

Wydział Podstawowych Problemów Techniki

**PRACA DYPLOMOWA**

**Zastosowanie sieci neuronowych do automatycznej segmentacji tkanki nowotworowej na obrazach histopatologicznych**

**Autor: Jakub Siembida nr alb. 236728**

**Opiekun: dr inż. Witold Dyrka**

Sieć neuronowa, obraz histopatologiczny, rak płuc

Aplikacja komputerowa oznaczająca miejsca obecności komórek nowotworowych płuc na obrazach histopatologicznych typu whole-slide-image przy pomocy sieci neuronowej trenowanej na zbiorze danych ACDC-Lung HP

1. Cel i zakres pracy

Nowotwory stanowią poważny problem społeczny. W ostatnich latach obserwuje się wzrost zachorowalności na te choroby. Stanowią one drugą najczęstszą przyczynę zgonów na świecie. Najczęściej diagnozowanymi przypadkami nowotworu są raki płuc. Stanowią one około 20% wszystkich zdiagnozowanych przypadków. Wpływ na tak dużą zachorowalność mają m.in. zły stan powietrza, czy palenie tytoniu. Ze względu na trudną diagnostykę, są to również najbardziej letalne nowotwory. Powodem tak dużej śmiertelności jest również nieodpowiedni dobór terapii[1][2].

W przeciągu ostatnich lat nastąpił duży wzrost zainteresowania sztuczną inteligencją, w szczególności sieciami neuronowymi. Dzięki nim możliwa jest implementacja zadań, które dla człowieka wydają się proste, lecz dla komputera są skomplikowane, np. rozpoznawanie twarzy, głosu, gra w szachy. Dzięki wykorzystaniu algorytmu uczenia maszynowego możliwe jest poprawne analizowanie przez sieć danych, które zostały po raz pierwszy tej sieci podane[3].

Zainteresowanie analizą danych medycznych przy pomocy algorytmów sztucznej inteligencji wzrasta z roku na rok. Spowodowane jest to m.in. brakiem dostatecznej liczby specjalistów z danej dziedziny, lub czasem trwania analizy danych przez lekarza. Opracowywane są coraz to nowsze algorytmy pozwalające na uzyskanie dokładniejszej oceny obrazów medycznych, pomiarów (np. EEG, EKG), porównywalnych z oceną specjalisty w danej dziedzinie, w krótszym czasie[4].

Celem niniejszej pracy jest stworzenie oprogramowania wspomagającego ocenę obrazów histopatologicznych płuc zapisanych jako obrazy całego preparatu histopatologicznego (ang. whole slide image, WSI)[XXXXXXXXXXXXXXXXXX][YYYYYYYY], wykorzystującego konwolucyjną sieć neuronową oznaczającą tkankę nowotworową na obrazie. Jako zestaw danych został wybrany zbiór udostępniony wraz z ACDC-Lung HP challenge[5]. Wybrano ten zestaw, gdyż składa się on z całych skanów skrawków płuc, dzięki czemu nie jest narzucony wymiar obrazu. Jako język programowania wybrano język Python 3.7, który dzięki zastosowaniu zewnętrznych modułów stanowi na dzień dzisiejszy jeden z najlepszych języków programowania do implementacji sieci neuronowej.

1. Nowotwór płuc

W ostatnich latach zanotowano znaczny wzrost zapadalności na nowotwory płuc. Według statystyk WHO z 2018r. stanowią drugą najczęstszą przyczynę zgonów na świecie. Najczęściej diagnozowanym i zarazem najbardziej śmiertelnym nowotworem jest rak płuc[1].

Rakiem płuc nazywamy nowotwór wywodzący się z tkanki nabłonkowej wyścielającej drogi oddechowe[ŹRÓDŁO]. Czynnikami sprzyjającymi powstawaniu zmian nowotworowych w płucach są m.in. palenie tytoniu, ekspozycja na duże dawki promieniowania, obecność azbestu w płucach[2]. W obrębie raka płuc wyróżnia się dwa typy: drobno- i niedrobnokomórkowy. Pierwszy z wymienionych rozwija się agresywniej, przez co wyleczenie pacjenta cierpiącego na ten rodzaj raka jest trudniejsze[6].

Nowotwór nie drobnokomórkowy płuc jest dzielony na trzy linie komórek rakowych: rak gruczołowy, płaskonabłonkowy oraz wielkokomórkowy. Pierwsza z wymienionych charakteryzuje się tworzeniem struktur podobnych do gruczołów, widocznych na obrazowaniu histopatologicznym oraz obecnością śluzu. Rak płaskonabłonkowy jest silnie związany z paleniem tytoniu, cechuje się występowaniem stanów przedrakowych, takich jak zmiana kształtu komórek i ich funkcji. Rak wielkokomórkowy kształtem przypomina komórki pozostałych dwóch linii komórek rakowych, lecz nie współdzieli ich cech. Komórki tej linii rakowej są zdecydowanie większe od komórek pozostałych dwóch linii[2].

1. Kliniczna diagnostyka raka płuc

Diagnostyka raka płuc polega na wykryciu obecności komórek nowotworowych w tkankach pacjenta. Ważną cechą diagnostyczną jest stopień rozsiania nowotworu w tkance. Aby dobrać odpowiedni typ leczenia, należy przeprowadzić badanie histopatologiczne. Jest to inwazyjna metoda polegająca na poborze tkanki i jej analizy przez lekarza histopatologa. Pozwala ona na dokładne rozpoznanie choroby nowotworowej poprzez przypisanie konkretnej linii rakowej, której obecność jest stwierdzona w próbce. Pozwala to dostosować terapię do potrzeb pacjenta zmniejszając ekspozycje na toksyczne właściwości cytostatyków[7][8][11].

