

RAPORT Z WYKONANIA PROJEKTU W RAMACH PRACOWNI PROJEKTOWEJ

Temat: Medical image analysis using machine learning

Opis: Pierwszym etapem projektu jest wykonanie przeglądu aktualnych prac badawczych w dziedzinie automatycznego przetwarzania obrazów medycznych z użyciem uczenia maszynowego, w szczególności z wykorzystaniem uczenia głębokiego. Drugim etapem będzie implementacja wybranych metod i ewaluacja ich skuteczności, a następnie określenie planu rozwijania wybranych algorytmów w trakcie przyszłych prac dyplomowych.

Opiekun: Filip Malawski

Realizacja:

Anna Banaszak

Natalia Organek

Wybrany temat projektu do implementacji:

HuBMAP - Hacking the Kidney <https://www.kaggle.com/c/hubmap-kidney-segmentation>

Tematem projektu jest analiza zdjęć pojedynczych komórek nerek w poszukiwaniu aktywnych funkcjonalnych jednostek kłębuszków (glomeruli FTU). Zadanie pojawiło się w ramach większego projektu tworzenia mapy ludzkiego ciała na poziomie funkcjonalnych jednostek tkanki kłębuszkowej - [Human BioMolecular Atlas Program](#) (HuBMAP) jako konkurs na platformie Kaggle.

Zbiór danych z platformy Kaggle:

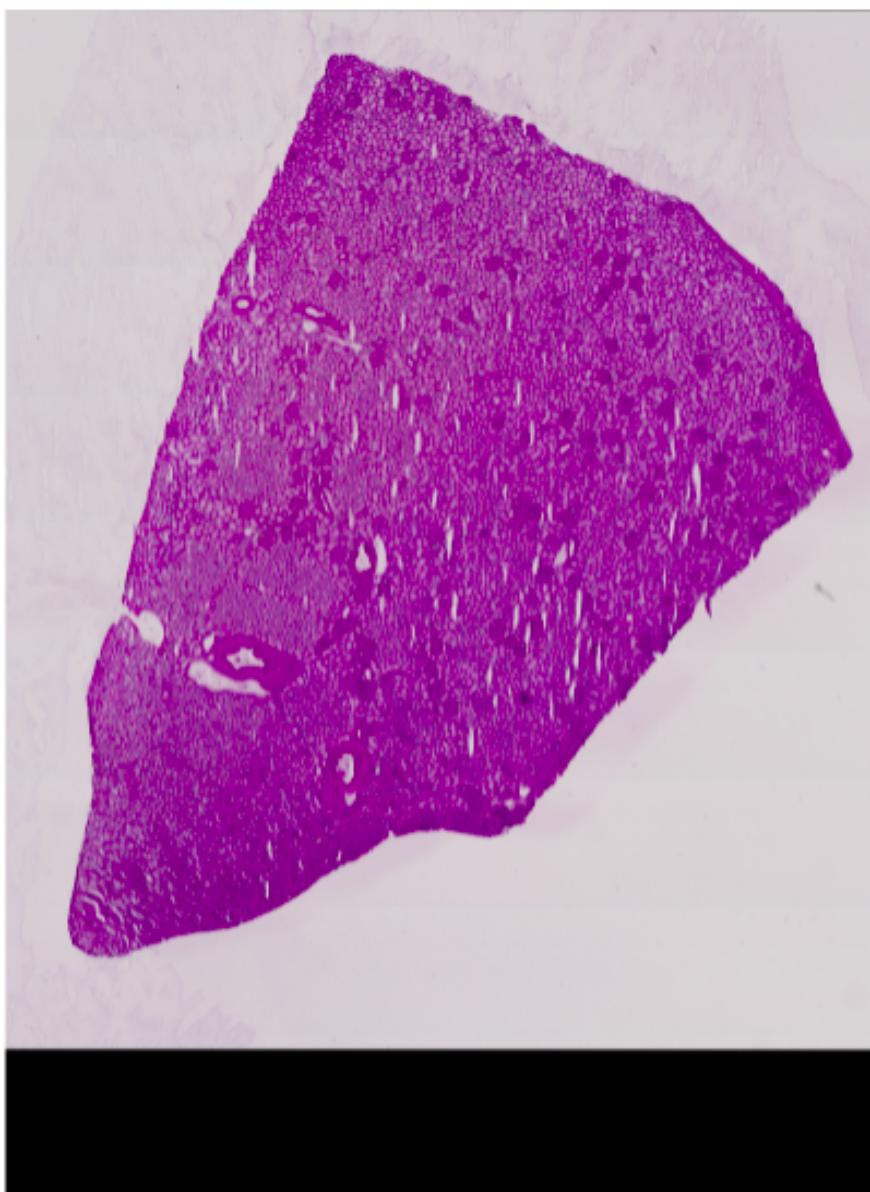
Jest to 13 zdjęć komórek nerkowych (część zamrożonych, część z formaliną). Są to duże zdjęcia (500MB - 5GB), o dużej rozdzielczości, zapisane w formacie .TIFF. Zbiór jest podzielony na część treningową i testową (8 vs 5 zdjęć). Opisy zdjęć (labels) podane są w postaci JSONów, które zawierają współrzędne wielokątów obrysowujących kłębuszki.

