

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ
І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ**

ІНСТИТУТ ЕНЕРГЕТИКИ, АВТОМАТИКИ І ЕНЕРГОЗБЕРЕЖЕННЯ

ДОПУСКАЄТЬСЯ ДО ЗАХИСТУ
Завідувач кафедри
автоматики та робототехнічних систем
ім. акад. І.І. Мартиненка
(назва кафедри)

К.Т.Н., доц. _____ О.О. Опришко
(підпис) (ПІБ)

" ____ " _____ 2025 р.

БАКАЛАВРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

**на тему "БІОМЕДИЧНА СИСТЕМА КОМП'ЮТЕРНОГО
АНАЛІЗУ МЕДИЧНИХ ЗОБРАЖЕНЬ ГІБРИДНОГО
ТОМОГРАФУ"**

Спеціальність: 163 - "Біомедична інженерія"

Гарант освітньої програми

Д.Т.Н., професор
(науковий ступінь та вчене
звання)

(підпис)

Никифорова Л.Є.
(П.І.Б.)

Керівник бакалаврської кваліфікаційної роботи

К.Т.Н., доцент
(науковий ступінь та вчене
звання)

(підпис)

Кіктев М.О.
(П.І.Б.)

Виконав

(підпис)

Халімоненко Є.Т.
(П.І.Б.)

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ
І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ
ІНСТИТУТ ЕНЕРГЕТИКИ, АВТОМАТИКИ І ЕНЕРГОЗБЕРЕЖЕННЯ**

ЗАТВЕРДЖУЮ

Завідувач кафедри

автоматики та робототехнічних систем

ім. акад. І.І. Мартиненка

(назва кафедри)

д.т.н., доц. О.О.Опришко

(підпис)

(ПБ)

" ____ " _____ 2025 р.

ЗАВДАННЯ

на виконання бакалаврської кваліфікаційної роботи студенту

Халімоненко Єгору Тарасовичу

(прізвище, ім'я, по батькові)

Спеціальність: 163 - "Біомедична інженерія"

1. Тема бакалаврської кваліфікаційної роботи: "Біомедична система комп'ютерного аналізу медичних зображень гібридного томографу", затверджена наказом ректора НУБіП України від "12" 11 2024 р. № 2023"С"

2. Термін подання завершеної роботи на кафедру — "1" червня 2025 р.

3. Вихідні дані до бакалаврської кваліфікаційної роботи:

3.1. Завдання кафедри на виконання бакалаврської кваліфікаційної роботи.

3.2. Нормативні документи по проектуванню біомедичних об'єктів .

3.3. Наукова література з тематики бакалаврської кваліфікаційної роботи.

3.4. Матеріали виробничої практики

4. Перелік питань, які необхідно розробити:

4.1. Аналіз стану медичної проблеми аналізу зображень гібридного томографу

4.2. Дослідження томографії як біологічного об'єкту .

4.3. Огляд відомих математичних моделей опрацювання кардіосигналу.

4.4. Вибір технічних засобів для розпізнавання і аналізу зображень томографу

4.5. Програмні засоби для аналізу зображень гібридного томографу

4.5. Схеми біомедичної системи (структурна, функціональна, принципова електрична).

4.6. Кошторисні розрахунки.

4.7. Техніка безпеки і охорона праці.

5. Перелік графічних документів:

5.1. Структурна і функціональна схеми біомедичної системи комп'ютерного аналізу медичних зображень гібридного томографу_5.2. Схема електрична принципова комп'ютерного аналізу медичних зображень гібридного томографу

5.3. Алгоритм аналізу зображень гібридного томографу з використанням згорткових нейронних мереж.

.

Дата видачі завдання "24" березня 2025 року

**Керівник
бакалаврської
кваліфікаційної
роботи**

(підпис)

Кіктєв М.О.
(П.І.Б.)

**Завдання прийняв до
виконання**

(підпис)

Халімоненко Є.Т.
(П.І.Б.)

РЕФЕРАТ

Дипломна робота «Біомедична система комп'ютерного аналізу медичних зображень гібридного томографу» виконана студентом кафедри автоматизації та робототехнічних систем ННІ енергетики, автоматики та енергозбереження *Халімоненко Єгором Тарасовичем* зі спеціальності 163 «Біомедична інженерія»

Робота присвячена дослідженню концепції побудови біомедичної системи комп'ютерного аналізу медичних зображень, отриманих з гібридних томографів, зокрема ПЕТ/КТ та ПЕТ/МРТ. У роботі розглянуто архітектуру гібридного томографа, апаратну та програмну структуру його складових, а також функціональну модель комп'ютерного аналізу зображень.

Проведено аналіз методів попередньої обробки, реєстрації, сегментації, виділення ознак та класифікації. Особливу увагу приділено алгоритмам машинного навчання та сучасному програмному забезпеченню, такому як 3D Slicer, MITK та OsiriX.

Метою роботи є вдосконалення якості розпізнавання зображень шляхом розробки архітектури штучної нейронної мережі для сегментації МРТ-зображень та її навчання.

Об'єкт дослідження – штучні нейронні мережі для сегментації МРТ-зображень хребта.

Предмет дослідження – методи розробки гібридних архітектур штучних нейронних мереж та застосування машинного навчання для сегментації МРТ-зображень хребта.

Результатом роботи є розроблення інтелектуальної біомедичної системи аналізу зображень томографу на основі гібридної архітектури згорткової нейронної мережі для сегментації МРТ-зображень та навчання моделі за допомогою методу Mean Teacher.

Ключові слова: *гібридна архітектура нейронної мережі, машинне навчання, сегментація зображень, МРТ хребта, інтелектуальна біомедична система.*

ABSTRACT

This thesis "*Biomedical system for computer analysis of medical images of a hybrid tomograph*" was completed by a student of the Department of Automation and Robotic Systems of the National Research Institute of Power Engineering, Automation and Energy Saving *Yegor Khalimonenko* from the specialty 163 "*Biomedical Engineering*"

The work is devoted to the study of the concept of building a biomedical system for computer analysis of medical images obtained from hybrid tomographs, in particular PET/CT and PET/MRI. The work considers the architecture of a hybrid tomograph, the hardware and software structure of its components, as well as the functional model of computer image analysis.

The analysis of methods of preprocessing, registration, segmentation, feature extraction and classification is carried out. Particular attention is paid to machine learning algorithms and modern software, such as 3D Slicer, MITK and OsiriX.

The aim of the work is to improve the quality of image recognition by developing the architecture of an artificial neural network for segmentation of MRI images and its training.

The object of the study is artificial neural networks for segmentation of MRI images of the spine.

The subject of the study is methods for developing hybrid architectures of artificial neural networks and applying machine learning for segmentation of MRI images of the spine.

The result of the work is the development of an intelligent biomedical system for analyzing tomograph images based on a hybrid architecture of a convolutional neural network for segmentation of MRI images and training a model using the Mean Teacher method.

Keywords: *hybrid architecture of a neural network, machine learning, image segmentation, MRI of the spine, intelligent biomedical system.*

ЗМІСТ

ПЕРЕДІК УМОВНИХ ПОЗНАЧЕНЬ	8
ВСТУП	9
РОЗДІЛ 1. МЕДИКО-ТЕХНІЧНЕ ОБҐРУНТУВАННЯ КОМП'ЮТЕРНОГО АНАЛІЗУ МЕДИЧНИХ ЗОБРАЖЕНЬ ГІБРИДНОЇ ТОМОГРАФІЇ	11
1.1. Сучасний стан гібридної томографії	11
1.2. Принцип роботи гібридних томографів	13
1.3. МРТ зображення та їх класифікація	17
РОЗДІЛ 2. ЗГОРТКОВІ НЕЙРОННІ МЕРЕЖІ ДЛЯ МЕДИЧНОЇ СЕГМЕНТАЦІЇ. РЕКОНСТРУКЦІЯ 3D МОДЕЛІ	20
2.1. Задача сегментації та її особливості	20
2.2. Огляд архітектури, методів та ресурсів для сегментації медичних зображень	21
2.3. Математичні моделі для сегментації медичних зображень	25
2.4. Відкриті набори даних сегментованих МРТ-зображень хребта	32
2.5. Опис обраних архітектур, методу та медичної системи	34
2.6. Інтелектуальна медична система діагностики	45
РОЗДІЛ 3. ОПИС ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	48
3.1 Використані технології та бібліотеки	48
3.2 Поетапний опис роботи програми	49
РОЗДІЛ 4. АНАЛІЗ РЕЗУЛЬТАТІВ ДОСЛІДЖЕНЬ	53
4.1. Кероване навчання модифікованих U-Net	53
4.2. Попереднє навчання VAFUNet	57
4.3. Напівкероване навчання VAFUNet	59
4.4. Навчання ResNet	62
РОЗДІЛ 5. ОХОРОНА ПРАЦІ	68
5.1 Характеристика приміщення та робочого місця	68

5.2. Оцінка небезпечних і шкідливих виробничих факторів	70
Висновки	73
Список використаних джерел	76
Додатки	80
Додаток 1	80
Додаток 2	81

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

ПЕТ – позитронно-емісійна томографія

КТ – комп'ютерна томографія

МРТ – магнітно-резонансна томографія

ПЕТ-КТ – гібридна система ПЕТ та КТ

ПЕТ-МРТ – гібридна система ПЕТ та МРТ

DICOM – Digital Imaging and Communications in Medicine (стандарт для зберігання та передачі медичних зображень)

АЦП – аналого-цифровий перетворювач

ФПМ – фотопомножувач

ЗНМ – згорткова нейронна мережа

ШІ – штучний інтелект

3D Slicer, MITK, OsiriX – програмні платформи для аналізу медичних зображень

ПЗ – програмне забезпечення

ККП – коректор коефіцієнта потужності

ПЗК – позиційне зондування координат

ВПП, ШП, ПВЧ, ПЗЧ – типи підсилювачів: вузькополосний, широкополосний, високої частоти, звукових частот відповідно

ВСТУП

У сучасній медичній практиці рання та точна діагностика є вирішальним фактором для успішного лікування багатьох захворювань, зокрема онкологічних, неврологічних та кардіологічних. Інтеграція кількох візуалізаційних методів у єдину систему — як у випадку з гібридною ПЕТ-КТ чи ПЕТ-МРТ — дозволяє поєднати морфологічну та функціональну інформацію про патологічні процеси в організмі.

Позитронно-емісійна томографія (ПЕТ) забезпечує високу чутливість до метаболічних змін ще до того, як з'являються анатомічні зміни, тоді як магнітно-резонансна томографія (МРТ) або комп'ютерна томографія (КТ) дають докладну анатомічну інформацію. Використання гібридної ПЕТ/МРТ візуалізації дозволяє отримати детальні функціонально-анатомічні карти органів і тканин, що особливо важливо в нейронауці, онкології, кардіології та ортопедії.

Згідно з дослідженням European Association of Nuclear Medicine (EANM), гібридні системи ПЕТ/МРТ все частіше використовуються у клінічній практиці завдяки зменшеному променевому навантаженню порівняно з ПЕТ/КТ та кращій візуалізації м'яких тканин. Проте, попри потужні діагностичні можливості гібридних томографів, їхнє повноцінне використання обмежене через складність обробки та аналізу отриманих зображень. Ручна інтерпретація візуальних даних вимагає високої кваліфікації, є часовитратною та суб'єктивною, що підвищує ризик помилок. Це створює нагальну потребу у створенні автоматизованих комп'ютерних систем аналізу медичних зображень.

Інформаційні технології, штучний інтелект і машинне навчання відкривають нові перспективи для точного, швидкого й відтворюваного аналізу медичних зображень. Алгоритми сегментації, реєстрації та класифікації вже довели свою ефективність у численних наукових дослідженнях. Тому розробка біомедичної системи комп'ютерного аналізу зображень з гібридних томографів є надзвичайно актуальним завданням.

Метою роботи є вдосконалення якості розпізнавання зображень шляхом розробки архітектури штучної нейронної мережі для сегментації

МРТ-зображень та її навчання.

Об'єкт дослідження – штучні нейронні мережі для сегментації МРТ-зображень хребта.

Предмет дослідження – методи розробки гібридних архітектур штучних нейронних мереж та застосування машинного навчання для сегментації МРТ-зображень хребта.

Результатом роботи є розроблення інтелектуальної біомедичної системи аналізу зображень томографу на основі гібридної архітектури згорткової нейронної мережі для сегментації МРТ-зображень та навчання моделі за допомогою методу Mean Teacher.

РОЗДІЛ 1

МЕДИКО-ТЕХНІЧНЕ ОБҐРУНТУВАННЯ КОМП'ЮТЕРНОГО АНАЛІЗУ МЕДИЧНИХ ЗОБРАЖЕНЬ ГІБРИДНОЇ ТОМОГРАФІЇ

1.1. Сучасний стан гібридної томографії

Гібридна томографія — це сучасний напрям медичної візуалізації, що об'єднує дані з різних методів сканування (наприклад, ПЕТ, КТ та МРТ), забезпечуючи повнішу клінічну картину. Такий підхід дозволяє одночасно отримати анатомічну (КТ, МРТ) та функціональну (ПЕТ) інформацію про організм пацієнта. Згідно з даними European Society of Radiology (2023), ПЕТ/КТ і ПЕТ/МРТ стали золотим стандартом при онкологічних дослідженнях, кардіології та неврології, дозволяючи виявити пухлини на ранніх стадіях, оцінити метаболічну активність тканин і спрогнозувати ефективність лікування.

