

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ТАРАСА
ШЕВЧЕНКА
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ВИСОКИХ ТЕХНОЛОГІЙ

Завідувач кафедри молекулярної біотехнології та біоінформатики

к.б.н., доцент Олексій Юрійович Нипорко

Протокол No ____ засідання кафедри

від “ ____ ” _____ 2023 р.

**МУЛЬТИМОДАЛЬНА СЕГМЕНТАЦІЯ ПУХЛИН ГОЛОВНОГО
МОЗКУ НА МРТ ЗОБРАЖЕННЯХ ЗА ДОПОМОГОЮ НЕЙРОННИХ
МЕРЕЖ**

Випускна кваліфікаційна робота магістра
студентки спеціальності 091 «Біологія»
ОП «Біоінформатика і структурна біологія»
Даниленко Єлизавети Станіславівни

Науковий керівник асистент кафедри
біотехнології та біоінформатики
ас. Рибалка Ігор Євгенович

Оцінка захисту роботи _____

Київ – 2023 р.

АНОТАЦІЯ

Даниленко Є.С. Мультимодальна сегментація пухлин головного мозку на МРТ зображеннях за допомогою нейронних мереж. – Випускна кваліфікаційна робота магістра за спеціальністю 091 Біологія ОП «Біоінформатика і структурна біологія».

У ході роботи було створено модель та проведено її оцінку на задачі мультимодальної сегментації злоякісних пухлин головного мозку на МРТ зображеннях. Модель була побудована на основі нейронної мережі архітектури U-Net [23].

Отримані результати можуть бути використані у майбутньому для впровадження автоматичного діагностування злоякісних пухлин головного мозку та проведення моніторингу протікання даного захворювання.

Ключові слова: нейронні мережі, глибоке машинне навчання, сегментація, пухлини головного мозку, гліобластома.

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

- MPT – магнітно-резонансна томографія
- DSC – Dice Coefficient – коефіцієнт Дайса
- ED – peritumoral edema – перитуморальний набряк (едема)
- ET – enhancing tumor – підсилена контрастом пухлина
- FLAIR – fluid attenuated inversion recovery MRI image – МРТ зображення з відновленням інверсії з ослабленням рідини
- IoU – Intersection-Over-Union – індекс схожості
- NCR/NET – necrotic and non-enhancing tumor core – некротичне не підсилене контрастом ядро пухлини
- T1 – T1 weighted MRI images – T1-зважене МРТ-зображення
- T2 – T2 weighted MRI images – T2-зважене МРТ-зображення
- T1CE – T1-weighted Contrast-enhanced - T1-зважене підсилене контрастом МРТ-зображення

ЗМІСТ

ВСТУП.....	5
РОЗДІЛ 1 ЗАГАЛЬНА ІНФОРМАЦІЯ ПРО ОНКОЛОГІЧНІ ЗАХВОРЮВАННЯ ГОЛОВНОГО МОЗКУ ТА МЕТОДИ ЇХ ДІАГНОСТУВАННЯ (ОГЛЯД ЛІТЕРАТУРИ).....	7
1.1. Онкологічні захворювання головного мозку	7
1.2. МРТ як основний підхід до діагностування злоякісних пухлин головного мозку	10
1.3. Актуальність розробки програмного забезпечення для діагностування злоякісних пухлин головного мозку.....	11
1.4. Основні підходи до діагностування пухлин головного мозку за допомогою програмного забезпечення	12
1.4.1. Традиційні методи сегментації	13
1.4.2. Методи машинного навчання для сегментації пухлин на МРТ зображеннях	14
1.4.3. Сегментація пухлин головного мозку за допомогою нейронних мереж.....	15
РОЗДІЛ 2 МАТЕРІАЛИ ТА МЕТОДИ ДОСЛІДЖЕНЬ	17
2.1. Перелік використаних бібліотек та модулів.....	17
2.2. Опис набору даних.....	17
2.3. Схема роботи	19
2.3. Попередня обробка даних для навчання нейронної мережі	21
2.4. Подача даних на вхід для навчання нейронної мережі	22
2.5. Архітектура нейронної мережі	23
2.6. Функція втрат	25
2.6. Методи оцінки якості моделі	26
РОЗДІЛ 3 ОЦІНКА ТА ОБГОВОРЕННЯ РЕЗУЛЬТАТІВ РОБОТИ.....	30
3.1. Оцінка автосегментації пухлин головного мозку за допомогою нейронної мережі	30
3.2. Обговорення результатів та перспективи майбутніх досліджень	34
ВИСНОВКИ	36
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	37
ДОДАТКИ	40

ВСТУП

Злоякісні пухлини головного мозку – небезпечне онкологічне захворювання, що асоційоване з високою ймовірністю негативних наслідків: у 70% пацієнтів спостерігається прогресування захворювання протягом першого року і тільки 5% пацієнтів житимуть впродовж 5 років з моменту встановлення діагнозу [1].

Одна з головних проблем цих захворювань полягає в їх складному діагностуванні та лікуванні. Задля ефективного лікування необхідно точно визначити розмір та місцезнаходження пухлини. Крім того, потрібно виявляти пухлину на ранніх стадіях розвитку та проводити моніторинг її реакції на лікування. Причому чим раніше захворювання буде діагностовано, тим більше ймовірність позитивного результату лікування.

Традиційний метод діагностики заснований на ручному сегментуванні пухлин на МРТ-зображеннях лікарем-рентгенологом. Проте цей метод має ряд обмежень.

Сьогодні не існує затвердженого методу автоматичного діагностування злоякісних пухлин головного мозку. Тому, розробка нових та ефективних методів сегментації пухлин головного мозку на МРТ зображеннях є критично важливою. Використання нейронних мереж та глибокого навчання вбачається досить багатообіцяючим напрямком досліджень, оскільки ці методи ймовірно зможуть пришвидшити, здешевити та поліпшити точність процесу сегментації.

У цій дипломній роботі запропоновано використання нейронних мереж для мультимодальної сегментації пухлин головного мозку на МРТ зображеннях. Наша модель базується на архітектурі U-Net, що є однією з найбільш ефективних та точних у галузі медичної сегментації [2].

Об'єктом дослідження стали злоякісні пухлини головного мозку, зокрема гліобластоми (GBM/HGG) і гліоми нижчого ступеня (LGG) із клінічно підтвердженим діагнозом.

Предметом дослідження є мультимодальне сегментування злоякісних пухлин головного мозку на МРТ зображеннях.

Актуальність даної теми полягає в тому, що впровадження методів автоматичної діагностики злоякісних пухлин головного мозку та моніторингу перебігу захворювання зможе нівелювати такі недоліки традиційного методу як суб'єктивність, ручні помилки, висока вартість і, особливо, високі часові затрати. Ці покращення є важливим для підвищення вірогідності позитивного результату лікування.

Мета роботи полягала розробці моделі діагностування та мультимодальної сегментації злоякісних пухлин головного мозку на МРТ зображеннях.

Відповідно до мети було поставлено наступні **завдання**:

1. Обрати архітектуру нейронної мережі для мультимодальної сегментації пухлин головного мозку.
2. Провести попередню обробку даних МРТ зображень злоякісних пухлин головного мозку.
3. Натренувати нейронну мережу для задачі мультимодальної сегментації злоякісних пухлин головного мозку.
4. Оцінити роботу нейронної мережі на валідаційному наборі даних.
5. Провести візуальний контроль результатів роботи нейронної мережі на валідаційному наборі даних.

РОЗДІЛ 1

ЗАГАЛЬНА ІНФОРМАЦІЯ ПРО ОНКОЛОГІЧНІ ЗАХВОРЮВАННЯ ГОЛОВНОГО МОЗКУ ТА МЕТОДИ ЇХ ДІАГНОСТУВАННЯ (ОГЛЯД ЛІТЕРАТУРИ)

1.1. Онкологічні захворювання головного мозку

Онкологічні захворювання головного мозку — це аномальне розростання клітин мозку, яке може бути як доброякісним так і злоякісним. Пухлини головного мозку складаються з аномально зростаючої тканини, що є результатом неконтрольованого розмноження клітин, і ця тканина не виконує фізіологічної функції в мозку. Пухлини мозку можуть походити з різних типів клітин мозку, включаючи як гліальні клітини, які виконують функцію підтримуючих, так і безпосередньо з нейронів, які є тими клітинами, що передають сигнали в головному мозку.

Точна причина виникнення пухлин головного мозку наразі невідома, але були виявлені деякі фактори ризику, включаючи вплив радіації, генетичні фактори та певні захворювання, такі як нейрофіброматоз і туберозний склероз, синдром Лі-Фраумені, ретинобластома та синдром Турко. [1]

Загалом ймовірність того, що протягом життя у людини виникне злоякісна пухлина головного або спинного мозку доволі низька та становить менше 1% [3]. Цей показник значно збільшиться, якщо будуть враховані доброякісні пухлини, проте вони не несуть значних ризиків для життя людини. Їх можна успішно лікувати за допомогою хірургічного видалення або інших методів. За оцінками Американського онкологічного товариства щодо пухлин головного та спинного мозку в Сполучених Штатах за 2023 рік буде діагностовано близько 24 810 злоякісних пухлин головного або спинного мозку (14 280 у чоловіків і 10 530 у жінок), а близько 18 990 осіб (11 020

чоловіків і 7 970 жінок) очікує смерть від пухлин головного і спинного мозку [3].

Гліоми є основним типом пухлини, який зараз привертає увагу дослідників пухлин головного мозку. Термін гліома описує різні типи гліом, починаючи від пухлин високого ступеня злжкисності (HG), які називаються мультиформною гліобластою (GBM), до пухлин низького ступеня злжкисності (LG), таких як олігодендрогліоми або астроцити.

Мультиформна гліобластома є найпоширенішою первинною злжкисною пухлиною головного мозку. На неї припадає 16% усіх первинних онкологічних новоутворень мозку та центральної нервової системи. Середній рівень захворюваності з поправкою на вік становить 3.2 випадки на 100 000 населення [4]. Спочатку вважалося, що гліобластоми походять виключно з гліальних клітин; проте дані свідчать про те, що вони можуть виникати з багатьох типів клітин із властивостями, подібними до нервових стовбурових клітин. Ці клітини знаходяться на кількох стадіях диференціювання від стовбурової клітини до нейрона та глії, з фенотипічними варіаціями, які здебільшого визначаються молекулярними змінами в сигнальних шляхах, а не відмінностями в типі клітинного походження [5].

