

17.06.2019  
WT 7:30

Laura Tokarz 236020  
Jakub Małecki 225774  
Jan Zieliński 235793  
Michał Wypustek 232339

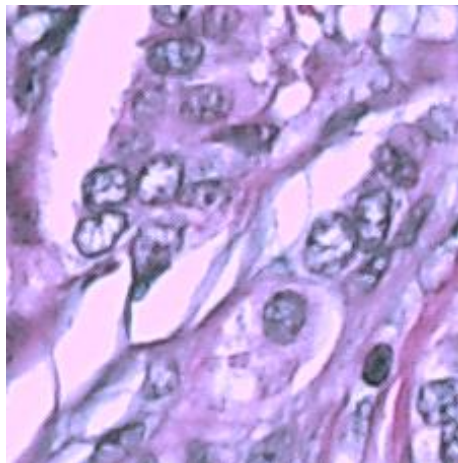
# **Sprawozdanie**

## **PROJEKT ZESPOŁOWY**

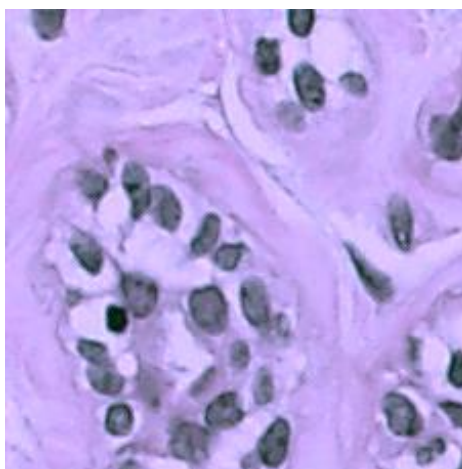
**Temat: Głęboka sieć neuronowa do wykrywania  
raka piersi na podstawie zdjęć histopatologicznych**

## 1. Cel projektu

Celem projektu było stworzenie głębokiej konwolucyjnej sieci neuronowej, która ma za zadanie klasyfikować zdjęcia histopatologiczne w zależności od wykrycia komórek rakowych. Klasyfikacja odbywa się w skali 0-1, tj. do kategorii Cancer lub Normal.



Rysunek 1. Przykładowe zdjęcie histopatologiczne zawierające komórki rakowe



Rysunek 2. Przykładowe zdjęcie histopatologiczne niezawierające komórek rakowych

## 2. Konwolucyjne sieci neuronowe – architektura i opis działania

Głębokie sieci konwolucyjne (CNN) potrafią stopniowo filtrować różne części danych uczących i wyodrębiać ważne cechy w procesie rozpoznawania lub klasyfikacji wzorców. Sieci konwolucyjne porządkują neurony w strukturze 3D: szerokość, wysokość i głębokość. Neurony w każdej warstwie są połączone do małego regionu warstwy poprzedniej. CNN redukują pełne obrazy do pojedynczych wektorów wyjściowych reprezentujących wyniki dla poszczególnych klas. Sieć konwolucyjna jest zwykle sekwencją warstw, które transformują jeden obraz danych do drugiego poprzez funkcję różniczkowalną (w celu umożliwienia wykorzystania algorytmu propagacji wstecznej błędów do dostrojenia parametrów sieci neuronowej). Wyróżniamy trzy typowe warstwy.

Pierwszą z nich jest warstwa konwolucyjna. Zawiera ona zbiór adaptowalnych filtrów np.  $[32 \times 32 \times 3]$ . Konwolucje pozwalają na ekstrakcję prostych cech, np. rozpoznają krawędzie o różnej orientacji lub różnokolorowe plamy, a następnie plastry, koła w kolejnych warstwach. Każda warstwa konwolucyjna zawiera cały zbiór filtrów (np. 8 filtrów), a każdy z nich generuje osobną mapę aktywacji 2D. Układamy te mapy aktywacyjne na sterce wzdłuż wymiaru głębokości sieci i produkujemy obraz wyjściowy.

