

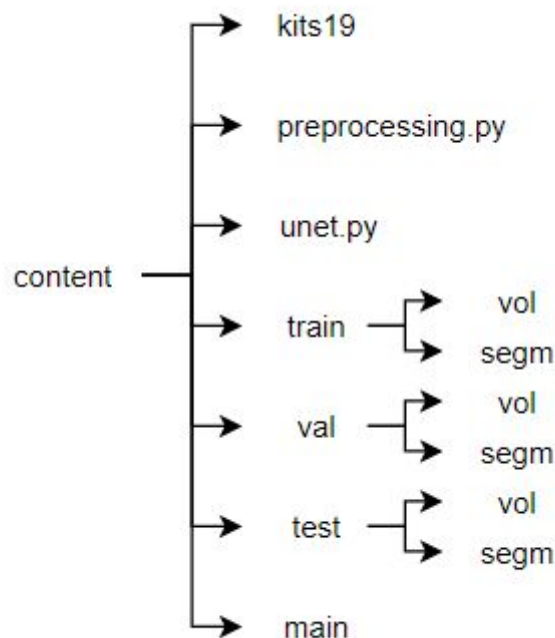
Raport 08.06.2020 r.

1. Wykonane zadania

1. Charakterystyka zbioru danych - podział na zbiór treningowy i testowy, format danych, wymiary zdjęć
2. Preprocessing
3. Zaimplementowanie architektury U-net
4. Implementacja walidacji DICE.

2. Przygotowanie danych

Opracowano strukturę danych:



Preprocessing obejmuje przeskalowanie obrazów do rozmiaru 256×256 , normalizację obrazów volume, zapisanie segmentacji w notacji one-hot oraz odpowiednie przygotowanie generatora danych (funkcja ImageDataGenerator).

3. Architektura

Do wykonania projektu, wykorzystano algorytm U-Net, która łączy warstwy enkodera z mapami obiektów dekodera tworząc strukturę przypominającą literę U. Cechą szczególną tej sieci jest to, że na każdym etapie umożliwia na zapoznanie się z cechami, które zostały utracone w wyniku działania warstwy łączącej.

Wymiary obrazu wejściowego: $256 \times 256 \times 1$

Contracting path: `Conv2D` → `Dropout` → `Conv2D` → `MaxPooling2D`

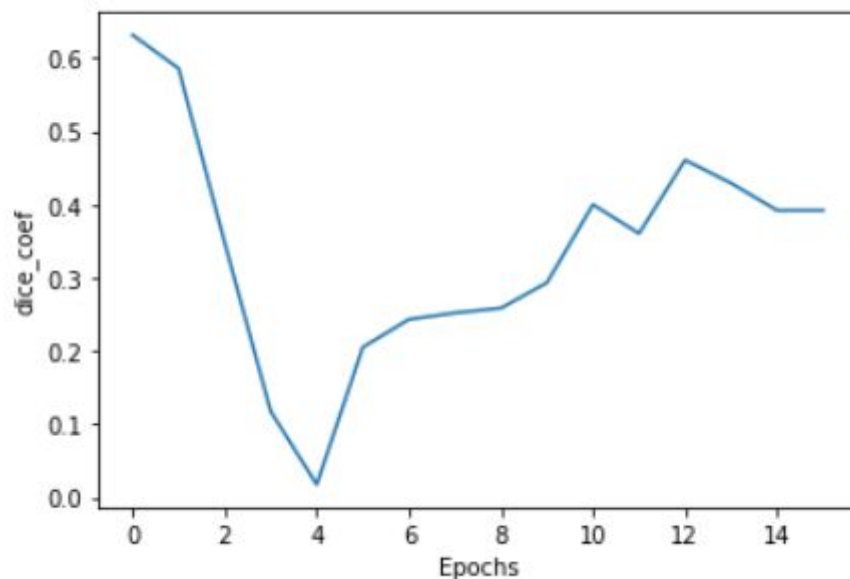
Expansive path: `Conv2DTranspose` → `concatenate` → `Conv2D` → `Dropout` → `Conv2D`

Jako funkcję utraty zastosowano *dice_coef_loss*, a wykorzystanym optimizer'em jest optimizer *adam*.

4. Efekt pracy i napotkane problemy

Wstępną segmentację przeprowadzono na zbiorze treningowym składającym się z 10 case'ów oraz zbiorze walidacyjnym z 5 case'ów. Wstępną ilością epok było 25, ale użycie EarlyStopping przerywa uczenie się sieci gdy wartość współczynnika spada.

Poniżej przedstawiono wykres wartości współczynnika DICE w zależności od epoki.



Niestety wyniki nie są wiarygodne - wiąże się to z potrzebą dostosowania funkcji współczynnika DICE do obecnego rozwiązania. Należy przyjrzeć się obliczaniu przez tę funkcję wartości dla notacji one-hot i ją dostosować.

Dalsze prace będą obejmowały:

- bugfix funkcji `dice_coef()`
- test na nowym zbiorze danych
- optymalizacja uczenia sieci
- optymalizacja wyników
- wizualizację wyników
- sporządzenie raportu