1. Stworzenie zbioru danych. (Anna Banaszak)

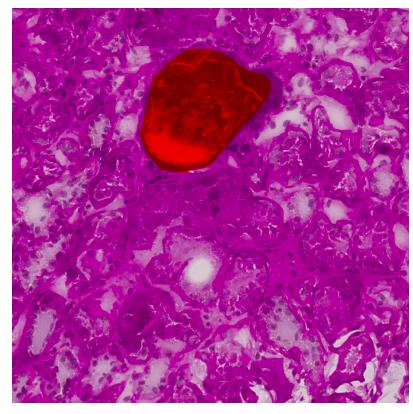
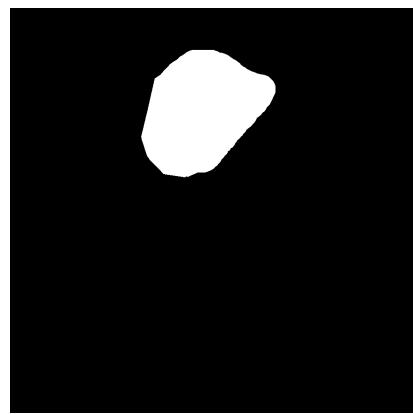
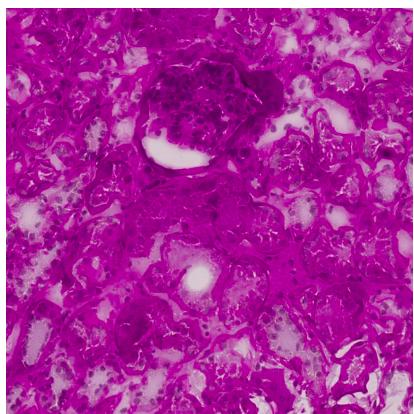
W celu zastosowania posiadanego zbioru danych do uczenia sieci neuronowej należało podzielić pliki TIFF na wiele mniejszych obrazów. Oryginalny obraz zawiera części mało istotne dla rozwiązywanego problemu, czyli czarne lub białe tło. Na podstawie histogramu koloru, do przetestowania dostępnych modeli, odfiltrowano fragmenty, na których znajdowało się mniej powierzchni kłębuszka niż zadany próg. Próg wyznaczono na 300 pikseli przy rozdzielczości 1024x1024.

Podział obrazków może być elastycznie dostosowany do wymagań rozmiaru oraz nakładania się na siebie obrazków. Zdecydowano się na wymiar 1024x1024 przeskalowany do 256x256 w celu redukcji zajmowanej pamięci.

Przykładowa zawartość pliku TIFF:



Przykładowy fragment, jego maska i nałożony wielobok:



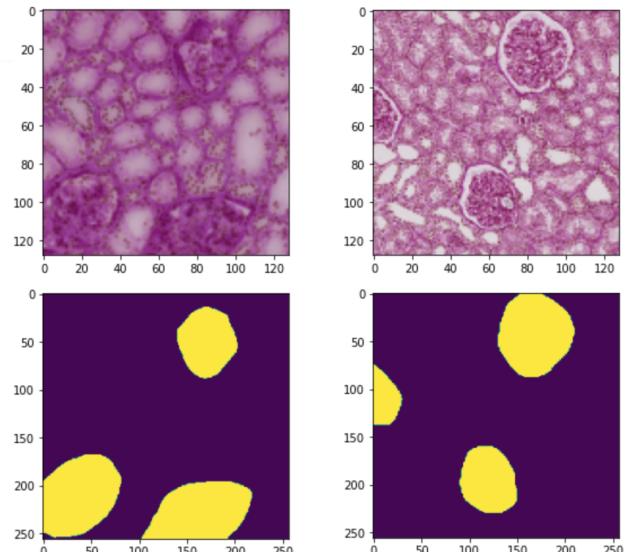
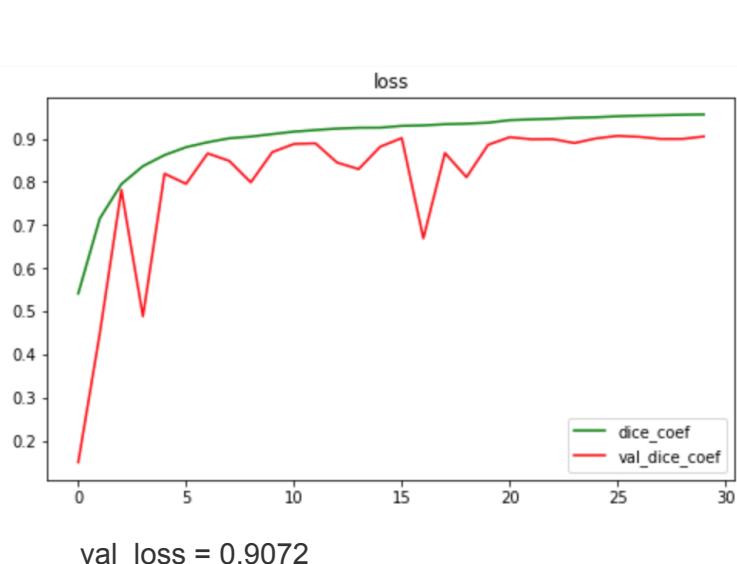
2. UNet