Kolejną warstwą jest warstwa łącząca (pooling layer) służy do progresywnej redukcji rozmiaru przestrzennego do zredukowania ilości cech i złożoności obliczeniowej sieci. Najczęściej w sieciach konwolucyjnych stosujemy warstwę MaxPool która przesuwając filtry  $2 \times 2$  przez całą macierz wyciągając największą wartość z okna filtra i zapisuje ją do następnej mapy. Najważniejszy powód stosowania warstw łączących jest uchronienie modelu przed przeuczeniem (overfitting).

Zwykle przy uczeniu sieci konwolucyjnych natrafiamy na problem zanikającego gradientu, ponieważ pochodne są z przedziału  $[0, 1]$ , więc wielokrotne przemnażanie prowadzi do bardzo małych liczb prowadząc do coraz mniejszych zmian wag w warstwach coraz dalej położonych od wyjścia sieci, stosując np. propagację wsteczną błędów. Możemy wyeliminować ten problem wykorzystując jednostki ReLU. Są one zdefiniowane jako:  $f(x) = \max(0, x)$  zamiast funkcji logistycznej. Strategia wykorzystująca jednostki ReLU oparta jest na uczeniu pewnych cech dzięki rzadszym aktywacjom tych jednostek. Inną korzyścią jest to, że proces uczenia przebiega zwykle szybciej.

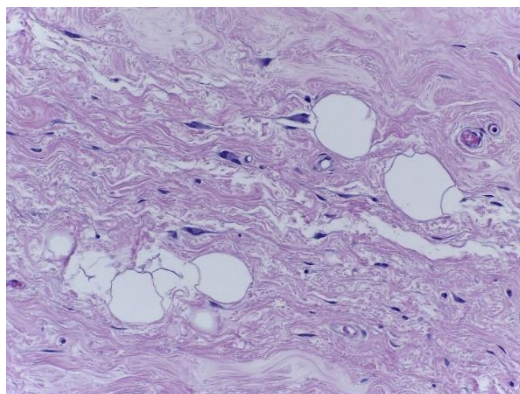
## 3. Stain Normalization

Pierwszym etapem pracy nad projektem było odpowiednie przygotowanie danych uczących. Przede wszystkim należało na danym zbiorze zastosować stain normalization, które polega na transferze rozkładu barw z jednego zdjęcia do drugiego. W przypadku projektu, wybrane zostało jedno zdjęcie pełniące rolę wzorca. Następnie dla wszystkich zdjęć znajdujących się w zbiorze uczącym dokonany został transfer rozkładu barw ze zdjęcia wzorcowego. Operację tę stosujemy się dlatego, iż zdjęcia histopatologiczne mogą być wykonywane różnymi technikami, za pomocą różnych przyrządów oraz przy wykorzystaniu różnych barwników.

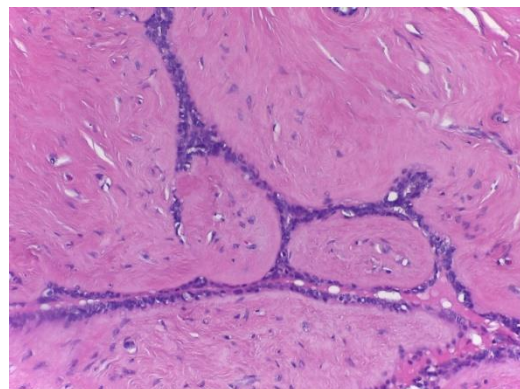
Jeżeli sieć zostałaby nauczona na zbiorze danych o identycznych właściwościach to mogła by ona działać niepoprawnie dla danych charakteryzujących się np. innymi zastosowanymi barwnikami. Dzięki stain normalization, zdjęcie nie pochodzące ze zbioru uczącego, stanie się bardziej do niego podobne.