У результаті геномного профілювання та проекту Cancer Genome Atlas було секвеновано понад 600 генів з більш ніж 200 зразків пухлин людини, що виявило складний генетичний профіль гліобластом і створило набір із трьох основних сигнальних шляхів, які зазвичай активуються: шлях пухлинного білка p53, сигнальний шлях рецепторної тирозинкінази/Ras/фосфоінозитид-3-кінази та шлях ретинобластоми.

Більшість первинних і вторинних гліобластом мають зміни в цих шляхах, що призводить до неконтрольованої клітинної проліферації та покращеного виживання клітин, а також дозволяє пухлинній клітині вийти з контрольних точок клітинного циклу, шляхів старіння та апоптозу [6]. Генетичні зміни, характерні для первинних гліом, включають надмірну експресію рецептора епідермального фактора росту (EGFR), мутації гомологів

фосфату та тензину (PTEN) і втрату хромосоми 10q. При вторинних гліобластомах часто спостерігаються мутації ізоцитратдегідрогенази 1 (IDH1), мутації p53 і втрата хромосоми 19q [5].

Сучасні варіанти лікування при постановці діагнозу є мультимодальними і включають хірургічне видалення, променеви та хіміотерапію. Поточна стандартна терапія включає максимально безпечну хірургічну резекцію з подальшим одночасним опроміненням темозоломідом, пероральним алкілюючим хіміотерапевтичним агентом, а потім ад'ювантною хіміотерапією [6].

Повна хірургічна резекція гліобластоми є важкою та майже неможливою, тому що ці пухлини часто є інвазивними і часто знаходяться в областях мозку, які контролюють мову, моторику та органи чуття. Через високий ступінь інвазивності радикальна резекція первинної пухлинної маси не є лікувальною, і інфільтровані пухлинні клітини незмінно залишаються в навколишньому мозку, що призводить до подальшого прогресування захворювання або рецидиву [6].

Навіть із прогресом у хірургічній резекції прогноз для пацієнтів з діагнозом гліобластома залишається поганим: середня виживаність 15 місяців. Незважаючи на початкову резекцію та мультимодальну терапію, приблизно у 70% пацієнтів прогресування захворювання відзначатиметься протягом одного року після встановлення діагнозу, при цьому менше 5% пацієнтів виживуть через п'ять років після встановлення діагнозу [3].

Раннє виявлення гліобластоми дозволяє медичним працівникам ідентифікувати пухлину на початковій стадії, коли вона може бути меншою та більш піддатливою для хірургічного видалення або інших терапевтичних втручань. Крім того, рання діагностика може допомогти запобігти прогресуванню пухлини та її поширенню в інші ділянки мозку чи системи кровообігу, таким чином збільшуючи ймовірність успішного лікування.

1.2. МРТ як основний підхід до діагностування злоякісних пухлин головного мозку

Рання діагностика пухлини відіграє важливу роль у підвищенні ймовірності одужання або переходу в ремісію.

Для діагностики онкологічних захворювань мозку найчастіше застосовують візуальні методи ідентифікації, а найбільш поширений метод при діагностиці та клінічному лікуванні пухлин головного мозку – це магнітно-резонансна томографія (МРТ).

Як багатопараметричний метод візуалізації МРТ має високу роздільну здатність м'яких тканин. МРТ допомагає отримати точну інформацію про зображення головного мозку, а отже є змога судити про патологічні та гістоморфологічні зміни, що корисно для оцінки уражень і лікування пухлин [8]. У МРТ зображення різних режимів можна отримати відповідно до різниці часу поперечної релаксації та часу поздовжньої релаксації, а зображення різних режимів мають специфічність інформації про зображення. Наприклад, T1-зважена послідовність зображень (T1-weighted, T1) може краще відобразити анатомічну структуру різних тканин мозку. T1-зважена візуалізація з посиленням контрасту (T1-weighted Contrast-enhanced, T1CE) дозволяє чіткіше розрізнити межі пухлин головного мозку. T2-зважені візуалізації (T2-weighted, T2) покращують видимість області ураження і часто використовуються для ідентифікації уражень і визначення типу пухлини. Відновлення інверсії з ослабленням рідини (Fluid Attenuated Inversion Recovery, FLAIR) пригнічує видимість внутрішньочерепної цереброспінальної рідини та дає змогу краще виявляти інформацію про високий сигнал у зоні ураження.

Отже, у наші часи основний метод діагностики онкологічних захворювань мозку – це магнітно-резонансна томографія з подальшим розміченням (сегментуванням) та встановленням точного діагнозу спеціалістом-рентгенологом.

1.3. Актуальність розробки програмного забезпечення для діагностування злоякісних пухлин головного мозку

Точна сегментація зображень пухлин головного мозку є життєво важливим кроком до точної діагностики та ефективного лікування пухлин головного мозку. Відповідно до різних ступенів втручання людини у цей процес сьогодні можна виділити ручну, напівавтоматичну та автоматичну сегментацію.

Традиційно ручна сегментація, тобто сегментація, що проводиться спеціалістом-рентгенологом, є основним шляхом постановки діагнозу онкологічних захворювань мозку. Цей метод наразі є найбільш точним, проте він вимагає багато часу (який є вкрай важливим під час діагностування), є трудоміським, а також несе велику долю суб'єктивності в оцінці кожного окремого спеціаліста. Також будь-яка робота людська робота завжди асоційована з певною кількістю помилок. Крім того, цей метод також вимагає від експертів як глибинних знань про зображення пухлин мозку, так і інших професійних знань з анатомії. Не можна не згадати той факт, що робота висококваліфікованого спеціаліста також є дороговартісною, що збільшує ймовірність недосяжності такого діагностування для вразливих категорій населення.

Саме для того, щоб нівелювати всі ці недоліки, увага багатьох дослідників останні десятиліття прикута до спроб зсунути вектор з ручного діагностування до напівавтоматичного або автоматичного (тобто до діагностування за допомогою програмного забезпечення). Це стосується не тільки проблеми діагностування пухлин головного мозку, зокрема поліморфних гліобластом, а й багатьох інших захворювань. Варто підкреслити, що попри ці спроби, сьогодні не існує рішення, яке б застосовувалося для діагностики пухлин головного мозку у лікарській практиці.

1.4. Основні підходи до діагностування пухлин головного мозку за допомогою програмного забезпечення

Наразі існує безліч підходів до напівавтоматичного чи автоматичного діагностування пухлин головного мозку. Центральна ідея полягає в тому, щоб створити певну математичну модель, яка буде сегментувати пухлини, а також всі складові патологічного процесу на МРТ-зображеннях. Результат сегментації моделі потрібно порівняти з результатом, виділеним лікарем-рентгенологом вручну. Ці два результати мають бути якомога більш схожими між собою. Важливо, що у зображеннях головного мозку кількість інформації, яку можна отримати за допомогою одномодальних МРТ-зображень, обмежена. Тобто вона не може дати точну інформацію лікарям. Комбінація різних модальних зображень, які детальніше були описані у підрозділі 1.2., може надати вичерпну інформацію для отримання морфологічного та патологічного уявлення про пухлини мозку. Загалом, існуючі методи сегментування на зображеннях можна поділити на традиційні або звичайні, методи з використанням машинного навчання та методи з використанням нейронних мереж (Deep Learning) [рис. 1.1.].



Рис. 1.1. Основні методи автоматичного сегментування на зображеннях

1.4.1. Традиційні методи сегментації

Найпростішим існуючим методом сегментації на зображеннях є порогова сегментація. Це метод, при якому кожен піксель зображення порівнюється з заданим пороговим значенням, і якщо його інтенсивність перевищує цей поріг, то піксель вважається частиною об'єкту, в іншому випадку - фоном. Якість результатів цього методу повністю залежить від вибору порогового значення, тому цей вибір є дуже важливим. Таким чином, метод корисний для сегментації простих зображень і не демонструє позитивних результатів у задачі сегментації пухлин на МРТ-зображеннях.

Іншим методом є метод сегментації на основі регіону. Основна ідея методу сегментації на основі регіону полягає в тому, щоб спочатку розділити зображення на набір регіонів, що перекриваються, які містять однорідні характеристики зображення, такі як кольори, текстури, яскравість тощо. Потім, ці регіони об'єднуються у більші регіони на основі певних критеріїв, таких як схожість кольорів, границі об'єктів тощо, щоб виділити окремі об'єкти на зображенні. Метод більше підходить для сегментації невеликих об'єктів, проте він чутливий до шуму і легко утворює отвори в виділеному регіоні, що робить його не надто точним під час сегментації пухлин головного мозку.

Також часто застосовуються методи сегментації на основі теорії нечіткої логіки. Основна ідея методу сегментації на основі теорії нечіткої логіки полягає в тому, щоб присвоїти кожному пікселю зображення значення приналежності до кожного з розглянутих об'єктів на зображенні. Значення приналежності визначається на основі деяких характеристик пікселя, таких як яскравість, текстура, кольори тощо, та порівнюється з пороговим значенням, щоб визначити, до якого об'єкту належить піксель. Отже, метод легко інтегрується з іншими, що також дозволяє підвищувати точність, проте при його застосуванні доводиться мати справу з великими даними, обсягами обчислень, а отже робота потребує багато часу та великих обчислювальних потужностей.

1.4.2. Методи машинного навчання для сегментації пухлин на МРТ зображеннях

Методи сегментації пухлин головного мозку на основі традиційного машинного навчання використовують попередньо визначені функції для навчання моделі класифікації. Традиційні алгоритми машинного навчання в основному включають метод k-найближчих сусідів (KNN, K-Nearest Neighbors), метод опорних векторів (SVM, Support Vector Machine), випадковий ліс (RF, Random Forest) тощо.

Метод k-найближчих сусідів є одним з найпростіших алгоритмів машинного навчання, що використовується для класифікації і регресії. Він використовується для знаходження класу об'єкта на основі найближчих об'єктів з навчальної вибірки даних. Алгоритм k-найближчих сусідів простий у реалізації, а точність прогнозування області пухлини головного мозку відносно висока, проте часові затрати на обчислення дуже високі [9].

Основна ідея методу опорних векторів полягає у тому, щоб знайти границю, яка максимально відділяє один клас даних від іншого, найбільш точно. Алгоритм SVM шукає оптимальну гіперплощину в багатовимірному просторі, яка найбільш точно розділяє дані двох класів, які знаходяться найближче до неї. Методу опорних векторів властиві відносно простий розрахунок і здатність до узагальнення, але він має вищі вимоги щодо вибору параметрів і функції ядра [10].

