

Projekt realizowany w ramach przedmiotu Techniki Obrazowania Medycznego 2020

Magdalena Bialik
Anna Kudzia
Fidelus Aleksander

1. Opis i cel projektu

Corocznie na świecie jest wykrywane 400 tys. nowych przypadków nowotworu nerki, gdzie w większości przypadków dochodzi do leczenia operacyjnego. Ze względu na światową wielkość problemu, cieszy się ono zainteresowaniem oraz współpracą w celu stworzenia niezawodnej metodologii automatycznej segmentacji nerek oraz ich nowotworu.

Celem naszego projektu było zaproponowanie oraz zrealizowanie systemu automatycznej segmentacji nerek oraz nowotworów nerek w obrazach tomografii komputerowej.

2. Przygotowanie danych

Zbiór danych składał się z wyników badania tomografii komputerowej jamy brzusznej dla 210 unikalnych pacjentów. Dla każdego z pacjentów dane zostały również opisane poprzez zaznaczenie konturem nerek oraz ewentualnego nowotworu. A więc dla każdego pacjenta zestaw danych składał się z wielu przekrojów poprzecznych wraz z plikami segmentacyjnymi zawierającymi dane o ewentualnym występowaniu oraz pozycji nerek lub nowotworu na danym przekroju.

W celu przygotowania danych do zastosowania uczenia maszynowego każdy obraz standaryzowano do skali Hounsfielda w celu pozbycia się artefaktów niekorzystnych do uczenia modelu, odcięcie ustawiono na wartości $[-700, 700]$ w ten sposób usuwając większość niechcianych elementów na obrazie co prowadzi do stworzenia dokładniejszego modelu oraz pozwala otrzymać bardziej satysfakcjonujące wyniki.

W celu zwiększenia zbioru danych (podwojenia), zastosowaliśmy augmentację danych. W tym celu każdy przekrój tomografii komputerowej został również zastosowany po operacji horyzontalnego odbicia lustrzanego. Tak przetworzone obrazy zdecydowaliśmy się zapisać w osobnych plikach .png w celu ułatwienia późniejszego wczytywania danych do modelu.

Dane zostały podzielone na zestawy train, development oraz test odpowiednio zawierające 80%, 10%, 10% danych, razem licząc 210 obiektów. Dla każdego obrazu zapisywaliśmy również odpowiadający plik .png który zawierał dostarczone wyniki segmentacji w którym można rozróżnić trzy wartości 0 oznaczające tło, 1 nerkę oraz 2 dla nowotworu nerki.

3. Architektura sieci

W naszym projekcie zastosowaliśmy sieć UNet, która została zaprojektowana do segmentacji obrazów biologicznych. Architekturę sieci można podzielić na dwie większe ścieżki. Pierwsza to tak zwany koder, służący do wyłapywania kontekstu w obrazie, a druga to symetryczny do kodera, dekodery. Druga część ścieżki służy do precyzyjnej lokalizacji przy użyciu transponowanych warstw. W wielu artykułach naukowych zespoły badawcze zastosowały właśnie tę sieć do problemu segmentacji, stąd decyzja o użyciu sieci Unet.

W naszym modelu koder jest zbudowany z 5 bloków. Każdy blok jest zbudowany z 2 warstw konwolucyjnych (Conv2D 3x3) o takich samych rozmiarach filtra i z warstwy MaxPooling. Wraz z głębokością sieci ilość filtrów rośnie od 32 do 512 reprezentując o kolejne potęgi dwójki. Po 5 blokach znajdują się 2 warstwy konwolucyjne z 1024 filtrami.

Następnie przechodzimy do dekodera. Zbudowanego z 5 bloków, w których występują identyczne pary warstw konwolucyjnych, z tą różnicą, że ilość filtrów maleje. Każdy blok rozpoczyna się od warstwy transponowanej, która wykorzystuje informacje zebrane przez sieć w części kodera.

Wszystkie przedstawione warstwy wykorzystywały jako funkcję aktywacyjną ReLu. Jedynie ostatnia warstwa, która decyduje o przynależności pixeli do jednej z trzech klas (tło, nerka, nowotwór nerki) oparta jest na funkcji softmax. Opisana architektura sieci jest bazowana na artykule [1] i za jej pomocą uzyskano najlepsze wyniki.