Гібридні томографи — це високотехнологічні системи, які об'єднують функціональну та анатомічну візуалізацію в єдиному апараті. Їх розробка є результатом тісної інтеграції досягнень ядерної медицини, цифрової електроніки, обчислювальної техніки та програмного забезпечення для злиття зображень.

Позитронно-емісійна томографія з комп'ютерною томографією (ПЕТ-КТ) — найрозвиненіший і найширше застосовуваний тип гібридного томографа. Він об'єднує в собі два апарати: ПЕТ для виявлення метаболічної активності (наприклад, за допомогою радіофармпрепарату ^{18}F -FDG) та КТ — для точного анатомічного позиціонування. Її технічні особливості:

- ✓ послідовне розташування ПЕТ і КТ у тунелі;
- ✓ синхронізоване програмне забезпечення, що дозволяє корекцію рухів пацієнта;
- ✓ автоматичне суміщення зображень двох методик.

Детектори ПЕТ — найчастіше сцинтилятори з лютецієвим оксидом (LSO) або гадолінієвим оксидом (GSO), які мають високу чутливість до гамма-випромінювання. *Наприклад*, система Biograph Vision (Siemens Healthineers)

забезпечує просторову роздільну здатність менше 4 мм і дозволяє проводити швидкі повнотілесні сканування за низької дози опромінення.

Позитронно-емісійна томографія з магнітно-резонансною томографією (ПЕТ-МРТ) — більш нова, але перспективна технологія. Вона дозволяє поєднати переваги високої роздільної здатності МРТ та функціональної візуалізації ПЕТ без додаткового іонізуючого опромінення.

Основні технічні особливості:

- сумісність детекторів ПЕТ з сильним магнітним полем МРТ (1,5 або 3 Тл);
- заміна традиційних фотопомножувачів на кремнієві фотопомножувачі (SiPM), які не реагують на магнітне поле;
- єдиний простір сканування — ПЕТ і МРТ працюють одночасно, що знижує похибки при реєстрації зображень.

Наприклад, система SIGNA PET/MR (GE Healthcare) є першою повноцінною інтегрованою ПЕТ-МРТ системою для клінічного використання з високою точністю суміщення.

Новими тенденціями в томографії є:

- розробка повнотілесних ПЕТ-сканерів (total-body PET), таких як uEXPLORER[®] (United Imaging), які дозволяють отримати повну карту метаболізму всього тіла за один скан, скоротивши час обстеження.
- інтеграція штучного інтелекту (ШІ) для покращення реконструкції зображень, зниження артефактів і автоматичного аналізу аномалій.
- застосування гібридних систем у персоналізованій медицині, де дані ПЕТ-КТ чи ПЕТ-МРТ використовуються для створення індивідуальних моделей лікування (радіотерапія, імунотерапія тощо).

Таким чином, гібридна томографія — це передова технологія медичної візуалізації, яка дозволяє покращити якість діагностики, скоротити час обстеження та підвищити ефективність клінічних рішень завдяки поєднанню кількох джерел зображень в одній системі (рис. 3.1).

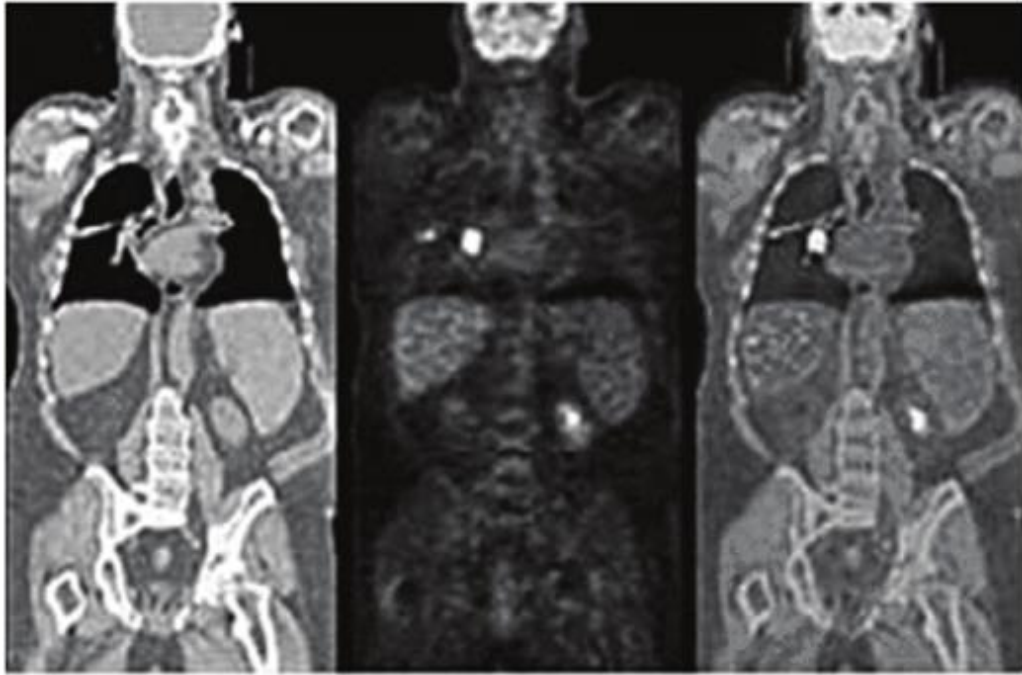


Рисунок 1.1. Схема поєднання зображень ПЕТ та КТ у гібридній системі.

1.2. Принцип роботи гібридних томографів

Важливою перевагою використання гібридних томографів є можливість отримання багаторівневої інформації про стан пацієнта, яка може бути проаналізована для створення тривимірної реконструкції (рис. 1.2.).

1. Введення радіофармпрепарату. Людині вводять речовину, яка мітиться радіоактивним ізотопом. Цей препарат розподіляється по тілу, концентрується в активних метаболічних зонах.

2. Розпад позитронів. Радіоактивний ізотоп випускає позитивні частинки - позитрони, які стикаються з негативними електронами в тканинах організму. У результаті цього зіткнення виділяються два гамма-фотони (вид електромагнітного випромінювання з надзвичайно малою довжиною хвилі).

3. Детектування (уловлювання) частинок. Гамма-фотони випускаються під кутом 180 градусів один до одного, вловлюються детекторами, розташованими навколо пацієнта.

4. Збір, обробка даних. Система обробки даних аналізує час, розташування кожного зареєстрованого зіткнення частинок.

5. Реконструкція зображення. Комп'ютер використовує складні алгоритми для реконструкції знімків. Ці скани показують розподіл радіофармпрепарату в тілі, що відображає метаболічну активність різних тканин.

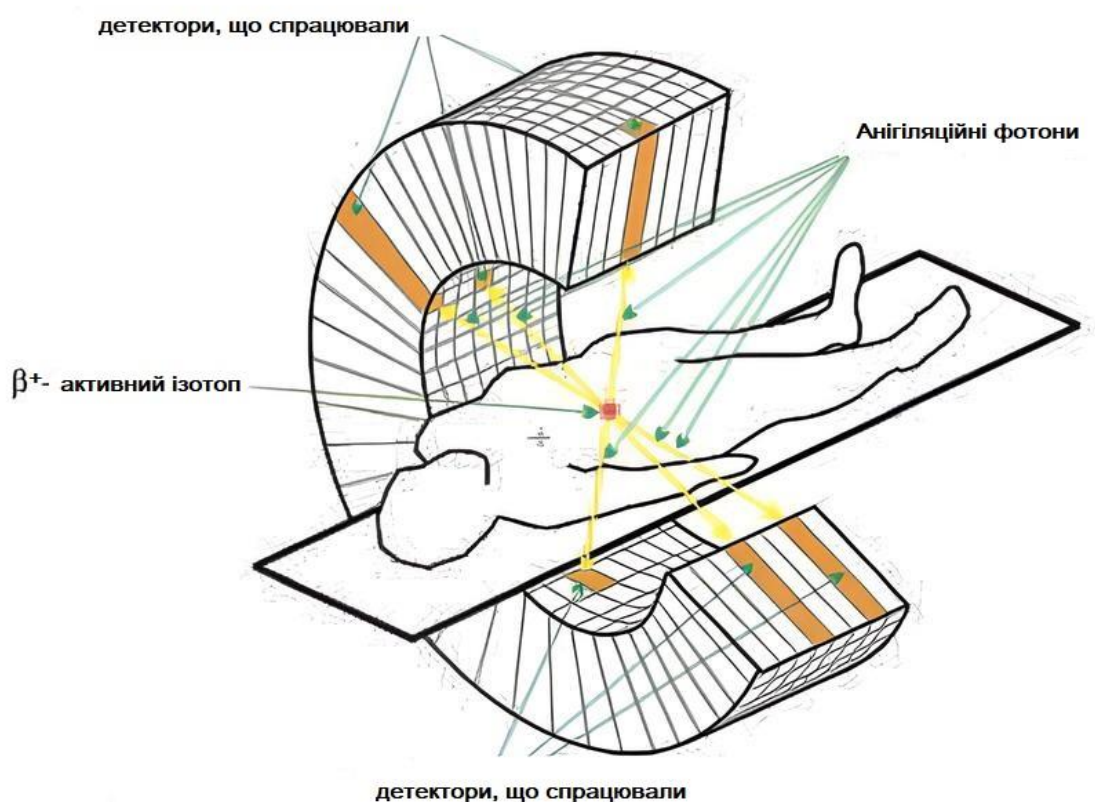


Рисунок 1.2. Принцип роботи ПЕТ сканера

При скануванні ПЕТ машина виділяє випромінювання, що генеруються радіотрейсером. Радіотрейсер складається з радіоактивного матеріалу, який пов'язаний з природньою хімічною речовиною на зразок глюкози.

Цей радіотрейсер вводиться в організм, де він потрапляє в клітини, які використовують глюкозу для отримання енергії. Чим більше енергії потрібно групі клітин, тим більше радіотрейсер буде накопичуватися в цьому місці. Це буде відображатися на зображеннях, які відтворюються комп'ютером.

Клітини або їх активність будуть відображатися як «гарячі точки» або «холодні точки». «Гарячі точки» є активними та яскравими на ПЕТ. Там, де клітинам потрібно менше енергії, області будуть менш яскравими — так виглядають «холодні точки».

У порівнянні з нормальними клітинами, ракові клітини дуже активно використовують глюкозу, тому радіотрейсер візуалізує пухлинні ділянки.

Прикладом радіоактивного індикатора на основі глюкози є фтордезоксиглюкоза (ФДГ). У ФДГ молекули радіоактивного фтору помічені на глюкозу, щоб зробити радіоактивний слід. ФДГ — це найбільш поширене джерело радіотрас, яке використовується сьогодні. Замість глюкози можна використовувати кисень.

Переваги та недоліки гібридної томографії

Гібридна томографія, завдяки поєднанню функціональної та морфологічної інформації, має низку переваг, які роблять її одним з найефективніших інструментів у сучасній діагностичній медицині.

Переваги:

- ✓ Суміщення функціонального та анатомічного зображення
- ✓ Підвищена точність діагностики
- ✓ Оптимізація планування лікування
- ✓ Зменшення часу дослідження
- ✓ Менше радіаційне навантаження (у разі ПЕТ-МРТ)
- ✓ Підвищення рівня виявлення прихованих патологій

Недоліки:

- ✓ Висока вартість обладнання та обстеження
- ✓ Необхідність використання радіо фармпрепаратів
- ✓ Висока доза опромінення (у ПЕТ-КТ)
- ✓ Високі вимоги до кваліфікації персоналу
- ✓ Обмеження за часом сканування

Застосування гібридної томографії в клінічній практиці

Гібридні томографи, що поєднують функціональну та анатомічну візуалізацію, знайшли широке застосування в різних галузях медицини. Основні

сфери використання включають: онкологія, кардіологія, неврологія, педіатрія, травматологія та ортопедія.

Гібридні томографи, особливо ПЕТ-КТ та ПЕТ-МРТ, широко використовуються для виявлення, стадіювання та моніторингу онкологічних захворювань. Вони дозволяють точно локалізувати пухлини, оцінити їх метаболічну активність та виявити метастази.

У кардіології гібридні томографи використовуються для оцінки життєздатності міокарда, виявлення ішемічних ділянок та запальних процесів. Це сприяє ранній діагностиці та ефективному плануванню лікування серцево-судинних захворювань.

У неврології гібридні томографи застосовуються для діагностики нейродегенеративних захворювань, таких як хвороба Альцгеймера, епілепсія та пухлини головного мозку. Вони дозволяють оцінити як структурні, так і функціональні зміни в мозку.

У педіатрії, де важливо мінімізувати дозу опромінення, гібридні томографи, особливо ПЕТ-МРТ, дозволяють отримати високоякісні зображення з меншою радіаційною експозицією.

Гібридні томографи допомагають виявляти приховані переломи, оцінювати стан кісткової тканини та виявляти запальні процеси в суглобах. Провідні світові клініки активно застосовують гібридну томографію в усіх вищенаведених напрямках. В Україні такі системи доступні в приватних медичних центрах, наприклад у «Lisod», «Tomoclinic», клініці «Оберіг». Дослідження, опубліковане в The Journal of Nuclear Medicine, також показує, що ПЕТ - МРТ значно зменшує загальне опромінення в порівнянні з ПЕТ / КТ - особливо корисно для пацієнтів дитячого та підліткового віку.

