

**Akademia Górniczo-Hutnicza im Stanisława Staszica Wydział
Automatyki, Elektroniki Informatyki i Inżynierii
Biomedycznej**



Techniki Obrazowania Medycznego

Prototyp realizacji projektu

**Krystian Strzałka, Jakub Chabior, Katarzyna Urbanek, Aleksandra Augustyn
Kraków, 7 czerwca 2020**

1 Wykonanie

Do segmentacji nerek oraz nowotworów nerek użyto biblioteki Tensorflow oraz jej modułu Keras. Skorzystano z architektury sieci U-net.

Analizując dane napotkano nie działające przypadki lub takie, których rozmiar nie wynosił 512x512. Wadliwymi przypadkami były przypadki wszystkie oprócz tych o indeksach: 158, 159, 160, 170, 202. Wybrano zbiór walidacyjny oraz treningowy. Walidacyjny zbiór zawierał przypadki od indeksu 179 do 209, zaś treningowy zbiór zawierał przypadki od 0 do 178.

Każdy przypadek w zbiorze treningowym zostaje wczytany oraz zmieniony w obraz binarny. Następnie porównywany on zostaje z losowym przypadkiem znajdującym się w zbiorze walidacyjnym.

W celu uruchomienia kodu zamieszczonego na platformie Github należy przygotować w folderze ze skryptami folder data, w którym umieszczone będą dane z kits19. W pliku predict.py znajduje się funkcja przewidująca przypadek o zadanym numerze oraz zapisująca go w podanym folderze z rozszerzeniem .nii.

Segmentacja nerek działa poprawnie, więc zostanie wykorzystana do odnajdywania nowotworu. Do segmentacji nowotworu zostanie stworzona druga sieć neuronowa. W następnej kolejności kod zostanie dostosowany do możliwości uruchamiania go z konsoli.

Tabela 1
Opis plików w repozytorium

Nazwa	Opis
NeuralNetwork.py	Skrypt zawierający architekturę modelu.
main.ipynb	Notatnik, w którym przebiega wczytywanie danych, ich obróbka oraz uczenie modelu.
predict.py	Zawiera funkcję, która wczytuje dane o numerze <i>case</i> , wczytuje model, generuje segmentację, wizualizuje dane oraz zapisuje segmentację do pliku .nii w folderze <i>save</i> .
visualizeSlider3Way.py	Zawiera funkcję, która wyświetla 3 trójwymiarowe macierze o tych samych wymiarach wraz ze sliderem (zamierzanie obraz z CT, segmentacja ręczna, segmentacja wygenerowana).

2 Ewaluacja

Ewaluacji wyników dokonywano z użyciem współczynnika podobieństwa Dice’a (DSC), porównującego obrazy binarne. Segmentację, z którą jest porównywany obraz treningowy

zmieniono na obraz binarny, a następnie obrazy poddano porównaniu z użyciem współczynnika Dice’a, opisanego wzorem (1).

$$DSC(X, Y) = \frac{2 \cdot |X \cap Y|}{|X| + |Y|} \quad (1)$$

gdzie X oraz Y zbiory pikseli porównywanych obrazów. Współczynnik Dice’a zawiera się w przedziale między 0 a 1, im bliżej wartości 1, tym lepiej.

Zapis funkcji do obliczenia współczynnika Dice’a:

```
def dice(i1, i2):  
    return (2*np.sum(np.logical_and(i1, i2)))/(np.sum(i1)+np.sum(i2))
```