Pobór tkanek do badań może odbyć się na drodze biopsji przezskórnej lub zabiegu chirurgicznego. Pierwsza z metod została zaprezentowana na rysunku 3.1. Polega ona na nakłuciu guza igłą biopsyjną i poborze tkanki. Podczas zabiegu wykonywane jest obrazowanie CT w celu umożliwienia lokalizacji guza w płucu. Druga z metod jest używana w momencie, gdy pacjent kwalifikuje się do operacji. Polega na torakotomii(otwarciu ściany klatki piersiowej) i wycięciu chorej tkanki[9].

# Diagram showing a percutaneous lung biopsy

# Rys. 3.1. Wizualizacja biopsji przezskórnej guza płuc. Źródło: [9]

Świeżo pobrane tkanki należy wybarwić przed analizą histopatologiczną. Zastosowanie barwienia ma za zadanie ułatwić lekarzowi rozpoznawanie struktur komórkowych, bądź całych komórek. Najczęściej stosowanym barwieniem jest barwienie hematoksyliną i eozyną. Pierwszy z wymienionych związków chemicznych jest utleniany przy pomocy tlenu z powietrza przy obecności światła (hematoksylina Boehmera), lub tlenku rtęci(II)(hematoksylina Harrisa) do hemateiny. Barwi ona w obecności jonów glinu jądro komórkowe i siateczkę śródplazmatyczną szorstką na kolor niebieski. Eozyna natomiast wybarwia cytoplazmę i włókna kolagenowe na kolor czerwony[10][11].

1. Sieci neuronowe
   1. Sieć neuronowa a mózg ludzki

W ostatnich latach można dostrzec duże zainteresowanie dziedziną sztucznej inteligencji. Jest to spowodowane chęcią automatyzacji procesów, mające na celu wspomaganie pracy człowieka, osiągnięcie powtarzalności wyników i zaoszczędzenie czasu. Jednym z działów sztucznej inteligencji są sieci neuronowe, które swoją strukturą niekiedy przypominają system połączeń neuronów i przesyłanie sygnału w mózgu ludzkim[12].

Na rysunku 4.1.1. przedstawiono porównanie komórki nerwowej z modelem matematycznym stanowiącym podstawową jednostkę sieci neuronowej. Każda ze struktur otrzymuje sygnały wejściowe od dendrytów (będących danymi wejściowymi do sieci lub wzmocnieniami neuronów z poprzedniej warstwy) i tworzy sygnał wyjściowy wysyłany aksonem. W modelu matematycznym każdy dendryt ma przypisaną do siebie wagę *wx*, powodującą wzmocnienie (*wx > 0*) lub osłabienie (*wx < 0*) sygnału. Sygnały od poszczególnych dendrytów są łączone wybraną operacją matematyczną. Wartość sygnału wyjściowego zależy od funkcji aktywacji, mapującej liczby z dziedziny liczb rzeczywistych na oczekiwany przedział (najczęściej [0, ∞) lub [0, 1])[12].

# 

# Rys. 4.1.1. Porównanie komórki nerwowej człowieka z modelem matematycznym neuronu sieci neuronowej. [12]

* 1. Zasada działania sieci

Aby zrozumieć jak działa sieć neuronowa, należy zastanowić się jak wygląda przesyłanie informacji między warstwami oraz algorytm nauki sieci.

Poprzez naukę sieci rozumie się ustawienie parametrów sieci (np. wag) na podstawie przekazywanych do sieci danych wejściowych oraz danych wzorcowych[12][13]. W celu nauki modelu tworzy się zestaw danych składający się z dwóch podzbiorów – zbiór danych do nauczania (na których sieć jest uczona), oraz zbiór danych testowych (na których przeprowadzana jest walidacja). Każdy ze zbiorów zawiera pary dane wejściowe – dane wzorcowe, reprezentujące oczekiwany wynik. Celem nauczania sieci jest ustawienie jej parametrów w taki sposób, aby podobieństwo między predykcją sieci, a danymi wzorcowymi było jak największe, oraz aby dla nigdy nie przetwarzanych przez sieć danych uzyskać jak najdokładniejsza, zgodną z rzeczywistością predykcję[3][12][13].

Pierwsza warstwa neuronów, na podstawie otrzymanych danych wejściowych, wytwarza sygnały (na podstawie algorytmu opisanego w rozdziale 5.1.), które są przekazywane do kolejnej warstwy jako sygnał wejściowy. Proces jest powtarzany do momentu osiągnięcia ostatniej warstwy. W tym miejscu predykcja sieci jest porównywana z danymi wzorcowymi przy pomocy określonej funkcji. Wynik porównania jest mnożony przez współczynnik uczenia, będący liczbą z zakresu [0, 1], determinującą szybkość zmian. Wyznaczana jest pochodna powyższego iloczynu jest dodawana do wag warstwy wcześniejszej. Powyższy proces jest powtarzany dla całego zestawu danych[3].

4.3. Rodzaje sieci neuronowych

Sieci neuronowe można klasyfikować na wiele sposobów. Najpopularniejszymi podziałami są podział ze względu na kierunkowość przepływu danych, rodzaj dokonywanych operacji[13]. W powyższym podrozdziale zostaną opisane trzy główne typy sieci- wielowarstwowy perceptron, sieć konwolucyjna i sieć rekurencyjna.

Najprostszym typem sieci jest wielowarstwowy perceptron. Składa się on z warstwy wejściowej, warstwy wyjściowej i przynajmniej jednej warstwy ukrytej (znajdującej się pomiędzy warstwą wejściową, a warstwą wyjściową). Neurony sąsiadujących warstw są połączone na zasadzie każdy z każdym, tzn. neuron *ai* należący do warstwy A jest połączony   
z każdym neuronem z warstwy B, natomiast każdy neuron *bj* należący B otrzymuje sygnał od każdego neuronu z warstwy A (Rys. 4.3.1.)[13].