Останніми роками активно впроваджується практика штучного інтелекту в обробці зображень ПЕТ/КТ та ПЕТ/МРТ, що значно полегшує інтерпретацію результатів для лікарів та знижує людський фактор у постановці діагнозу.

1.3. МРТ зображення та їх класифікація

Магнітно-резонансна томографія (МРТ) служить основним діагностичним інструментом для шванном головного мозку. Він забезпечує високий рівень деталізації м'яких тканин, що уможлиблює отримати точне уявлення про стан черепних нервів, особливості пухлини та прилеглих структур. Шванноми на МРТ зазвичай виглядають як чітко окреслені інкапсульовані утворення, що уможлиблює відрізнити їх від інших типів пухлин або патологій. Для підвищення точності діагностики використовують контрастні речовини, що посилюють яскравість пухлини на отриманому зображенні. МРТ полегшує як первинну діагностику, дозволяючи знайти пухлину, так і довгострокове лікування. Лікарі можуть протягом певних інтервалів оцінювати динаміку росту, вплив на сусідні структури тощо.

Існує декілька режимів роботи МРТ апаратів, кожен із яких виділяє різні фізичні особливості обстежуваних тканин. Так, наприклад, T1-зважені зображення залежать від часу поздовжньої релаксації протонів водню після збудження магнітним імпульсом. На цих зображеннях тканини, що містять багато жиру, виглядають яскравими, оскільки у жиру короткий час релаксації; навпаки, у тканинах, багатих на воду, час релаксації довший, тому на зображеннях вони виглядають темнішими.

T1+C режим є вдосконаленою модифікацією базового режиму T1. Дана візуалізація виконується після введення в тканини контрастного препарату на основі гадолінію. Контрастна речовина скорочує час T1-релаксації у тканинах, де вона накопичується, завдяки чому ці ділянки виглядають яскравішими.

Режим T2 має інший принцип дії. T2-зважена візуалізація вимірює поперечне розслаблення, фокусуючись на процесі дефазування протонів водню після збудження. На відміну від режиму T1, тканини, багаті на воду, виглядають яскравими, а тканини, що містять жир, відносно темними. Проте інтенсивний сигнал від води може іноді приховувати важливу інформацію,

наприклад при спробах виявити ураження у заповнених рідиною ділянках.

Цю проблему вирішує режим T2 FLAIR (Fluid Attenuated Inversion Recovery). У цій методиці перед отриманням типового T2-зваженого зображення МРТ-сканер генерує радіочастотний імпульс на 180 градусів. Цей імпульс тимчасово змінює магнітне вирівнювання, забезпечуючи точно синхронізований інверсійний час, протягом якого первинний сигнал майже повністю пригнічується. Коли первинний сигнал "вимкнений" або заглушений, МРТ створює T2-зважене зображення.

T2 3D DRIVE вирішує іншу проблему. У типових T2-зображеннях сигнал швидко згасає, особливо в тканинах із середнім або коротким T2-часом розслаблення. Це згасання може ускладнити отримання дуже деталізованих зображень, особливо при вивченні дрібних структур або незначних відхилень. Для цього T2 3D DRIVE використовує техніку під назвою "керована рівновага" (driven equilibrium). Після першого імпульсу, який збуджує тканину, МРТ додає додаткові підготовчі імпульси для "рефокусування" та посилення T2-зваженого сигналу. Фактично, це спосіб прискорити та ефективніше повернути магнетизацію до природного стану, що уможливорює пристрою вловити більше сигналів навіть від тканин, які зазвичай швидко згасають. Частина "3D" у назві стосується здатності цієї послідовності отримувати зображення з дуже тонкими зрізами та високою просторовою роздільною здатністю у трьох вимірах.

Аналізуючи всю вищенаведену інформацію, неможливо переоцінити важливість МРТ апаратів як основного діагностичного інструменту для лікування пацієнтів із вестибулярними шваномами. Утім, знімки на даних апаратах не завжди відповідають задовільним якісним характеристикам. Це може бути зумовлено технічними обмеженнями МРТ апаратів або їх неправильним налаштуванням. А це, у свою чергу, загрожує створити помилки з боку медичного персоналу у трактуванні результатів МРТ знімків. Інтерпретація отриманих зображень, значною мірою залежить від досвіду і компетенції медичного персоналу.

Висновки

У першому розділі було розглянуто теоретичні основи функціонування гібридних томографів, зокрема ПЕТ-КТ та ПЕТ-МРТ, а також проаналізовано сучасні методи комп'ютерного аналізу медичних зображень, що застосовуються у цих системах.

Встановлено, що гібридні томографи поєднують в собі переваги морфологічної (КТ, МРТ) та функціональної (ПЕТ) візуалізації, що забезпечує високу діагностичну цінність, особливо у випадках онкологічних, кардіологічних та неврологічних захворювань. Основними перевагами таких систем є можливість одночасного отримання структурної і метаболічної інформації, що дає змогу точніше визначати локалізацію і характер патологічних змін.

МРТ апарати на фізичному рівні взаємодіють із протонами водню в головному мозку. Отримані метрики відрізняються режимами, у яких проводяться обстеження. Було розглянуто п'ять режимів роботи МРТ апаратів: T1, T1+C, T2, T2 FLAIR, T2 3D DRIVE. В режимі T1 використовується швидкість T1-релаксації протонів. Покращенням цього режиму є режим T1+C, в якому застосовується контрастна рідина для більш яскравих зображень пухлин. Режим T2 використовує швидкість T2-релаксації, а його покращення T2 FLAIR та T2 3D DRIVE використовують обробку сигналів для кращого виділення ознак на МРТ зображеннях.

Таким чином, подальші дослідження мають бути спрямовані на розробку ефективних, інтерпретованих та масштабованих моделей комп'ютерного аналізу медичних зображень, адаптованих до специфіки гібридної візуалізації.

РОЗДІЛ 2

ЗГОРТКОВІ НЕЙРОННІ МЕРЕЖІ ДЛЯ МЕДИЧНОЇ СЕГМЕНТАЦІЇ. РЕКОНСТРУКЦІЯ 3D МОДЕЛІ

2.1. Задача сегментації та її особливості

Сегментація зображень – одна із ключових задач комп'ютерного зору. Вона передбачає розподіл зображення на окремі частини або сегменти з метою виділення об'єктів, областей або інших значущих елементів. Суть сегментації полягає у тому, щоб кожному пікселю зображення було призначено певну категорію або властивість.

Сегментація може варіюватись залежно від задачі. Бінарна сегментація є одним із найпростішим видів, коли зображення ділиться на дві категорії: об'єкт та фон. Це корисно в задачах, де важливо виділити один або декілька об'єктів із загального фону. Семантична сегментація є більш складним підходом і передбачає, що кожному пікселю призначається певний клас із наперед визначеного набору. Наприклад, у контексті вуличного зображення семантична сегментація може позначати окремі пікселі як частини дороги, автомобілів, тротуарів чи пішоходів.

Однією з головних особливостей сегментації є залежність від природи даних та області застосування. У медичних зображеннях, наприклад, використовують сегментацію для виявлення органів або патологічних змін у тканинах. Тут важлива надзвичайно висока точність — навіть незначні похибки можуть призвести до помилкових висновків чи некоректної діагностики. Навпаки, у випадку автономних автомобілів акцент робиться на швидкості й загальній точності, оскільки потрібно ідентифікувати великі області, такі як дороги, будівлі чи інші транспортні засоби, у реальному часі.

2.2. Огляд архітектури, методів та ресурсів для сегментації медичних зображень

Згідно з дослідженням [1], у період 2015-2021 років глибоке навчання в аналізі зображень хребта розглядалося в 79 публікаціях, при цьому в 40.5% з них використовувалися МРТ-зображення. Задача діагностування вирішувалася в 35.4% від усіх публікацій, а задача сегментації в 30.4%, причому найбільш популярною архітектурою для сегментації медичних зображень була U-Net та її модифікації.

У процесі аналізу літератури були знайдені роботи [2-6], в яких розглядалася задача сегментації МРТ-зображень, публікації [7-9], які присвячені використанню методів напівкерованого навчання щодо сегментації медичних зображень, а також відкриті набори даних сегментованих МРТ-зображень хребта [10-12].

2.2.1. Архітектури нейронних мереж для сегментації

Серед літератури, в якій розглядалися архітектури штучних нейронних мереж для сегментації МРТ-зображень, за останні роки найбільш цікавими виявилися публікації [2-6]:

- в роботі [2] використовувалися архітектури U-Net++ та Yolov5x, при цьому було застосоване трансферне навчання: для нейронних мереж використовувалися попередньо підготовлені вагові коефіцієнти на наборах даних IMAGENET для U-Net++ і COCO 2017 для Yolov5x;

- в дослідженні [3] пропонується використання SCRL (Sequential Conditional Reinforcement Learning) – послідовного умовного

- навчання з підкріпленням, та 3 мереж: мережі анатомічного моделювання для фокусування уваги, повнозв'язної залишкової мережі ResNet для виявлення обмежувальної рамки, та мережі Y-Net для сегментації;

– в роботі [4] розглядається Spine Explorer – програма, що використовує U-Net для сегментації тіл хребців, міжхребцевих дисків та хребетного каналу;

– в роботі [5] пропонується MMCL-Net (Multi-task Multi-structure Correlation Learning Network) – багатозадачна багатоструктурна мережа кореляційного навчання, що надає можливість одночасно виявляти, сегментувати та класифікувати (визначати ступінь пошкодження) тіла хребців, міжхребцеві диски та хребетний канал;

– в статті [6] розглядається система для мультимодальної двовимірної сегментації тіл хребців і міжхребцевих дисків, що може виконувати сегментацію МРТ, КТ та рентгенівських знімків, в основі системи лежить адаптована архітектура Mask R-CNN.

Розглянемо більш детально архітектури нейронних мереж, що застосовувалися в цих публікаціях.

Сегментація та виявлення за допомогою U-Net++ та Yolov5x. В роботі [2] зазначалося, що через нестачу тренувальних даних була помічена схильність нейронних мереж до перенавчання навіть у випадку застосування аугментації даних. Також було припущено, що наявність шумів на вхідних зображеннях негативно вплинуло на точність. У випадку сегментації МРТ-зображень хребта, U-Net++ продемонструвала на валідаційній вибірці значення коефіцієнту Дайса 0.93 при сегментації тіл хребців та 0.96 для міжхребцевих дисків.

Послідовне умовне навчання з підкріпленням (SCRL) та сегментація за допомогою Y-Net. Підхід, розглянутий в роботі [3], використовує навчання з підкріпленням для створення анатомічної моделі хребта перед виконанням сегментації хребців та міжхребцевих дисків. Окрім сегментації модель виконує класифікацію виявлених хребців. Запропонована модель досягає таких результатів: індекс Жаккара: 0.923, коефіцієнт Дайса: 0.926. Менша точність в порівнянні з результатами статті [2] може бути пояснена меншим розміром

набору даних та його більшою різноманітністю, оскільки в дослідженні [3] використовувалися МРТ-зображення від апаратів МРТ різних моделей.

Spine Explorer. Програма, запропонована в роботі [4] цікава тим, що здатна виконувати сегментацію МРТ-зображень хребта разом з обчисленням певних показників, які можуть бути корисними в діагностиці. При тренуванні U-Net в [4] використовувався набір даних лише з 50 МРТ-зображень для тренування та 50 для тестів. При цьому були отримані непогані результати – індекс Жаккара: 0.947 для хребців і 0.926 для дисків.

Багатозадачна багатоструктурна мережа кореляційного навчання MMCL-Net. Особливістю мережі, яка була запропонована в роботі [5], є те, що вона здатна одночасно з сегментацією виконувати класифікацію частин хребта за ступенем ураження (нормальний стан або легкі, виражені чи важкі ураження). MMCL-Net складається з 3 модулів: щільно зв'язаної розширеної залишкової мережі Densely Dilated ResNet, множини рівнів глибокої згортки Deep Convolution Level Set та модуль злиття ознак екземпляру Instance Feature Merge. Також застосовуються 3 додаткові підмодулі: пірамідна мережа ознак Feature Pyramid Network, мережа регіональних пропозицій Region Proposal Network та вирівнювання регіону інтересу ROI Align. В результаті тренування такої складної мережі на наборі даних з 200 МРТ-зображень були отримані результати: коефіцієнт Дайса: 0.9060, точність класифікації: 0.9067.

Система для мультимодальної двовимірної сегментації тіл хребців і міжхребцевих дисків. В дослідженні [6] пропонується система, яка здатна виконувати сегментацію МРТ, КТ та рентгенівських знімків. В основі цієї системи – адаптована архітектура Mask R-CNN, що складається з трьох основних частин: мережа створення ознак (Feature Generation Network), мережа розпізнавання регіонів (Region Recognition Network) і мережа виявлення орієнтирів (Landmark Detection Network). При навчанні використовувалися набори даних з 898 МРТ-зображень, 110 КТ-зображень та 387 рентгенівських знімків, однак розмір тренувальної вибірки був в 5 разів більший за рахунок аугментації даних. При сегментації МРТ-зображень були

отримані результати – коефіцієнт Дайса: 0.958 для тіл хребців та 0.959 для міжхребцевих дисків.

