytm są nawet dwukrotnie większe. Prostata znajduje się między 9 a 18 warstwą, jednak algorytm segmentacji wykrył obecność prostaty już na warstwie 4.

Dla przypadków, w których maski są znalezione na tych samych warstwach (to znaczy początek i koniec na obrazach o tym samym indeksie w przypadku danych konkursowych oraz algorytmu segmentacji), jak na przykład *Prostate3T-01-0007* uzyskujemy wyniki zbliżone do wartości źródłowej. Błąd tutaj wynika z niedokładności algorytmu segmentacji.



Rysunek 5.10: Porównanie metod liczenia

Tabela 5.2 przedstawia statystyki porównujące opisywane algorytmy z danymi źródłowymi. Jak widać najlepszą metodą przybliżenia jest otoczka wypukła wokół wszystkich elementów maski. Nie występuje w nim niedoszacowanie, które można zaobserwować w algorytmie znajdującym otoczę tylko wokół największego kształtu maski. Algorytm aproksymujący kształt maski prostokątem następnie sumujący voxele na każdej warstwie okazał się rozwiązaniem wysoce nieskutecznym z uwagi na duże przeszacowanie. Algorytm daje złe wyniki w szczególności gdy maska nie jest spójna. Wówczas występuje duże przeszacowanie.

Tablica 5.2: Analiza statystyczna algorytmów ( $cm^3$ )

Algorytm	Różnice średniej (%)	Różnice odchylenie (%)	Min	Max
Otoczka element (1)	28%	19%	9	145
Wpisanie w prostokąt (2)	-56%	-213%	24	638
Otoczka cała (3)	-2%	-34%	18	248

Z zaproponowanych algorytmów jedynym który osiągnął większą dokładność niż algorytm aproksymujący elipsoidą jest algorytm znajdujący otoczkę wypukłą dla całej maski. Pierwszy miał około 15 procentowe przeszacowania, natomiast drugi około 2% niedoszacowanie względem danych źródłowych. Pokazuje to, że istnieją lepsze algorytmy niż obecnie najczęściej stosowany.

#### 5.2.4. Alternatywne podejścia

Algorytmy opisane powyżej są jednymi z wielu możliwości poprawienia efektywności badania objętości prostaty w sposób analityczny. Jak widać w dużej mierze objętość opiera się o dokładność segmentacji. Wartymi do rozważenia modyfikacjami jest również

Pole powierzchni prostokąta na którym można opisać największy spójny element uzyskany w procesie segmentacji. Z uwagi na małe zróżnicowanie względem opisanych przez nas metod



### 5.3. MODUŁ BACKEDOWY

nie uwzględniliśmy jej w algorytmach udostępnianych przez naszą aplikację

Wybór elementów wokół których szukamy otoczki wypukłej na podstawie pewnej funkcji. Jest to zagadnienie skomplikowane, ponieważ możliwych funkcji jest bardzo dużo. Na przykład moglibyśmy szukać otoczki wokół wszystkich elementów maski znajdujących się w określonej odległości od środka największego elementu maski.

Przy poprawianiu maski uwzględniamy wszystkie warstwy. Czyli w przypadku gdy aktualna maska jest niespójna z poprzedzającą oraz następującą po sobie, wówczas zamiast niej bierzemy maskę będącą średnią dwóch wymienionych.

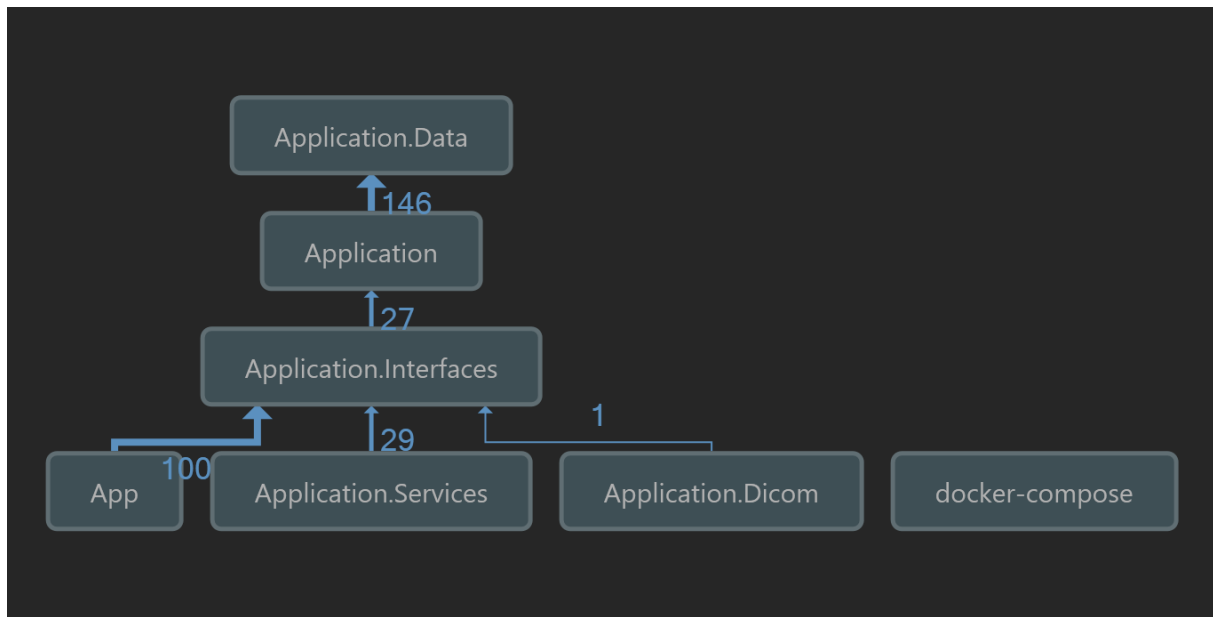
### 5.3. Moduł backedowy

Moduł zaimplementowany w .NET Framework odpowiada za komunikację z warstwą prezentacji, komunikację między resztą serwisów oraz dostępem do bazy danych. Podstawowe funkcje

1. Udostępnia API do komunikacji z naszym modułem frontendowym oraz umożliwiającą integrację z dowolnym innym systemem, który mógłby wykorzystywać nasze API. Interfejs zapewnia pełen dostęp do tabel czyli wszystkie operacje *CRUD* jakie mogą być potrzebne przy rozwoju systemu. Dodatkowo daje zapewnia komunikację użytkownika ze wszystkimi kontenerami. W tym przypadku zakres operacji jest ograniczony do minimum.
2. Obsługę bazy danych. W tym obsługę dodawania nowych pacjentów z plików oraz systemu PACS, migrację bazy danych oraz obsługę kwerend i procedur.
3. Komunikację z pozostałymi komponentami backendowymi.