# 

# Rys. 4.3.1. Dwa modele sieci typu wielowarstwowy perceptron- z jedną ukrytą warstwą (po lewej), z dwiema ukrytymi warstwami (po prawej). [13]

Zasadniczą wadą wielowarstwowego perceptronu jest ignorowanie wzajemnego położenia danych, które często jest istotnym czynnikiem definiującym dane wejściowe (np. podczas analizy funkcji matematycznych, obrazu). W takich sytuacjach zdecydowanie lepszym wyborem jest konwolucyjny model sieci neuronowej. Zasada przejścia danych z jednej warstwy do drugiej opiera się na operacji splotu – z tego powodu sąsiadujące warstwy nie są w pełni połączone- tylko pewien region warstwy A jest połączony z neuronem warstwy B (Rys. 4.3.2.)[13]. Funkcja splotu osiąga lokalne maksimum w sytuacji, gdy dane wejściowe są podobne do struktury wag (wektora w przypadku funkcji, macierzy w przypadku obrazu)[3][13]. W tym typie sieci wykorzystywana jest tez operacja poolingu, polegająca na mapowaniu grupy wzmocnień neuronów warstwy A na wzmocnienie jednego neuronu warstwy B. Najczęściej stosowanym typem poolingu jest pooling maksymalny, który jako wzmocnienie neuronu następnej warstwy wybiera wartość maksymalną z grupy wzmocnień z warstwy aktualnej[3][13][14].

# 

# Rys. 4.3.2. Operacja splotu macierzą 3x3 na macierzy 4x4.[14]

Oba omówione wcześniej modele nie są najlepszym wyborem w przypadku danych, gdzie informacja o poprzedniku jest niezbędna do poprawnej predykcji danych następcy. W takich sytuacjach najlepiej sprawdza się rekurencyjna sieć neuronowa, która poprzez pętlę sprzężenia zwrotnego w swojej architekturze przechowuje informację o poprzednim przejściu danych przez sieć (Rys. 4.3.3.).

# 3. Graph of a recurrent neural network.Â

# Rys. 4.3.3. Przykładowy model sieci rekurencyjnej.[15]

1. zbiór danych

Zbiorem danych wykorzystanych w niniejszej pracy jest zbiór udostępniony wraz z konkursem „ACDC- Lung HP” na stronie www.grand-challenges.org. Celem konkursu było stworzenie algorytmu lub sieci segmentującej skrawki histologiczne płuc[<https://acdc-lunghp.grand-challenge.org/>]. [TODO też trochę o pochodzeni datasetu : [https://arxiv.org/ftp/arxiv/papers/1803 /1803.05471.pdf](https://arxiv.org/ftp/arxiv/papers/1803%20/1803.05471.pdf)]

* 1. Opis ilościowy zbioru danych

Dane zostały udostępnione jako 150 skanów skrawków do uczenia, oraz 50 skrawków do walidacji. Na każdy skrawek zbioru służącego do nauki składają się dwa pliki: o rozszerzeniu .tif (ang. tagged image file)[źródło: wikipedia] zawierające całe skany histologiczne oraz .xml (ang. extensible markup language)[źródło: wikipedia] zawierające oznaczenia miejsc występowania tkanki nowotworowej na obrazie. Zbiór do walidacji zawierał jedynie obrazy skrawków histologicznych. Skany zostały zapisane przy użyciu zautomatyzowanego mikroskopu Olympus VS120[[https://arxiv.org/ftp/arxiv/papers/1803 /1803.05471.pdf](https://arxiv.org/ftp/arxiv/papers/1803%20/1803.05471.pdf)].

Każdy skan histologiczny składa się z około 80 000 pomniejszych skanów[<https://acdc-lunghp.grand-challenge.org/>dataset], które ułożone w kolejności zapisu oddają rzeczywistą geometrię próbki. Dzięki zastosowaniu formatu .tif, umożliwiającego na zapis kilku przybliżeń próbki w jednym pliku[ŹRÓDŁO], dostarczone obraz można wyświetlić w rzeczywistej skali, jak i w skali komórkowej.

* 1. Opis jakościowy zbioru danych

# czy to zamieszczać?

##Wśród obrazów preparatów histologicznych można wyróżnić dwie grupy: zawierającą jeden duży skrawek oraz zawierającą kilka skanów tej samej tkanki. W drugiej grupie kluczowym jest wybranie tylko tego skanu, który został oznaczony. W przeciwnym razie dany fragment zostanie wycięty tylokrotnie, ile takich samych skanów jest obecnych na obrazie. W efekcie otrzymamy kilka obrazów tego samego fragmentu wycinka, lecz tylko jeden z nich będzie poprawnie oznaczony. Zaistnienie takiej sytuacji może realnie wpłynąć na jakość predykcji sieci. ##

Kolejnym aspektem jest stosunek powierzchni tkanki do powierzchni całego obrazu. W pierwszej grupie tkanka wypełnia obraz w dużym procencie, co przekłada się na dużą liczbę danych do uczenia sieci pozyskanych z jednego slajdu. W drugiej grupie procent tkanki na obrazie jest zdecydowanie mniejszy- większość obrazu wypełnia białe tło. Powoduje to mniejsza ilość pozyskanych danych w porównaniu do grupy pierwszej. [TODO warto pokazać]

Slajdy preparatów histologicznych zawierają liczne zabrudzenia, czy artefakty. Wśród nich wymienić można m.in. plamy mieszaniny barwiącej w miejscach, gdzie nie znajduje się tkanka, makroskopowe oznaczenia na obrazach, liczne, drobne fragmenty tkanek, czy nachodzenie na siebie dwóch warstw preparatu. Są to typowe artefakty powstające podczas pracy przy preparatach histologicznych. Istotną kwestią jest, aby predykcja sieci neuronowej nie obejmowała obszarów ich występowania[ŹRÓDŁO]. [może pokazać?]