2.2.2. Методи напівкерованого навчання

Роботи [7-9] присвячені використанню методів напівкерованого навчання щодо сегментації медичних зображень, в них розглядалися такі методи:

в дослідженні [7] пропонується новий метод напівкерованого навчання – кореляційне взаємонавчання (Correlation Aware Mutual Learning, CAML) для сегментації медичних зображень, яке засновано на перехресному псевдоконтролі (CPS);

в статті [8] пропонується ще один метод напівкерованого навчання для сегментації медичних зображень – MLB-Seg (Meta-Learning for Bootstrapping Medical Image Segmentation) та алгоритм покращення псевдоміток (Pseudo Label Enhancement, PLE);

в роботі [9] пропонується модифікація методу Mean Teacher за рахунок використання двонаправленого копіювання маркованих та немаркованих даних.

Розглянемо більш детально ці методи.

CAML для сегментації медичних зображень. В роботі [7] ключовою ідеєю є врахування кореляції між вибірками при напівкерованій сегментації, що вирішується за рахунок додавання модуля взаємної уваги між вибірками та модуля омні-кореляційної узгодженості. В результаті спостерігається покращення результатів в порівнянні з іншими методами напівкерованого навчання. Зазначається, що для набору з 20% вмістом маркованих даних використання CAML може забезпечити досягнення майже такої точності, що і у разі використання моделі з керованим навчанням на 100% маркованих даних.

MLB-Seg та PLE. Метод з дослідження [8] передбачає навчання моделі сегментації на невеликому наборі маркованих зображень для створення початкових міток для немаркованих даних. Крім цього вводиться система відображення ваг пікселів, алгоритм покращення псевдоміток і застосовується метод Mean Teacher з метою зменшення шуму та стабілізації ваг.

Двонаправлене копіювання маркованих та немаркованих даних в методі Mean Teacher. В роботі [9] розглядається варіація методу Mean Teacher, де для навчання мережі-учня виконується аугментація вхідних даних шляхом копіювання випадкових частин маркованого зображення (переднього плану) на немарковане зображення (фон) і, навпаки, копіювання випадкових частин немаркованого зображення (переднього плану) на марковане зображення (фон). В такому випадку два змішаних зображення допомагають мережі вивчити спільну семантику між маркованими та немаркованими даними двонаправлено та симетрично. Також зазначається, що використання цього підходу допомагає в певній мірі запобігти перенавчанню на малих вибірках.

2.3. Математичні моделі для сегментації медичних зображень

2.3.1. Задача сегментації та її особливості

Сегментація зображень – одна із ключових задач комп'ютерного зору. Вона передбачає розподіл зображення на окремі частини або сегменти з метою виділення об'єктів, областей або інших значущих елементів. Суть сегментації полягає у тому, щоб кожному пікселю зображення було призначено певну категорію або властивість.

Сегментація може варіюватись залежно від задачі. Бінарна сегментація є одним із найпростішим видів, коли зображення ділиться на дві категорії: об'єкт та фон. Це корисно в задачах, де важливо виділити один або декілька об'єктів із загального фону. Семантична сегментація є більш складним підходом і передбачає, що кожному пікселю призначається певний клас із наперед визначеного набору. Наприклад, у контексті вуличного зображення семантична сегментація може позначати окремі пікселі як частини дороги, автомобілів, тротуарів чи пішоходів.

Однією з головних особливостей сегментації є залежність від природи даних та області застосування. У медичних зображеннях, наприклад, використовують сегментацію для виявлення органів або патологічних змін у

тканинах. Тут важлива надзвичайно висока точність — навіть незначні похибки можуть призвести до помилкових висновків чи некоректної діагностики. Навпаки, у випадку автономних автомобілів акцент робиться на швидкості й загальній точності, оскільки потрібно ідентифікувати великі області, такі як дороги, будівлі чи інші транспортні засоби, у реальному часі.

2.3.2. Огляд традиційних підходів до сегментації медичних зображень

Методи сегментації зображень на основі інтенсивності пікселів

Традиційні методи сегментації пухлин головного мозку використовують різноманітні методи обробки зображень таким чином, щоб виділити межі пухлин на основі інтенсивності пікселів. До таких методів відносяться порогова сегментація, контурний аналіз та виділення особливостей на основі яскравості і контрасту.

Метод порогової сегментації виділяє об'єкти на зображенні на основі інтенсивності пікселів. Він визначає область пухлини шляхом розподілу пікселів на дві групи: з інтенсивністю, що перевищує певне порогове значення, і з інтенсивністю, що нижче цього значення. Порогову сегментацію можна записати формулою:

$$I(x, y) = \begin{cases} 1, & \text{if } f(x, y) \geq T \\ 0, & \text{if } f(x, y) < T, \end{cases}$$

де $I(x, y)$ – результат сегментації (1 для області пухлини, 0 для фону);

$f(x, y)$ – інтенсивність пікселя в позиції (x, y) ;

T – порогове значення.

Іншим традиційним методом є контурний аналіз. Контурний аналіз знаходить межі об'єкта на основі градієнта інтенсивності. Основна ідея полягає у виявленні швидких змін інтенсивності між пухлиною і нормальними тканинами. Для обчислення градієнта зображення застосовуються різні

оператори, наприклад оператор Собеля або оператор Каннінга. Розглянемо більш детально оператор Собеля. Він визначається двомаядрами для горизонтального G_x і вертикального G_y напрямків:

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, \quad G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}.$$

Для кожного пікселя виконується обчислення горизонтального та вертикального градієнтів

$$G_x * f(x, y), G_y * f(x, y),$$

де $*$ – операція згортки.

Результуючий градієнт $|\nabla f(x, y)|$ для кожного пікселя (x, y) визначається як:

$$|\nabla f(x, y)| = \sqrt{(G_x * f(x, y))^2 + (G_y * f(x, y))^2}.$$

Високі значення $|\nabla f(x, y)|$ вказують на значні зміни яскравості, що властиво контурам пухлин. Натомість низькі значення $|\nabla f(x, y)|$ більше властиві однорідним зонам з мінімальними змінами яскравості, тобто, ділянкам без пухлин. Окрім того, контурний аналіз можливо покращити за допомогою ряду підходів, що полягають в збільшенні контрасту між пухлиною та навколишньою тканиною за допомогою перетворень, що акцентують яскраві пікселі. Одним із популярних підходів є гістограмне вирівнювання, яке перетворює інтенсивність пікселів на зображенні таким чином, щоб оптимізувати контраст на всьому зображенні. Для цього використовується

кумулятивна функція розподілу, яка визначає перетворення інтенсивності f у нове значення f' :

$$f'(x, y) = \text{CDF}(f(x, y)) \cdot (L - 1)$$

де $f(x, y)$ – початкова інтенсивність пікселя;

L – максимальне значення інтенсивності (зазвичай 255 для 8-бітових зображень);

CDF – кумулятивна функція розподілу інтенсивностей.

Методи сегментації зображень на основі морфологічних операцій

Методи сегментації на основі морфологічних операцій застосовують власне математичну морфологію для розділення зображення на регіони, що відповідають різним анатомічним структурам, наприклад, пухлинам. Для сегментації пухлин використовують як базові морфологічні операції (ерозія, дилатація, розкриття і закриття), так і їхні похідні (морфологічний градієнт, топ-хет трансформація, трансформація «вододіл»). Кожна з цих операцій виконується над бінарними зображеннями I , а структурний елемент K (наприклад, круг або квадрат) задає геометричний шаблон для змін. Ерозія (erosion) видаляє пікселі з меж об'єктів, зменшуючи їх розмір. Вона використовується для видалення дрібних шумів і зменшення розмірів об'єктів. Формально, ерозія визначається як множина точок, для яких структурний елемент K , центрований навколо певної точки (x, y) , повністю входить в об'єкт I :

$$I_{\text{eroded}}(x, y) = \{(x, y) \mid K \subseteq I(x, y)\}.$$

Ерозію записують наступним чином:

$$I_{\text{eroded}} = I \ominus K.$$

Дилатація (dilation) додає пікселі до меж об'єктів, збільшуючи їх розмір. Вона використовується для заповнення прогалін і з'єднання розірваних компонентів. Формально, дилатація визначається як множина точок, де структурний елемент K , розташований навколо певної точки (x, y) , перетинається з об'єктом I :

$$\text{dilated}(x, y) = \{(x, y) \mid K \cap I(x, y) \neq \emptyset\}.$$

Дилатацію записують наступним чином:

$$I_{\text{dilated}} = I \oplus K.$$

Розкриття (opening) є послідовним застосуванням ерозії, а потім дилатації. Воно використовується для видалення дрібних об'єктів, які не входять до форми K , і згладжування меж:

$$I_{\text{opened}} = (I \ominus K) \oplus K.$$

Закриття (closing) є послідовним застосуванням дилатації, а потім ерозії. Ця операція заповнює прогалини всередині об'єктів і згладжує межі, закриваючи дрібні отвори:

$$I_{\text{closed}} = (I \oplus K) \ominus K.$$

Морфологічний градієнт визначається як різниця між дилатованим і еродованим зображенням. Він використовується для виявлення меж об'єктів. У результаті яскраві області відповідають краям об'єктів, а фон стає темним.

$$I_{\text{gradient}} = I \oplus K - I \ominus K.$$

Топ-хет трансформація виділяє об'єкти, розмір яких менший за структурний елемент K , і уможлиблює виявляти локальні яскраві області на неоднорідному фоні. Існує два типи топ-хет трансформації: білий і чорний. Для білого топ-хета трансформація обчислюється як різниця між вихідним зображенням I і результатом розкриття:

$$I_{\text{white_top-hat}} = I - (I \ominus K) \oplus K.$$

Цей підхід виділяє яскраві деталі, менші за структурний елемент. У свою чергу, для чорного топ-хета трансформація визначається як різниця між результатом закриття і вихідним зображенням, і дана операція виділяє темні деталі на яскравому фоні :

$$I_{\text{black_top-hat}} = (I \oplus K) \ominus K - I.$$

2.3.3. Постановка задачі сегментації медичних зображень

Нехай набір даних МРТ зображень представлений множиною зображень $\{X_i\}$, де кожне $X_i \in \mathbb{R}^{h \times w \times d}$ позначає зріз МРТ з висотою h , шириною w та d рівнями інтенсивності. Кожному зрізу зображення відповідає сегментаційна маска-розмітка $Y_i \in \{0, 1\}^{h \times w}$, де значення кожного пікселя вказує, чи належить він до області пухлини чи до фону. Задачу сегментації можна сформулювати як функцію $f_\theta : \mathbb{R}^{h \times w \times d} \rightarrow \{0, 1\}^{h \times w}$, параметризовану θ , яка відображає зріз зображення МРТ X_i у передбачувану сегментаційну маску $\hat{Y}_i : \hat{Y}_i = f_\theta(X_i)$. Метою є навчити f_θ таким чином, щоб \hat{Y}_i максимально точно відповідало Y_i , мінімізуючи розбіжності між передбачуваними та розміченими масками на всьому наборі даних.

2.3.4. Огляд існуючих підходів до побудови 3D моделі із 2D зрізів

Мультипланарна реконструкція (MPR) – метод, який уможливорює створювати тривимірні моделі на основі двовимірних зрізів, отриманих за допомогою таких технологій, як МРТ або КТ. Основний принцип MPR полягає у відновленні тривимірного об'єкта шляхом поєднання двовимірних зрізів у просторі вздовж певної осі. У разі великої відстані між зрізами застосовується інтерполяція, яка забезпечує плавність моделі. Перевагами цього підходу є простота реалізації, низькі обчислювальні витрати та можливість швидкої візуалізації. Однак якість моделі залежить від щільності та якості вихідних зрізів.

Воксельне представлення базується на перетворенні кожного пікселя сегментаційної маски в тривимірний елемент – воксель. Це уможливорює створити об'ємну сітку, в якій кожен воксель відповідає певній інтенсивності або приналежності до області. Такий підхід забезпечує високу точність і деталізацію моделі, даючи змогу виконувати аналіз об'єму, форми та

структури. Водночас він вимагає значних обчислювальних ресурсів і великого обсягу пам'яті, що ускладнює його реалізацію для моделей з високою роздільною здатністю.

2.3.5. Постановка задачі утворення 3D моделі із зображень пухлин

Нехай набір МРТ-зображень, що представляють різні зрізи в одній площині, позначається як $\{X_i\}$, де кожне $X_i \in \mathbb{R}^{h \times w \times d}$ – це зріз МРТ із висотою h , шириною w та d рівнями інтенсивності. Для кожного зрізу передбачена сегментаційна маска пухлини $\hat{Y}_i \in \{0, 1\}^{h \times w}$, де значення кожного пікселя позначає приналежність до області пухлини (1 – пухлина, 0 – фон). Метою є створення тривимірної моделі пухлини шляхом послідовного об'єднання передбачених сегментаційних масок $\{\hat{Y}_i\}$ у тривимірному просторі. У такому випадку, можливо визначити 3D модель як функцію:

$$\mathcal{T} = \bigcup_{i=1}^N \{(x, y, z) \mid (x, y) \in \hat{Y}_i \text{ and } z = z_i\},$$

де $\mathcal{T} \subset \mathbb{R}^3$ – тривимірна область, що описує форму органу, Z_i – координата зрізу X_i у просторі.

Алгоритм побудови 3D моделі:

1. Для кожного зрізу X_i , визначити передбачену маску \hat{Y}_i .
2. Розмістити сегментаційні маски \hat{Y}_i у відповідних координатах z_i у тривимірному просторі.
3. Об'єднати отримані площини у суцільну тривимірну структуру.

У даному розділі було проведено дослідження задачі сегментації та побудови 3D моделі на основі 2D знімків.