Początkowo próbowano zastosować transfer learning oraz sieć Unet bazującą na znanych sieciach takich jak: VGG16, MobileNet czy ResNet 34 korzystając z publicznego repozytorium do segmentacji obrazów [2]. Jednak otrzymywane wyniki były niezadowolające, a przeuczenie sieci bardzo duże. Dlatego postanowiono zaimplementować sieć opisaną w artykule [1], początkowo jedynie za pomocą 4 bloków, już wtedy wyniki były lepsze niż dla wcześniej wspomnianych architektur. Natomiast zastosowanie 6 bloków skutkowało bardzo długim czasem uczenia i przeuczeniem sieci. W związku z tym, końcowa, opisana wcześniej architektura składa się z 5 bloków. Opisany etap został wykonany przez Anna Kudzia.

4. Pobieranie danych

Dane wykorzystywane do uczenia sieci w formacie png zajmują ponad 6GB pamięci bez augmentacji. Biorąc pod uwagę rozmiar danych oraz obszerną architekturę sieci załadowanie od razu wszystkich danych oraz przeprowadzenie na nich procesu uczenia byłoby niemożliwe na normalnym komputerze. Sieć można by uczyć na ograniczonej ilości danych, ale to skutkowało by gorszymi wynikami, które chcieliśmy jak najbardziej zmaksymalizować. Zamiast ograniczyć ilość danych, zwiększyliśmy ją poprzez augmentacje.

W celu przeprowadzenia uczenia na tak dużym zbiorze danych wykorzystano klasę ImageDataGenerator, która umożliwia pobieranie danych ze ścieżki w zadanych porcjach. Wykorzystując tę klasę dokonano normalizacji danych. Obrazy CT podzielono na 255 w celu uzyskania wartości 0-1, a obrazy segmentacji podzielono przez 127 zapewniając postać całkowitą w celu otrzymania wartości 0 dla tła, 1 dla nerki i 2 dla nowotworu nerki. ImageDataGenerator pozwala pobierając dane od razu zmienić ich wymiar. Obrazy

przesyłane do sieci mają wymiary 256x256. Gdy były większe zbierały za dużo pamięci, a gdyby je jeszcze zmniejszyć mogłaby nastąpić utrata pewnych informacji. Ze względu na możliwości pamięci użyto `batch_size=32`, co mocno spowalnia model, ale pozwala na wykorzystanie wszystkich dostępnych danych. Proces uczenia przeprowadzono za pomocą Google Colab umożliwiającego wykonywanie obliczeń na GPU, co pozwoliło przyspieszyć proces uczenia z 2,5 godziny na epokę dla zwykłego laptopa, do 20 minut na epokę. Umożliwiło to przetestowanie większej ilości architektur sieci oraz różnych hiperparametrów w celu uzyskania jak najlepszych wyników. Opisany etap został wykonany przez Magdalena Bialik.

5. Zastosowana funkcja kosztu, metryka i optymalizator

Do uczenia modelu wykorzystano funkcję kosztu dice loss, bardzo popularną dla problemu segmentacji i wykorzystaną z obu artykułów, na których bazuje nasze rozwiązanie [1][3]. Początkowo próbowano zastosować gotową funkcję z repozytorium dla segmentacji obrazów [2], jednak była ona odpowiednia jedynie dla problemu posiadającego 2 klasy. Wartości dice score była większa od 1. W związku z tym postanowiono napisać własną funkcję do obliczania dice loss oraz dice score.