Дерево рішень (DT, Decision Tree) – метод, що включає в себе побудову дерева з різних рівнів, де кожен вузол у дереві представляє питання, на яке можна відповісти «так» або «ні», а кожна гілка від вузла відповідає можливій відповіді на це питання. Метод випадкового лісу – це метод машинного навчання, який використовує ансамбль дерев для класифікації або регресії. Використання методу випадкового лісу може вирішити проблему надлишкової сегментації, обробляти кілька типів даних і демонструє хорошу ефективність захисту від шуму. При застосуванні методу можливо проводити

паралельні операції, а отже скоротити час. Але алгоритм погано працює на даних про пухлини невеликого розміру [11].

1.4.3. Сегментація пухлин головного мозку за допомогою нейронних мереж

Однією з найпоширеніших архітектур нейронних мереж, що використовуються у сегментації зображень, є згорткові нейронні мережі (CNN, Convolutional neural network). Основна ідея полягає в тому, щоб зменшити кількість параметрів, що потрібні для навчання мережі, за допомогою згортки та пулінгу, що дозволяє мережі виділяти риси на зображенні незалежно від їх положення на зображенні. Загалом, можна сказати, що алгоритм сегментації на основі згорткових мереж може автоматично екстрагувати ознаки та обробляти високорозмірні дані, але в процесі пулінгу може втрачати інформацію, та має низьку інтерпретованість. Зазвичай, згорткові нейронні мережі призначені для класифікації зображень, тобто, вони визначають, які об'єкти існують на зображенні, але не вказують їх точне місцезнаходження.

Іншим підходом є використання так званих повністю згорткових мереж (FCN, Fully Convolutional Network), які краще працюють для задачі семантичної сегментації. У FCN замість повнозв'язних шарів використовуються тільки згорткові шари, в них FCN використовує розріджену згортку, яка зменшує кількість параметрів в мережі та дозволяє зберігати більше інформації про зображення. Дослідники [12] запропонували поєднання FCN з CRF для сегментації головного мозку. Метод навчає двовимірні зрізи в поперечному, корональному та сагітальному напрямках відповідно, а потім використовує стратегію фузії для поєднання сегментованих зображень головного мозку. У порівнянні з попередньо описаними методами сегментації, швидкість сегментації є вищою, а ефективність - кращою.

Одним з найкращих сьогодні вважається підхід використання енкодер-декодер нейронної мережі. Архітектура складається з двох частин: енкодеру

та декодеру. Модель U-Net, яка базується на FCN, наразі є однією з найбільш багатообіцяючих моделей сегментації пухлин мозку. Структура мережі складається з енкодера та декодера, і з'єднання декодерів U-Net з вхідними шляхами з кодування, використовуються для отримання характеристик фігури на відповідній позиції на декодувальному шляху, щоб отримати характеристики прямого вибіркового оцінювання в фазі кодування в декодувальній фазі, тим самим навчаючи більш детальні характеристики. Дослідники [13] запропонували багаторівневу глибоку мережу, яка може отримувати багаторівневу інформацію зображення, додавши допоміжні класифікатори на багаторівневу глибоку медичну (MLDM) та U-Net, щоб зробити сегментацію зображень. Результати DSC, PPV та TPR становлять відповідно 83%, 73% та 85%.

Підсумовуючи, сегментація мозкових пухлин на основі традиційних методів в основному проста та легко реалізовна, але важко обробляти складні зображення, та точність сегментації загалом низька. Методи сегментації на основі традиційного машинного навчання теоретично легко зрозуміти, але великі дані важко обробляти. Методи сегментації, засновані на глибокому навчанні, можуть виділяти глибоку інформацію з зображення, але їх інтерпретованість є низькою.

Загалом, сьогодні методи та техніки глибокого навчання поступово займають домінуючу позицію в галузі сегментації пухлин мозку на МРТ-зображеннях. Найбільш актуальними напрямками розробки за минулі роки була сегментація з використанням згорткових, повністю згорткових та енкодер-декодер нейронних мереж, зокрема архітектура U-Net показала хороші результати точності та є широко досліджуваною сьогодні. Однак варто зазначити, що навіть тоді, коли дослідники використовують однакову архітектуру нейронної мережі на однаковому наборі даних, їх результати можуть відрізнятися, оскільки на це сильно впливає попередня обробка даних, яка сильно варіюється [14, 15].

РОЗДІЛ 2

МАТЕРІАЛИ ТА МЕТОДИ ДОСЛІДЖЕНЬ

2.1. Перелік використаних бібліотек та модулів

Робота була проведена в інтерактивному хмарному середовищі Google Colaboratory PRO PLUS [16] з використанням з використанням мови програмування Python [17].

Нейронна мережа була розроблена та натренована з використанням фреймворку машинного навчання PyTorch [18]. Ця бібліотека підтримує роботу з графічними процесорами (GPU), що дозволяє значно прискорити обчислення в машинному навчанні. Для обробки зображень була використана бібліотека SimpleITK [19], що базується на бібліотеці Insight Segmentation and Registration Toolkit (ІТК). Ця бібліотека надає засоби для роботи з медичними зображеннями, такими як рентгенівські знімки, томограми, магнітно-резонансні зображення тощо. За допомогою модулю `rescale_intensity` з фреймворку `skimage` було нормовано значення пікселів. Для аугментації було використано модуль `torchvision.transforms`. Ітератор був реалізований за допомогою вбудованого модуля `Chain`, що доступний у бібліотеці `Itertools`.

Дані у нейронну мережу подавались у вигляді NumPy-масивів. Для візуалізацій використовувався фреймворк `Matplotlib`.

Також були використані бібліотеки `Pandas`, `Random`, `Os` як допоміжні.

2.2. Опис набору даних

У роботі було використано датасет щорічного конкурсу мультимодального сегментування пухлин головного мозку 2020 (`BraTS2020`,

Multimodal Brain Tumor Segmentation Challenge 2020) [20-22]. Набір даних містить регулярні клінічні передопераційні мультимодальні МРТ-сканування гліобластоми (GBM/HGG) і гліоми нижчого ступеня (LGG) із патологічно підтвердженим діагнозом. Всі МРТ-зображення робочого датасету були попередньо розмічені спеціалістами. Робочий датасет містить дані про 369 випадки пухлин головного мозку. Валідаційний датасет містить дані про 125 випадки пухлин головного мозку.

Дані зберігаються у форматі NII (NIFTI). Це широко вживаний формат для багатовимірних даних нейровізуалізацій. Дані зберігаються у 3D форматі з розмірністю (240, 240, 155) у вигляді стеку зрізів, тобто послідовності 2D зображень.

Датасет є мультимодальним через те, що кожен випадок пухлини головного мозку представлений у 4-х режимах, кожен з яких може надати різні подробиці про захворювання: T1, T1CE, T2, FLAIR. Приклад одного патологічного випадку у BraTS2020 датасеті у 2D зрізі центру наведено на рисунку 2.1.

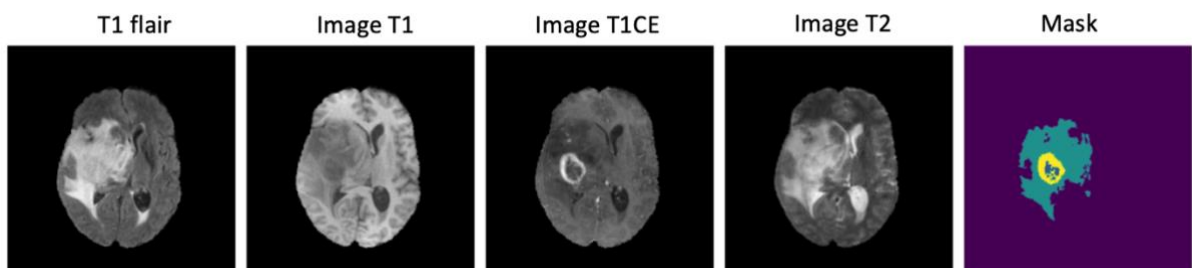


Рис. 2.1. Приклад одного випадку пухлини головного мозку у 4-х режимах [20-22]: Mask - зображення, що містить в собі інформацію про розмітку контурів пухлин головного мозку, а також її складових, спеціалістом.

Крім того, що кожен випадок представлений 4-ма режимами МРТ-сканувань, зображення, що розмічене фахівцем (рис. 2.1., Mask) містить в собі

4 складові, які також потрібно правильно розпізнати та виділити: нейтральний клас або фон, некротичне ядро пухлини (NCR), набряк (ED), ядро пухлини (ET). Приклад наведено на рисунку 2.2.

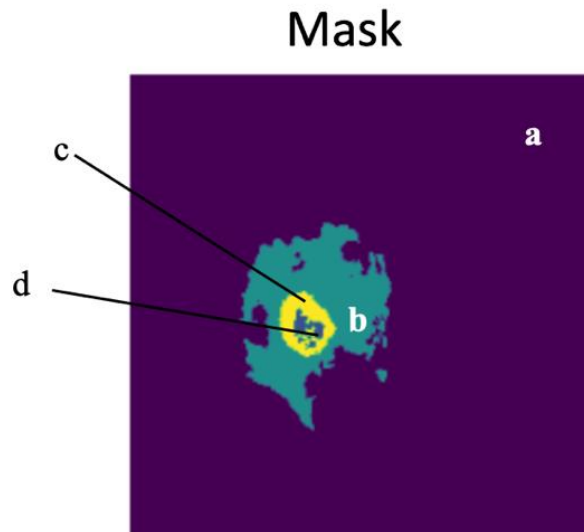


Рис. 2.2. Приклад розміченої пухлини головного мозку спеціалістом [20-22]: а (фіолетовий колір) – фон або нейтральний клас, b (зелений колір) - підсилена контрастом пухлина (ET), c (жовтий колір) - перитуморальний набряк (ED, едема), d (темно-синій колір) - некротичне ядро пухлини (NCR/NET)

2.3. Схема роботи

Роботу можна умовно поділити на 6 етапів: попередня обробка даних, створення конвеєру (pipeline) для завантаження даних у модель, тренування моделі, вибір найкращої моделі, оцінка якості моделі на тестовій вибірці та візуалізація результатів. На рисунку 2.1. наведено схему проведення роботи з виділенням найважливіших кроків.