Було розглянуто проблему сегментації зображень та її особливості. Ця проблема вирішується різними способами. Традиційні методи сегментації зображень включають методи на основі інтенсивності пікселів (метод порогової сегментації чи контурний аналіз) або методи на основі

морфологічних операцій (ерозія, дилатація, відкриття та закриття).

Для подальшої роботи було поставлено формальну задачу до сегментації пухлин головного мозку.

Також було розглянуто методи побудови 3D моделі із 2D знімків. Мультипланарна реконструкція є більш простим методом, що має меншу точність. Побудова моделі із вокселі потребує більше обчислювальних ресурсів та має вищу складність, але дозволяє створювати якісніші моделі.

2.4. Відкриті набори даних сегментованих МРТ-зображень хребта

Як вже зазначалося раніше, на даний момент часу більшість наборів медичних даних не знаходяться у відкритому доступі. Однак в результаті пошуку наборів даних з сегментованими МРТ-зображеннями хребта вдалося знайти відкриті набори [10-12]:

✓ в статті [10] розглядається відкритий набір сегментованих вручну МРТ-зображень хребта 34 пацієнтів у форматі NIfTI. В наборі присутня інформація про клінічні показання, які розділені на п'ять

✓ класів: 1 – післяопераційна/контрольна візуалізація пухлини після резекції, 2 – біль у попереку із або без радикулопатії внаслідок дегенеративних змін, 3 – злоякісне новоутворення з (підозрюваними) метастазами, 4 – спондилодисцит або інше запалення/інфекція, 5 – травма/перелом. Для роботи з форматом NIfTI пропонуються Python бібліотеки NiBabel та SimpleITK;

✓ в статті [11] розглядається відкритий набір даних МРТ-зображень хребта 23 пацієнтів. Формат – NifTI. Розмір зображень: $39 \times 305 \times 305$, розмір вокселів: $2 \times 1.25 \times 1.25$ мм³;

✓ в роботі [12] розглядається великий відкритий набір даних МРТ-зображень хребта 218 пацієнтів, формат даних – MHA. В наборі також присутні радіологічні оцінки для кожного міжхребцевого диску: наявність змін хребців за класифікацією Modic (типи I, II та III); наявність змін верхньої та нижньої

замикальних пластинок/грижи Шморля, спондилолітезу, грижи, звуження та випинання диска; наявність дегенеративних змін за класифікацією Pfirrmann (ступені від 1 до 5). Ці оцінки можуть бути використані для визначення ступені тяжкості дегенеративних змін кожного міжхребцевого диску. Дані зібрані з чотирьох різних лікарень, на всіх зображеннях тіла хребців, міжхребцевих дисків та хребетний канал були сегментовані ітеративним напівавтоматичним методом, при якому початкова сегментація генерувалася автоматично, а потім перевірялася та коригувалася вручну.

Було вирішено обрати набір даних [12], який є найбільшим серед знайдених і при цьому має радіологічні оцінки. Цей набір складається з 447 3D МРТ-зображень хребта в форматі МНА, однак для двовимірної сегментації були зроблені 2D зрізи, які використовувалися для навчання та перевірки моделей. На рисунку 1.1 можна побачити приклад зрізу МРТ-зображення з цього набору разом з виділеними масками, де позначені тіла хребців, міжхребцевих дисків та хребетний канал. В цьому наборі даних усі тіла промарковані окремо, що дає змогу застосовувати набір для класифікації кожного тіла хребців та міжхребцевого диску.

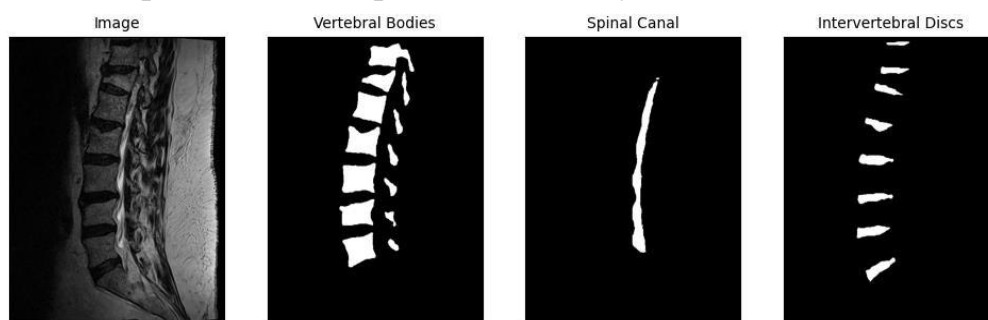


Рисунок 2.1 – Приклад декількох зрізів МРТ-зображення з набору [12]

В роботі [12] цей набір пропонується розділяти на дві вибірки. Розподіл радіологічних оцінок для такого розділення можна побачити в таблиці 2.1.

Таблиця 2.1 – Розподіл радіологічних оцінок в наборі [12]

Класифікація	Тренувальна вибірка		Валідаційна вибірка	
	Міжхребцеві диски (1240)	Пацієнти (179)	Міжхребцеві диски (280)	Пацієнти (39)
Modic I	4 (0.3%)	3 (1.7%)	0 (0.0%)	0 (0.0%)
Modic II	376 (30.3%)	115 (64.2%)	127 (45.4%)	31 (79.5%)
Modic III	6 (0.5%)	5 (2.8%)	1 (0.4%)	1 (2.6%)
Зміни верхньої замикальної пластинки	472 (38.1%)	133 (74.3%)	142 (50.7%)	34 (87.2%)
Зміни нижньої замикальної пластинки	477 (38.5%)	138 (77.1%)	146 (52.1%)	33 (84.6%)
Спондилолітез	33 (2.7%)	30 (16.8%)	9 (3.2%)	9 (23.1%)
Грижа диска	62 (5.0%)	55 (30.7%)	10 (3.6%)	10 (25.6%)
Звуження диска	440 (35.5%)	158 (88.3%)	104 (37.1%)	35 (89.7%)
Випинання диска	592 (47.5%)	163 (91.1%)	154 (55.0%)	37 (94.9%)
Pfirmann 1	241 (19.4%)	66 (36.9%)	45 (16.1%)	16 (41.0%)
Pfirmann 2	260 (21.0%)	106 (59.2%)	81 (28.9%)	14 (35.9%)
Pfirmann 3	348 (28.1%)	133 (74.3%)	70 (25.0%)	30 (76.9%)
Pfirmann 4	240 (19.4%)	92 (51.4%)	51 (18.2%)	25 (64.1%)
Pfirmann 5	151 (12.2%)	79 (44.1%)	51 (18.2%)	16 (41.0%)

2.5. Опис обраних архітектур, методу та медичної системи

Як вже згадувалося в попередньому розділі, для створення моделі для сегментації МРТ-зображень хребта було вирішено обрати архітектуру U-Net, модифікувати її, та застосувати одну з варіацій методу напівкерованого навчання Mean Teacher, що розглядалася в дослідженні [9]. Для діагностування деформації міжхребцевих дисків було вирішено обрати архітектуру ResNet. Розглянемо більш детально ці архітектури, метод напівкерованого навчання та блоки, які застосовуються для створення гібридних архітектур на основі U-Net.

2.4.1. Архітектура U-Net

U-Net [13] є однією з найбільш відомих архітектур для сегментації

медичних зображень. Ця архітектура є повністю згортковою і складається з кодувальника (encoder) та декодувальника (decoder). Кодувальник виділяє ознаки шляхом застосування шарів згортки та Max Pooling, зменшуючи роздільну здатність, а декодувальник збільшує роздільну здатність виділених ознак. Також в U-Net застосовуються пропускі з'єднання (skip-connections) на відповідних рівнях роздільної здатності, що сприяє збереженню просторової інформації. В якості функції активації пропонується використовувати ReLU. Оскільки в U-Net замість повнозв'язних шарів використовуються згорткові шари, то це зменшує кількість параметрів.

U-Net є найпоширенішою архітектурою сегментації зображень завдяки своїй гнучкості та оптимізованому модульному дизайну [14]. З роками U-Net отримала значну увагу з боку дослідників, що призвело до появи численних гібридних рішень. Введення нових блоків або інших архітектурних модифікацій сприяє вирішенню специфічних проблем, з якими стикаються в біомедичних задачах, і підвищенню ефективності сегментації. Найпростішою модифікацією U-Net є додавання шарів нормалізації пакетів (batch normalization) після кожного згорткового шару. Саме такий підхід пропонувався в дослідженні [15]. Цей підхід показав непогані результати при обробці МРТ-зображень і навчанні на наборах даних невеликих розмірів. Схематичне зображення архітектури U-Net можна побачити на рисунку 2.1, де видно, що архітектура має форму латинської літери «U».

2.4.2. Варіація методу Mean Teacher

В класичному методі напівкерованого навчання Mean Teacher використовуються дві копії однієї нейронної мережі: вчитель та учень [16]. Мережа-вчитель генерує псевдомітки для немаркованих даних, а мережа-учень навчається на маркованих та псевдомаркованих даних. На кожному кроці виконується навчання мережі-учня на аугментованих даних. Після того, як було виконано оновлення вагових коефіцієнтів мережі-учня, вагові коефіцієнти мережі-вчителя оновлюються як експоненціальне ковзне середнє

(EMA) вагових коефіцієнтів учня.

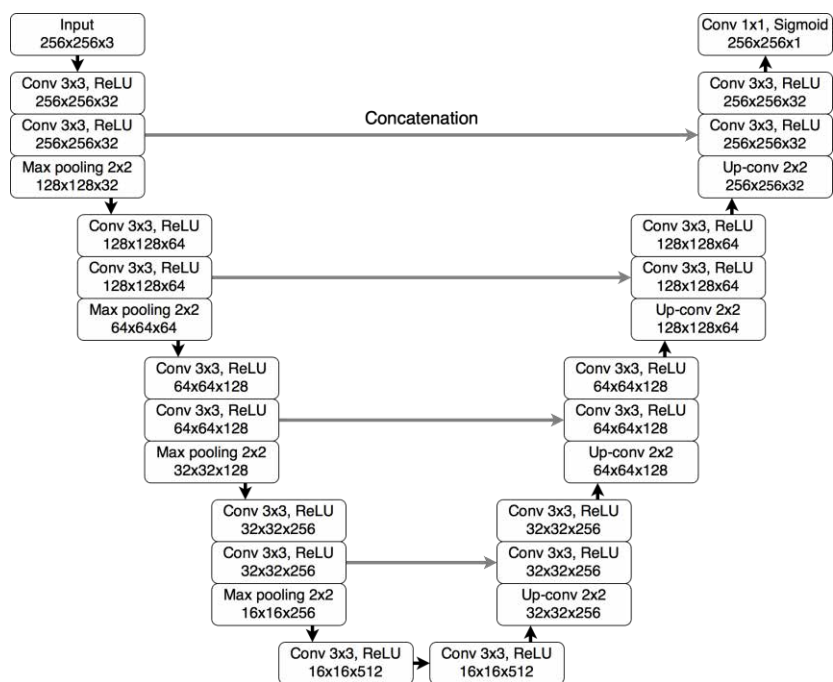


Рисунок 2.1 – Архітектура U-Net [15]

Існує варіація цього методу з використанням двонаправленого копіювання та вставки (BCP) [9]. Вона полягає в генерації вхідних даних для мережі-студента шляхом змішування маркованих та псевдомаркованих частин зображень. Схематичне зображення варіації методу Mean Teacher з використанням двонаправленого копіювання та вставки можна побачити на рисунку 2.2. Отримані змішані зображення (X^{out} , X^{in}) та змішані мітки (Y^{out} , Y^{in}) використовуються для навчання мережі-учня. При цьому застосовується модифікована функція втрат: значення loss окремо обчислюється для маркованої та псевдомаркованої частини, а результуюче значення функції втрат є сумою помножених на відповідні коефіцієнти значень, де коефіцієнт для псевдомаркованої частини менше за коефіцієнт для маркованої частини. Після оновлення вагових коефіцієнтів мережі-учня, вагові коефіцієнти мережі- вчителя оновлюються як експоненціальне ковзне середнє (EMA) вагових коефіцієнтів учня.

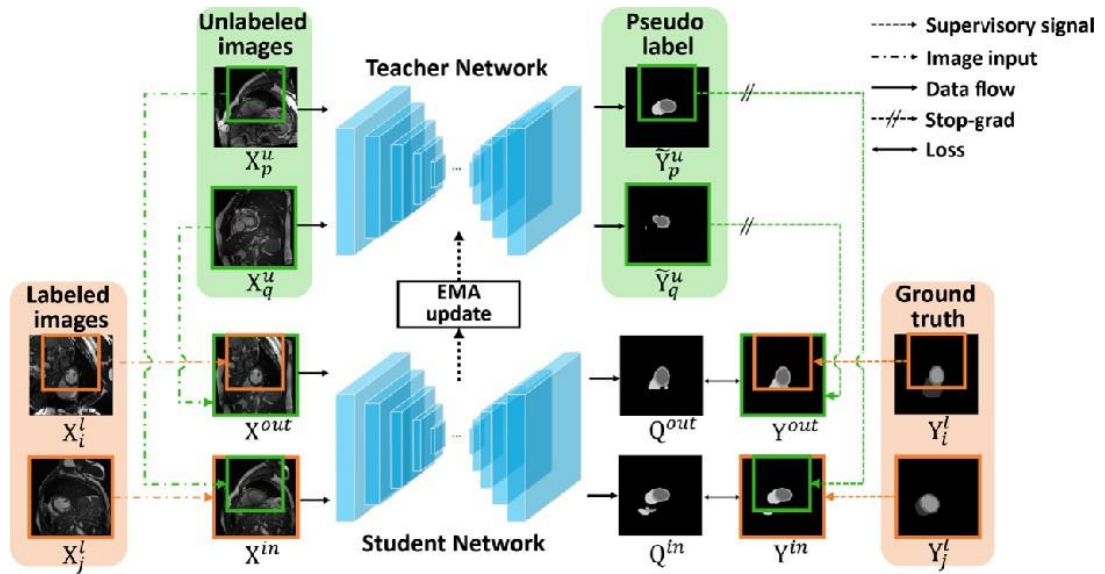


Рисунок 2.2 – Двонаправлене копіювання та вставка в Mean Teacher [9]

У роботі [9] зазначається, що для підвищення точності моделі замість використання мереж з випадковою ініціалізацією ваг, рекомендується виконати попереднє навчання мережі-вчителя на маркованому наборі даних. При цьому дані повинні бути аугментовані шляхом копіювання та вставки сегментів.