1. Narzędzia
   1. Język programowania Python

Jako wiodący język programowania projektu został wybrany Python w wersji 3.7.3. Jest to język obiektowy, zajmujący czwarte miejsce w rankingu najczęściej używanych języków programowania oraz drugie miejsce w rankingu najbardziej lubianych języków przez programistów[<https://insights.stackoverflow.com/survey/2019>].

Główną przewagą Pythona nad innymi wysokopoziomowymi językami, takimi jak Java, C#, jest fakt, iż programy o identycznej funkcjonalności napisane w innych językach statystycznie zawierają więcej linii kodu źródłowego. Powoduje to, iż utrzymanie programu napisanego w Pythonie jest łatwiejsze od oprogramowania napisanego np. w Javie[ŹRÓDŁO]. Kolejną zaletą tego języka jest obecność dobrze przetestowanych, intuicyjnych w użyciu zewnętrznych modułów do m.in. analizy danych, obróbki zdjęć, czy uczenia maszynowego[<https://www.python.org/about/apps/>].

[TODO może trochę więcej]

* 1. ASAP- Automated Slide Analysis Platform

Wraz z zestawem danych dołączone zostało oprogramowanie napisane w języku C++, w którym histopatolodzy oznaczali zmiany nowotworowe – Automated Slide Analysis Platform[<https://github.com/computationalpathologygroup/ASAP>]. Pozwala ono na otwarcie slajdów preparatów histologicznych oraz wczytanie i nałożenie na obraz oznaczeń. Dostarczono również fasadę API w/w programu w języku Python pozwalającą na odczyt slajdu, fragmentu slajdu, czy adnotacji w kodzie programu oraz na generowanie masek prawdy na podstawie adnotacji[<https://acdc-lunghp.grand-challenge.org/>]. Pozwala to na dowolność przy doborze wielkości analizowanych obrazów i zastosowanie własnej obróbki danych.

* 1. Keras i Tensorflow

Jako główne narzędzie do tworzenia sieci neuronowej użyto biblioteki Keras. Jest to jedna z najpopularniejszych bibliotek do tworzenia, nauki i walidacji konwolucyjnych i rekurencyjnych modeli sieci neuronowych w języku Python. Zaletą tej biblioteki jest możliwość użycia jednego z kilku ~~backendów~~ modułów matematycznych, które wykonują obliczenia niezbędne do nauki, czy walidacji sieci. Tworzenie modelów odbywa się w kodzie Pythona, bez udziału plików konfiguracyjnych. Keras zapewnia prostotę tworzenia modelu. Ilość kodu potrzebna do zaimplementowania danej sieci w tej bibliotece jest niewielka w porównaniu do pozostałych bibliotek[<https://keras.io/>].

Tensorflow to moduł matematyczny implementujący podstawowe operacje niezbędne do stworzenia modelu sieci neuronowej, napisany w języku Python. Jest często stosowany jako backend biblioteki Keras, lecz może służyć również jako niezależne środowisko do tworzenia modeli sieci, które oferuje większą kontrolę nad modelem, umożliwia projektowanie bardziej złożonych procesów nauki. Dzięki zastosowaniu rozszerzenia Tensorboard możliwa jest wizualizacja procesu uczenia na wykresach, jak i śledzenie zmian parametrów uczenia. Tensorflow jest używany przez wiele międzynarodowych firm, takich jak Google, Intel, Airbus[<https://www.tensorflow.org/about>][ <https://www.tensorflow.org/tensorboard/get_started>].

* 1. Pozostałe użyte biblioteki

Podstawową, użytą na potrzeby projektu biblioteką matematyczną w celu obróbki danych przechowywanych w wielowymiarowych tablicach jest NumPy. Umożliwia ona łatwe zarządzanie tablicami o liczbie wymiarów większej niż 3. Biblioteka ta zawiera również zoptymalizowaną czasowo implementację podstawowych operacji algebry liniowej, rachunku różniczkowego i statystyki. Dostarczanie danych przechowywanych w tablicach NumPy jest wspierane przez wiele popularnych bibliotek do edycji obrazów, obróbki danych tabelarycznych, czy uczenia maszynowego[<https://numpy.org/>, <https://pandas.pydata.org/pandas-docs/stable/getting_started/overview.html>, <https://scikit-image.org/>, <https://matplotlib.org/users/installing.html>].

Jako narzędzia do wizualizacji i obróbki obrazów wybrano Pyplot z biblioteki Matplotlib, Scikit-image. Pierwszy z wymienionych służy głównie do wyświetlania obrazów zapisanych w wielowymiarowej tablicy NumPy. Biblioteka ta umożliwia także generowanie wykresów punktowych, histogramów. Możliwe jest przedstawienie kilku wykresów, czy obrazów w jednym oknie, co znacząco wpływa na komfort analizy danych. Biblioteka Scikit-image zapewnia implementację podstawowych operacji wykorzystywanych do obróbki zdjęć, czy miar podobieństwa obrazów[<https://scikit-image.org/>].

1. Przygotowanie zbioru danych

Zbiór danych został dostarczony w postaci całych skanów histologicznych wraz z opisem fragmentów tkanek, uznawanych za nowotworowe, w pliku XML. Pozwala to na dobór dowolnej szerokości obrazów i sposobu pozyskiwania danych.