Staraliśmy się w pracy zastosować wszystkie najlepsze praktyki wykorzystywane przy projektowaniu oprogramowania takie jak zasady *SOLID* oraz wzorce projektowe jak *Dependency Injection*. Za spełnienie tego ostatniego w aplikacji odpowiada kontener Autofac.

### 5.3.1. Diagram zależności



Rysunek 5.11: Diagram zależności

Jak widać podstawowe projekty w systemie to

1. Application.Data - który jest warstwą dostępu do danych w aplikacji, definiuje encje oraz zarządza połączeniem z bazą danych.
2. Application - definiuje podstawowe obiekty współdzielone między modułami takie jak modele, wyjątki, logowanie, mapowanie obiektów.
3. Application.Interfaces - projekt definiuje wszystkie kontrakty za pomocą których klasy komunikują się ze sobą
4. Application.Services - zawiera implementację interfejsów zdefiniowanych w projekcie *Interfaces*.
5. Application.Dicom - odpowiada za obsługę plików DICOM. Moglibyśmy zaimplementować go w projekcie *Services* jednak z uwagi na złożoność zdecydowaliśmy umieścić go w osobnym module.

Projektując architekturę rozwiązania zależało nam aby komponenty były ze sobą luźno powiązane, przez co unikniemy niepotrzebnych nad którymi trudno zapanować. Moduły podstawowe są zależne tylko od projektu *Interfaces* co zapewnia oczekiwane luźne łączenie i komunikację pomiędzy nimi. Warto zauważyć, że projekt który udostępnia nasze REST API jest zależny tylko od projektu *Interfaces*.

### 5.3.2. Baza danych

Plik wejściowy jest plikiem binarnym zawierającym wiele informacji o pacjencie, w przypadku aplikacji, którą tworzymy przy każdym odwołaniu do kolejnych warstw przetwarzanie pliku jest bardzo nieefektywne. Dlatego postanowiliśmy plik przetwarzać tylko raz przy dodawaniu, a przy odwołaniach zwracać dane z bazy danych.

### 5.3. MODUŁ BACKEDOWY

Ponieważ według standardu DICOM [3] zawiera ponad 4 tysiące tagów. Po przeanalizowaniu plików wejściowych zauważyliśmy, że w naszym przypadku uzupełniona jest tylko część danych.

W bazie danych przechowujemy wybrane dane istotne z punktu widzenia działania aplikacji oraz potrzeb lekarza. Wybraliśmy strukturę gdzie dla jednego pacjenta informację na temat obrazów oraz danych są wspólne i nie zmieniają się dla kolejnych warstw obrazu. Dlatego jedyne informacje dotyczące obrazu są przechowywane dla każdego pliku osobno. Poniżej zamieszczono diagram encji jakie znajdują się w bazie danych.



Rysunek 5.12: Schemat bazy danych

## 5.4. Moduł prezentacji

Aplikacja jest stworzona w architekturze SPA (Single Page Application). Wykorzystanie tego podejścia pozwala na wyświetlanie wielu widoków bez konieczności pobierania ponownie tych samych informacji, dzięki temu zachowując płynne działanie strony internetowej. Stosując te podejście zyskujemy na płynności działania oraz lepsze odczucia użytkownika przy korzystaniu z serwisu.

Naszą aplikację stworzyliśmy przy pomocy biblioteki React. Ułatwia ona myślenie o logice biznesowej aplikacji dzieląc stronę na komponenty. Każdy komponent składa się z co najmniej jednego HTML tagu i jest odpowiedzialny za inną część widoczną na stronie. Granulacja komponentów sprzyja pisaniu złożonych aplikacji poprzez separację stanów każdego z komponentów. Naszą aplikację stworzyliśmy przy pomocy biblioteki React. Ułatwia ona myślenie o logice biznesowej aplikacji dzieląc stronę na komponenty. Każdy komponent składa się z co najmniej jednego HTML tagu i jest odpowiedzialny za inną część widoczną na stronie. Granulacja komponentów sprzyja pisaniu złożonych aplikacji poprzez separację stanów każdego z komponentów.

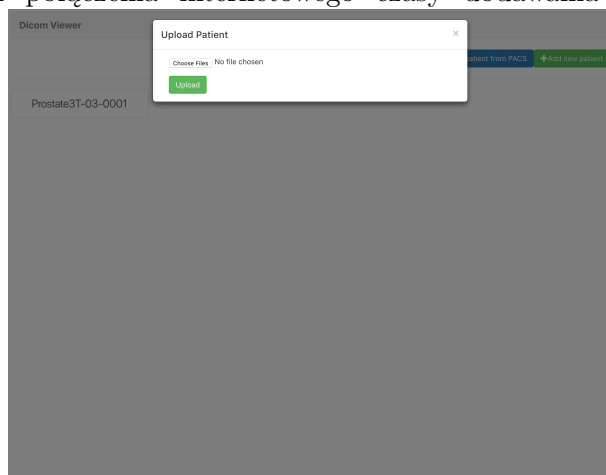
### 5.4.1. Widok startowy

Pierwszy ekran jaki jest wyświetlany w naszej spełnia trzy podstawowe funkcjonalności.