2.4.3. Гібридні рішення для U-Net

Гібридні архітектури нейронних мереж є комбінованими архітектурами, які поєднують кілька парадигм або структур (блоків) для досягнення кращих результатів у певних задачах. Вони об'єднують переваги різних архітектур, що дає змогу обійти недоліки однієї структури завдяки перевагам іншої, і в результаті отримати вищу точність. Для покращення U-Net часто додають або замінюють певні блоки цієї архітектури. В U-Net виділяють такі складові, до яких застосовують заміну та додавання блоків: кодувальник, декодувальник, bottleneck та пропускні з'єднання.

DENSE-INception U-net. В роботі [17] пропонується використання

Inception-Res блоку в кодувальнику та декодувальнику, Dense-Inception блоку в bottleneck, а також блоків зменшення та збільшення роздільної здатності. При цьому Dense-Inception блок складається з декількох Inception-Res блоків, наприклад, з чотирьох. Кожен з блоків містить пропускні з'єднання та шари нормалізації пакетів. Загальну архітектуру Inception-Res блоку можна побачити на рисунку 2.3. Серед недоліків DENSE-INception U-net є те, що велика кількість блоків Dense-Inception може призвести до суттєвого збільшення кількості параметрів, що ускладнить і уповільнить навчання моделі, однак при цьому може не забезпечити кращих результатів.

FusionU-Net. Відповідно до дослідження [18], між кодувальником і декодувальником існує семантичний розрив, який може негативно вплинути на продуктивність моделі при використанні класичних пропускних з'єднань в U-Net. Для вирішення цієї проблеми можна виконати модифікацію пропускних з'єднань шляхом додавання модуля злиття (fusion module) для зменшення семантичних розривів.

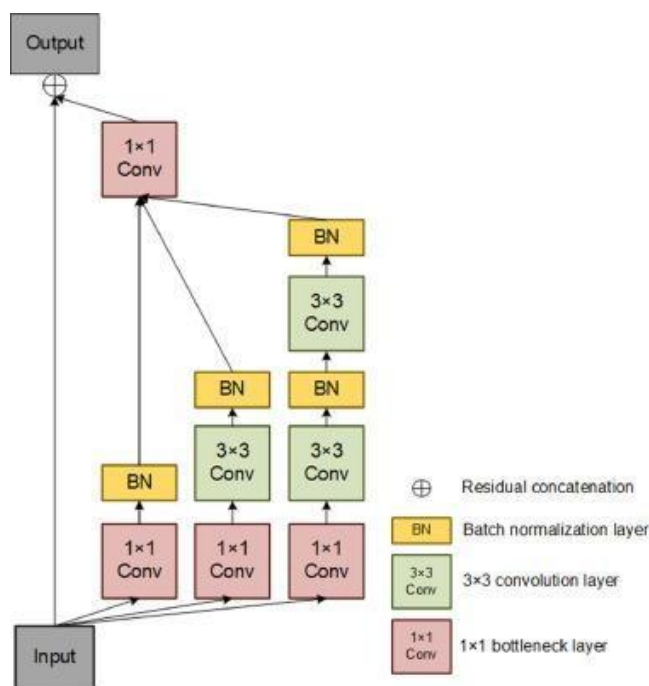


Рисунок 2.3 – Архітектура Inception-Res блоку [17]

В роботі [18] пропонується двораундовий модуль злиття, який враховує локальну релевантність між суміжними виходами рівня кодувальника та потребу в двонаправленому обміні інформацією між кількома рівнями. Цей модуль може складатися з декількох блоків злиття (FuseBlock), кожен з яких складається зі спеціалізованих блоків зменшення та збільшення роздільної здатності DownFuse та UpFuse. Архітектуру цього блоку злиття можна побачити на рисунку 2.4.

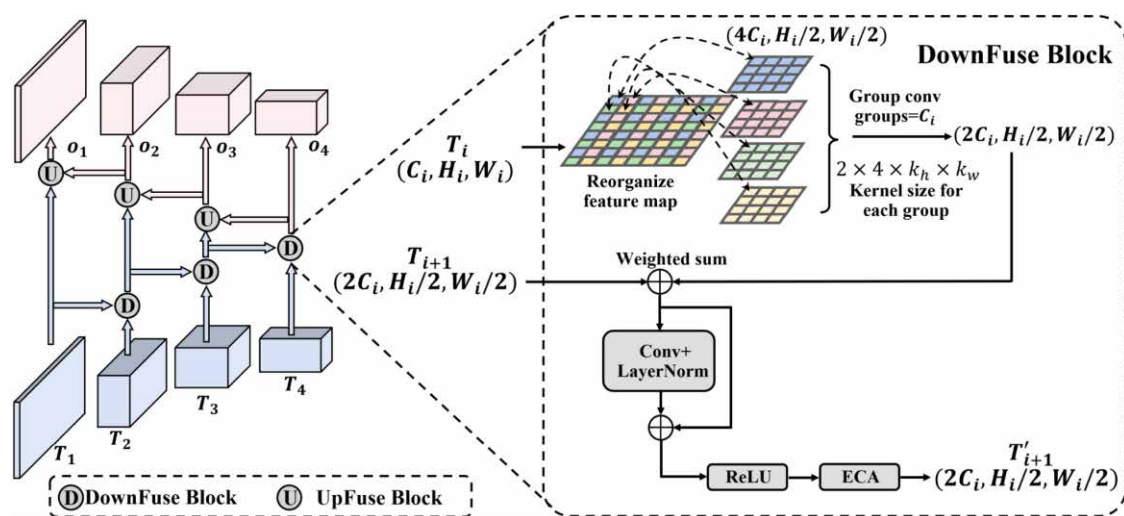


Рисунок 2.4 – Архітектура блоку FuseBlock [18]

HDA-ResUNet. Модифікація пропускних з'єднань за допомогою додавання блоків уваги пропонується в роботі [19]. В *HDA-ResUNet* на пропускних з'єднаннях використовуються блоки уваги каналу (channel attention (CA) block), а в якості bottleneck – гібридний згортковий шар розширеної уваги (hybrid dilated attention convolutional (HDAC) layer). Блок уваги каналу допомагає більше зосередитися на каналах, які містять достатню кількість інформації, одночасно обмежуючи вплив несуттєвих каналів. Архітектуру цього блоку можна побачити на рисунку 2.5.

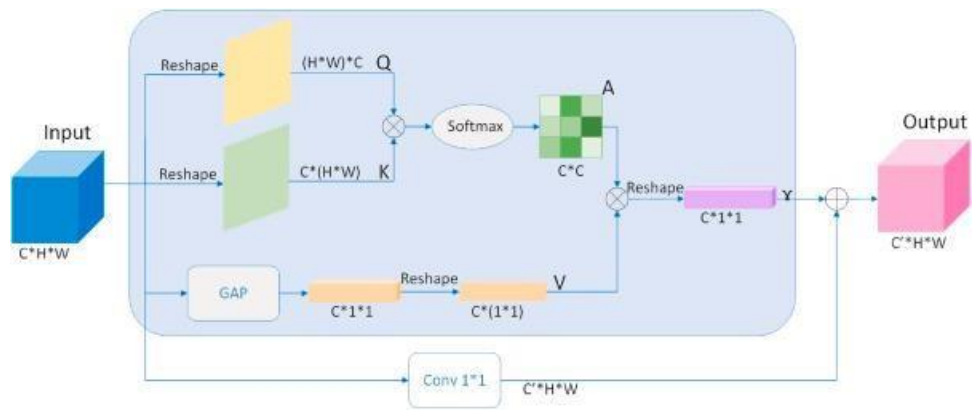


Рисунок 2.5 – Архітектура СА-блоку [19]

Гібридний згортковий шар розширеної уваги використовується в HDA-ResUNet для вилучення об'єктів різних масштабів і контекстної інформації. Архітектура цього шару продемонстрована на рисунку 2.6. Як видно, СА-блок також є частиною HDAC-шару.

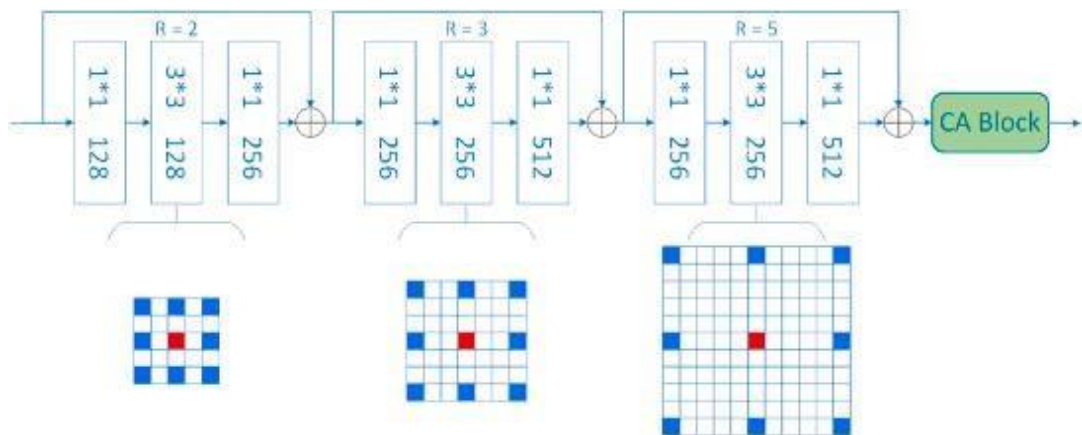


Рисунок 2.6 – Архітектура HDAC-шару [19]

ResUNet++. В роботі [20] для покращення архітектури пропонується використання залишкових блоків (Residual blocks, Res), блоків стиснення та збудження (Squeeze and Excitation, SE), розширеного просторового пірамідального об'єднання (Atrous Spatial Pyramidal Pooling, ASPP) і блоків уваги (Attention). Зі схематичного зображення архітектури ResUNet++ на рисунку 2.7 видно, що в кодувальнику використовуються блоки Res та SE, в bottleneck – ASPP, в декодувальнику – блоки Res та ASPP, також наявні Attention-блоки, які модифікують пропускні з'єднання. Зазначається, що ця

архітектура краще працює у випадку невеликої кількості зображень, ніж класична U-Net.

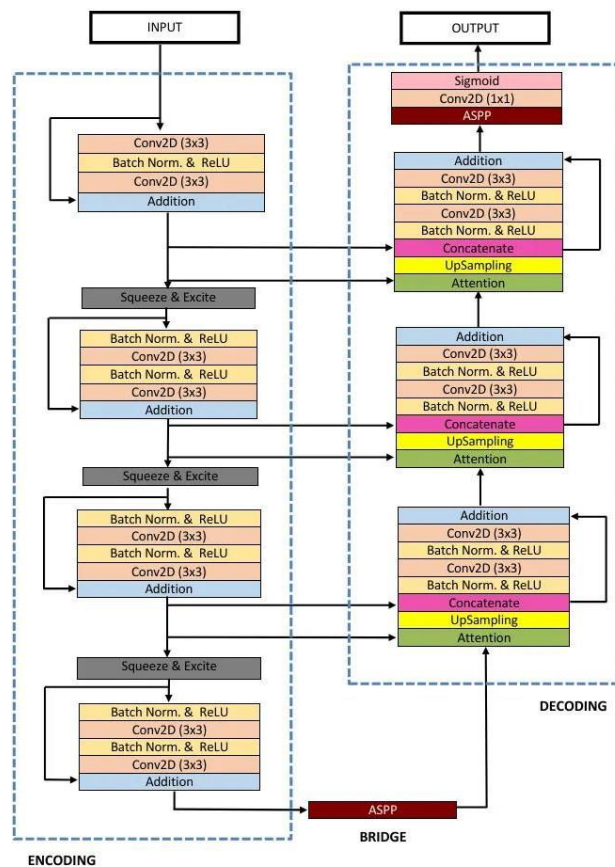


Рисунок 2.7 – Архітектура ResUNet++ [20]

MS-TransUNet++. Існують публікації, де архітектура U-Net модифікується шляхом введення трансформерів. Наприклад, в дослідженні [21] пропонується використання блоку ефективного трансформера в якості bottleneck. Архітектуру цього блоку можна побачити на рисунку 2.8. Також в *MS-TransUNet++* пропонується модифікація пропускових з'єднань, що реалізує гнучку схему об'єднання ознак між кодувальником і декодувальником на різних рівнях.