Aby przetestować architekturę UNet, szeroko stosowaną przy problemie segmentacji na obrazach medycznych, utworzono projekt Google Collab z wykorzystaniem biblioteki keras oraz silnika tensorflow na GPU.

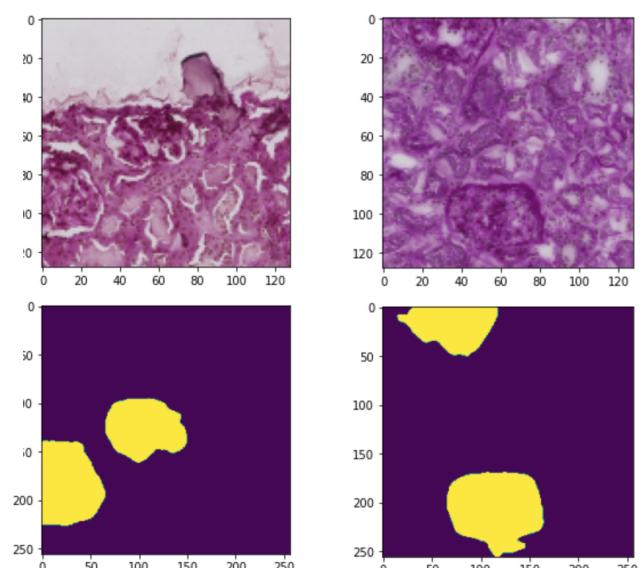
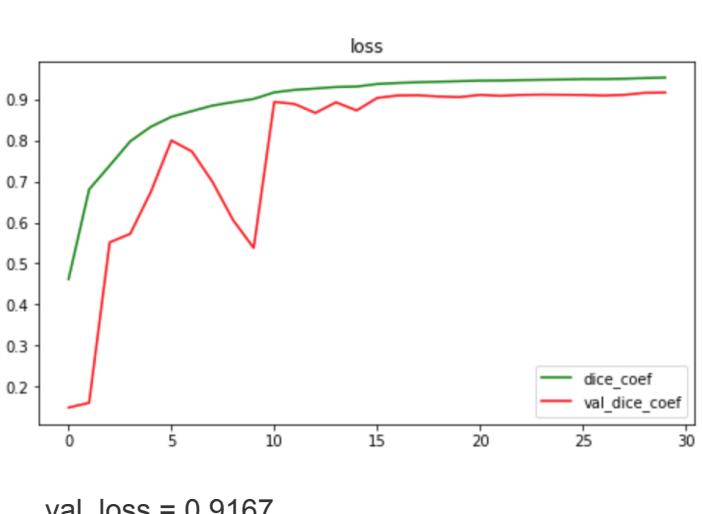
UNet został zdefiniowany dla obrazów rgb rozmiaru 128x128x3. Zastosowano redukcję learning rate o połowę przy braku poprawy `val_loss` o więcej niż 1e-5 przez 4 epoki. Zapisywano wagi dla najlepszych wyników aby możliwe było wznowienie uczenia lub wykonanie predykcji dla zapisanego modelu.

Ewaluacje modelu przeprowadzono za pomocą metryki Dice score.

Wykres historii uczenia modelu z funkcją aktywacji relu i rozmiarem batcha 8 oraz przykładowe predykcje:



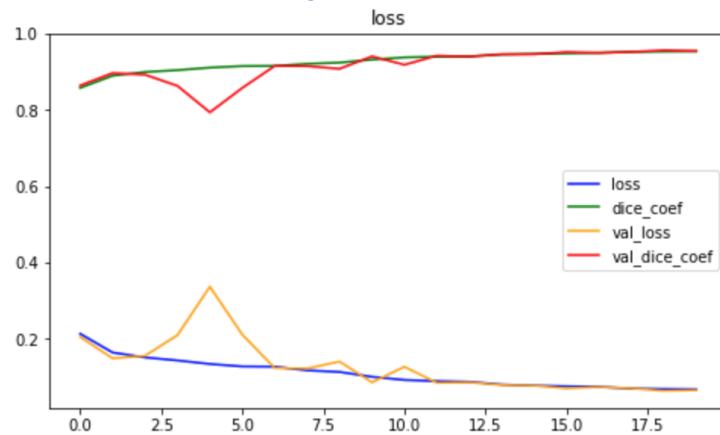
Wykres historii uczenia modelu z funkcją aktywacji leaky relu i rozmiarem batcha 8 oraz przykładowe predykcje:



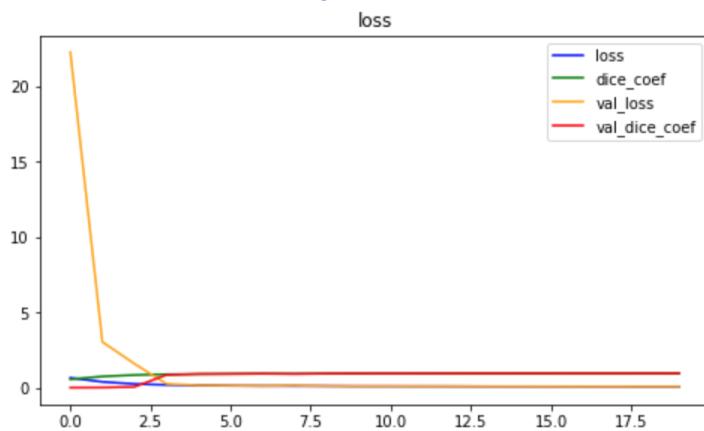
model i ewaluacja - Anna Banaszak
ładowanie danych generatorem - Anna Banaszak
ładowanie danych TFDataset - Natalia Organek

3. Test rozszerzeń architektury UNet:

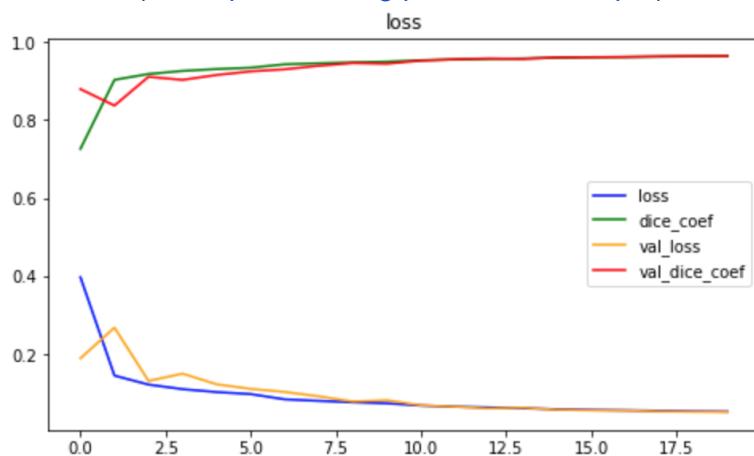
- a. UNet++ (ref. <https://arxiv.org/pdf/1807.10165.pdf>)



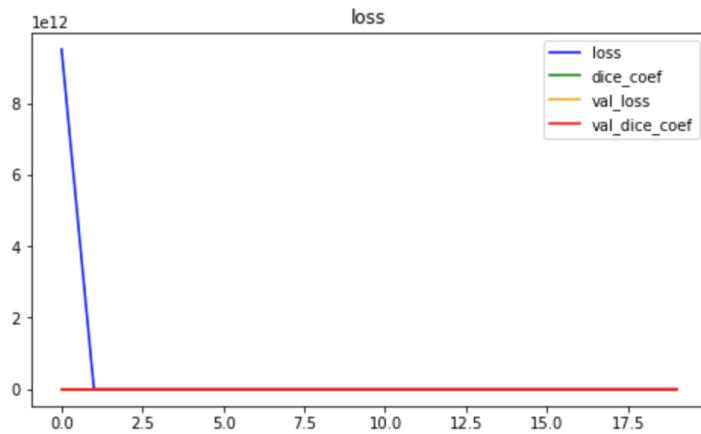
- b. RESNet (ref. <https://arxiv.org/pdf/1711.10684.pdf>)



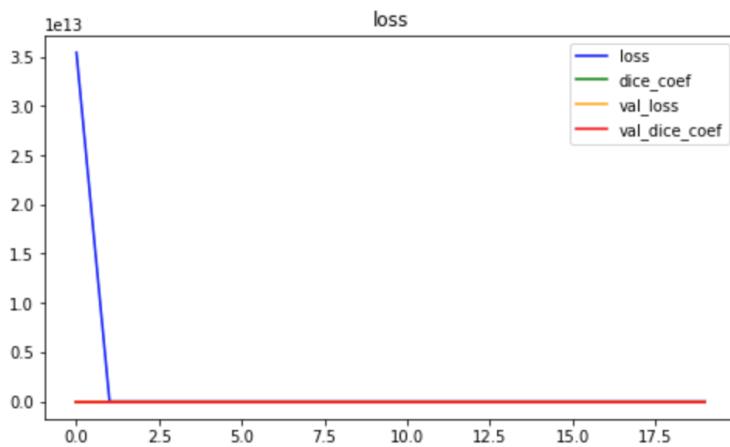
- c. Dense UNet (ref. <https://arxiv.org/pdf/1709.07330.pdf>)



d. R2UNet (ref. <https://arxiv.org/pdf/1802.06955.pdf>)