Przykład działania:

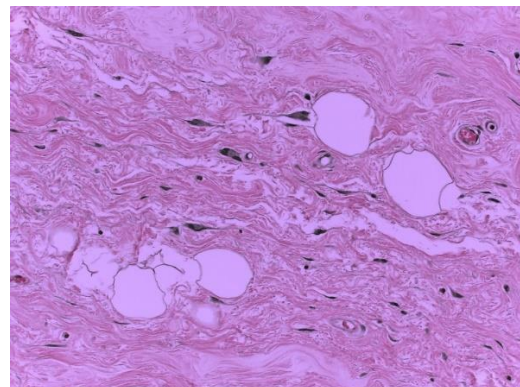
Obraz wejściowy:



Obraz wzorcowy:



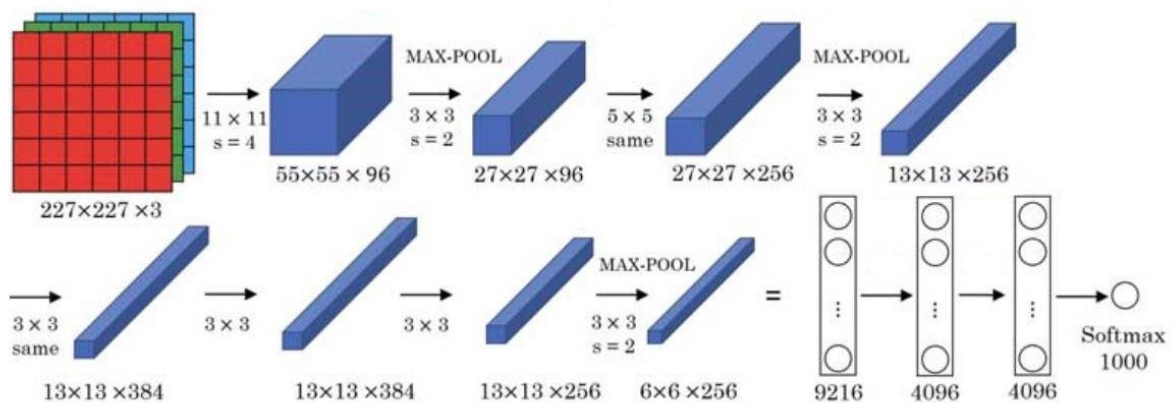
Obraz uzyskany:



#### 4. Sieć neuronowa AlexNet

Wykorzystano głęboką konwolucyjną sieć neuronową AlexNet. Jest to sieć już wytrenowana na ponad milionie obrazów. Posiada ona 8 warstw - 5 konwolucyjnych (convolutional layers) i 3 gęste (fully connected layers).

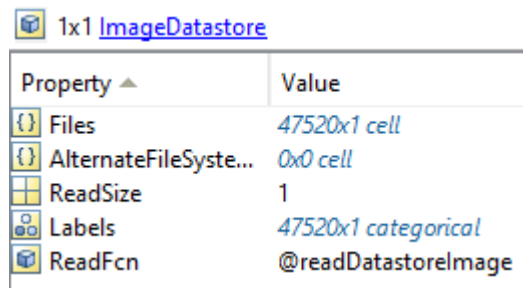
Na wejściu AlexNet przyjmuje obraz RGB o wymiarach  $227 \times 227 \times 3$ . Przechodzi on przez pierwszą warstwę konwolucyjną z 96 mapami cech lub filtrami o rozmiarze  $11 \times 11$  i przesunięciem równym 4. Wymiary obrazu zmieniają się na  $55 \times 55 \times 96$ . Po tych operacjach przechodzimy do warstwy MaxPool z filtrem o rozmiarze  $3 \times 3$  i krokiem 2, zmieniającej rozdzielczość naszego obrazu do  $27 \times 27 \times 96$ . Kolejna warstwa również jest konwolucyjna, posiada jednak 256 map cech lub filtrów o rozmiarze  $5 \times 5$  z przesunięciem równym 1. Za nią znowu występuje warstwa MaxPool, jest ona taka sama jak poprzednia, jednak posiada 256 map cech, co na wyjściu daje nam wymiary obrazu  $13 \times 13 \times 256$ . Trzecia i czwarta warstwa to warstwy konwolucyjne o filtrze  $3 \times 3$  i kroku 1. Posiadają one po 384 mapy cech, co daje nam wymiary obrazu  $13 \times 13 \times 384$ . Ostatnią warstwą konwolucyjną jest warstwa piąta. Jest identyczna jak dwie poprzednie, z tą różnicą, że posiada 256 filtrów. Następnie występuje kolejna warstwa MaxPool z filtrem  $3 \times 3$ , krokiem równym 2 i mapami cech w ilości 256. Po tych operacjach obraz ma wymiary  $6 \times 6 \times 256$ . Warstwa szósta jest warstwą gęstą (fully connected), posiada 9216 map cech, każda o rozmiarze  $1 \times 1$ . Dwie kolejne warstwy również są gęste o 4096 jednostkach. Wszystkie warstwy używają funkcji aktywacji ReLU. Ostatnia warstwą jest warstwa wyjściowa (o aktywacji Softmax), która może przyjmować 1000 wartości.