Pierwszym podjętym krokiem była konwersja XML-owych oznaczeń na jednokanałowe obrazy. Każdej komórce z zaznaczonego obszaru przypisano wartość 1, a pozostałym komórkom wartość 0- w wyniku otrzymano czarno-białą maskę. Konwersji dokonano przy pomocy dedykowanego przez dostawcę zbioru danych oprogramowania.

Krokiem milowym w obróbce zbioru danych było napisanie algorytmu selekcjonującego fragment pełnego skanu histologicznego zawierającego tkankę. W niektórych skanach stosunek tkanki do tła na obrazie był niewielki. W innych natomiast występowały dwie kopie tej samej tkanki, z czego tylko jedna była oznaczona. Próba selekcji danych podczas obróbki powodowałaby wykorzystanie pamięci i czasu procesora do przetwarzania danych, które w dużym procencie nie kwalifikowały się do wykorzystania do nauki sieci. [PROP może schemat blokowy, ale to raczej zbyt skomplikowane; TODO policz złożoność] Algorytm można podzielić na dwa odrębne zagadnienia:

* odnalezienie współrzędnych ramki zawierającej wszystkie adnotacje
* rozszerzenie współrzędnych z pierwszego podpunktu w taki sposób, aby zawierały całą oznaczoną tkankę.

W pierwszym podpunkcie jako dana wejściowa podawana jest maska. W wyniku otrzymujemy wektor zawierający cztery współrzędne. Wczytywany jest pierwszy wiersz. Co tysięczny piksel jest porównywany do 1- gdy porównanie zwróci *True*, porównywane są współrzędne obecnie przetwarzanego piksela z współrzędnymi z poprzednich iteracji- zmniejszane są współrzędne początkowe jeżeli współrzędne iterowanej komórki są mniejsze, zwiększane współrzędne końcowe, gdy przetwarzany piksel znajduje się w komórce o większych indeksach. Wszystkie w/w czynności są powtarzane dla co tysięcznego wiersza. Gdy współrzędne wynikowe zostały zainicjalizowane, a w obrębie całego rzędu nie znaleziono ani jednej białej komórki, zwracane są współrzędne. W drugim podpunkcie idea algorytmu jest bardzo podobna. Daną wejściową są otrzymane z wcześniejszej części algorytmu współrzędne i cały skan histologiczny. Każda współrzędna jest rozszerzana o 1024 (od początkowej 1024 jest odejmowane, do końcowej- dodawane). Jest to powtarzane tak długo, jak średnia arytmetyczna z kanałów danego piksela jest różna od 255 (gdy piksel jest koloru białego). Gdy komórka zostanie znaleziona, współrzędna jest cofana o iterację. Wykonywane jest wyszukiwanie binarne pierwszego piksela o kolorze białym. Pozwala ono na odnalezienie pierwszego wystąpienia białej komórki. Zapisywany jest indeks. Całość jest powtarzana dla każdej współrzędnej. Wynikiem funkcji jest wektor zawierający koordynaty najbardziej wysuniętych punktów na oznakowanym obrazie histologicznym.

Celem pierwszej części algorytmu jest identyfikacja oznakowanej kopii obrazu. Nie zawsze pierwszy obraz, iterując od góry, jest tym oznakowanym. Dzięki temu jedynie adnotowane przez histopatologa próbki są obecne w zbiorze danych. Krok iteracji równy 1000 ma za zadanie przyspieszyć przetwarzanie obrazu. Pełne skany histologiczne cechują się wymiarami rzędu 105 pikseli[Źródło do WSI]. Druga część rozszerza okno o tkankę zdrową (nieoznakowaną). Gdyby ten krok pominąć, powierzchnia zaznaczona jako występowanie komórek nowotworowych mogła by przeważać nad powierzchnią oznaczającą brak tej cechy (zdrowa tkanka i obszary niebędące tkanką)[ŹRÓDŁO coś w prepare dataset poszukaj]. Wynik działania algorytmu przedstawiono na rycinie [NUMER].

[TODO wrzuć wycięte i screen z asapa]

Ostatnim krokiem było pocięcie obrazów na mniejsze, o zadanej szerokości, równej 256 pikseli. Wyeliminowano pozostałe fragmenty białego tła uśredniając obraz i zapisując tylko te obrazy, dla których średnia wynosiła mniej, niż 240. Eliminowało to także te fragmenty, w których większość pikseli była biała, a tylko niewielka część była fragmentami tkanek.

1. Model
   1. U-Net

Jednym z kluczowych aspektów projektowania sieci neuronowej jest wybór architektury. Dobór nieodpowiedniej struktury sieci może mieć znaczący wpływ na jakość predykcji[François Chollet - Deep Learning with Python (2018)]. Jako, iż zadaniem projektowanej sieci jest segmentacja obrazów, jako model sieci wybrano architekturę U-Net. Jest to w pełni konwolucyjny model sieci zaproponowany w 2015 roku przez Olafa Ronnebergera, Philippa Fischera i Thomasa Broxa z uniwersytetu w Freiburg w celu segmentacji obrazów medycznych[<https://arxiv.org/pdf/1505.04597.pdf>, <https://towardsdatascience.com/review-u-net-biomedical-image-segmentation-d02bf06ca760>]. Jego nazwa została zainspirowana pierwszym, graficznym przedstawieniem modelu, przypominającym literę „U”. Został on użyty po raz pierwszy do segmentacji struktur komórek nerwowych na obrazach mikroskopii elektronowej oraz do oznaczania zmian próchnicznych zębów na obrazach rentgenowskich osiągając lepsze wyniki od innych architektur[<https://arxiv.org/pdf/1505.04597.pdf>, <https://lmb.informatik.uni-freiburg.de/people/ronneber/isbi2015/>].