1. Wyświetlenie listy pacjentów dostępnych w aplikacji.
2. Dodanie nowego użytkownika do bazy danych.
3. Integracja z systemem PACS.

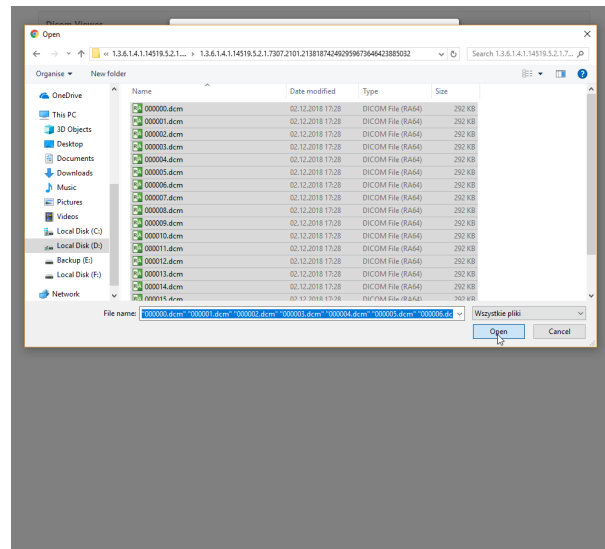
Aby dodać nowego pacjenta, należy nacisnąć przycisk *Add New Patient*. Pojawi się nowe okno w którym użytkownik powinien wybrać wszystkie pliki z danymi pacjenta, następnie nacisnąć *Upload*. Po udanym procesie należy zamknąć okienko.

Typowym scenariuszem jest dodawanie pacjenta wraz ze wszystkimi obrazami jakie mamy na jego temat. Pliki średnio ważą około 10 MB, w zależności od ilości warstw obrazu oraz szybkości połączenia internetowego czasu dodawania mogą być wydłużone.

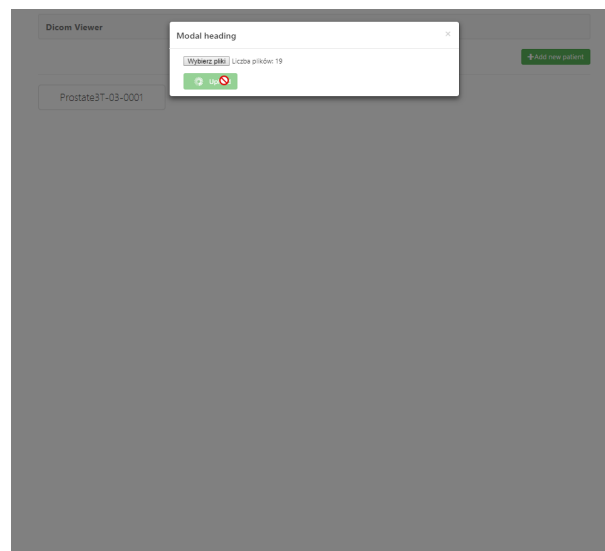


Rysunek 5.13: Dodawanie pacjenta

## 5.4. MODUŁ PREZENTACJI



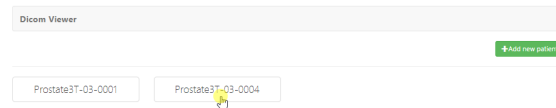
Rysunek 5.14: Wybór plików



Rysunek 5.15: Proces zakończony

### 5.4.2. Widok pacjenta

Lista pacjentów wyświetla ich identyfikatory w systemie. Aby przejść do widoku wyświetlającego wszystkie informacje o pacjencie należy nacisnąć na wybranego pacjenta.



Rysunek 5.16: Przejście do widoku pacjenta

Widok pacjenta przedstawia nam wszystkie informacje jakie posiadamy na temat danego przypadku badanego. W szczególności:

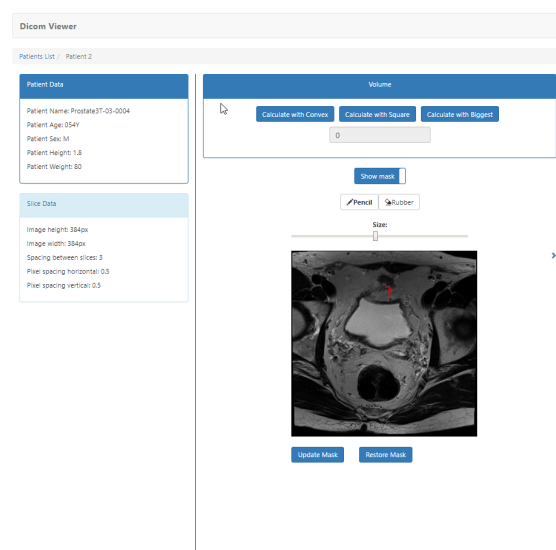
Dane pacjenta z pliku DICOM, takie jak: nazwa, wiek, płeć, wzrost, waga oraz inne.

Wyświetlamy tylko informacje, które w danym pliku się znajdowały. Jeżeli jakieś dane miały wartość pustą pomijamy je. Jeżeli dane podlegały wcześniejszej anonimizacji dane wrażliwe jakie zobaczymy na interfejsie użytkownika nie będą danymi odpowiadającymi tym z systemu szpitalnego.

Informacje na temat obrazu, takie jak: wymiary obrazu (szerokość i wysokość), odległość między kolejnymi obrazami w badaniu, odległości między konkretnymi pixelami na obrazie.

Informacje na temat objętości oraz przyciski do kalkulacji objętości według wybranego algorytmu.