Rysunek 3. Architektura sieci AlexNet

#### Tworzenie bazy danych:

Nasza baza danych zawierała ok. 200 zdjęć histopatologicznych piersi, ale o innych wymiarach. Na początku zastosowano normalizację na podstawie jednego wzorca, aby każde zdjęcie miało identyczną przestrzeń barw. Następnie podzielono każde ze zdjęć na 54 mniejsze wycinki o wymiarach  $227 \times 227$  i każdy z nich obrócono w różne strony i odbito. Zwiększyło to ilość zdjęć w naszej bazie do prawie 50 tysięcy i zmniejszyło prawdopodobieństwo przeuczenia sieci (overfitting).



Property ▲	Value
Files	47520x1 cell
AlternateFileSyste...	0x0 cell
ReadSize	1
Labels	47520x1 categorical
ReadFcn	@readDatastoreImage

Rysunek 4. Zestaw danych przechowywany w ImageDatastore

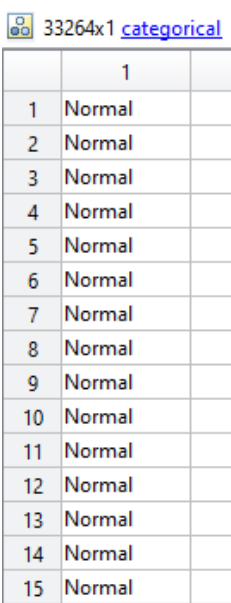
### Konfigurowanie sieci:

Utworzono dwie kategorie: Cancer i Normal. Obie zawierają identyczną ilość zdjęć.

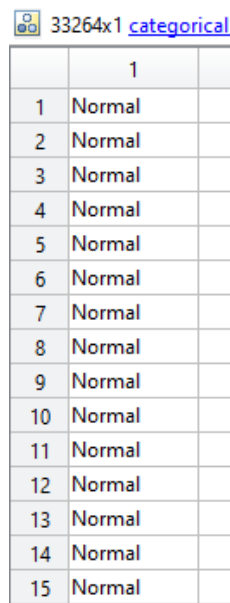
Label	Count
Cancer	23760
Normal	23760

Rysunek 5. Podział na kategorie

Następnie załadowano wytrenowaną sieć AlexNet. Podzielono dane na zestaw do uczenia i zestaw do testowania w stosunku 30:70. Funkcję aktywacji zastosowano na warstwie gęstej (fully connected) fc7. Kolejnym krokiem było utworzenie przewidywanych i testowych etykiet. Przewidywane otrzymano dzięki funkcji fitcecoc oraz predict. Testowe etykiety były nam znane.



a)



b)

Rysunek 6. a) etykiety przewidywane, b) etykiety testowe

Następnie porównano oba rodzaje etykiet dzięki funkcji confusionmat i przekonwertowano na procenty używając funkcji bsxfun. Pozostało jedynie wyliczyć średnią skuteczność tak skonfigurowanej sieci AlexNet. Wyniosła ona 76 %.

### Testowanie:

Do testowania wykorzystano zdjęcia, które nie były używane do uczenia. Ponownie na każdym z takich zdjęć zastosowano normalizację za pomocą tego samego wzorca, którego użyto do normalizacji wszystkich poprzednich zdjęć. Następnie podzielono każde zdjęcie na wycinki o rozmiarach 227x227. Testowano każde z 54 otrzymanych zdjęć i klasyfikowano do kategorii Cancer lub Normal. Wyniki zapisywano w wektorze, dla kategorii Cancer była to wartość 1, a dla kategorii Normal wartość 2.