# Obraz zawierający zrzut ekranu, mapa Opis wygenerowany automatycznie

# Rys. 6.1. Przykładowy schemat modelu sieci typu U-Net. Źródło: [<https://arxiv.org/pdf/1505.04597.pdf>] Adaptacja

Przykładowy schemat modelu sieci został przedstawiony na rycinie 6.1. Jest to architektura typu auto-enkoder. Warunkami koniecznymi do klasyfikacji sieci do tej grupy jest zastosowanie kompresji danych poprzez zmniejszanie rozmiaru kolejnych warstw (na lewo od zielonej linii), dekompresja danych poprzez ponowne zwiększanie rozmiarów warstw (na prawo od zielonej linii) oraz podobieństwo między danymi wejściowym, a danymi wyjściowymi sieci. Podczas procesu kompresji, selekcjonowane są najistotniejsze informacje, pozwalające na spełnienie zaprojektowanego zadania (klasyfikacji, detekcji, czy segmentacji)[ <http://web.stanford.edu/class/cs294a/sparseAutoencoder_2011new.pdf>].

Model U-Net jest w pełni konwolucyjny. Oznacza to, iż nie występuje w nim ani jedna w pełni połączona warstwa neuronów, a główną operacją wykonywaną na danych jest operacja splotu. Dane wejściowe podawane są w tensorze reprezentującym obraz. Przy pomocy dwóch operacji splotu macierzą 3x3 selekcjonowane są wybrane cechy i następnie przy pomocy operacji poolingu maksymalnego macierzą 2x2 z krokiem 2 obraz jest zmniejszany czterokrotnie. W ten sposób pozyskiwane są szczegółowe informacje o występowaniu cechy kosztem zatracenie informacji o jej położeniu na obrazie. Po zmniejszeniu szerokości do żądanej wielkości, wykonywana jest operacja próbkowania w górę, polegająca na przetworzeniu jednej komórki w obrazie zadanym na cztery komórki w obrazie wynikowym, co powoduje czterokrotne zwiększenie rozmiaru obrazu. Dodatkowo- wykonywana jest operacja splotu macierzą 2x2 powodującą dwukrotne zmniejszenie liczby filtrów. Cechą wyróżniającą architekturę U-Net od innych, głębokich auto-enkoderów jest operacja dołączenia warstwy pochodzącej z procesu detekcji. Celem tego zabiegu jest ponowne pozyskanie informacji o położeniu danej cechy na obrazie. Następnie wykonywane są dwie operacje splotu macierzą 3x3. Kroki opisane od próbkowania w górę są powtarzane dokładnie tyle razy, ile razy została wykonana operacja maksymalnego poolingu. Na sam koniec wykonywana jest operacja splotu przy wykorzystaniu jednej wartości w celu przetworzenia filtrów obrazu zadanego do ilości filtrów wymaganych do spełnienia zadania. Dzięki zastosowaniu funkcji aktywacji typu sigmoida można uzyskać rozkład prawdopodobieństwa występowania danej cechy[źródło], lub dla skoku jednostkowego - maskę prawdy określającą miejsce występowania danej cechy na obrazie[<https://arxiv.org/pdf/1505.04597.pdf>, <https://towardsdatascience.com/review-u-net-biomedical-image-segmentation-d02bf06ca760>].

# Wykres 6.1. Zależność szerokości warstwy oraz ilości filtrów od indeksu warstwy w modelu przedstawionym na rycinie 6.1. Twórczość własna.

Na wykresie 6.1. przedstawiono zależności ilości filtrów oraz szerokości warstwy od jej indeksu w modelu. Opisane zależności są przeciwnie monotoniczne. Wraz ze wzrostem numeru warstwy ilość filtrów rośnie, a szerokość warstwy maleje. Obie charakterystyki osiągają ekstremum dla indeksu równemu połowie indeksu maksymalnego. Po osiągnięciu warstwy o indeksie 15, szerokość warstwy jest proporcjonalna do indeksu warstwy, a ilość filtrów- przeciwnie proporcjonalna. Charakterystyka ilości filtrów od indeksów warstwy dla przedstawionego modelu jest symetryczna, a zależność szerokości warstwy od pozycji warstwy jest asymetryczna. Powodem tego jest fakt, iż obraz wejściowy jest większy, niż obraz wyjściowy z powodu zastosowania operacji splotu

W architekturze U-Net można wyróżnić dwie grupy operacji- pierwszą opisującą proces dwukrotnego zmniejszenia szerokości obrazu oraz dwukrotnego zwiększenia liczby filtrów, oraz drugą, opisującą proces dwukrotnego zwiększenia szerokości obrazu i dwukrotnego zredukowania liczby filtrów.

* 1. Zaproponowany model sieci

Zastosowano gotową implementację szkieletu modelu udostępnioną przez użytkownika zhixuhao na licencji MIT na portalu github. Wykorzystano dwie funkcje: o nazwach „down” i „up”. Implementacja pierwszej z nich zawiera operacje niezbędne do dwukrotnego zmniejszenia szerokości obrazka i dwukrotnego zwiększenia ilości filtrów, a drugiej- do uzyskania przeciwstawnego efektu. Zastosowanie tych dwóch funkcji upraszcza tworzenie modeli o różnej głębokości.