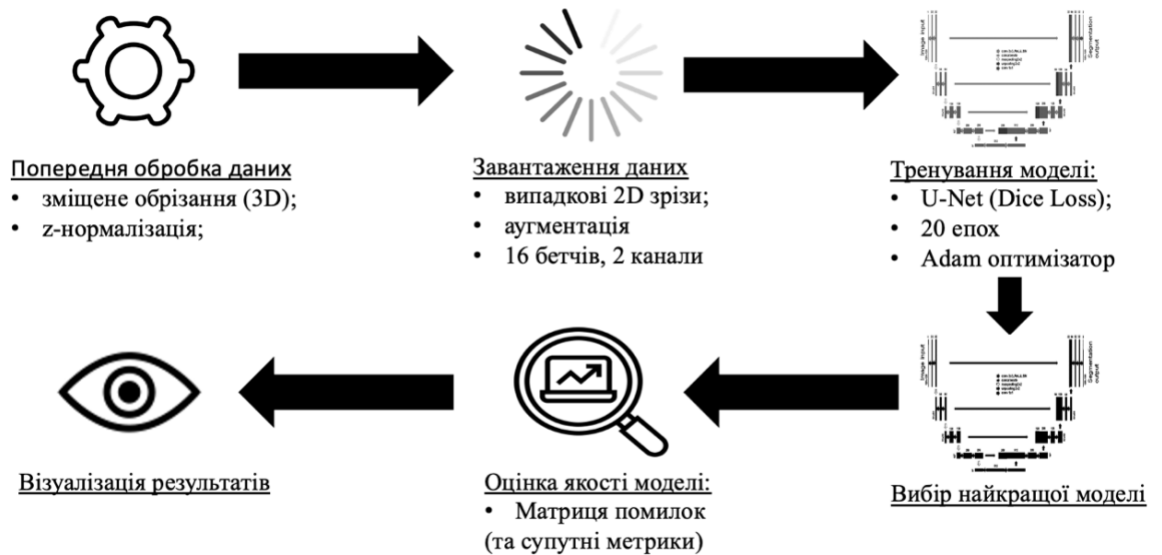


Рис. 2.1. Схема проведення роботи

Детальніше кожен з цих етапів описаний у наступних підрозділах. В загальному, була проведена попередня обробка даних із застосуванням техніки зміщеного обрізання та z-нормалізація відмінних від нейтрального класів. Надалі була проведена аугментація та для того, щоб знизити потреби у обчислювальних потужностях, у модель подавались рандомізовані 2D-зрізи у 2 каналах та 16 бетчах. Далі було проведено тренування U-Net нейронної мережі з використанням коефіцієнта Дайса як функції втрат та Adam оптимізатора. На тренування було відведено 20 епох з використання оптимізатора Адама та швидкістю навчання 10^{-4} , після чого на основі метрик, головною з яких був коефіцієнт Дайса, було обрано найкращу модель. Цю модель перевіряли на валідаційному наборі даних за допомогою таких метрик: середня точність (Mean Precision), середня чутливість (Mean Recall), середня специфічність (Mean Specificity), F1-оцінка (Mean F1-score), індекс схожості (Mean IoU, Intersection-Over-Union). Ці метрики підраховувались як для сегментації пухлини загалом, так і для кожної з її складових. В кінці була проведена візуалізація отриманих результатів.

2.3. Попередня обробка даних для навчання нейронної мережі

Попередня обробка даних є важливим кроком у навчанні нейронних мереж, оскільки від неї напряду залежить результат. Як вже зазначалося раніше, якщо подати в одну й ту саму архітектуру нейронної мережі дані, що були оброблені по-різному, можна отримати зовсім різні результати.

У попередній обробці даних було використано техніку зміщеного обрізання для виділення фрагментів з випадковою глибиною. Зрізи випадкової глибини використовуються в глибинному навчанні для зменшення обчислювальних витрат та зменшення перенавчання моделі. Замість того, щоб обробляти всі зображення з повною глибиною, випадково обрані зрізи глибини використовуються як вхідні дані для моделі. Це дозволяє зменшити кількість даних, які несуть незначну інформацію для моделі, та зменшити обчислювальні витрати для обробки великих об'ємів даних.

Спочатку було проведено розархівування даних та читання формату NII, в якому зберігаються зображення. Далі для кожної групи вхідних зображень було застосовано техніку зміщеного обрізання, щоб виділити 3D фрагменти з випадковою глибиною. Для того щоб збільшити рівень надійності моделей потрібно забезпечити, щоб всі такі фрагменти завжди мали зрізи, які містять як пухлинні, так і здорові частини мозку. На першому етапі попередньої обробки кожен зразок має форму $(Z, 240, 240)$, що відповідає формату тензора (Канали, Висота, Ширина), де Z - випадкове число від 14 до 155, що означає кількість зрізів. На другому кроці трохи зменшується висота і ширина кожної фрагменту до 224. На останньому етапі для кожної модальності кожної вибірки було обчислено середнє значення і стандартне відхилення в межах не нульових ділянок мозку, потім нормалізовано вибірку за допомогою z -нормалізації, тобто шляхом віднімання середнього значення і ділення на стандартне відхилення. Нормалізується лише область мозку, фонові інтенсивності залишаються рівними нулю. Нормалізовані масиви було збережено у форматі numpy.

2.4. Подача даних на вхід для навчання нейронної мережі

Після попередньої обробки всіх модальностей для кожної навчальної вибірки було створено простий конвеєр (pipeline) завантаження даних. Замість того, щоб завантажувати в моделі створені 3D-фрагменти, було прийняте рішення ще зменшити обчислювальне навантаження, випадково вибравши 2D-фрагменти і використавши їх як вхідні дані.

У роботі було використано фреймворк Pytorch для створення класу, необхідного для генерації вхідних даних. При кожному виклику обиралася випадкова вибірка, з якої було обрано випадковий зріз. Під час випадкового семплування використовувалися ймовірності, щоб вибрати зрізи з більшою ймовірністю відповідно до їх частоти появи у наборі даних. На цьому етапі також було проведено аугментацію (техніка обробки даних, яка використовується в машинному навчанні для створення нових прикладів даних шляхом застосування різних видів перетворень до наявних навчальних прикладів) та розподіл даних на тренувальний, тестувальний та валідаційний набори.

Було складено чотири модальності відповідного зрізу, щоб отримати тензор форми (3, 224, 224). Зазвичай моделі оцінюються на основі результатів їх сегментації на 3 окремі мітки, що злегка перекриваються. Щоб створити ці мітки (ціла пухлина (NCR + ED + ET), ядро пухлини (NCR + ET) і набряк (ED)), було спочатку створено one-hot представлення заданих анотацій, а потім просто підсумовано необхідні класи. Дані різних модальностей (FLAIR, T1, T1CE, T2) оброблялися окремо, а потім об'єднувалися в один вхідний тензор для моделі. Тобто, мультимодальні дані перетворювалися на одну спільну модальність перед подачею до моделі для сегментування.

Остаточна форма цільової маски сегментації мала вигляд (16, 3, 224, 224). Це означає, що для кожного пакета розміром 16 було присутніх 3 канали маски і розмір кожної маски становив 224x224 пікселів.

2.5. Архітектура нейронної мережі

У роботі була використана нейронна мережа архітектури U-Net. Архітектура цієї мережі була розроблена О. Ронненбергером, Ф. Фішером та Т. Брудманом в 2015 році, коли вони працювали в Університеті Фрайбурга в Німеччині [23]. Основна відмінність U-Net полягає в тому, що вона має "U"-подібну структуру (рис.2.3.), що дозволяє мережі більш точно визначати границі об'єктів на зображенні.

Архітектура мережі проілюстрована на рисунку 2.3. Вона складається з звужуючого шляху (ліва сторона) та розширюючого шляху (права сторона). Звужуючий шлях відповідає типовій архітектурі згорткової мережі. Він складається з повторного застосування двох згорток 3x3 (без заповнення), за якими слідує функція активації ReLU (Формула 1, Rectified Linear Units) та операція максимального пулінгу 2x2 зі стрибком 2 для зменшення розміру.

[1]

$$f(x) = \max(0, x)$$

На кожному етапі зменшення розміру подвоюється кількість каналів ознак. Кожен етап розширюючого шляху включає в себе збільшення розміру карти ознак за допомогою методу збільшення розміру (upsampling), за яким слідує згортка 2x2 (up-convolution), що зменшує кількість каналів ознак удвічі, з'єднання з відповідно обрізаною картою ознак зі звужуючого шляху, та дві згортки 3x3, кожна за якою слідує функція активації ReLU. Обрізка є необхідною через втрату граничних пікселів на кожній згортці. На останньому шарі використовується згортка 1x1 для відображення кожного 64-компонентного вектора ознак на бажану кількість класів. Всього в мережі є 23 згорткових шари [23].

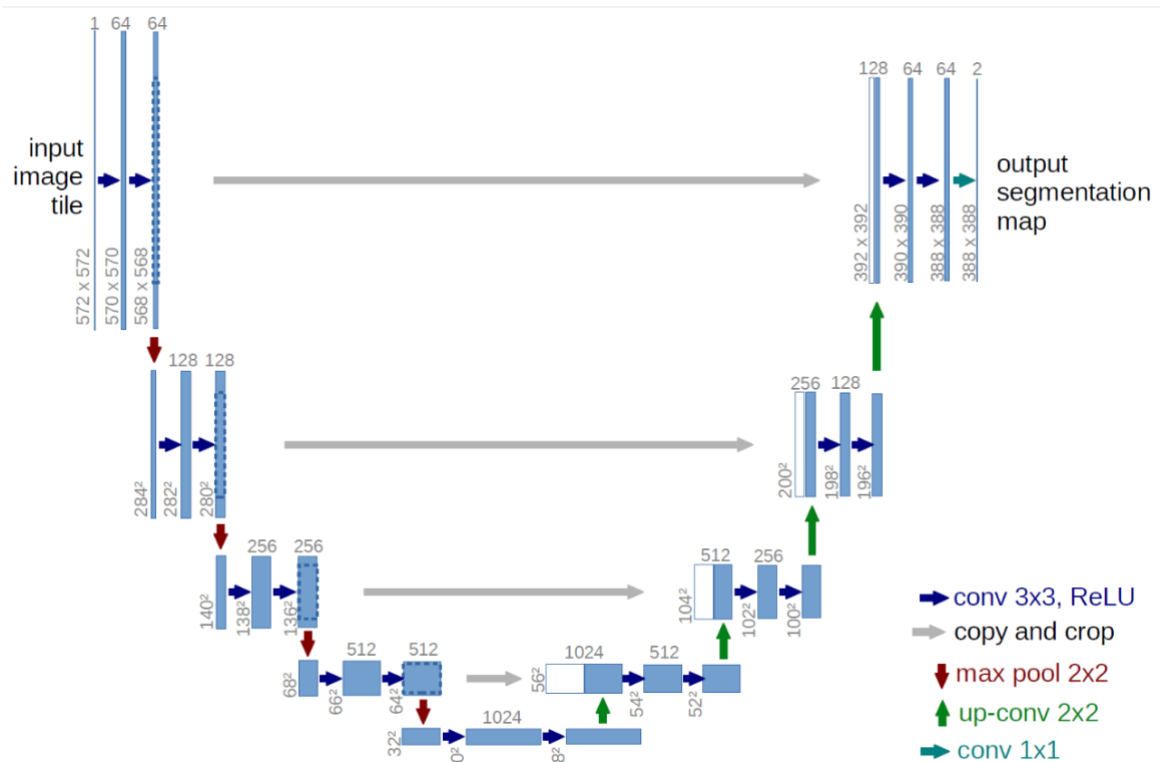


Рис.2.3. Архітектура нейронної мережі U-Net [21]: кожна синя рамка відповідає багатоканальній карті ознак. Кількість каналів позначена зверху на рамці. Розмір у координатах x-y наведений у нижньому лівому кутку рамки. Білі рамки представляють скопійовані карти ознак. Стрілки позначають різні операції.