6x9 double

	1	2	3	4	5	6	7	8	9
1	1	2	2	1	2	2	2	2	2
2	2	1	2	2	2	1	2	2	2
3	2	2	2	2	2	2	2	2	2
4	1	2	2	2	1	2	2	2	2
5	2	2	2	2	2	2	2	2	1
6	1	2	1	2	2	1	2	2	1

Rysunek 7. Wektor wyników

Na podstawie danych w wektorze wyliczano średnią. Jeżeli średnia była mniejsza od 1.5 (więcej zdjęć sklasyfikowanych jako zawierające komórki rakowe), to klasyfikowano całe oryginalne zdjęcie do kategorii Cancer. Natomiast jeśli średnia była większa od 1.5 (więcej zdjęć sklasyfikowanych jako niezawierające komórek rakowych), to klasyfikowano całe oryginalne zdjęcie do kategorii Normal.

```
srednia =

    1.7963

ans =

    'The loaded 1 image belongs to Normal class'
```

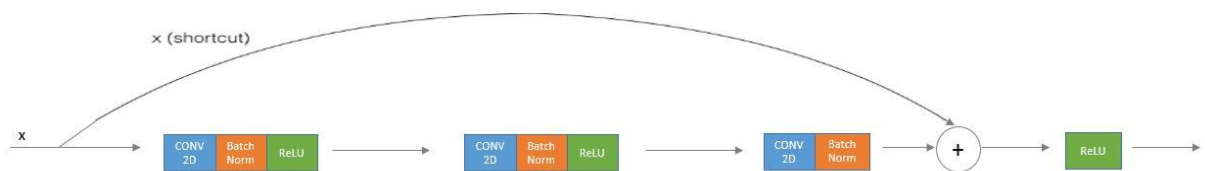
Rysunek 8. Klasyfikacja zdjęcia do kategorii Normal



## 5. Sieć neuronowa ResNet-50

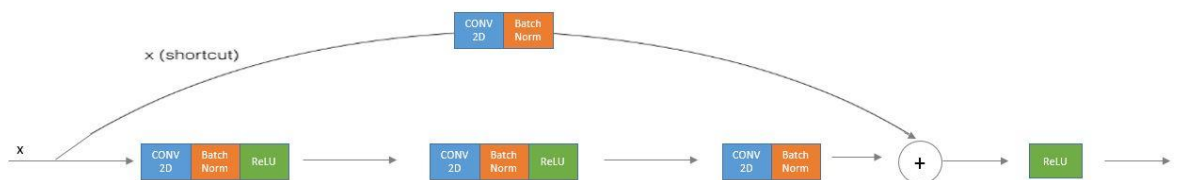
Wykorzystano głęboką konwolucyjną sieć neuronową ResNet-50. Jest to sieć już wytrenowana na ponad milionie obrazów. Posiada ona 50 warstw.

W przypadku sieci ResNet możemy wyróżnić 2 bloki: tożsamości i konwolucyjny. Pierwszy z nich odpowiada przypadkowi, gdzie aktywacja wejścia ma taki sam wymiar jak aktywacja wyjścia. Poniżej znajduje się przykład bloku tożsamości, gdzie górna ścieżka to „ścieżka skrótu”, a dolna to „ścieżka główna”.



Rysunek 9. Blok tożsamości sieci ResNet

Drugim blokiem jest blok konwolucyjny. Dodajemy go w momencie, gdy wymiary wejściowe i wyjściowe nie pasują do siebie.