Architektura użyta w niniejszej pracy nieznacznie różni się od zaprezentowanej w poprzednim rozdziale. Jako dane wejściowe do sieci wybrano obrazy o wymiarach 256x256px, kodowane w 3 kanałach. Zdecydowano się na zastosowanie splotu macierzą 3x3 z jednostkową wyściółką zer w celu zachowania szerokości warstwy. Pozwoliło to na uzyskanie predykcji w postaci obrazu o takiej samej wielkości, jak obraz wejściowy przy równej liczbie operacji poolingu i próbkowania w górę.

Na wykresie 7.2. przedstawiono zależności ilości filtrów i szerokości warstwy w zależności od jej położenia w zaproponowanym modelu. W przeciwieństwie do charakterystyk zilustrowanych na wykresie 7.1., zarówno jedna jak i druga zależność jest symetryczna względem środkowego indeksu. Spowodowane jest to zastosowaniem splotu macierzą 3x3 z jednostkowym paddingiem zerami, który gwarantuje zachowanie szerokości obrazu w trakcie operacji splotu. [TODO narysuj schemat sieci]

# Wykres 7.2. Zależność szerokości warstwy i ilości filtrów od indeksu warstwy w modelu sieci użytej w niniejszej pracy. Twórczość własna.

* 1. Metryki

Aby móc oceniać jakościowo poziom wyuczenia sieci, wprowadza się pojęcie metryki. Jest to funkcja dwóch zmiennych- wartości oczekiwanych i predykcji sieci, zwracająca skalar reprezentujący jakość dopasowania[Źródło].

Jedną z podstawowych metryk stosowanych do porównywania segmentacji obrazów jest współczynnik Sørensena-Dice’a[ŹRÓDŁO]. Dla dwóch zbiorów X i Y jest zdefiniowany jako:

gdzie |X| i |Y| są liczbami kardynalnymi zbiorów X i Y, a |X | jest liczbą kardynalną zbioru wspólnych elementów X i Y. Wartość tej metryki silnie zależy od ilości elementów występujących w obu zbiorach- im więcej wspólnych elementów jest posiadają oba zbiory, tym wartość metryki jest większa. Antydziedziną współczynnika Sørensena-Dice’a jest przedział [0, 1]. Metryka ta jest też często zapisywana w procentach[ŹRÓDŁO, to z wiki].

W celu porównania danych boolowskich, współczynnik Sørensena-Dice’a opisywany jest zależnoscią:

gdzie TP to ilość porównań pozytywnie dodatnich (część wspólna zbiorów), FP to liczba porównań fałszywie pozytywnych (elementy zbioru porównywanego niewystępujące w zbiorze odniesienia), a FN to liczba pozostałych porównań[ŹRÓDŁO, to z wiki].

Inną metryką szeroko stosowaną do porównywania jakości segmentacji obrazów jest odległość Hausdorffa. Nie analizuje ona powierzchni, jak metryka Sørensena-Dice’a, lecz obwiednie kształtów. Algorytm wyznaczenia odległości Hausdorffa dla dwóch zbiorów punktów leżących na obwiedniach kształtów można opisać w kilku punktach:

1. wyznaczenie dla każdego punktu ze zbioru X odległości do najbliższego punktu ze zbioru Y (najmniejszej odległości z obwiedni X do obwiedni Y)
2. odnalezienie największej z najmniejszych odległości ze zbioru X do zbioru Y
3. wyznaczenie dla każdego punktu ze zbioru Y odległości do najbliższego punktu ze zbioru X (najmniejszej odległości z obwiedni Y do obwiedni X)
4. odnalezienie największej z najmniejszych odległości ze zbioru Y do zbioru X
5. wynikiem jest większa z wartości z punktów 2. i 4.

# 

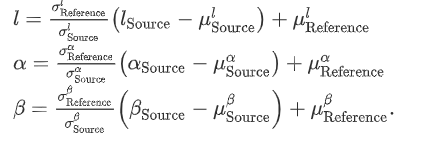
# Rys. 213123. Sposób wyznaczania odległości Hausdorffa dla dwóch zbiorów X i Y. Źródło: [<https://en.wikipedia.org/wiki/Hausdorff_distance#/media/File:Hausdorff_distance_sample.svg>]

Matematyczna definicja tej metryki dla dwóch zbiorów X i Y jest opisana równaniem 3671223, a wizualizacja wyznaczonych największych z najmniejszych odległości od obwiedni do obwiedni została przedstawiona na rycinie 213123 [Tu mógłbym napisać, że szukanie najmniejszej odległości to jak szukanie promienia koła o środku w punkcie x, które byłoby styczne do obwiedni Y i nie przecinałoby obwiedni Y, lecz ciężko o źródło na to, może jako obserwacja (tak jest, bo szukamy minimum funkcji promienia od y, a koło jest równoodległe od środka dla każdego punktu][ <https://en.wikipedia.org/wiki/Hausdorff_distance>].

* 1. Preprocessing

Preprocessing to szereg operacji wykonywanych na danych przed ich udostępnieniem sieci neuronowej w celu zwiększenia jakości predykcji, czy wspomagania procesu uczenia[Źródło]. Do stworzenia modelu na potrzeby niniejszej pracy użyto transformację koloru Reinharda oraz filtr HED.

Intensywność wybarwienia próbek hematoksyliną i eozyną jest zależna m.in. od stężenia barwników i czasu ekspozycji[<https://www.leicabiosystems.com/knowledge-pathway/he-basics-part-4-troubleshooting-he/>]. Powoduje to, iż uzyskanie dwóch identycznie wybarwionych próbek jest trudne[[http://nsh.org/si tes/default/files/Guidelines \_For\_Hematoxylin\_and\_Eos in\_Staining.pdf](http://nsh.org/si%20tes/default/files/Guidelines%20_For_Hematoxylin_and_Eos%20in_Staining.pdf)]. W celu znormalizowania odcieni barw zastosowano transformację koloru Reinharda. Pozwala ona na zmodyfikowanie koloru obrazu na podstawie obrazu odniesienia. Dane z obu obrazów są konwertowane do przestrzeni barw *lαβ*, która minimalizuje korelacje między poszczególnymi kanałami. Kolejnym krokiem jest wyznaczenie wartości średniej i odchylenia standardowego obu obrazów. Wartości kanałów *lαβ* obrazu wynikowego opisują zależności przedstawione na równaniu [NUMER].