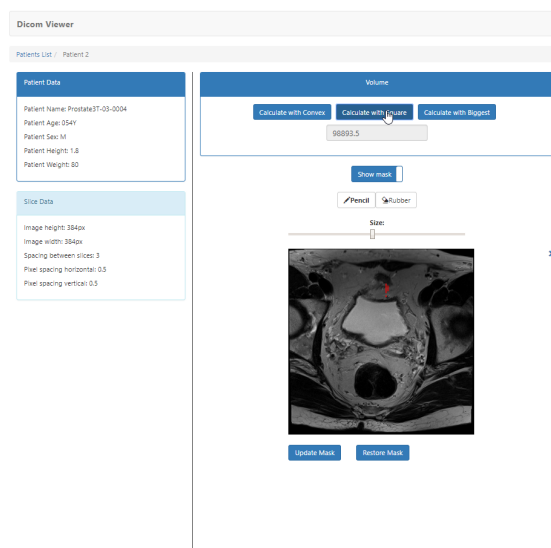
Obraz z rezonansu magnetycznego oraz maskę znaną w procesie segmentacji wraz z możliwością edycji.



Rysunek 5.17: Widok pacjenta

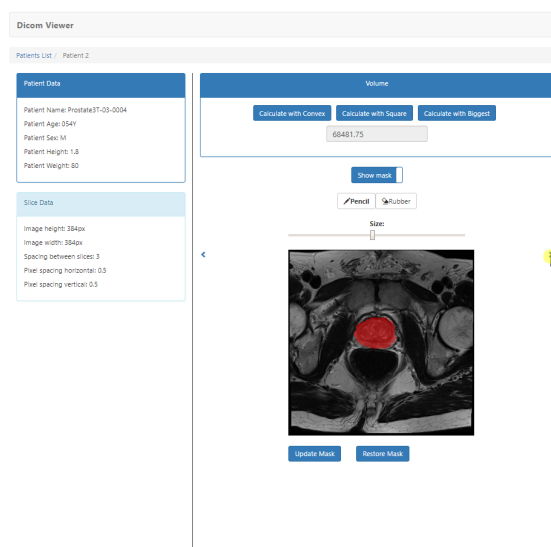
## 5.4. MODUŁ PREZENTACJI

Aby obliczyć objętość na podstawie masek znajdujących się obecnie w bazie danych należy nacisnąć odpowiedni przycisk znajdujący się na górze strony. Udostępniamy trzy alternatywne metody opisane w poprzednim rozdziale. Po naciśnięciu przycisku objętość jest obliczana oraz automatycznie aktualizowana na stronie tak jak i w bazie danych.



Rysunek 5.18: Kalkulacja objętości

Aby przełączać się pomiędzy poszczególnymi warstwami należy nacisnąć przycisk po prawej lub lewej stronie obrazu.



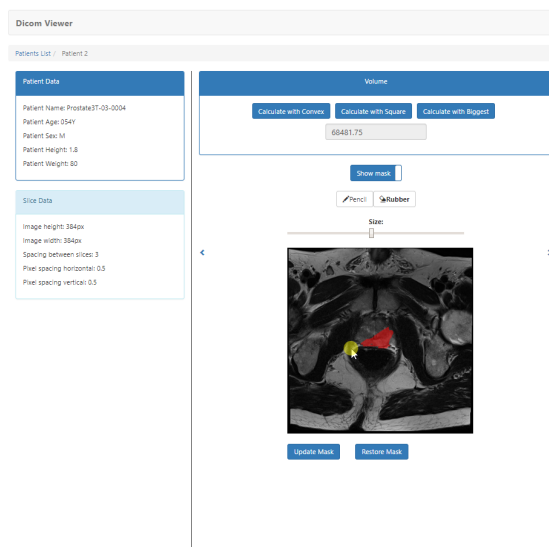
Rysunek 5.19: Przechodzenie pomiędzy kolejnymi obrazami

### 5.4.3. Edycja obrazu

Zapewniamy możliwość edycji masek obliczonych w procesie segmentacji. Udostępniamy opcje rysowania *Pencil* oraz usuwania *Rubber*. Po wybraniu narzędzia użytkownik może rysować po obrazie.



Rysunek 5.20: Rysowanie maski



Rysunek 5.21: Usuwanie maski

Po zakończeniu edycji należy nacisnąć przycisk *Update mask*.



## 5.4. MODUŁ PREZENTACJI



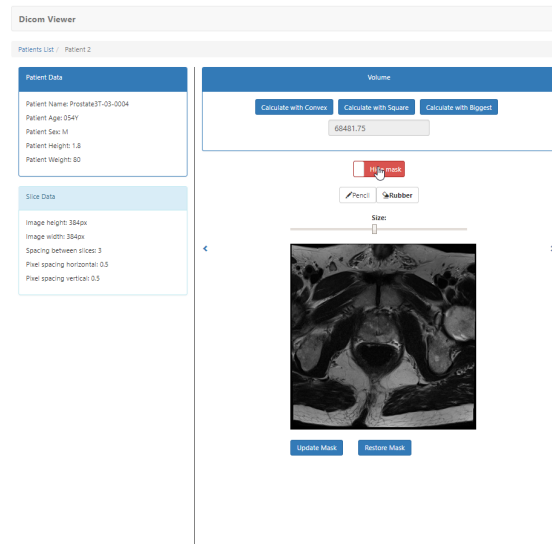
Rysunek 5.22: Aktualizacja mask

W przypadku gdy chcemy przywrócić pierwotną maskę znalezioną przez nasz algorytm segmentacji należy wybrać opcję *Restore mask*



Rysunek 5.23: Resetowanie maski

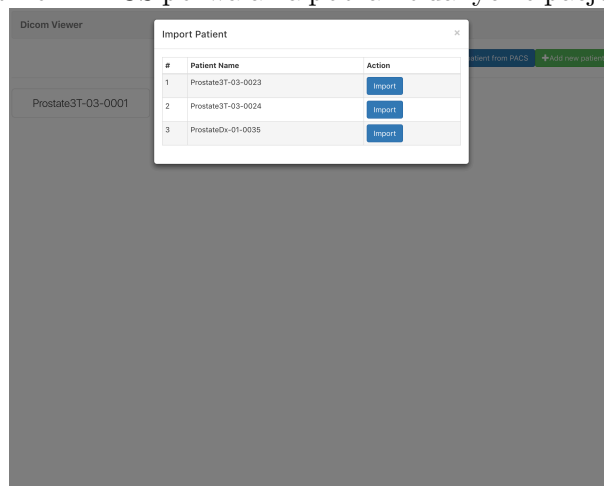
Dodatkowo zapewniamy możliwość ukrycia maski w przypadku aby lekarz mógł przeglądać oryginał zdjęcia bez przysłaniania go maską.