Отже, архітектура U-Net складається з двох частин - енкодера та декодера, які забезпечують глибоку зв'язність між вхідним та вихідним зображеннями. Енкодер містить послідовність згорткових та пулінгових шарів, які виконують зменшення розмірності зображення та вилучення його основних ознак. Декодер складається з послідовності згорткових та розгорткових шарів, які здійснюють збільшення розмірності зображення та відновлення деталей зображення. Особливістю архітектури U-Net є використання пропускних з'єднань між енкодером та декодером, які забезпечують зв'язність між різними рівнями глибини мережі та допомагають уникнути втрати інформації під час зменшення розміру зображення. Також, U-Net використовує функцію активації ReLU для активації згорткових шарів

та функцію активації сигмоїд для вихідного шару, що забезпечує бінарну сегментацію зображення.

2.6. Функція втрат

Функція втрат - це функція, яка вимірює рівень помилок між фактичними значеннями і передбаченими значеннями в нейронній мережі. Основна мета цієї функції полягає в тому, щоб дати зворотний зв'язок моделі, що допомагає їй коригувати ваги нейронів під час навчання. Часто для цього використовують функцію кросс-ентропії, проте враховує тільки втрату на мікрорівні, а не розглядає її глобально, що недостатньо для передбачення на рівні зображення. У роботі було використано функцію втрат Дайса (DL, Dice Loss), оскільки вона довела свою ефективність в завданнях медичної сегментації [20].

[1]

$$DSC = \frac{2 \sum_i^N p_i g_i}{\sum_i^N p_i^2 + \sum_i^N g_i^2}$$

Формула 1 показує рівняння DL, в якому p_i та g_i є парами відповідних значень пікселів прогнозу та істинної інформації відповідно.

Замість того, щоб розглядати кожен мітку як окремий клас та розраховувати середню втрату по всіх класах, ця функція розраховує втрату на виході тензора без такої диференціації. У задачах виявлення границь, границі, що відповідають правильному результату, та границі, передбачені моделлю, можна розглядати як два набори. За допомогою DL, обидва набори тренуються на те, щоб перетинатися поступово. Як показано на рис. 2.4., знаменник враховує загальну кількість пікселів границі на глобальному рівні, тоді як чисельник враховує перекриття між двома наборами на локальному рівні.

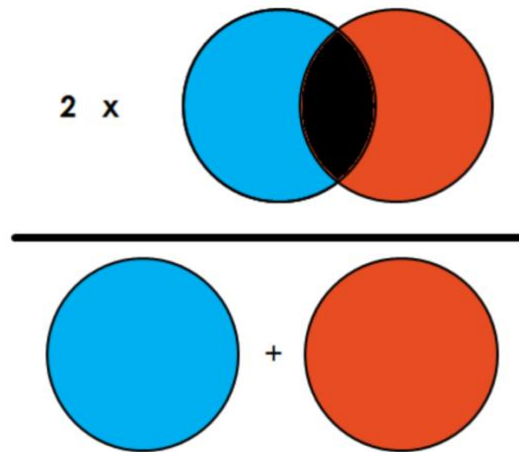


Рис 2.4. Схематичне представлення коефіцієнта Дайса

Таким чином, функція втрат Дайса враховує інформацію про втрати як локально, так і глобально, що є критичним для отримання високої точності, яка важлива для задачі медичної сегментації.

2.6. Методи оцінки якості моделі

Методи оцінки якості моделі в нейронних мережах - це способи вимірювання ефективності моделі, що використовуються для оцінки її здатності до роботи з новими даними. Оцінка якості моделі є важливим кроком в процесі навчання нейронних мереж і дозволяє оцінити, наскільки точно модель прогнозує результати для нових даних.

У роботі було використано такі метрики оцінки якості моделі: середня точність (Mean Precision), середня чутливість (Mean Recall), середня специфічність (Mean Specificity), F1-оцінка (Mean F1-score), індекс схожості (Mean IoU, Intersection-Over-Union).

Загальноживаним підходом до оцінки якості класифікаційних моделей є матриця помилок (confusion matrix). У найпростішому варіанті матриця помилок допомагає зрозуміти та візуалізувати, чи робить модель помилки в класифікації між двома класами (рис.2.5.).

		Прогноз моделі	
		Так	Ні
Реальні значення таргету	Так	True Positives (TP)	False Negatives (FN) (Помилка другого роду)
	Ні	False Positives (FP) (Помилка першого роду)	True negatives (TN)

Рис.2.5. Матриця помилок

Проте матриця помилок може бути використана і для більше, ніж 2-х класів. У цьому випадку вона представляється у вигляді $N \times N$ матриці, де N - кількість класів. Кожен ряд і стовпець у матриці відповідає конкретному класу, а кожен елемент матриці показує кількість прикладів, які були класифіковані відповідно до реального класу та передбаченого класу.

Матриця помилок складається з чотирьох окремих метрик:

- Істинно-позитивні значення (TP, True Positives) - це кількість правильних прогнозів моделі, коли вона передбачає позитивний клас (1), і насправді це так.
- Істинно-негативні значення (TN, True Negatives) - це кількість правильних прогнозів моделі, коли вона передбачає негативний клас (0), і насправді це так.
- Хибно-позитивні значення (FP, False Positives) - це кількість неправильних прогнозів моделі, коли вона передбачає позитивний клас (1), але насправді це негативний клас (0).
- Хибно-негативні значення (FN, False Negatives) - це кількість неправильних прогнозів моделі, коли вона передбачає негативний клас (0), але насправді це позитивний клас (1).

На основі цих метрик можна порахувати додаткові метрики, що зазначені вище.

Середня точність відображає, наскільки надійні прогнози моделі для позитивного класу. У випадку високої точності моделі, це означає, що будь-який зразок, передбачений як позитивний, є істинним. Середня точність рахується за формулою:

[2]

$$\text{Точність} = \frac{TP}{TP + FP}$$

Чутливість - це співвідношення кількості позитивних зразків, що були правильно класифіковані як позитивні, до загальної кількості фактично позитивних зразків. Чим вища чутливість, тим більше фактично позитивних зразків було знайдено. Ця метрика не залежить від того, як передбачаються негативні зразки.

[3]

$$\text{Чутливість} = \frac{TP}{TP + FN}$$

Специфічність фокусується на фактичному негативному класі, аналогічно тому, як чутливість фокусується на фактичному позитивному класі.

[4]

$$\text{Специфічність} = \frac{TN}{TN + FP}$$

Призначення F1-оцінки полягає в поєднанні оцінок чутливості та точності в одну метрику. Ця метрика обчислюється згідно з наступним рівнянням:

[5]

$$F1 \text{ оцінка} = 2 * \frac{\text{Точність} * \text{Чутливість}}{\text{Точність} + \text{Чутливість}}$$

Значення F1 оцінки лежать в діапазоні між 0 та 1, де 1 є ідеальним значенням, що вказує на повну точність та повну чутливість моделі.

Індекс схожості (Intersection-Over-Union, IoU) - це відношення площі перетину між передбачуваною сегментацією та істинною до площі їхнього об'єднання. Знаходиться за такою формулою:

[6]

$$\text{Індекс схожості} = \frac{TP}{TP + FP + FN}$$

РОЗДІЛ 3

ОЦІНКА ТА ОБГОВОРННЯ РЕЗУЛЬТАТІВ РОБОТИ

3.1. Оцінка автосегментації пухлин головного мозку за допомогою нейронної мережі

У ході роботи було натреновано нейронну мережу архітектури U-Net для розпізнавання та сегментації злоякісних пухлин головного мозку, а саме гліобластом та гліом нижчого ступеня. Програмний код з реалізацією моделі мультимодальної сегментації пухлин головного мозку можна побачити за посиланням: <https://github.com/infel17/SegmentationBrainTumor1>.

На виході було отримано інформацію про загальні кордони пухлини, а також її складових окремо. Саме тому оцінка моделі також була проведена як оцінка визначення пухлини загалом, так і визначення окремих її частин. Таким чином було обчислено середній DSC, а також DSC повної пухлини, ядра, підсиленої контрастом пухлини та DSC едеми.

Метрики найкращої моделі наведені у Таблиці 3.1.

Таблиця 3.1.

Метрики найкращої моделі сегментації пухлин головного мозку після тренування нейронної мережі

Метрика	Значення
Середній DSC	0.8916
Повна пухлина DSC	0.8957
Ядро пухлини DSC	0.8675
Підсилена контрастом пухлина DSC	0.8149
Едема DSC	0.7713

З Таблиці 3.1. можна зробити висновок, що моделі притаманна достатньо висока точність: в середньому 89.16% пікселів класифіковано правильно. Під час тренування модель добре справлялася з задачею виділення повної пухлини на зображенні, а також сегментуванні ядра пухлини. Дещо гірше сегментувалась ділянка підсиленої контрастом пухлини. Сегментування ж едеми характеризується найгіршим результатом, хоча він все ще не є критично поганим: 77.13% пікселів, на яких розташовується перитуморальний набряк, класифікуються правильно.

Для того, щоб оцінити цю модель, в неї було подано валідаційний датасет та обчислено такі метрики: середня точність, середня чутливість, середня специфічність, F1-оцінка, IoU. Результати також обчислювались для кожної з складових пухлини окремо, а результати наведено у Таблиці 3.2.

Таблиця 3.2.