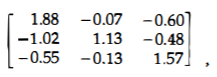


[TODO przerobić na równania] Źródło: [https://www.scienc edirect.com/s cience/articl e/pii/S089571770600032X#fd1]

Ryciny 123-23123 przedstawiają przykłady transformacji koloru na reprezentantach próbek histopatologicznych ze zbioru danych użytego w niniejszej pracy.

[TODO zdjęcia 3-4 próbek, wyżej może być opis]

Informacja o wybarwionych strukturach biologicznych jest rozproszona pomiędzy trzema kanałami w kodowaniu RGB. W celu ułatwienia nauki, wykonano konwersję obrazu z przestrzeni barw RGB do przestrzeni HED. Konwersja powoduje zmianę kanałów kolorów na kanały przechowujące informacje o występowaniu trzech podstawowych barwników histopatologicznych- hematoksyliny, eozyny i diaminobenzydyny[https://pdfs.semanticscho lar.org/e0a0/dc42714dd8add295cb43c15110d86af95cf9.pdf, https://scikit-image.org /docs/0 .14.x/a pi/skimage.color.html#skimage.color.rgb2hed]. Przejście z jednego systemu barw do drugiego zachodzi na drodze konwolucji obrazu z macierzą:

przerób to

Kolumny reprezentują kolejne kanały systemu RGB, a wiersze- systemu HED. Macierz ta jest ortogonalna do znormalizowanej macierzy gęstości optycznych poszczególnych kanałów.

Na rycinach 5678-231 przedtsawiono wynik konwersji obrazów histopatologicznych do przestrzeni barw HED. [TODO opisz wskazując na zdjęcia]

[TODO 4-5 obrazów]

Ryciny 2313-12312 przedstawiają zastosowanie obu metod preprocessingu na obrazach histopatologicznych. ##Tak jest, porównaj ze zdjDzięki zastosowaniu transformacji kolorów wszystkich próbek do jednolitej palety barw, informacje o barwieniu są podobne dla tych samych struktur biologicznych na róznych obrazach. ##

[TODO 4-5 obrazków]

* 1. Eksperymentalne dobieranie parametrów sieci

W celu maksymalizacji jakości predykcji sieci funkcja aktywacji i krotność zmniejszenia szerokości obrazka zostały dobrane eksperymentalnie. W tym celu wylosowano 900 zdjęć ze zbioru danych, służących do nauki sieci, oraz 100 zdjęć do walidacji.

# <https://towardsdatascience.com/review-u-net-biomedical-image-segmentation-d02bf06ca760> unet

Testowanie parametrów modelu(Nie mieszać opisu tabeli z opisem danych)

Test modelu

wnioski

Literatura

[1] https://www.who.int/news-room/fact-sheets/detail/cancer (dostęp z dnia 20.05.2019)

[2] http://onkologia.org.pl/nowotwory-pluca-oplucnej-tchawicy/ (dostęp z dnia 20.05.2019)

[3] François Chollet - Deep Learning with Python (2018)

[4] https://grand-challenge.org/ (dostęp z dnia 05.06.2019)

[XXXXXXXXXXXXXXXX] <http://www.script.home.pl/pbkom/roczniki/pdf2011/pbk%2011-3/s475-490.pdf>

[YYYYYYYYYYYYYYYY] <https://docplayer.pl/25857891-Statystyczna-analiza-obrazow-morfologicznych-i-jej-mozliwosci-zastosowania-w-mikroskopii-wirtualnej.html>

[5] https://acdc-lunghp.grand-challenge.org/ (dostęp z dnia 05.06.2019)

[6]https://www.cancer.org/cancer/lung-cancer/prevention-and-early-detection/risk-factors.html (dostęp z dnia 20.05.2019)

[7] https://www.researchgate.net/profile/Lauren\_Collins2/publication/6576120\_Lung\_Cancer  
\_Diagnosis\_and\_Management/links/00b4952d3f9c7b8434000000/Lung-Cancer-Diagnosis-and-Management.pdf (dostęp z dnia 24.05.2019)

[8] https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2797332/ (dostęp z dnia 24.05.2019)

[9] https://www.cancerresearchuk.org/about-cancer/lung-cancer/getting-diagnosed/tests-diagnose/biopsy-through-skin (dostęp z dnia 25.05.2019)

[10] http://www.kolchem.pl/instrukcje/he\_instrukcja.pdf (dostęp z dnia 01.06.2019)

[11] https://www.histology.leeds.ac.uk/what-is-histology/H\_and\_E.php?fbclid=IwAR1yA3pw0\_M\_yTGU9SjaXdOLclvIPC5j1gh6D7b2B903tyPk5fKDAOtBrww (dostęp z dnia 01.06.2019)

[12] http://cs231n.github.io/neural-networks-1/ (dostęp z dnia 20.05.2019)

[13] Nikhil Ketkar - Deep Learning with Python - A Hands-on Introduction - 1E (2017)

[14] Vincent Dumoulin, Francesco Visin - A guide to convolution arithmetic for deep learning (2018)

[15]https://www.researchgate.net/figure/Graph-of-a-recurrent-neural-network\_fig3\_234055140 (dostęp z dnia 21.05.2019)