Rysunek 5.24: Ukryj maskę

#### 5.4.4. Integracja z systemem PACS

Przycisk Import patient from PACS pozwala na pobranie danych o pacjencie z systemu PACS.



Rysunek 5.25: Import z systemu PACS

Połączenie z systemem jest konfigurowalne w plikach konfiguracyjnych aplikacji fronowej. Import polega na pobraniu zanonimizowanych danych z serwera PACS i załadowaniu ich do naszej bazy danych.

## 6. Podsumowanie rezultatów

Rozdział jest poświęcony przedstawieniu rezultatów pracy nad omawianym systemem. Omówione zostaną testy, które zostały przeprowadzone w celu zbadania, czy stworzone rozwiązanie odpowiada założeniom, przedstawionym w rozdziale *Specyfikacja*.

Głównym założeniem stworzenia aplikacji było wspieranie pracy lekarza poprzez zapewnienie systemu, który zapewniałby informację o predycji objętości prostaty z obrazów DICOM. Informacja o wielkości gruczołu ma dać lekarzowi szybką informację o potencjalnym wystąpieniu nowotworu prostaty. Cele zostały osiągnięte poprzez zaprojektowanie aplikacji, która daje możliwość przeglądania, segmentacji oraz liczenia objętości prostaty z różnych źródeł jakimi są pliki płaskie oraz szpitalny system PACS. Dodatkowo zapewniliśmy możliwość poprawy wyliczenia objętości poprzez ręczną możliwość edycji segmentacji.

### 6.1. Integracja z systemem informatycznym

Podstawowym wymaganiem dotyczącym naszej pracy jest możliwość integracji i używania go razem z obecnymi w szpitalu systemami. Podczas naszej współpracy ze szpitalem niestety doszliśmy do wniosku, że nie jest możliwa integracja z systemem używanym na co dzień w szpitalu. System ten jest napisany przez firmę Philips i nie ma w nim możliwości instalacji wtyczek, a cały kod źródłowy jest utajniony. Przez co w naszym systemie proponujemy obejście tego problemu poprzez łączenie się z siecią szpitalną przez przeglądarkę i w ten sposób pobieranie danych. Rozwiązanie umożliwiające łatwą integrację z infrastrukturą szpitalną bez rozszerzania funkcjonalności przeglądarek DICOM. Alternatywnie można podłączyć serwer do szpitalnego systemu PACS przy użyciu sieci VPN. Rozwiązanie przy użyciu klienta po stronie przeglądarki ma kilka zalet, takich jak na przykład redukcję kosztów po stronie obsługi IT i umożliwia wprowadzanie zmian po stronie klienta bez konieczności manualnej instalacji oprogramowania. Co ma istotne znaczenie w używanych systemach szpitalnych.

Aplikacja powstała przy wykorzystaniu najnowszych dostępnych technologii stąd w środowisku produkcyjnym (w szczególności w systemie szpitalnym) mogą pojawiać się problemy, wynikające z różnic pomiędzy naszym środowiskiem deweloperskim a szpitalną infrastrukturą IT. Wszystkie biblioteki jakich użyliśmy w pracy są stale rozwijane i zapewniają dobre wsparcie dla deweloperów.

### 6.2. Trudności

W trakcie realizacji projektu napotkaliśmy kilka trudności. Największymi z nich były:

1. Problemy z integracją z siecią szpitalną. Wniosek jaki powstał po etapie analizy biznesowej, to, że uzyskanie odpowiednich uprawnień oraz dostępów do celu realizacji projektu jest niemożliwe. Stąd w naszej pracy integrację wykonaliśmy z lokalną instancją bazy PACS. Może to wprowadzać problemy przy realnej próbie integracji z systemem szpitalnym, gdyby okazało się, że systemy różnią się między sobą metodami dostępu w szczególności metodami udostępnianymi przez REST API.
2. Największym problemem związanym z Pozyskanie danych szpitalnych był fakt, iż każdy system używa innego formatowania dokumentów, więc często występowały trudności z odczytaniem istotnych w naszym badaniu informacji.
3. Dane testowe jakie użyliśmy do uczenia sieci pochodzą z konkursu i zbiór testowy jest bardzo ograniczony. Może to powodować problemy z działaniem na realnych danych szpitalnych.
4. Nie byliśmy w stanie sprawdzić dokładności algorytmu segmentacji z danymi realnych pacjentów szpitala, z uwagi na brak dostępu do danych. W analizie wyników brakowało nam najdokładniejszej informacji jaką jest objętość prostaty po zabiegu prostatektomii. Stąd nie jesteśmy w stanie zweryfikować wyników względem faktycznych wartości.

Trudności spowodowały rozwiązywanie problemów w alternatywne sposoby niż początkowo zakładaliśmy. Jednak aby dopuścić rozwiązanie do użytku produkcyjnego powinny być rozwiązane.

### 6.3. Wnioski

W trakcie realizacji pracy pojawiły się problemy, które zmusiły nas do zmiany początkowego podejścia. Finalna wersja spełnia wszystkie założenia przyjęte na początku projektu. Przedstawiony system może poprawić komfort pracy lekarza radiologa oraz zapewnić istotne informacje przy wystawianiu diagnozy pacjenta poprzez informację o przybliżonej wartości objętości gruczołu prostaty. Z uwagi na dużą złożoność systemu jest wiele możliwości wprowadzania usprawnień.

