

Techniki obrazowania medycznego

Skład osobowy:

Górak Kamil
Kozioł Bartosz
Kwater Jakub
Mermon Gabriela

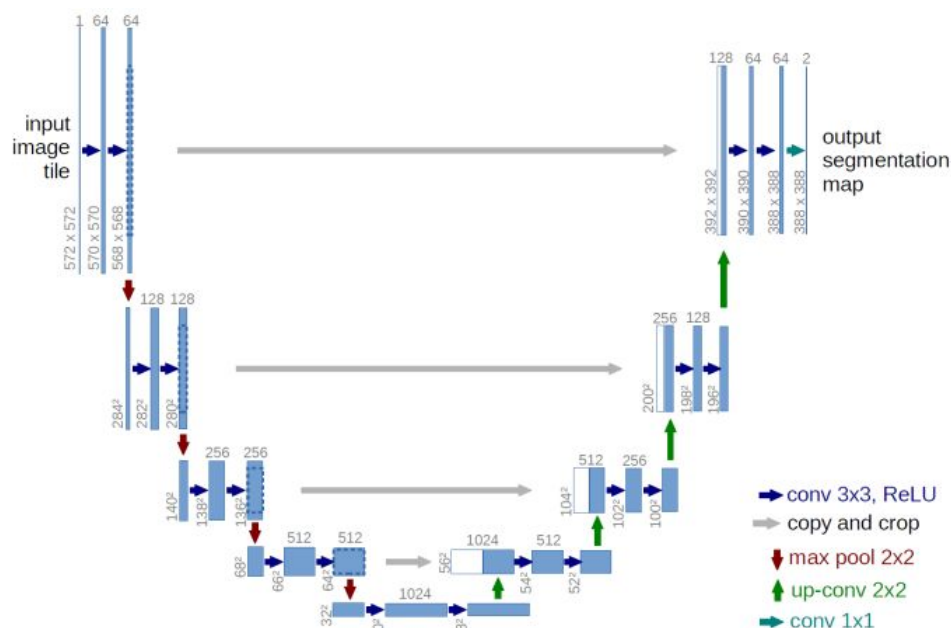
Zastosowane rozwiązania:

Wizualizacja danych:

Pierwszym krokiem było pobranie danych z publicznego repozytorium ... na githubie gdzie otrzymaliśmy dane dla 300 pacjentów. Posługując się funkcją `load_case()` możliwe jest wczytanie danych (obrazów CT i obrazów wysegmentowanych nerek i guzów w nich obecnych) dla danego pacjenta. Korzystając z metody `get_fdata()` z biblioteki `nibabel` pozyskano dane dotyczące liczby wykonanych przekrojów. Za pomocą funkcji `visualize()` pobrano zdjęcia w formacie `.png` i sprawdzono czy są one w skali szarości czy też RGB. Wyświetlono obrazy CT i obrazy segmentacji aby lepiej zwizualizować dane na których będzie się uczyć sieć.

U-net:

Na początku zdefiniowano wymiary obrazów, uzyskanych w poprzednim kroku, które będą dalej przesyłane do sieci jako dane treningowe. Do budowy sieci wykorzystano framework `tensorflow`, przez wzgląd na dobrze rozbudowaną dokumentację oraz bazę pomocy naukowych w postaci przykładowych kodów i tutoriali. Proponowana sieć składa się ze ścieżki kodującej i dekodującej. Zaimplementowana struktura jest charakterystyczna dla tej sieci neuronowej i stanowi realizację poniżej przedstawionego modelu:



Trening i ewaluacja:

Została podjęta próba wczytania obrazów CT i obrazów segmentacji dla 4 pacjentów, jednak sposób ich implementacji do proponowanej sieci nie jest prawidłowy. Problemem jest różnica w liczbie przekrojów dla każdego z pacjentów. W planach jest dokonanie treningu za pomocą metody "*model.fit()*" do której zostaną podane parametry "*callbacks*" takie jak "*EarlyStopping*" i "*TensorBoard*". Pierwsza zapobiegnie overfittingowi, a druga umożliwi podgląd krzywych procesu uczenia sieci i lepsza analize uzyskanych wyników. Testy zostaną wykonane na pacjentach od numeru 210 gdzie dane nie zawierają już obrazów segmentacji. W tym celu planowane jest wykorzystanie funkcji "*model.predict()*".