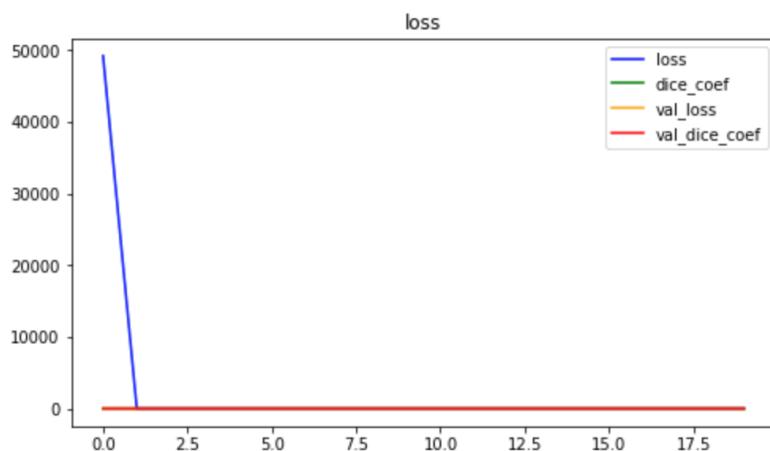


e. Attention R2UNet (ref. <https://arxiv.org/pdf/1802.06955.pdf>)



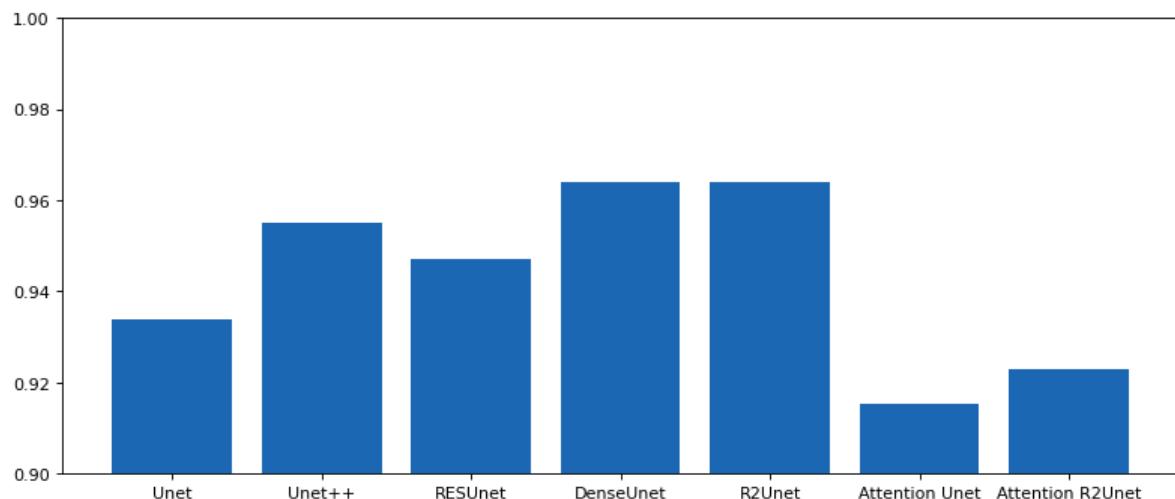
f. Attention UNet (ref.

https://smcdonagh.github.io/papers/attention_u_net_learning_where_to_look_for_the_pancreas.pdf)



	Unet++	RESUnet	DenseUnet	R2Unet	Attention Unet	Attention R2Unet
Współczynnik Dice'a	0.9549	0.9470	0.9639	0.9641	0.9153	0.9229
Końcowy loss	0.0651	0.0776	0.0521	0.0511	0.1234	0.1122

Tabela: Końcowe współczynnik Dice'a i loss dla porównywanych rozszerzeń architektur sieci Unet.



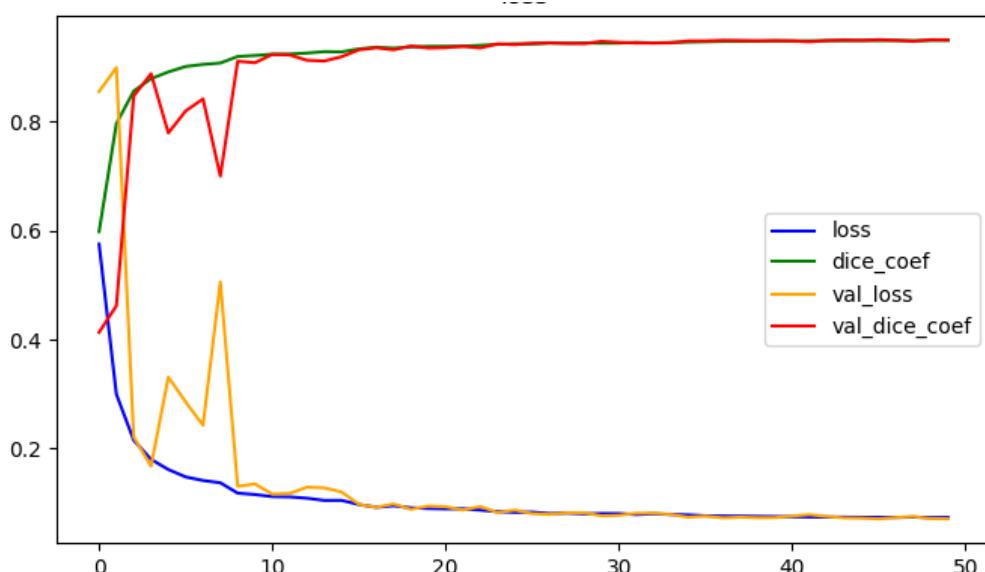
Wykres: Porównanie współczynnika Dice'a dla badanych rozszerzeń architektur.