Najpierw obliczono wartości dice coefficient dla każdej klasy osobno, a następnie obliczono ich średnią ważoną. Wagi nadano ręcznie najmniejszą dla tła - 0.1, gdyż na obrazach tła jest zdecydowanie najwięcej, a jego rozpoznawanie najmniej nas interesuje. Klasie 1, czyli nerce nadano wagę 0.3. Klasie 2, nowotworowi nerki, przypisano wagę 0.6, ponieważ jest to najmniejszy obszar na obrazach i najtrudniejszy do segmentacji. Funkcja dice z różnymi wagami dla klas została wykorzystana w artykule [3]. Do optymalizacji modelu wybrano optymalizator Adam o standardowych wartościach $\beta_1 = 0.9$ oraz $\beta_2 = 0.999$. Ustawiono wskaźnik uczenia na $\alpha = 3 \cdot 10^{-4}$, a spadek szybkości uczenia na $decay = 0.2$. Dodatkowo podczas uczenia zastosowano Early Stopping oraz 30 epok. W wyniku uczenia uzyskano wartości dice score dla przeszczepów dla 54% train set i 49% dla dev set. Opisany etap został wykonany przez Magdalena Bialik i Aleksander Fidelus. Punkty 3, 4 oraz 5 zostały zrealizowane w programie `training_network`.

6. Ewaluacja modelu

Ewaluację przeprowadzono na zbiorze testowym złożonym z 22 pacjentów, wcześniej nie widzianych przez sieć, co gwarantuje uzyskanie wiarygodnych wyników. Wynikiem segmentacji powinna być maska 3D. Nasza sieć była uczona na przekrojach i również przekroje zwracała predykcja wykonana za pomocą modelu. Stąd wykonano osobno predykcję dla każdego pacjenta, a następnie otrzymane przekroje połączono ze sobą w celu otrzymania siatki 3D. Funkcja aktywacyjna softmax zwraca rozkład prawdopodobieństwa przynależenia do konkretnej klasy, chcąc przeprowadzić ewaluację wyników za pomocą tego każdy piksel przypisano do najbardziej prawdopodobnej klasy. Wynik predykcji porównano z danymi z segmentacji ręcznej za pomocą wcześniej wspomnianej funkcji dice score z wagami dla każdej klasy. W celu ewaluacji modelu należało zmienić implementację funkcji

dice score. Podczas uczenia korzystano z obrazów 2D, które były opisane za pomocą tablicy 4 wymiarowej, gdzie ostatni wymiar odpowiadał za przynależność do jednej z 3 klas. Natomiast po predykcji danych, zamianie prawdopodobieństwa na konkretną klasę oraz połączeniu przekrojów w obraz wolumetryczny, uzyskano tablice 3 wymiarową, a każdy element tablicy posiadał wartość (0,1,2) świadczącą o przynależności do klasy. Pierwotne dane Nifti wczytano i zmieniono na dane typu Numpy Array. Obrazy uzyskane dzięki predykcji należało przeskalować (sieć zwracała obrazy 256x256, a dane z segmentacji były wymiarów 512x512), aby rozmiary obrazów z predykcji i segmentacji ręcznej były sobie równe. W tym celu wykorzystano funkcję zoom z biblioteki scipy.ndimage. Etap wykonany przez Anna Kudzia.