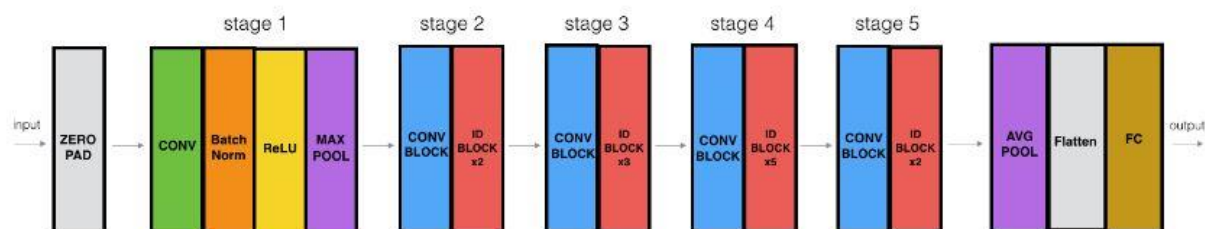


Rysunek 10. Blok konwolucyjny sieci ResNet

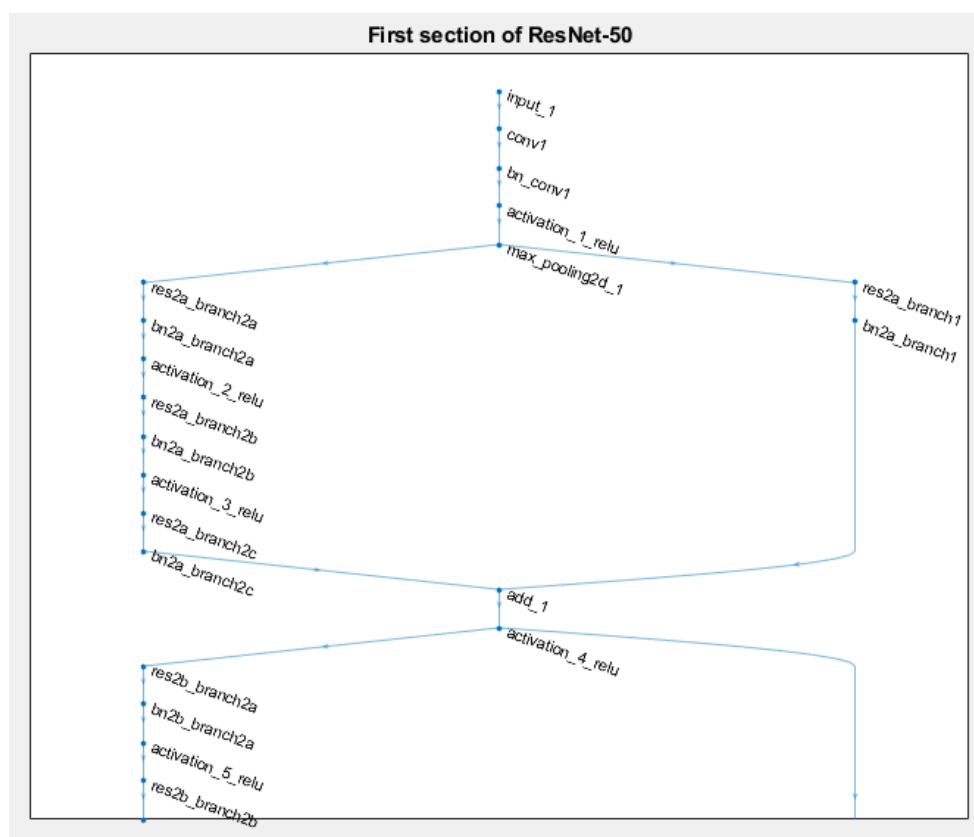
Na wejściu podajemy zdjęcie w rozmiarach 224x224x3. Pierwszy etap zawiera warstwę konwolucyjną z 64 filtrami o rozmiarze 7x7 i krokiem przesunięcia równym 2. Następnie stosujemy normalizację Batch, która dostosowuje i skaluje aktywacje. Kolejną warstwą jest MaxPool, z filtrem 3x3 i krokiem 2. Etap drugi zawiera blok konwolucyjny z 3 zestawami filtrów 64x64x256 i krokiem równym 1 oraz dwa bloki tożsamości z takim samym zestawem filtrów jak poprzedni blok. Etap trzeci również zawiera blok konwolucyjny z 3 zestawami filtrów, ale o rozmiarze 128x128x512 i przesunięciu 2. Za nim występują trzy bloki tożsamości o takich samych zestawach filtrów. Etap czwarty zawiera jeden blok konwolucyjny i 4 bloki tożsamości, wszystkie o rozmiarze filtrów 256x256x1024.