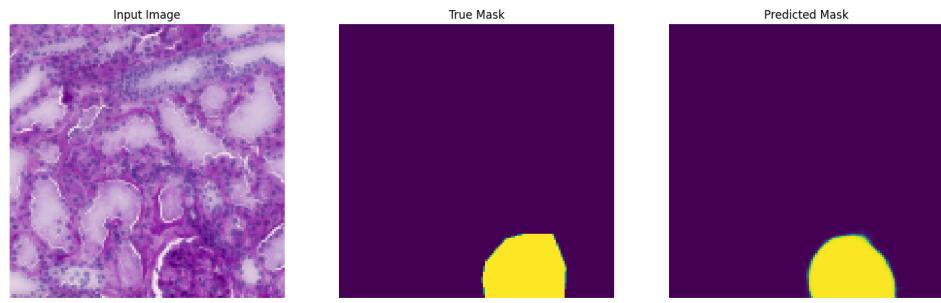
modele i ewaluacja - Natalia Organek

4. Trenowanie sieci na Prometeuszu - Anna Banaszak

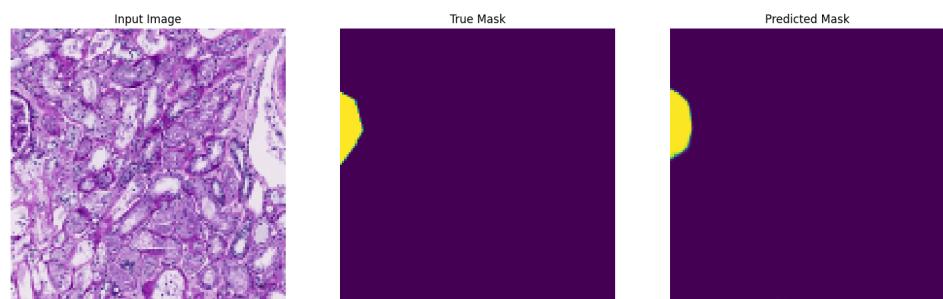
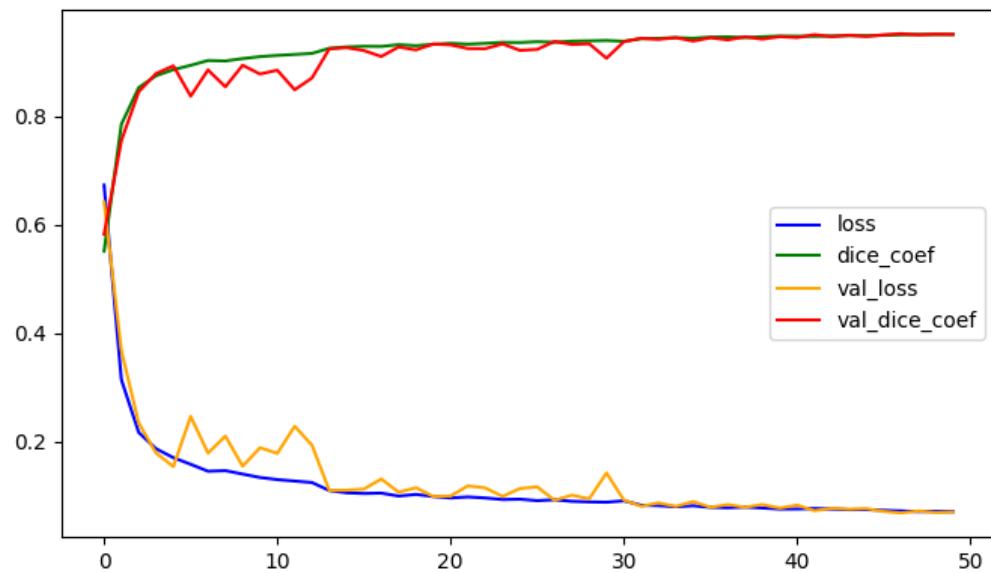
Wykorzystane modele pochodzą z modułu keras-unet-collection.