7. Ostateczne wyniki

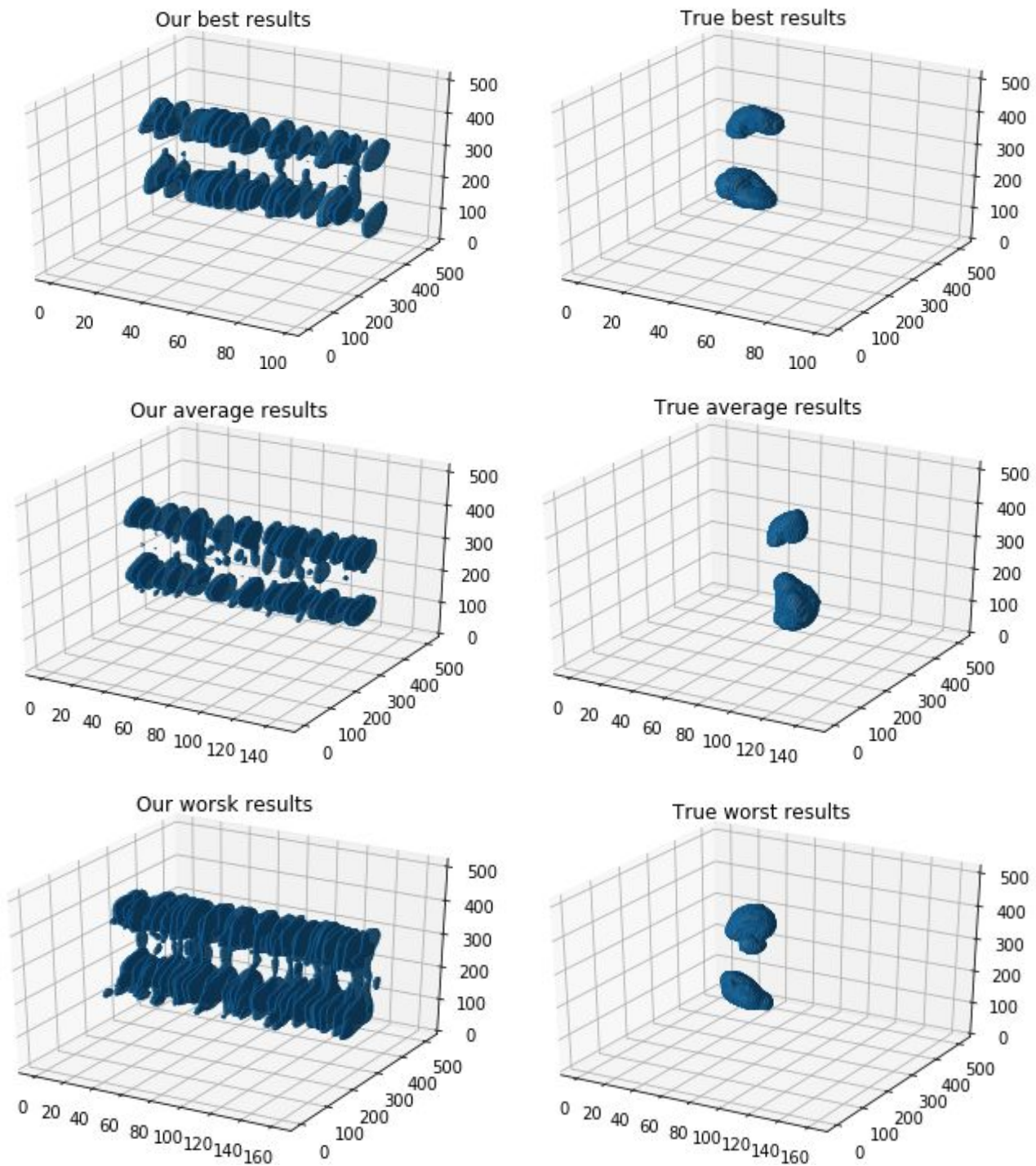
W tabeli 1 przedstawiono wyniki otrzymane podczas ewaluacji modelu dla całego problemu segmentacji oraz osobno dla segmentacji nerki i nowotworu nerki.

Tabela 1. Uzyskane wyniki dice score.

	Weighted Dice Score dla całego obrazu	Dice Score dla nerki	Dice Score dla nowotworu nerki
wartość średnia [%]	13,8	8,9	2,1
odchylenie standardowe [%]	2,2	3,1	2,8
wartość maksymalna[%]	21,2	14,5	11,6
wartość minimalna[%]	11,1	3,8	0

Jak widać uzyskane wyniki nie są najlepsze. Najgorzej wyglądają wyniki dla segmentacji nerki. W niektórych przypadkach nie udało się, w ogóle prawidłowo oznaczyć nerki.

Poniżej przedstawiono porównanie segmentacji ręcznej z segmentacją wykonaną przez zespół. Zaprezentowano najlepszy, najgorszy oraz średni wynik. Punkty 6 i 7 zostały zrealizowane w programie evaluation.



8. Bibliografia

- [1] Zhao, Wenshuai, and Zengfeng Zeng. "Multi Scale Supervised 3D U-Net for Kidney and Tumor Segmentation." *arXiv preprint arXiv:1908.03204* (2019).
- [2] https://github.com/qubvel/segmentation_models
- [3] Santini, Gianmarco, Noémi Moreau, and Mathieu Rubeaux. "Kidney tumor segmentation using an ensembling multi-stage deep learning approach. A contribution to the KiTS19 challenge." *arXiv preprint arXiv:1909.00735* (2019).