

Wydział Matematyki i Nauk Informacyjnych
Politechnika Warszawska

Ewaluacja Zwycięskiego Modelu BraTS 2021

Raport częściowy

Warsztaty badawcze -
Wyzwania w przetwarzaniu danych obrazowych MRI mózgu

Filip Szlingiert, Krzysztof Tkaczyk, Michał Zajączkowski

9 maja 2025

Spis treści

1	Wstęp	3
1.1	Problem	3
1.2	Rozwiązanie zwycięzców	3
2	Realizacja projektu	4
2.1	Środowisko techniczne	4
2.2	Przygotowanie pipeline'u	4
2.3	Dane	5

1 Wstęp

1.1 Problem

Brain Tumor Segmentation Challenge (BraTS) jest odbywającym się co roku wyzwaniem mającym na celu automatyczną segmentację guzów mózgu w obrazach MRI.

1.2 Rozwiązanie zwycięzców

Zespół KAIST MRI Lab wykorzystał zmodyfikowaną wersję nnUNet. Główne modyfikacje wprowadzone do baseline'u obejmowały:

- **Asymetryczne powiększenie architektury:** Zwiększenie liczby filtrów w warstwach enkodera przy zachowaniu oryginalnej liczby filtrów w dekodерze. Maksymalna liczba filtrów została zwiększona z 320 do 512.
- **Zastosowanie Group Normalization:** Zamiana standardowej normalizacji wsadowej (Batch Normalization) na Group Normalization, co pozwoliło na efektywniejsze trenowanie przy mniejszych rozmiarach batchy (2 zamiast 5).
- **Dekoder z uwagą osiową (Axial Attention):** Wprowadzenie mechanizmu uwagi działającego niezależnie na każdej z osi przestrzennych (wysokość, szerokość, głębokość) w dekodерze. Rozwiązanie to zastosowano dla 4 niższych rozdzielczości w sieci.

Dodatkowe istotne elementy rozwiązania to:

- Region-based training - przewidywanie trzech nakładających się regionów (enhancing tumor, tumor core, whole tumor) zamiast wzajemnie wykluczających się podregionów
- Głęboka superwizja z dodatkowymi wyjściami sigmoidalnymi
- Augmentacja danych obejmująca losowe obroty, skalowanie, deformacje sprężyste i modyfikacje jasności
- Funkcja straty będąca kombinacją binary cross entropy i Dice loss

Wyniki pokazały, że proponowane modyfikacje przyniosły niewielką, ale zauważalną poprawę w metrykach Dice i Hausdorff Distance w porównaniu do baseline'owego nnUNet. Najlepsza konfiguracja (BL + L + GN) osiągnęła średni Dice score 88.36% na zbiorze walidacyjnym.

2 Realizacja projektu

2.1 Środowisko techniczne

Projekt zrealizowany został w środowisku Google Colab w podstawowej wersji z następującymi parametrami maszyny:

- **System:** Python 3 Google Compute Engine backend
- **Pamięć RAM:** 12.7 GB
- **GPU:** NVIDIA Tesla T4
- **Pamięć GPU:** 15.0 GB
- **Dysk:** 112.6 GB
- **Środowisko wykonawcze:** Jupyter Notebook

2.2 Przygotowanie pipeline'u

Nasz pipeline wykorzystuje framework nnUNet, który pozwala zautomatyzować procesy trenowania sieci i przetwarzania danych. Cały pipeline został zaimplementowany w środowisku Google Colab i w jego skład wchodzi następujące etapy:

- Pobranie i instalacja nnUNet - sklonowanie i zainstalowanie w trybie edycji repozytorium GitHub.
- Przygotowanie środowiska - utworzenie wymaganych katalogów oraz zmiennych środowiskowych.
- Pobranie danych
- Wstępne przetwarzanie danych - konwersja do wymaganego formatu.
- Pobranie wag wytrenowanego modelu
- Instalacja odpowiedniej wersji PyTorch - ze względu na problemy z kompatybilnością z GPU zainstalowano wersję PyTorch 2.3.0 z obsługą CUDA 12.1.
- Przewidywanie na danych testowych.

Przygotowany przez nas pipeline umożliwia pełne przygotowanie danych oraz wykonanie predykcji przy użyciu pretrenowanego modelu nnUNet.

2.3 Dane

Do modelowania użyliśmy danych przygotowanych przez twórców rozwiązania, zamieszczonych na dysku Google dostępnych [1].

Bibliografia

- [1] . *Dysk zawierający zbiory danych treningowych*. URL: <https://drive.google.com/drive/folders/1HqEgzS8BV2c7xYNrZdEAnrHk7osJJ--2>.