Przesunięcie dla bloku konwolucyjnego wynosi 2. Piąty etap ma filtry 256x256x2048, jeden blok konwolucyjny o kroku równym 2 i dwa bloki tożsamości. Następnie wykorzystywana jest warstwa AveragePooling, później warstwa spłaszczona, a na końcu warstwa gęsta (fully connected), która zmniejsza rozmiar swojego wejścia do ilości klas, używając aktywacji Softmax.



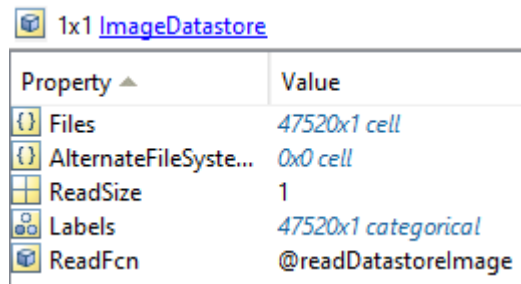
Rysunek 11. Architektura sieci ResNet-50



Rysunek 12. Architektura sieci ResNet-50 uzyskana za pomocą funkcji plot w Matlabie

## Tworzenie bazy danych:

Tworzenie bazy danych przebiegało dokładnie tak jak dla sieci AlexNet, z jedyną różnicą, że zdjęcia były dzielone na wycinki o wymiarach 224x224.



1x1 ImageDatastore

Property ▲	Value
Files	47520x1 cell
AlternateFileSyste...	0x0 cell
ReadSize	1
Labels	47520x1 categorical
ReadFcn	@readDatastoreImage

Rysunek 13. Zestaw danych przechowywany w ImageDatastore

## Tworzenie sieci:

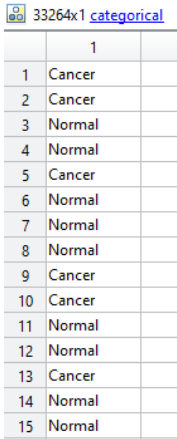
Utworzono dwie kategorie: Cancer i Normal. Obie zawierają identyczną ilość zdjęć.

Label	Count
Cancer	23760
Normal	23760

Rysunek 14. Podział na kategorie

Następnie załadowano wytrenowaną sieć ResNet-50. Podzielono dane na zestaw do uczenia i zestaw do testowania w stosunku 30:70. Funkcję aktywacji zastosowano na warstwie gęstej (fully connected) fc1000. Kolejnym krokiem było utworzenie przewidywanych i testowych etykiet. Podobnie jak w przypadku sieci AlexNet przewidywane otrzymano dzięki funkcji fitcecoc oraz predict. Testowe etykiety również były nam znane.

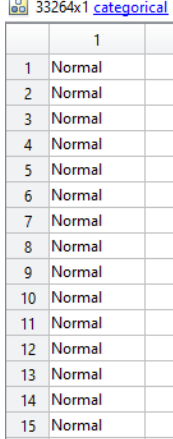
a)



33264x1 categorical

	1
1	Cancer
2	Cancer
3	Normal
4	Normal
5	Cancer
6	Normal
7	Normal
8	Normal
9	Cancer
10	Cancer
11	Normal
12	Normal
13	Cancer
14	Normal
15	Normal

b)



33264x1 categorical

	1
1	Normal
2	Normal
3	Normal
4	Normal
5	Normal
6	Normal
7	Normal
8	Normal
9	Normal
10	Normal
11	Normal
12	Normal
13	Normal
14	Normal
15	Normal

Rysunek 15. a) etykiety przewidywane, b) etykiety testowe

Następnie porównano oba rodzaje etykiet dzięki funkcji confusionmat i przekonwertowano na procenty używając funkcji bsxfun. Pozostało jedynie wyliczyć średnią skuteczność dla tak skonfigurowanej sieci ResNet. Wyniosła ona 89 %.

### Testowanie:

Do testowania wykorzystano zdjęcia, które nie były używane do uczenia. Ponownie na każdym z takich zdjęć zastosowano normalizację za pomocą tego samego wzorca, którego użyto do normalizacji wszystkich poprzednich zdjęć. Następnie podzielono każde zdjęcie na wycinki o rozmiarach 224x224. Testowano każde z 54 otrzymanych zdjęć i klasyfikowano do kategorii Cancer lub Normal. Wyniki zapisywano w wektorze, dla kategorii Cancer była to wartość 1, a dla kategorii Normal wartość 2.