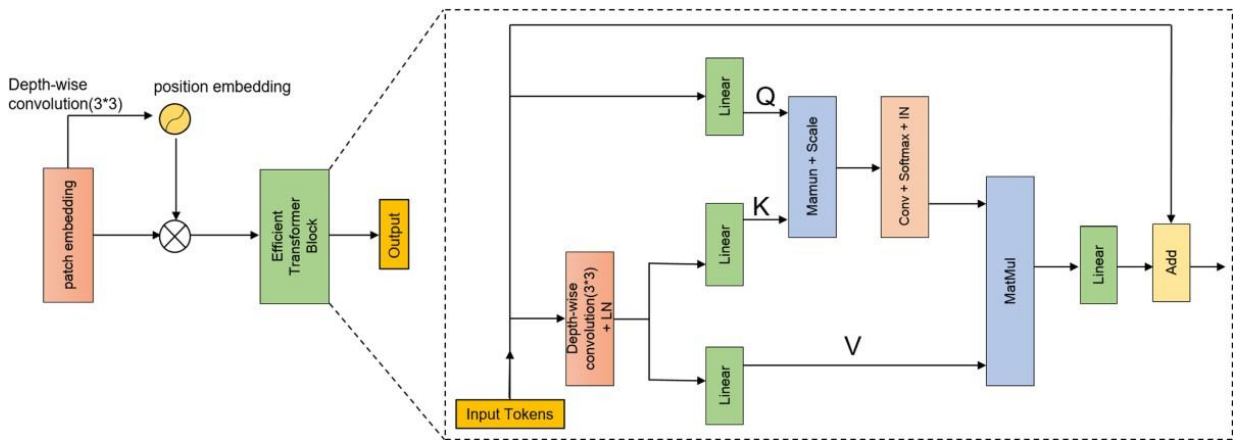


Рисунок 2.8 – Архітектура ефективного трансформера [21]

2.4.4. Гібридна архітектура BAFUNet

На основі розглянутих гібридних рішень було вирішено розробити власну гібридну архітектуру для сегментації зображень BAFUNet (Bottleneck ASPP & Fusion U-Net). В якості основи було вирішено взяти чотирирівневу U-Net з шарами нормалізації пакетів після кожного згорткового шару [15], архітектура якої зображена на рисунку 2.1. Далі bottleneck цієї архітектури було модифіковано шляхом заміни на блок ASPP, який застосовувався в ResUNet++ з дослідження [20]. Для зменшення семантичних розривів було виконано модифікацію пропускних з'єднань шляхом додавання двораундового модуля злиття, який використовувався у FusionU-Net з роботи [18], з двома блоками злиття FuseBlock. В результаті була отримана гібридна архітектура, яка зображена на рисунку 2.9. Планувалося, що така архітектура зможе успішно вилучати об'єкти різних масштабів і об'єкти зі складними або нечіткими межами, а також буде здатна працювати на малих наборах даних.

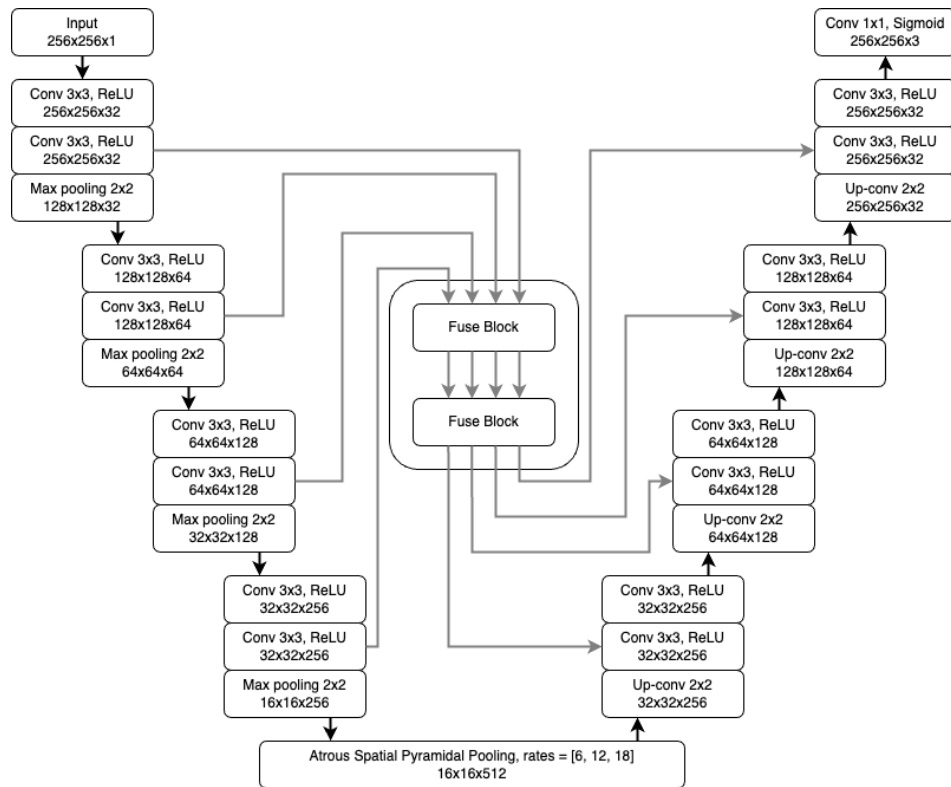


Рисунок 2.9 – Архітектура BAFUNet

2.4.5. Архітектура ResNet

Як виявилось при аналізі публікацій, різновиди архітектури ResNet демонструють гарні результати в задачах діагностування на медичних зображеннях [3, 5]. ResNet є глибокими нейронними мережами, які використовують залишкові блоки з пропусковими з'єднаннями [22], побудову яких зображено на рисунку 2.10. Класичний залишковий блок складається з двох або трьох згорткових шарів, які реалізують залишкову функцію F , та пропускового з'єднання, в якості функції активації використовується ReLU. Таким чином, в результаті застосування блоку до входу x на виході блоку буде отримано значення $y = F(x) + x$. Використання цих блоків дає змогу тренувати мережі з великою кількістю шарів, оскільки застосування

пропускних з'єднань допомагає уникати проблеми затухаючого градієнта. Це знижує ризик перенавчання і деградації точності в глибоких мережах, що спрощує навчання ResNet.

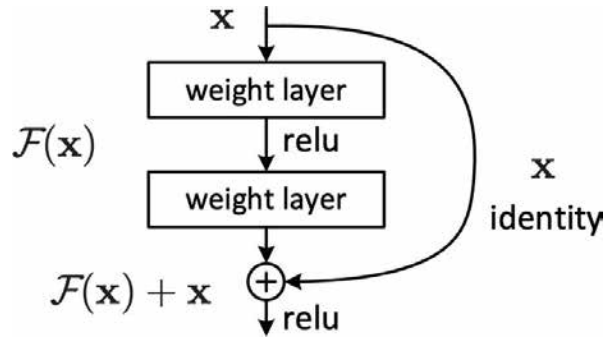


Рисунок 2.10 – Залишковий блок з пропускними з'єднаннями [22]

Сімейство ResNet включає базові моделі ResNet-18, ResNet-34, ResNet-50, ResNet-101 та ResNet-152. Для задач медичного діагностування, наприклад, в роботі [23], використовують архітектуру ResNet-18, яка зображена на рисунку

2.11. ResNet-18 є однією з найменших базових моделей сімейства ResNet, що містить лише 18 шарів із параметрами. ResNet-18 містить 17 згорткових шарів та 1 повнозв'язний шар, при цьому 16 з 17 згорткових шарів об'єднані в 4 групи залишкових блоків. ResNet-18 є популярною для навчання з нуля на невеликих наборах даних завдяки своїй простоті та швидкості. При цьому вона демонструє високу ефективність у завданнях класифікації.

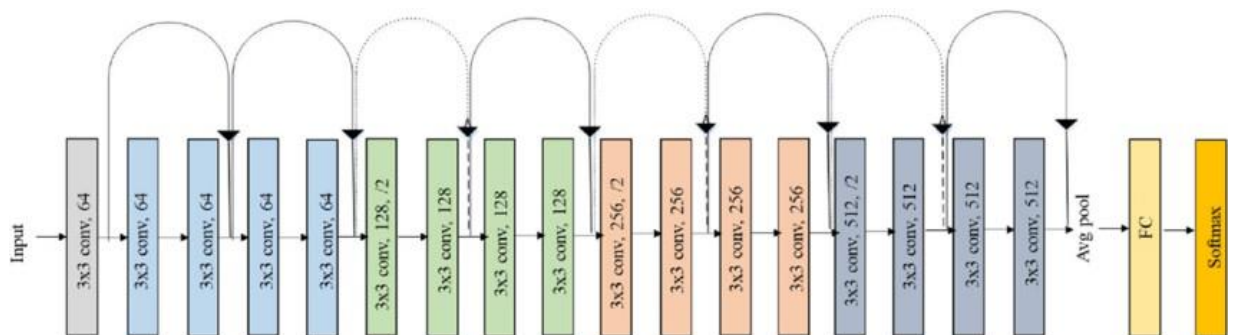


Рисунок 2.11 – Архітектура ResNet-18 [23]

2.6. Інтелектуальна медична система діагностики

Використання сучасних архітектур глибокого навчання, таких як U-Net для сегментації та ResNet для класифікації, робить можливим створення комплексних рішень для аналізу медичних зображень. На основі цих моделей було вирішено розробити медичну систему діагностики деформації міжхребцевих дисків у вигляді двох модулів: модуля сегментації та модуля діагностування. Модульна структура інтелектуальної медичної системи, яка зображена на рисунку 2.12, дає змогу розширювати функціональність за рахунок додавання нових модулів діагностування. При цьому потреба в перенавчанні мережі для сегментації може виникнути лише у разі зміни типу вхідних зображень, наприклад, при використанні КТ- та рентгенівських знімків.

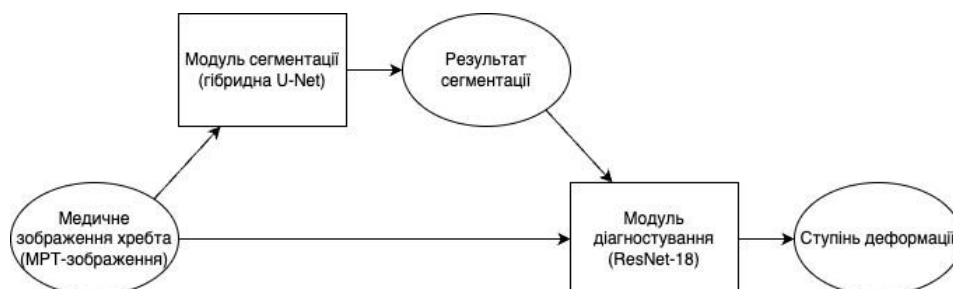


Рисунок 2.12 – Передача даних між модулями медичної системи діагностики

В інтелектуальній медичній системі діагностики деформації міжхребцевих дисків модуль діагностування застосовується після того, як відпрацює модуль сегментації: на вхід модуля діагностування подається не тільки початкове зображення, але й власне результат сегментації. Внутрішня реалізація модуля і тип вихідних даних можуть відрізнятися в залежності від потреб. Наприклад, в системі з модулем діагностування деформації міжхребцевих дисків на виході буде декілька значень (по одному класу деформації для кожного виявленого міжхребцевого диска). Отже, на вхід

ТАКОГО

модуля подаються 3 канали: початкове МРТ-зображення, а також два канали від модуля сегментації (тіла хребців та міжхребцеві диски), додатково перед застосуванням ResNet-18 виконується виділення регіонів інтересу, кожен з яких містить один диск. Далі ці регіони подаються на вхід нейронної мережі ResNet- 18, що виконує класифікацію деформації кожного міжхребцевого диску окремо.

Висновки

В даному розділі було виконано огляд архітектур нейронних мереж для сегментації МРТ-зображень, методів напівкерowanego навчання, які використовуються при сегментації медичних зображень, та відкритих наборів даних сегментованих МРТ-зображень хребта. Виявилось, що архітектура U-Net здатна демонструвати високу точність сегментації та при цьому є відносно простою для реалізації та модифікації. Було вирішено при розробці власної моделі для сегментації 2D МРТ-зображень хребта в якості основи узяти архітектуру U-Net, після чого модифікувати її. Також було вирішено застосувати метод напівкерowanego навчання Mean Teacher з двонаправленим копіюванням та вставкою, оскільки він допомагає запобігти перенавчанню на малих вибірках. В якості набору даних було вирішено взяти відкритий набір з 447 3D МРТ-зображень хребта, після чого зробити 2D зрізи і виконати навчання моделей на отриманому наборі 2D МРТ-зображень.

Обраний набір даних містив інформацію про радіологічні оцінки, що забезпечує можливість використання цього набору для діагностики деформації міжхребцевих дисків. Таким чином, було вирішено розробити прототип медичної системи, яка виконує сегментацію 2D МРТ-зображень хребта за допомогою моделі модифікованої архітектури U-Net, навченої з використанням напівкерowanego навчання, та виконує діагностику деформації міжхребцевих дисків на основі результатів сегментації.