**Метрики оцінки якості натренованої моделі для сегментації
пухлин головного мозку**

Метрика	Повна пухлина	Ядро пухлини	Підсилена пухлина	Едема
Точність	0.9003	0.9053	0.8374	0.7580
Чутливість	0.8945	0.8551	0.8290	0.8425
Специфічність	0.9885	0.9892	0.9894	0.9870
F1-оцінка	0.8957	0.8675	0.8149	0.7713
IoU	0.8472	0.7844	0.7095	0.6594

З Таблиці 3.2. можна зробити висновок, що сегментація злоякісних пухлин та її складових знаходиться на високому рівні.

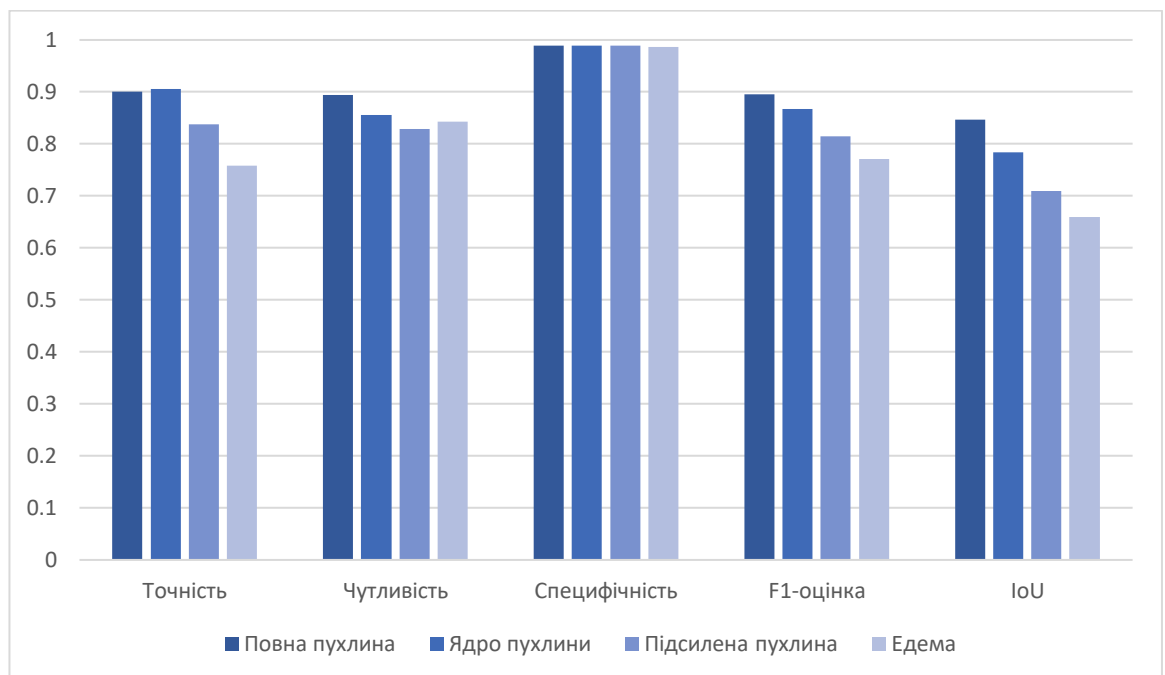
Всі метрики досягають значень більше 0.7, що свідчить про високу точність та надійність моделі. Отримані результати показали високу точність та достовірність моделі у сегментації повної пухлини Метрики F1-Score та IoU для цієї задачі перевищує 0.8, що свідчить про високу якість моделі. Ядро

пухлини також сегментується з високою точністю: F1-Score та IoU досягають значень 0.8675 та 0.7844 відповідно. Існує тенденція до дещо гіршого сегментування ділянки підсиленої контрастом пухлини, а едема або перитуморальний набряк сегментується найгірше. Проте оцінки якості моделі для цих задач все ще залишаються задовільними. Варто відмітити високі значення оцінки специфічності під час сегментування всіх складових. Це говорить про високу здатність моделі правильно визначати негативні класи.

Для більшої наглядності результати оцінки натренованої моделі також винесено на графік 3.1.

Графік 3.1.

**Метрики оцінки якості натренованої моделі для сегментації
пухлин головного мозку**



Як зазначалось вище, сьогодні найбільш поширений метод діагностики та моніторингу захворювання – візуальний контроль та сегментування лікарем-рентгенологом. Тому результати роботи моделі також було візуалізовано. Результати візуалізації результатів сегментації злоякісних

пухлин головного мозку за допомогою нейронної мережі архітектури U-Net наведені на рис.3.1.

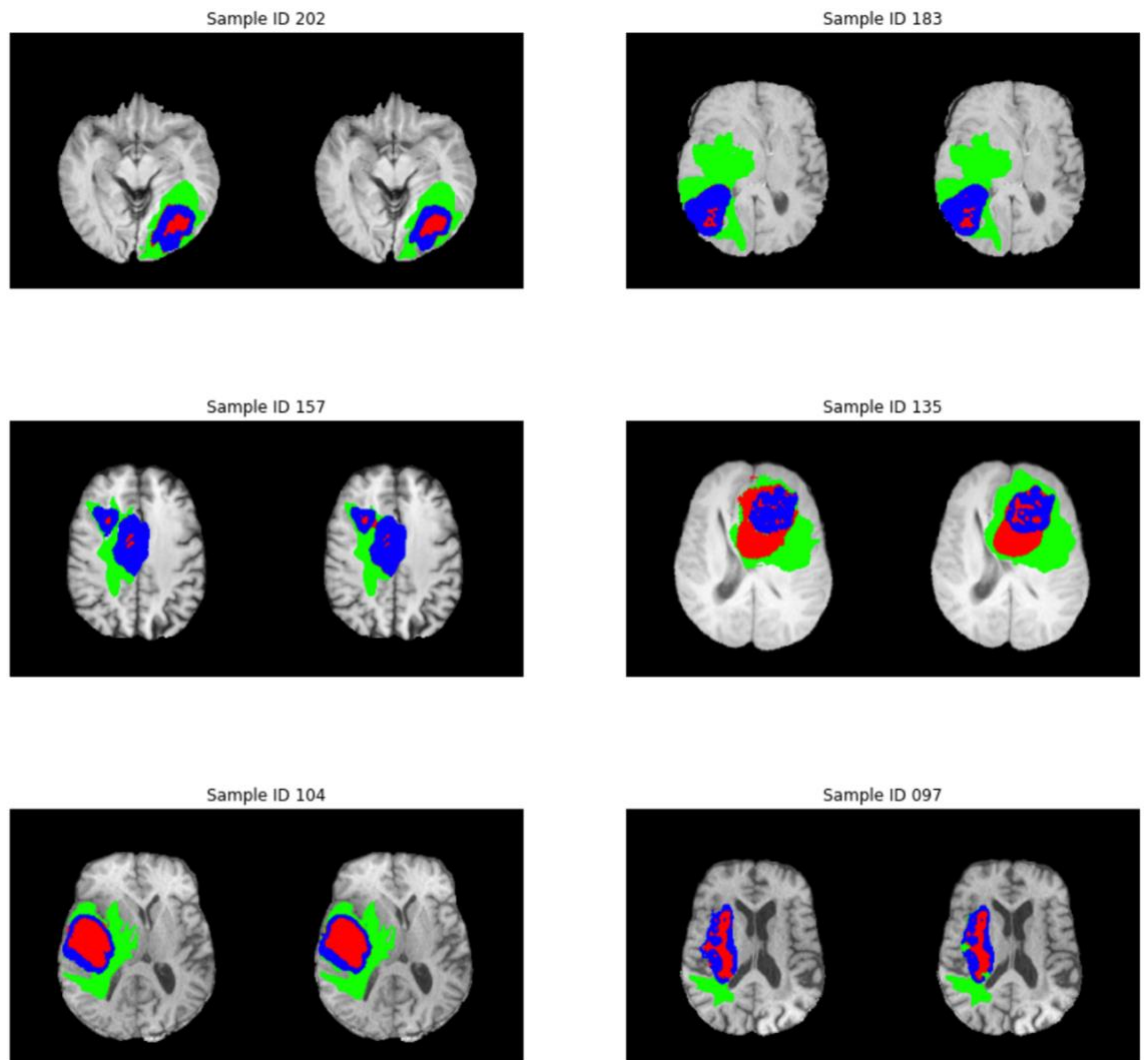


Рис. 3.1. Візуалізація сегментації вибірки зрізів з валідаційного датасету: зображення зліва – сегментація, проведена спеціалістом, зображення справа – результат роботи моделі.

Отже, використання моделі U-Net показало високі результати у сегментації пухлин головного мозку, зокрема у визначенні повної пухлини, ядра пухлини та підсиленої контрастом пухлини. Це дозволяє використовувати отриману модель для сегментування пухлин головного мозку з високою достовірністю. Метрики для едеми не досягають таких високих значень, але все ще відповідають стандартам для більшості випадків.

Таким чином, можна стверджувати, що отримана модель демонструє високі результати сегментування пухлин головного мозку та може бути корисною для клінічних досліджень та діагностики.

3.2. Обговорення результатів та перспективи майбутніх досліджень

В попередньому підрозділі було зазначено метрики якості натренованої моделі та зроблено висновок, що вони є високими, а результати сегментування достатньо достовірними. Для того, щоб отримати ширше розуміння цих результатів потрібно також порівняти їх з сучасними дослідженнями інших авторів.

Під час змагання Brain Tumor Segmentation 2020 перше місце посіла команда на чолі з Ф. Ізензі з Німеччини з результатами DSC 0.8895, 0.8506 та 0.8203 для повної пухлини, ядра пухлини та підсиленої пухлини відповідно [24]. Науковий прогрес не стоїть на місці і в наступного року багато інших дослідників зосередилась над тим, щоб покращити отримані результати. Часто вдавалось покращити сегментацію певної складової пухлини на фоні погіршення сегментації іншої. Наприклад, робота команди на чолі з Х.Джіа продемонструвала такі метрики: 0.9130, 0.8553, 0.7881 для повної пухлини, ядра пухлини та підсиленої пухлини відповідно [25].

Проте роботи, опубліковані у 2022-2023 роках, демонструють вищі показники якості сегментації пухлин головного мозку. Наприклад, у роботі команди на чолі з З.Шаукатом був досягнутий такий результат: середній DSC дорівнює 0.9558. Група дослідників на чолі з К.Р. Лаукампом у 2021 році опублікувала наступні результати: DSC 0.9100 для повного об'єму пухлини [26].