6x9 double

	1	2	3	4	5	6	7	8	9
1	1	1	1	1	1	1	1	1	1
2	1	1	1	1	1	1	1	1	1
3	1	1	1	1	1	1	1	1	2
4	1	1	1	1	1	1	2	2	2
5	1	1	1	1	1	1	1	2	2
6	2	1	1	1	1	2	2	2	1

Rysunek 16. Wektor wyników

Na podstawie danych w wektorze wyliczano średnią. Jeżeli średnia była mniejsza od 1.5 (więcej zdjęć sklasyfikowanych jako zawierające komórki rakowe), to klasyfikowano całe oryginalne zdjęcie do kategorii Cancer. Natomiast jeśli średnia była większa od 1.5 (więcej zdjęć sklasyfikowanych jako niezawierające komórek rakowych), to klasyfikowane całe oryginalne zdjęcie do kategorii Normal.

```
srednia =

    1.1852

ans =

    'The loaded 1 image belongs to Cancer class'
```

Rysunek 17. Klasyfikacja zdjęcia do kategorii Cancer

## 6. Wnioski:

Typowa sztuczna sieć neuronowa wykorzystująca połączenia neuronów na zasadzie każdy-z-każdym może w łatwy sposób ulegać przeuczeniu (overfitting) dla średniej wielkości i dużych obrazów, gdyż np. dla obrazu o wielkości  $227 \times 227 \times 3$  (gdzie  $227 \times 227$  to wielkość obrazu w pikselach, a 3 to ilość kanałów kolorów R, G i B) uzyskujemy potężną ilość parametrów wolnych, które muszą zostać wytrenowane:  $(227 \times 227 \times 3 = 154587)$  w stosunku do ilości uczonych obrazów (wzorców, obiektów). Natomiast w strukturze konwolucyjnej CNN, każdy neuron połączony jest tylko do lokalnego regionu obrazu wejściowego. Lokalne obszary (regiony) zdefiniowane są przez szerokość i wysokość, zaś głębokość przebiega przez całą warstwę obrazu wejściowego. Zakres połączeń wzdłuż osi głębokości CNN jest zawsze równy głębokości obrazu wejściowego (warstwy poprzedniej).

Możemy więc śmiało stwierdzić, że algorytmy uczenia głębokiego, dają nam o wiele więcej możliwości niż zwykłe sieci neuronowe, a między innymi:

- uczenie hierarchicznych kaskad warstw lub podsieci,
- wykorzystanie różnych nadzorowanych i nienadzorowanych strategii uczenia poszczególnych warstw oraz stopniowy rozwój struktury sieci,
- stopniowe uczenie/adaptację neuronów lub innych jednostek obliczeniowych i kolejnych warstwach,
- łączenie neuronów pomiędzy różnymi warstwami, a nie tylko pomiędzy kolejnymi warstwami

Porównano również stworzone przez nas sieci głębokie AlexNet (skuteczność 76%) i ResNet-50 (skuteczność 89%). Wyliczone średnie skuteczności wskazują na lepsze działanie sieci ResNet-50. Przeprowadzone ręcznie testowanie na osobnych zdjęciach również to potwierdza. Prawdopodobnie wynika to z większej ilości warstw jaką posiada sieć ResNet-50 (o 42 warstwy więcej niż AlexNet) oraz z faktu że sieć AlexNet powstała w 2012 roku, natomiast ResNet w roku 2015. Można na tej podstawie stwierdzić, że ResNet będzie miała ogólnie lepsze działanie, bo jest bardziej dopracowana i udoskonalona. Wskazuje na to nawet zamieszczone poniżej porównanie.

CONVOLUTIONAL NEURAL NETWORKS ARCHITECTURES			
Architecture	Top-1 Accuracy	Top-5 Accuracy	Year
Alexnet	57.1	80.2	2012
Resnet-50	75.2	93	2015

Rysunek 18. Porównanie skuteczności sieci ResNet i AlexNet