## Bibliografia

- [1] *Strona internetowa Rynek Zdrowia*, <http://www.rynekzdrowia.pl/Serwis-Onkologia/Eksperci-wzrasta-z>
- [2] *Dane konkursowe*, <https://wiki.cancerimagingarchive.net/plugins/servlet/contentId=23691656/#content>
- [3] *Standard DCIOM*, <https://www.dicomstandard.org/>
- [4] *Open CV* <https://opencv.org/>
- [5] *Przetwarzanie danych w medycynie* <http://www.image-net.org/>
- [6] *Dane do uczenia sieci* <https://wiki.cancerimagingarchive.net/display/Public/NCI-ISBI+2013+Challenge>
- [7] *Zgłoszenie do konkursu*, [https://github.com/mirzaevinom/prostate\\_segmentation](https://github.com/mirzaevinom/prostate_segmentation)
- [8] *Serwer PACS* <https://www.orthanc-server.com/>
- [9] *Aplikacje dostarczane przez firmę Philips* <https://www.philips.com.eg/healthcare/resources/support-documents>
- [10] *Future Processing* <https://www.future-processing.com/software-solutions/healthcare/>
- [11] *norma PSA* <https://www.webmd.com/prostate-cancer/guide/psa#1>
- [12] *norma PSAD jako dokładniejsza norma* [www.researchgate.net/publication/40688578\\_Does\\_prostate\\_s](http://www.researchgate.net/publication/40688578_Does_prostate_s)

## Wykaz symboli i skrótów

HTTP	Hypertext Transfer Protocol
IDE	Integrated Development Environment
np.	Na przykład
tzw.	Tak zwany

## Spis rysunków

2.1	Korelacja między objętością z MR i objętością z USG . . . . .	21
2.2	PSA względem wieku . . . . .	22
2.3	PSA względem Gleasona . . . . .	22
2.4	PSAD względem Gleasona . . . . .	23
2.5	PSA względem PIRADS . . . . .	23
3.1	Diagram przypadków użycia . . . . .	27
4.1	Schemat architektury aplikacji . . . . .	33
5.1	Architektura sieci U-Net . . . . .	36
5.2	Zaznaczone regiony prostaty na zdjęciu z badania MR . . . . .	37
5.3	Przykładowy obraz z badania MRI . . . . .	38
5.4	Przykładowy obraz z badania MRI . . . . .	38
5.5	Architektura sieci . . . . .	40
5.6	Przykłady wyników . . . . .	42
5.7	Średnia objętość uzyskana przy obliczaniu objętości przybliżając elipsoidą oraz przy sumowaniu voxelu . . . . .	44
5.8	Przykłady wyników . . . . .	46
5.9	Przykłady wyników . . . . .	47
5.10	Porównanie metod liczenia . . . . .	48
5.11	Diagram zależności . . . . .	50
5.12	Schemat bazy danych . . . . .	51
5.13	Dodawanie pacjenta . . . . .	52
5.14	Wybór plików . . . . .	53
5.15	Proces zakończony . . . . .	53
5.16	Przejsie do widoku pacjenta . . . . .	54
5.17	Widok pacjenta . . . . .	54
5.18	Kalkulacja objętości . . . . .	55
5.19	Przechodzenie pomiędzy kolejnymi obrazami . . . . .	55
5.20	Rysowanie maski . . . . .	56
5.21	Usuwanie maski . . . . .	56
5.22	Aktualizacja mask . . . . .	57
5.23	Resetowanie maski . . . . .	57
5.24	Ukryj maskę . . . . .	58
5.25	Import z systemu PACS . . . . .	58

# Spis tabel

2.1	Dane szpitalne . . . . .	20
2.2	Podstawowe statystyki . . . . .	21
3.1	Przypadki użycia . . . . .	28
3.2	Wymagania niefunkcjonalne . . . . .	29
5.1	Wartości objętości ( $cm^3$ ) . . . . .	46
5.2	Analiza statystyczna algorytmów ( $cm^3$ ) . . . . .	48



## Spis załączników

1. Załącznik 1 - Opis API

#### 6.4. Załącznik 1 - Opis API

---

# API

Prostate segmentation rest API

---

## api

Folder for api

---

### GET http:///api/Dicom

http:///api/Dicom

#### HEADERS

---

##### **Accept**

text/plain, application/json, text/json

#### Example Request

http:///api/Dicom

```
curl --request GET \  
  --url http:/api/Dicom \  
  --header 'Accept: text/plain, application/json, text/json'
```

---

### POST http:///api/Dicom

http:///api/Dicom

#### HEADERS

---

##### **Content-Type**

application/json-patch+json

#### Example Request

http:///api/Dicom

```
curl --request POST \  
  --url http:/api/Dicom \  
  --header 'Content-Type: application/json-patch+json'
```

---

## GET http:///api/Dicom/:id

http:///api/Dicom/:id

#### HEADERS

##### Accept

text/plain, application/json, text/json

#### PATH VARIABLES

##### id

{{id}}

#### Example Request

http:///api/Dicom/:id

```
curl --request GET \  
  --url http:/api/Dicom/:id \  
  --header 'Accept: text/plain, application/json, text/json'
```

---

## PUT http:///api/Dicom/:id

http:///api/Dicom/:id

#### HEADERS

## Content-Type

application/json-patch+json

## PATH VARIABLES

---

**id**

{{id}}

## Example Request

```
http:///api/Dicom/:id
```

```
curl --request PUT \  
  --url http:/api/Dicom/:id \  
  --header 'Content-Type: application/json-patch+json'
```