Отже, можна зробити висновок, що дослідникам загалом і нашій групі зокрема вдалось покращити метрики якості сегментації злоякісних пухлин головного мозку на МРТ зображеннях порівняно до 2020 року, коли

розпочалась робота в цьому напрямку на прикладі набору даних BraTS2020. Оцінка якості нашої моделі показала, що модель наразі не є найбільш точною в світі, проте її якість можна оцінити як високу. Велика кількість даних є ключем до розуміння та покращення якості і точності глибоких мереж, тому задля покращення результатів даної моделі у майбутньому можна збільшити кількість даних. Також для тренування можна використати більш потужне обчислювальне обладнання. Таке покращення дозволить додати додаткові шари до нейронної мережі, з'єднання між енкодером та декодером, збільшити роздільну здатність поданих на вхід даних. Це все може позитивно корелювати зі збільшенням точності моделі.

ВИСНОВКИ

1. Було проведено попередню обробку даних, найголовнішими кроками якої було застосування алгоритму зміщеного обрізання та з-стандартизацією.
2. Було здійснено навчання нейронної мережі. Найкращий варіант моделі продемонстрував таке значення середнього DSC: 0.8916. Найкраще модель навчилася сегментувати повну пухлину (DSC = 0.8957) та ядро пухлини (DSC = 0.8675). Гірше модель справлялась з сегментуванням частини підсиленої контрастом пухлини (DSC = 0.8149), а найменш точно сегментувала едему (DSC = 0.7713). Загалом, не зважаючи на розбіжності у числах, всі ці метрики можна назвати високими.
3. Роботу найкращу моделі було оцінено на валідаційному наборі даних. Результати демонструють високу точність та достовірність моделі в сегментації повної пухлини, з F1-Score та IoU більше 0.8. Крім того, ядро пухлини також сегментується з високою точністю, з F1-Score та IoU значеннями 0.8675 та 0.7844 відповідно. Хоча сегментація ділянки підсиленої контрастом пухлини та едеми може бути дещо гіршою, оцінки якості моделі для цих задач залишаються задовільними.
4. Візуальний контроль показав високу точність моделі.
5. Сукупність вищенаведених фактів вказує на те, що запропонований метод може бути покращений та застосований у майбутньому для автоматичного діагностування та моніторингу зляжисних пухлин головного мозку.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Thakkar, J.P. *et al.* (2014) ‘Epidemiologic and molecular prognostic review of glioblastoma’, *Cancer Epidemiology, Biomarkers & Prevention*, 23(10), pp. 1985–1996. doi:10.1158/1055-9965.epi-14-0275.
2. *ISBI 2015*. Available at: <https://biomedicalimaging.org/2015/>.
3. American Cancer Society (2023) ‘Cancer Facts & Figures 2023’. Atlanta: American Cancer Society.
4. Ellor, S.V., Pagano-Young, T.A. and Avgeropoulos, N.G. (2014) ‘Glioblastoma: Background, standard treatment paradigms, and supportive care considerations’, *Journal of Law, Medicine & Ethics*, 42(2), pp. 171–182. doi:10.1111/jlme.12133.
5. Phillips, H.S. *et al.* (2006) ‘Molecular subclasses of high-grade glioma predict prognosis, delineate a pattern of disease progression, and resemble stages in neurogenesis’, *Cancer Cell*, 9(3), pp. 157–173. doi:10.1016/j.ccr.2006.02.019.
6. Wilson, T.A., Karajannis, M.A., and Harter, D.H. (2014). Glioblastoma multiforme: State of the art and future therapeutics. *Surgical Neurology International*, 5, 64–62. doi:10.4103/2152 -7806.132138
7. Chen, J., McKay, R.M., and Parada, L.F. (2012). Malignant glioma: Lesions from genomics, mouse models, and stem cells. *Cell*, 149, 36–47. doi:10.1016/j.cell.2012.03.009
8. Nabors, L.B. *et al.* (2015) ‘Central Nervous System Cancers, version 1.2015’, *Journal of the National Comprehensive Cancer Network*, 13(10), pp. 1191–1202. doi:10.6004/jnccn.2015.0148.
9. Havaei, M., Jodoin, P.-M. and Larochelle, H. (2014) ‘Efficient Interactive Brain Tumor segmentation as within-brain knn classification’, *2014 22nd International Conference on Pattern Recognition* [Preprint]. doi:10.1109/icpr.2014.106.

10. Zöllner, F.G., Emblem, K.E. and Schad, L.R. (2010) ‘Support Vector Machines in DSC-based glioma imaging: Suggestions for optimal characterization’, *Magnetic Resonance in Medicine*, 64(4), pp. 1230–1236. doi:10.1002/mrm.22495.
11. Mahmood, Q. and Basit, A. (2016) ‘Automatic ischemic stroke lesion segmentation in multi-spectral MRI images using random forests classifier’, *Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries*, pp. 266–274. doi:10.1007/978-3-319-30858-6_23.
12. Zhao, X. *et al.* (2018) ‘A deep learning model integrating fcnn and CRFS for brain tumor segmentation’, *Medical Image Analysis*, 43, pp. 98–111. doi:10.1016/j.media.2017.10.002.
13. Zhang, W. *et al.* (2021) ‘Overview of multi-modal brain tumor mr image segmentation’, *Healthcare*, 9(8), p. 1051. doi:10.3390/healthcare9081051.
14. McKinley, R. *et al.* (2018a) ‘Pooling-free fully convolutional networks with dense skip connections for semantic segmentation, with application to brain tumor segmentation’, *Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries*, pp. 169–177. doi:10.1007/978-3-319-75238-9_15.
15. Anita, J.N. and Kumaran, S. (2022) ‘A deep learning architecture for meningioma brain tumor detection and segmentation’, *Journal of Cancer Prevention*, 27(3), pp. 192–198. doi:10.15430/jcp.2022.27.3.192.
16. Bisong, E. (2019) ‘Google colaboratory’, *Building Machine Learning and Deep Learning Models on Google Cloud Platform*, pp. 59–64. doi:10.1007/978-1-4842-4470-8_7.
17. Van Rossum, G. & Drake Jr, F.L., 1995. *Python reference manual*, Centrum voor Wiskunde en Informatica Amsterdam
18. Paszke, A. *et al.* (2019) ‘PyTorch: An Imperative Style, High-Performance Deep Learning Library’, *Advances in Neural Information Processing Systems*.
19. R. Beare, B. C. Lowekamp, Z. Yaniv (2018) ‘Image Segmentation, Registration and Characterization in R with SimpleITK’, *J Stat Softw*, 86(8), [doi:10.18637/jss.v086.i08](https://doi.org/10.18637/jss.v086.i08)

20. B. H. Menze, A. Jakab, S. Bauer, J. Kalpathy-Cramer, K. Farahani, J. Kirby, et al. (2015) ‘The Multimodal Brain Tumor Image Segmentation Benchmark (BRATS)’, *IEEE Transactions on Medical Imaging* 34(10), 1993-2024 DOI:10.1109/TMI.2014.2377694
21. S. Bakas, H. Akbari, A. Sotiras, M. Bilello, M. Rozycki, J.S. Kirby, et al. (2017) "Advancing The Cancer Genome Atlas glioma MRI collections with expert segmentation labels and radiomic features", *Nature Scientific Data*, 4:170117 DOI:10.1038/sdata.2017.117
22. S. Bakas, M. Reyes, A. Jakab, S. Bauer, M. Rempfler, A. Crimi, et al. (2018) ‘Identifying the Best Machine Learning Algorithm for Brain Tumor Segmentation, Progression Assessment, and Overall Survival Prediction in the BRATS Challenge’, arXiv preprint arXiv:1811.02629
23. Ronneberger, O., Fischer, P., Brox, T. (2015). ‘U-Net: Convolutional Networks for Biomedical Image Segmentation’. *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015. Lecture Notes in Computer Science()*, vol 9351. Springer, Cham.
24. Isensee, F., Jäger, P.F., Full, P.M., Vollmuth, P., Maier-Hein, K.H. (2021). ‘nnU-Net for Brain Tumor Segmentation’. In: Crimi, A., Bakas, S. (eds) *Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries. BrainLes 2020. Lecture Notes in Computer Science()*, vol 12659. Springer, Cham. https://doi.org/10.1007/978-3-030-72087-2_11
25. Jia H., Cai W., Huang H., Xia Y. (2021). ‘H2NF-Net for Brain Tumor Segmentation Using Multimodal MR Imaging: 2nd Place Solution to BraTs Challenge 2020 Segmentation Task’. *Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries. BrainLes 2020*. <https://doi.org/10.48550/arXiv.2012.15318>
26. Laukamp, K.R. et al. (2020) ‘Automated meningioma segmentation in multiparametric MRI’, *Clinical Neuroradiology*, 31(2), pp. 357–366. doi:10.1007/s00062-020-00884-4.

ДОДАТКИ

Додаток А

ПОПЕРЕДНЯ ОБРОБКА ДАНИХ

```

import random
from skimage.exposure import rescale_intensity

def get_tumor(modality, segmentation):
    random.seed(100)
    mask = [any(element in [1, 2, 4] for element in i)
            for i in segmentation.reshape((segmentation.shape[0], -1))]
    mask = [i for i in range(len(mask)) if mask[i] == True]
    z_min, z_max = min(mask), max(mask)

    epsilon = random.choice([-1, 0, 1]) * 14
    if z_min + epsilon < 0:
        epsilon = -z_min
    if z_max + epsilon > 153:
        epsilon = 153 - z_max

    return (modality[z_min + epsilon : z_max + epsilon + 1 :, :],
            segmentation[z_min + epsilon : z_max + epsilon + 1 :, :])

def fix_labels(volume):
    volume[volume == 4.] = 3.
    return volume

def crop_volumes(v, m):
    return (v[:, 8: 232, 8: 232], m[:, 8: 232, 8: 232])

def normalize_volume(volume):
    p10 = np.percentile(volume, 10)
    volume = rescale_intensity(volume, in_range=(p10, volume.max()))

    brain_mask = np.zeros(volume.shape, dtype=bool)
    brain_mask[volume > 0.] = True
    volume /= 255.
    mean = volume[brain_mask].mean()
    std = volume[brain_mask].std()
    volume = (volume - mean) / (std + 1e-8)
    volume[np.logical_or(brain_mask == False, volume < 0.)] = 0.

    return volume

def preprocess(volume, segmentation):
    volume_patch, seg_patch = get_tumor(volume, segmentation)
    volume_patch, seg_patch = crop_volumes(volume_patch, seg_patch)
    volume_patch, seg_patch = volume_patch.astype(np.float32), seg_patch.astype(np.uint8)
    volume_patch = normalize_volume(volume_patch)
    seg_patch = fix_labels(seg_patch)
    return volume_patch, seg_patch

volume = sitk.GetArrayFromImage(sitk.ReadImage(path + "/BraTS20_Training_001_flair.nii"))
segmentation = sitk.GetArrayFromImage(sitk.ReadImage(path + "/BraTS20_Training_001_seg.nii"))
preprocessed_volume, preprocessed_seg = preprocess(volume, segmentation)