Unet z funkcją aktywacji relu:

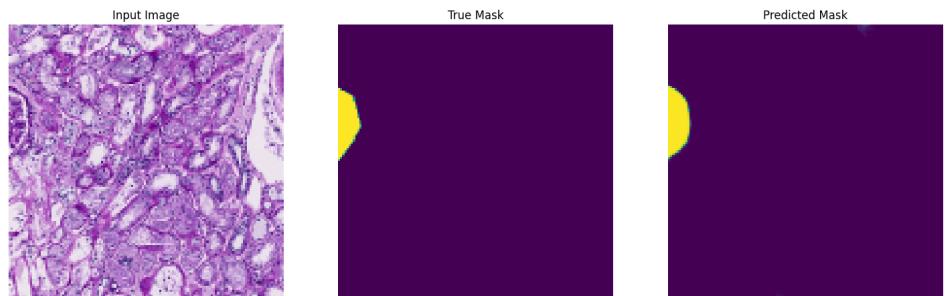
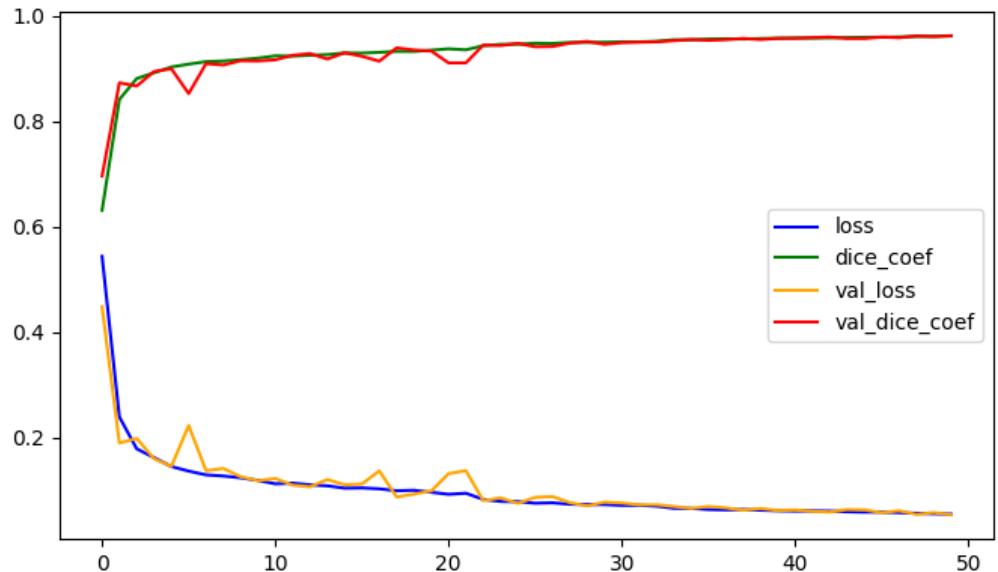




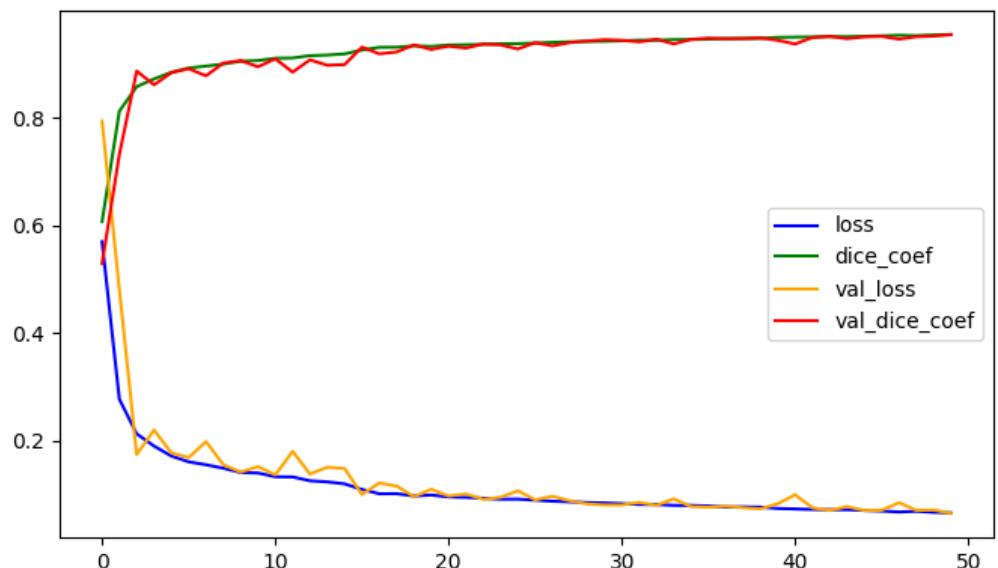
Unet z funkcją aktywacji leaky relu:

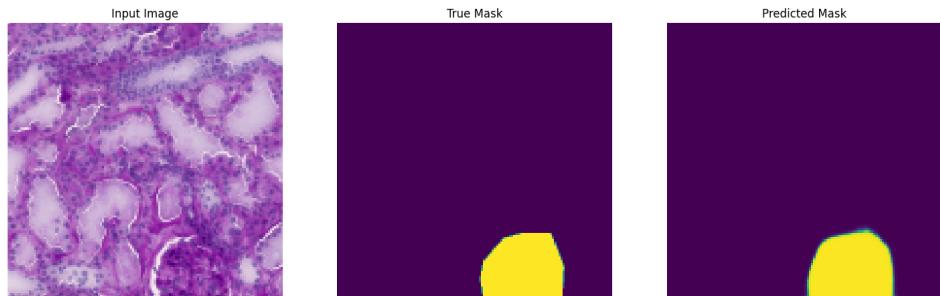


R2Unet z funkcją aktywacji relu:



Unet++ z funkcja aktywacją leaky relu:





W celu weryfikacji poprawności działania modeli pliki tiff podzielono na zbiór treningowy (3400 przykładów) i testowy (1088 przykładów). Następnie utworzono z nich datasety plików png, których wytrenowane i przetestowane zostały sieci z wynikami przedstawionymi w tabeli:

Model	Unet z funkcją aktywacji ReLu	Unet z funkcją aktywacji Leaky ReLu	Unet ++	R2Unet	ResUnet	Unet 3+
Testowy Współ. Dice'a	0.8958	0.8966	0.8933	0.9008	0.9003	0.8988

Model R2Unet został następnie wytrenowany na całym zbiorze treningowym w celu przetestowania na platformie Kaggle.

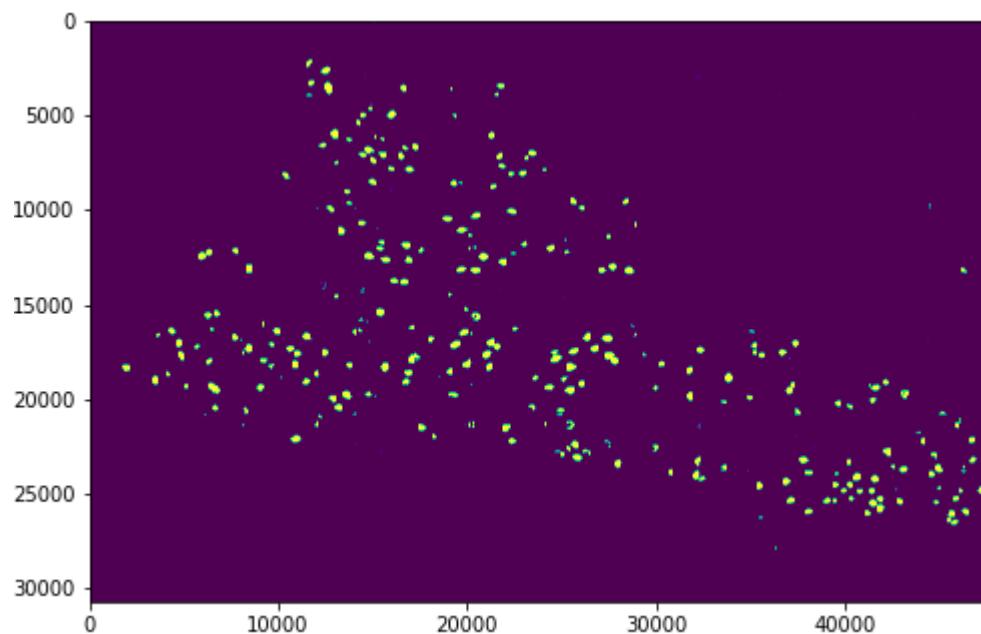
4. Test modelu na platformie Kaggle - Natalia Organek

Do sprawdzenia jakości modelu, organizatorzy udostępnili dane testowe (publiczne - widoczne dla uczestników konkursu i prywatne - niewidoczne dla uczestników). Ponieważ używałyśmy modelu z biblioteki kerasa (R2Unet - przy testach na Prometeuszu okazał się najlepszy), nie mogłyśmy go użyć bezpośrednio w notebooku konkursowym. W związku z tym nie mogłyśmy sprawdzić modelu na prywatnych danych. Obeszłyśmy to, wyliczając wcześniej wyniki dla publicznego zbioru danych w oddzielnego notebookach i podając je do notebooka konkursowego. Dla publicznych danych testowych uzyskałyśmy skuteczność modelu około 87%.

Submission and Description	Status	Private Score	Public Score
notebook07442b6547 (version 8/8) 4 days ago by Natalia Organek Notebook notebook07442b6547 Version 8	Succeeded	0.0000	0.8670

Wynikiem był plik csv z zakodowaną w rle maską całego pliku tiff. Oryginalny plik tiff był dzielony na obszary (tak jak robiłyśmy to podczas trenowania modelu), dla których model R2Unet (z wagami wyznaczonymi na Prometeusz) wyznaczał maski, które z kolei były składane w całość, a następnie kompresowane do rle.

Przykładowa maska dla całego pliku tiff:



Notebook dla jednego z przykładów testowych:

https://colab.research.google.com/drive/1_P7Sqa9GJT1LT1zUN4j2PK4j_TDkqPvc?usp=sharing.