---

## DEL http:///api/Dicom/:id

```
http:///api/Dicom/:id
```

## PATH VARIABLES

---

**id**

{{id}}

## Example Request

```
http:///api/Dicom/:id
```

```
curl --request DELETE \  
  --url http:/api/Dicom/:id
```

---

## GET http:///api/Dicom/Indexes/:id

```
http:///api/Dicom/Indexes/:id
```

## HEADERS

---

**Accept**

text/plain, application/json, text/json

PATH VARIABLES

---

**id**

{{id}}

## Example Request

```
http:///api/Dicom/Indexes/:id
```

```
curl --request GET \  
  --url http://api/Dicom/Indexes/:id \  
  --header 'Accept: text/plain, application/json, text/json'
```

**GET** http:///api/Dicom/File/:id

```
http:///api/Dicom/File/:id
```

HEADERS

---

**Accept**

text/plain, application/json, text/json

PATH VARIABLES

---

**id**

{{id}}

## Example Request

```
http:///api/Dicom/File/:id
```

```
curl --request GET \  
  --url http://api/Dicom/File/:id \  
  --header 'Accept: text/plain, application/json, text/json'
```

**GET** http:///api/Image/:dicomId

---

http:///api/Image/:dicomId

## HEADERS

---

### Accept

text/plain, application/json, text/json

## PATH VARIABLES

---

### dicomId

{{dicomId}}

## Example Request

http:///api/Image/:dicomId

```
curl --request GET \  
  --url http://api/Image/:dicomId \  
  --header 'Accept: text/plain, application/json, text/json'
```

## GET http:///api/Image/:dicomId&:sliceId

http:///api/Image/:dicomId&:sliceId

## HEADERS

---

### Accept

text/plain, application/json, text/json

## PATH VARIABLES

---

### dicomId

{{dicomId}}

### sliceId

{{sliceId}}

## Example Request

http:///api/Image/:dicomId&:sliceId

```
curl --request GET \  
  --url 'http://api/Image/:dicomId&:sliceId' \  
  --header 'Accept: text/plain, application/json, text/json'
```

---

## PUT http:///api/Image/:dicomId&:sliceId

http:///api/Image/:dicomId&:sliceId

---

### HEADERS

#### Content-Type

application/json-patch+json

---

### PATH VARIABLES

#### dicomId

{{dicomId}}

#### sliceId

{{sliceId}}

---

### Example Request

http:///api/Image/:dicomId&:sliceId

```
curl --request PUT \  
  --url 'http://api/Image/:dicomId&:sliceId' \  
  --header 'Content-Type: application/json-patch+json'
```

---

## DEL http:///api/Image/:dicomId&:sliceId

http:///api/Image/:dicomId&:sliceId

---

### PATH VARIABLES

#### dicomId

{{dicomId}}

#### sliceId

{{sliceId}}



#### Example Request

```
http://api/Image/:dicomId&:sliceId
```

```
curl --request DELETE \  
  --url 'http://api/Image/:dicomId:sliceId'
```

---

## GET http:///api/Mask/:dicomId

```
http:///api/Mask/:dicomId
```

---

#### HEADERS

##### **Accept**

text/plain, application/json, text/json

---

#### PATH VARIABLES

##### **dicomId**

{{dicomId}}

#### Example Request

```
http:///api/Mask/:dicomId
```

```
curl --request GET \  
  --url http://api/Mask/:dicomId \  
  --header 'Accept: text/plain, application/json, text/json'
```

---

## GET http:///api/Mask/:dicomId&:sliceId

```
http:///api/Mask/:dicomId&:sliceId
```

---

#### HEADERS

##### **Accept**

text/plain, application/json, text/json

## PATH VARIABLES

---

### **dicomId**

{{dicomId}}

### **sliceId**

{{sliceId}}

## Example Request

```
http:///api/Mask/:dicomId&:sliceId
```

```
curl --request GET \  
  --url 'http://api/Mask/:dicomId&:sliceId' \  
  --header 'Accept: text/plain, application/json, text/json'
```

---

## PUT http:///api/Mask/:dicomId&:sliceId

```
http:///api/Mask/:dicomId&:sliceId
```

## HEADERS

---

### **Content-Type**

application/json-patch+json

## PATH VARIABLES

---

### **dicomId**

{{dicomId}}

### **sliceId**

{{sliceId}}

## Example Request

```
http:///api/Mask/:dicomId&:sliceId
```

```
curl --request PUT \  
  --url 'http://api/Mask/:dicomId&:sliceId' \  
  --header 'Content-Type: application/json-patch+json'
```

---

## DEL http:///api/Mask/:dicomId&:sliceId

```
http:///api/Mask/:dicomId&:sliceId
```

### PATH VARIABLES

---

#### **dicomId**

{{dicomId}}

#### **sliceId**

{{sliceId}}

### Example Request

```
http:///api/Mask/:dicomId&:sliceId
```

```
curl --request DELETE \  
  --url 'http:/api/Mask/:dicomId:sliceId'
```

## POST http:///api/Mask/Recalculate/:dicomId&:sliceId

```
http:///api/Mask/Recalculate/:dicomId&:sliceId
```

### PATH VARIABLES

---

#### **dicomId**

{{dicomId}}

#### **sliceId**

{{sliceId}}

### Example Request

```
http:///api/Mask/Recalculate/:dicomId&:sliceId
```

```
curl --request POST \  
  --url 'http:/api/Mask/Recalculate/:dicomId:sliceId'
```

## POST http:///api/NewDicom

```
http:///api/NewDicom
```

### HEADERS

---

#### Accept

text/plain, application/json, text/json

#### Content-Type

application/json-patch+json

### Example Request

```
http:///api/NewDicom
```

```
curl --request POST \  
  --url http:/api/NewDicom \  
  --header 'Accept: text/plain, application/json, text/json' \  
  --header 'Content-Type: application/json-patch+json'
```

## POST http:///api/NewDicom/:id

```
http:///api/NewDicom/:id
```

### HEADERS

---

#### Content-Type

application/json-patch+json

### PATH VARIABLES

---

#### id

{{id}}

### Example Request

```
http:///api/NewDicom/:id
```

```
curl --request POST \  
  --url http://api/NewDicom/:id \  
  --header 'Content-Type: application/json-patch+json'
```