```

РЕАЛІЗАЦІЯ МОДЕЛІ U-NET

```

from collections import OrderedDict
from torchsummary import summary

class UNet(nn.Module):
    def __init__(self, in_channels, out_channels, base_features=64):
        super(UNet, self).__init__()

        self.encoder1 = UNet_block(in_channels, base_features, name="enc1_")
        self.pool1 = nn.MaxPool2d(kernel_size=2, stride=2)
        self.encoder2 = UNet_block(base_features, base_features * 2, name="enc2_")
        self.pool2 = nn.MaxPool2d(kernel_size=2, stride=2)
        self.encoder3 = UNet_block(base_features * 2, base_features * 4, name="enc3_")
        self.pool3 = nn.MaxPool2d(kernel_size=2, stride=2)
        self.encoder4 = UNet_block(base_features * 4, base_features * 8, name="enc4_")
        self.pool4 = nn.MaxPool2d(kernel_size=2, stride=2)

        self.bottleneck = UNet_block(base_features * 8, base_features * 16, name="bottleneck_")

        self.upconv4 = nn.ConvTranspose2d(base_features * 16, base_features * 8, kernel_size=2, stride=2)
        self.decoder4 = UNet_block((base_features * 8) * 2, base_features * 8, name="dec4_")
        self.upconv3 = nn.ConvTranspose2d(base_features * 8, base_features * 4, kernel_size=2, stride=2)
        self.decoder3 = UNet_block((base_features * 4) * 2, base_features * 4, name="dec3_")
        self.upconv2 = nn.ConvTranspose2d(base_features * 4, base_features * 2, kernel_size=2, stride=2)
        self.decoder2 = UNet_block((base_features * 2) * 2, base_features * 2, name="dec2_")
        self.upconv1 = nn.ConvTranspose2d(base_features * 2, base_features * 2, kernel_size=2, stride=2)
        self.decoder1 = UNet_block(base_features * 2, base_features, name="dec1_")

        self.final = nn.Conv2d(in_channels=base_features, out_channels=out_channels, kernel_size=1)

    def forward(self, x):
        enc1 = self.encoder1(x)
        enc2 = self.encoder2(self.pool1(enc1))
        enc3 = self.encoder3(self.pool2(enc2))
        enc4 = self.encoder4(self.pool3(enc3))

        bottleneck = self.bottleneck(self.pool4(enc4))

        dec4 = self.decoder4(torch.cat((self.upconv4(bottleneck), enc4), dim=1))
        dec3 = self.decoder3(torch.cat((self.upconv3(dec4), enc3), dim=1))
        dec2 = self.decoder2(torch.cat((self.upconv2(dec3), enc2), dim=1))
        dec1 = self.decoder1(torch.cat((self.upconv1(dec2), enc1), dim=1))

        return torch.sigmoid(self.final(dec1))

    @staticmethod
    def _block(in_channels, features, name):
        return nn.Sequential(OrderedDict([
            (name + "conv1", nn.Conv2d(in_channels=in_channels, out_channels=features,
                                     kernel_size=3, padding=1, bias=False)),
            (name + "bnorm1", nn.BatchNorm2d(num_features=features)),
            (name + "relu1", nn.ReLU(inplace=True)),
            (name + "conv2", nn.Conv2d(in_channels=features, out_channels=features,
                                     kernel_size=3, padding=1, bias=False)),
            (name + "bnorm2", nn.BatchNorm2d(num_features=features)),
            (name + "relu2", nn.ReLU(inplace=True))
        ]))

summary(UNet(4, 3), input_size=(4, 224, 224), verbose=False)

```

ТРЕНУВАННЯ МОДЕЛІ

```

from utils import *
import torch.optim as optim

epochs = 20
lr = 1e-4
weights = "./"
in_channels = 4
out_channels = 3

def train_validate_multi():

    device = torch.device("cpu" if not torch.cuda.is_available() else "cuda:0")
    print("Device used: {}".format(device))
    loader_train, loader_validation = data_loaders(batch_size, workers)
    loaders = {"train": loader_train, "validation": loader_validation}
    model = UNet(in_channels, out_channels, base_features=64)
    model.to(device)
    model.apply(weights_init)
    criterion = MultiClassDiceLoss()
    best_validation_dsc = 0.
    best_WT, best_TC, best_ET = 0., 0., 0.
    best_NEC, best_ED = 0., 0.
    optimizer = optim.Adam(model.parameters(), lr=lr)

    loss_train = []
    loss_validation = []

    step = 0

    for epoch in range(epochs):
        for phase in ["train", "validation"]:
            if phase == "train":
                model.train()
            else:
                model.eval()

            validation_pred = []
            validation_true = []

            for i, sample in enumerate(loaders[phase]):
                if phase == "train":
                    step += 1
                    x, y_true = sample
                    x, y_true = x.to(device), y_true.to(device)
                    optimizer.zero_grad()

                    with torch.set_grad_enabled(phase == "train"):
                        y_pred = model(x)
                        loss = criterion(y_pred, y_true)

                    if phase == "validation":
                        loss_validation.append(loss.item())
                        y_pred_np = y_pred.detach().cpu().numpy()
                        validation_pred.extend(y_pred_np[s] for s in range(y_pred_np.shape[0]))
                        y_true_np = y_true.detach().cpu().numpy()
                        validation_true.extend(y_true_np[s] for s in range(y_true_np.shape[0]))
                    if phase == "train":
                        loss_train.append(loss.item())
                        loss.backward()
                        optimizer.step()

            if phase == "train":
                log_loss_summary(loss_train, epoch, prefix="train_")
                loss_train = []

            if phase == "validation":
                log_loss_summary(loss_validation, epoch, prefix="validation_")
                mean_dsc = np.mean(dsc_per_volume(validation_pred, validation_true,
                                                  loader_validation.dataset.patient_slice_index))
                WT_list, TC_list, ET_list, NEC_list, ED_list = dsc_per_volume(validation_pred, validation_true,
                                                                              loader_validation.dataset.patient_slice_index)
                mean_WT, mean_TC, mean_ET = np.mean(WT_list), np.mean(TC_list), np.mean(ET_list)
                mean_NEC, mean_ED = np.mean(NEC_list), np.mean(ED_list)
                log_scalar_summary("val_dsc", mean_dsc, epoch)
                log_class_summary(mean_WT, mean_TC, mean_ET, epoch)
                log_class_orig_summary(mean_NEC, mean_ED, epoch)
                if mean_dsc > best_validation_dsc:
                    best_validation_dsc = mean_dsc
                    best_WT, best_TC, best_ET = mean_WT, mean_TC, mean_ET
                    best_NEC, best_ED = mean_NEC, mean_ED
                    torch.save(model.state_dict(), os.path.join(weights, "UNET.pt"))
                loss_validation = []

```

ОЦІНКА МОДЕЛІ

```

from skimage.io import imread, imsave

device = torch.device("cpu" if not torch.cuda.is_available() else "cuda:0")
print("Device used: {}".format(device))
_, loader_validation = data_loaders(batch_size, workers)
model = UNet(in_channels, out_channels, base_features=64)
model.to(device)

state_dict = torch.load(os.path.join(weights, "UNET.pt"))
model.load_state_dict(state_dict)
model.eval()

input_list = []
pred_list = []
true_list = []

for i, sample in enumerate(loader_validation):
    x, y_true = sample
    x, y_true = x.to(device), y_true.to(device)

    with torch.set_grad_enabled(False):
        y_pred = model(x)

        x_np = x.detach().cpu().numpy()
        y_true_np = y_true.detach().cpu().numpy()
        y_pred_np = y_pred.detach().cpu().numpy()

        input_list.extend([x_np[s] for s in range(x_np.shape[0])])
        true_list.extend([y_true_np[s] for s in range(y_true_np.shape[0])])
        pred_list.extend([y_pred_np[s] for s in range(y_pred_np.shape[0])])

volumes = postprocess_per_volume(input_list, pred_list, true_list,
                                 loader_validation.dataset.patient_slice_index,
                                 loader_validation.dataset.patients)

WT_metrics = []
TC_metrics = []
ET_metrics = []
NEC_metrics = []
ED_metrics = []

for p in volumes:
    x = volumes[p][0]
    y_pred = volumes[p][1]
    y_true = volumes[p][2]
    original_y_pred = np.zeros_like(y_pred).astype(int)
    original_y_true = np.zeros_like(y_true).astype(int)
    original_y_true[:, 0, ...] = y_true[:, 1, ...] - y_true[:, 2, ...]
    original_y_true[:, 1, ...] = y_true[:, 0, ...] - y_true[:, 1, ...]
    original_y_true[:, 2, ...] = y_true[:, 2, ...]

    original_y_pred[:, 0, ...] = y_pred[:, 1, ...] - y_pred[:, 2, ...]
    original_y_pred[:, 1, ...] = y_pred[:, 0, ...] - y_pred[:, 1, ...]
    original_y_pred[:, 2, ...] = y_pred[:, 2, ...]

    WT_metrics.append(list(Get_Metrics(y_pred[:, 0, ...], y_true[:, 0, ...]).values()))
    TC_metrics.append(list(Get_Metrics(y_pred[:, 1, ...], y_true[:, 1, ...]).values()))
    ET_metrics.append(list(Get_Metrics(y_pred[:, 2, ...], y_true[:, 2, ...]).values()))
    NEC_metrics.append(list(Get_Metrics(original_y_pred[:, 0, ...], original_y_true[:, 0, ...]).values()))
    ED_metrics.append(list(Get_Metrics(original_y_pred[:, 1, ...], original_y_true[:, 1, ...]).values()))

    for s in range(x.shape[0]):
        labeled_pred = colour_labels(gray2rgb(x[s, 1]), np.round(original_y_pred[s]))
        labeled_true = colour_labels(gray2rgb(x[s, 1]), original_y_true[s])
        stacked_image = np.hstack((labeled_true, labeled_pred))
        filename = "{}-{}.png".format(p, str(s).zfill(2))
        filepath = os.path.join(".", filename)
        imsave(filepath, stacked_image)

WT_metrics = np.array(WT_metrics).mean(axis=0)
TC_metrics = np.array(TC_metrics).mean(axis=0)
ET_metrics = np.array(ET_metrics).mean(axis=0)
NEC_metrics = np.array(NEC_metrics).mean(axis=0)
ED_metrics = np.array(ED_metrics).mean(axis=0)

```