## POST http:///api/NewDicom/UploadList

http:///api/NewDicom/UploadList

### HEADERS

#### Accept

text/plain, application/json, text/json

#### Content-Type

application/json-patch+json

### Example Request

http:///api/NewDicom/UploadList

```
curl --request POST \  
  --url http://api/NewDicom/UploadList \  
  --header 'Accept: text/plain, application/json, text/json' \  
  --header 'Content-Type: application/json-patch+json'
```

## GET http:///api/PatientData

http:///api/PatientData

### HEADERS

#### Accept

text/plain, application/json, text/json

### Example Request

http:///api/PatientData

```
curl --request GET \  
  --url http://api/PatientData \  
  --header 'Accept: text/plain, application/json, text/json'
```

## GET http:///api/PatientData/:dicomId

http:///api/PatientData/:dicomId

### HEADERS

#### Accept

text/plain, application/json, text/json

### PATH VARIABLES

#### dicomId

{{dicomId}}

### Example Request

http:///api/PatientData/:dicomId

```
curl --request GET \  
  --url http://api/PatientData/:dicomId \  
  --header 'Accept: text/plain, application/json, text/json'
```

## PUT http:///api/PatientData/:dicomId

http:///api/PatientData/:dicomId

### HEADERS

#### Content-Type

application/json-patch+json

### PATH VARIABLES

#### dicomId

{{dicomId}}

#### Example Request

```
http:///api/PatientData/:dicomId
```

```
curl --request PUT \  
  --url http://api/PatientData/:dicomId \  
  --header 'Content-Type: application/json-patch+json'
```

---

## POST http:///api/PatientData/:dicomId

```
http:///api/PatientData/:dicomId
```

#### HEADERS

---

##### **Content-Type**

application/json-patch+json

#### PATH VARIABLES

---

##### **dicomId**

{{dicomId}}

#### Example Request

```
http:///api/PatientData/:dicomId
```

```
curl --request POST \  
  --url http://api/PatientData/:dicomId \  
  --header 'Content-Type: application/json-patch+json'
```

---

## DEL http:///api/PatientData/:dicomId

```
http:///api/PatientData/:dicomId
```

#### PATH VARIABLES

---

##### **dicomId**

{{dicomId}}

#### Example Request

```
http://api/PatientData/:dicomId
```

```
curl --request DELETE \  
  --url http://api/PatientData/:dicomId
```

---

## GET http:///api/Slice/:dicomId

```
http:///api/Slice/:dicomId
```

---

#### HEADERS

##### **Accept**

text/plain, application/json, text/json

---

#### PATH VARIABLES

##### **dicomId**

{{dicomId}}

#### Example Request

```
http:///api/Slice/:dicomId
```

```
curl --request GET \  
  --url http://api/Slice/:dicomId \  
  --header 'Accept: text/plain, application/json, text/json'
```

---

## GET http:///api/Slice/:dicomId&:sliceId

```
http:///api/Slice/:dicomId&:sliceId
```

---

#### HEADERS

##### **Accept**

text/plain, application/json, text/json

---

#### PATH VARIABLES



**dicomId**

{{dicomId}}

**sliceId**

{{sliceId}}

## Example Request

```
http:///api/Slice/:dicomId&:sliceId
```

```
curl --request GET \  
  --url 'http://api/Slice/:dicomId&:sliceId' \  
  --header 'Accept: text/plain, application/json, text/json'
```

**POST** http:///api/Slice/:dicomId&:sliceId

```
http:///api/Slice/:dicomId&:sliceId
```

## HEADERS

**Content-Type**

application/json-patch+json

## PATH VARIABLES

**dicomId**

{{dicomId}}

**sliceId**

{{sliceId}}

## Example Request

```
http:///api/Slice/:dicomId&:sliceId
```

```
curl --request POST \  
  --url 'http://api/Slice/:dicomId&:sliceId' \  
  --header 'Content-Type: application/json-patch+json'
```

---

## DEL http:///api/Slice/:dicomId&:sliceId

http:///api/Slice/:dicomId&:sliceId

---

### PATH VARIABLES

#### **dicomId**

{{dicomId}}

#### **sliceId**

{{sliceId}}

---

### Example Request

http:///api/Slice/:dicomId&:sliceId

```
curl --request DELETE \  
  --url 'http://api/Slice/:dicomId&:sliceId'
```

---

## PUT http:///api/Slice/:id

http:///api/Slice/:id

---

### HEADERS

#### **Content-Type**

application/json-patch+json

---

### PATH VARIABLES

#### **id**

{{id}}

---

### Example Request

http:///api/Slice/:id

```
curl --request PUT \  
  --url http://api/Slice/:id \  
  --header 'Content-Type: application/json-patch+json'
```

**GET** http:///api/Volume/Calculate/:id?dicomId={{dicomId}}

http:///api/Volume/Calculate/:id?dicomId={{dicomId}}

#### HEADERS

##### Accept

text/plain, application/json, text/json

#### PARAMS

##### dicomId

{{dicomId}}

#### PATH VARIABLES

##### id

{{id}}

#### Example Request

http:///api/Volume/Calculate/:id?dicomId={{dicomId}}

```
curl --request GET \  
  --url 'http://api/Volume/Calculate/:id?dicomId={{dicomId}}' \  
  --header 'Accept: text/plain, application/json, text/json'
```

**GET** http:///api/Volume/:id?dicomId={{dicomId}}

http:///api/Volume/:id?dicomId={{dicomId}}

#### HEADERS

##### Accept

text/plain, application/json, text/json

PARAMS

---

**dicomId**

{{dicomId}}

PATH VARIABLES

---

**id**

{{id}}

Example Request

```
http:///api/Volume/:id?dicomId={{dicomId}}

curl --request GET \
  --url 'http://api/Volume/:id?dicomId={{dicomId}}' \
  --header 'Accept: text/plain, application/json, text/json'
```