Розглянуто архітектури нейронних мереж: U-Net, її модифікацій та ResNet, класичний метод напівкерowanego навчання Mean Teacher та його

модифікацію, а також запропоновано модульну медичну систему для діагностики деформації міжхребцевих дисків на основі сегментації МРТ-зображень. Розглянуто архітектуру U-Net та її складові. Порівняно класичний метод напівкерowanego навчання Mean Teacher з його модифікацією на основі двонаправленого копіювання та вставкою. Розглянуто деякі гібридні архітектури U-Net, виявлено, що U-Net модифікують за допомогою додавання та заміни блоків в кодувальнику, декодувальнику, bottleneck та/або пропускних з'єднаннях цієї архітектури. Запропонована власна гібридна архітектура для сегментації зображень BAFUNet. Також було розглянуто архітектуру ResNet та можливість застосування ResNet-18 для задач медичного діагностування.

В результаті було запропоновано модульну інтелектуальну медичну систему діагностики деформації міжхребцевих дисків по МРТ-зображеннях хребта, що складається з модуля сегментації та модуля діагностування. Для модуля сегментації МРТ-зображень хребта було вирішено розробити власну гібридну мережу на основі архітектури U-Net, після чого провести її навчання з використанням методу напівкерowanego навчання Mean Teacher з двонаправленим копіюванням та вставкою. Для модуля діагностування деформації міжхребцевих дисків було вирішено застосувати мережу ResNet-18.

РОЗДІЛ 3

ОПИС ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1. Використані технології та бібліотеки

При розробці програмного забезпечення було вирішено використати інтерактивне середовище Jupyter Notebook [24]. Це середовище надає можливість створення комірок, в яких може розміщуватися код на мові програмування Python [25]. Код в комірках можна запускати, а результат роботи буде виведено під коміркою. Приклад вигляду комірки можна побачити на рисунку 3.1. Детальний опис предметного середовища наведено в додатку А.

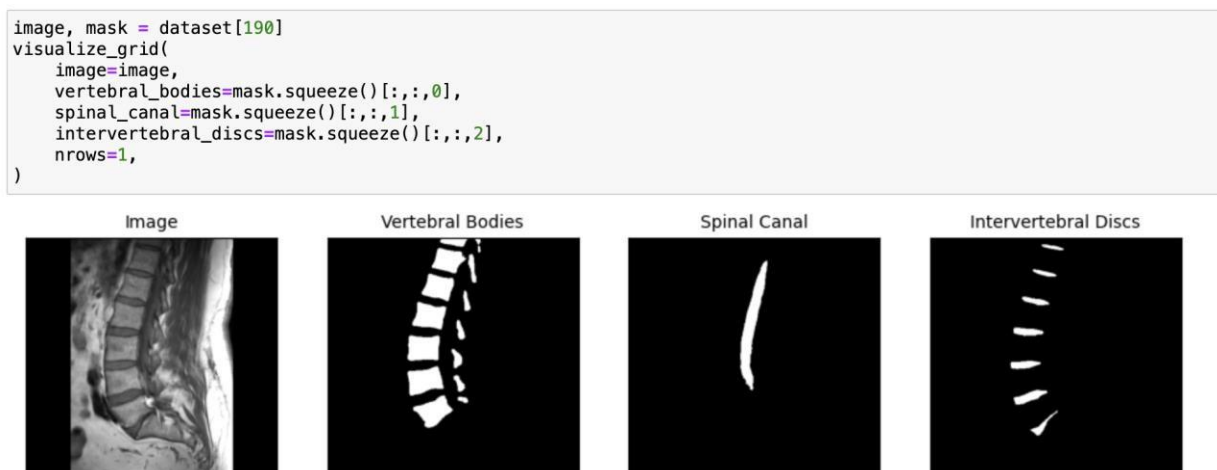


Рисунок 3.1 – Комірка Jupyter Notebook з результатом виконання

Для реалізації моделі була використана бібліотека для машинного навчання PyTorch [26] з пакетом torchvision, які часто застосовуються в задачах комп'ютерного зору. Для попередньої обробки даних використовувалися популярні бібліотеки для наукових обчислень NumPy [27], Pandas [28] та scikit-learn [29]. Бібліотека NumPy забезпечує підтримку багатовимірних масивів і матриць, а також надає велику колекцію математичних функцій для роботи з ними. Pandas є бібліотекою для обробки та аналізу даних, побудованою поверх NumPy. Pandas надає спеціальні структури даних та операції для маніпулювання числовими таблицями,

що знадобилося в цій роботі. Бібліотека `scikit-learn` також може працювати з масивами `NumPy`, а ще містить різні алгоритми класифікації, регресії та кластеризації, інструменти для попередньої обробки, вибору моделі та оцінки.

Графіки будувалися за допомогою бібліотеки `Matplotlib` [30]. Ця бібліотека часто використовується для візуалізації даних двовимірною та тривимірною графікою. Отримані за допомогою неї графіки можна побачити в розділі 4. Для роботи із 2D зображеннями використовувалася бібліотека `OpenCV` [31]. Ця бібліотека комп'ютерного зору призначена для аналізу, класифікації та обробки зображень. Для вивантаження і обробки МРТ-зображень використовувалася бібліотека `Tiger` [32]. Ця бібліотека обробки медичних зображень створена Групою Діагностичного Аналізу Зображень, яка є частиною Медичного центру університету Радбауд. `Tiger` забезпечує роботу з МРТ-зображеннями в форматі МНА, виконуючи їхню попередню обробку, засновуючись на метаданих файлів.

Для аугментації 2D зображень застосовувалася бібліотека `Albumentations` [33] та пакет `torchvision`, який входить в `PyTorch`. Бібліотека `Albumentations` може бути використана для налаштування трансформацій, таких як віддзеркалення, зміщення, поворот, зміна розміру, обрізання частини зображення, зміна яскравості, контрастності, додавання розмиття, шумів тощо. За допомогою `Albumentations` на кожному кроці навчання нейронних мереж використовувалися зображення із різною аугментацією з метою запобігання перенавчання. Додатково також застосовувалася бібліотека `Segmentation Models` [34], яка містить найбільш поширені архітектури для сегментації зображень і корисні функції для їхнього навчання, обчислення функцій втрат та метрик.

3.2. Поетапний опис роботи програми

Повний вихідний код розробленої програми можна побачити в додатку Б. Увесь код можна умовно поділити на декілька частин, які відповідають таким етапам роботи програми:

3.2.1. завантаження обраного набору даних 3D МРТ-зображень та інформації про набір за допомогою бібліотек Pandas та Tiger;

3.2.2. підготовка даних для сегментації (формування 2D зрізів, виділення трьох класів з масок: хребців, міжхребцевих дисків та хребетного каналу) та збереження 2D зображень за допомогою OpenCV;

3.2.3. підготовка трансформацій для аугментації даних з використанням бібліотеки Albumentations;

3.2.4. розділення набору даних на тренувальний та тестувальний;

3.2.5. створення та навчання різних моделей (U-Net, ResUNet++ та BAFUNet) на повному наборі тренувальних даних з використанням PyTorch та Segmentation Models, побудова графіків через Matplotlib;

3.2.6. розділення тренувального набору даних за допомогою scikit-learn на маркований та немаркований для напівкерованого навчання;

3.2.7. виконання попереднього навчання нової BAFUNet мережі-вчителя на маркованій частині тренувального набору даних;

3.2.8. застосування методу напівкерованого навчання Mean Teacher з двонаправленим копіюванням та вставкою для навчання мереж- учня та вчителя архітектури BAFUNet;

3.2.9. виведення результатів для найкращої моделі;

3.2.10. підготовка даних для класифікації ступенів деформації міжхребцевих дисків (виділення регіонів інтересу, кожен з яких містить міжхребцевий диск) та збереження 2D зображень з використанням OpenCV;

3.2.11. розділення отриманого набору на тренувальний та тестувальний;

3.2.12. підготовка трансформацій для аугментації даних з використанням пакету torchvision;

3.2.13. навчання моделей (ResNet-18) на наборі тренувальних даних для діагностування деформації міжхребцевих дисків за різними класифікаціями а також виведення результатів для кожної моделі.

Отже, розроблена програма виконує навчання декількох моделей. Спочатку створюються мережі різних архітектур та виконується їхнє кероване

навчання (усі дані є розміченими). Моделі порівнюються, обирається найкраща архітектура. В подальшому найкраща модель використовується для порівняння результатів з відповідними результатами при застосуванні напівкерованого навчання. Далі готується новий тренувальний набір даних, в якому більшість даних є немаркованими. Цей набір вже застосовується для напівкерованого навчання. Створюється мережа-вчитель і, відповідно до методу, описаному в підрозділі 2.2, виконується її попереднє навчання на маркованому наборі тренувальних даних. При цьому, оскільки розмір маркованих даних в цьому випадку є значно меншим за розмір відповідного набору для першої мережі, що навчалася шляхом керованого навчання, то мережа-вчитель після попереднього навчання буде демонструвати помітно гірші результати. Далі створюється мережа-учень з ваговими коефіцієнтами, які є точною копією вагових коефіцієнтів мережі-вчителя. Після цього застосовується метод Mean Teacher з двонаправленим копіюванням. В результаті цього отримана мережа, навчена за допомогою напівкерованого навчання, буде демонструвати результати, які повинні бути кращими за початкові результати мережі-вчителя після попереднього навчання, але зазвичай ці результати будуть гіршими, ніж результати від повністю керованої моделі.

Окрім моделей для сегментації також створюються моделі для класифікації. Ці моделі демонструють можливість використання модуля діагностування в інтелектуальній медичній системі діагностики деформації міжхребцевих дисків. Оскільки мережа модуля діагностування працює не з повним МРТ-зображенням, а лише з регіонами інтересу з міжхребцевими дисками, то додатково виконується етап попередньої обробки, на якому виділяються ці регіони. Таким чином, мережі для класифікації ступеней деформації навчаються на наборах невеликих регіонів. В результаті отримуються мережі, які здатні виконувати діагностування деформації міжхребцевих дисків за різними класифікаціями.

Висновки

В третьому розділі описано технології та бібліотеки, які використовувалися при розробці програмного забезпечення прототипу інтелектуальної медичної системи діагностики деформації міжхребцевих дисків. Було розроблено програмне забезпечення на мові програмування Python з використанням бібліотеки для машинного навчання PyTorch та інших допоміжних бібліотек. Реалізовано модуль сегментації та модуль діагностування інтелектуальної медичної системи.

Описано етапи роботи розробленого програмного забезпечення. Програма виконує підготовку даних, навчання моделей та виведення результатів застосування навчених моделей для сегментації 2D МРТ-зображень та діагностування деформації міжхребцевих дисків за різними класифікаціями відповідно до обраних налаштувань. В результаті були отримані навчені моделі, які можуть бути використані в якості модулів інтелектуальної медичної системи. Частина вихідного коду програмного забезпечення також можуть бути використані окремо: наприклад, програмна реалізація архітектури BAFUNet може бути застосована до вирішення задачі сегментації не тільки МРТ-зображень, але й інших зображень при навчанні на відповідному наборі даних.

РОЗДІЛ 4

АНАЛІЗ РЕЗУЛЬТАТІВ ДОСЛІДЖЕНЬ

4.1 Кероване навчання модифікованих U-Net

Початковий набір даних складався з 447 МРТ-зображень: 360 зображень для тренування та 87 зображень для тестування – цей розподіл рекомендувався в роботі [12]. Навчання виконувалося на 40 епохах з використанням оптимізатору Adam, початкова швидкість навчання: 0.001 зі зменшенням в 10 разів кожні 20 епох, розмір пакету (batch size) дорівнював 8. Застосовувалася функція втрат JaccardLoss, що заснована на індексі Жаккара (IoU, перетин над об'єднанням), а після кожної епохи зберігалася модель з найвищим значенням коефіцієнту подібності Дайса (F_1 -міри) на тестувальній вибірці.

Першим кроком було вирішено навчити мережі архітектур U-Net, ResUNet++ та BAFUNet зі 100% маркованих даних з метою обрання найкращої архітектури та подальшого порівняння результатів з найкращою моделлю. Зміну значення функції втрат при повністю керованому навчанні можна побачити на рисунку 4.1.

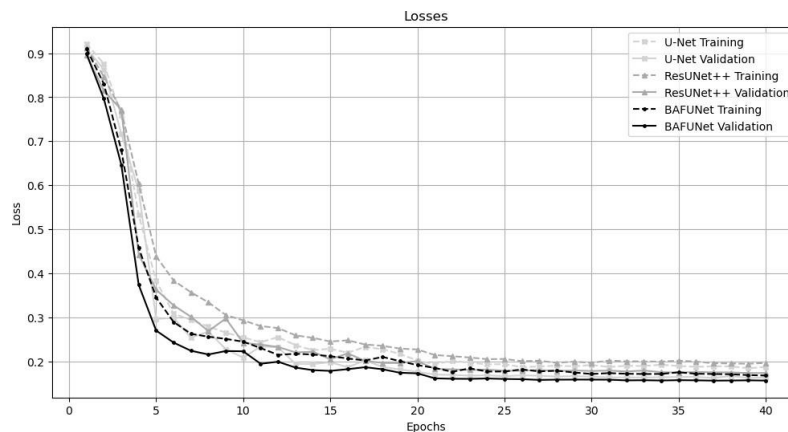


Рисунок 4.1 – Кероване навчання, значення функції втрат Жаккара

На рисунку 4.2 видно залежність коефіцієнту подібності Дайса від епохи. Як можна побачити з рисунків 4.1-4.2, на останніх епохах помітно вихід на

плато для усіх моделей. При цьому архітектура BAFUNet продемонструвала кращі результати за ту ж саму кількість епох: модель цієї архітектури виявилася більш точною – коефіцієнт подібності Дайса приблизно на 0.003 був більше за відповідне значення другої за точністю мережі.

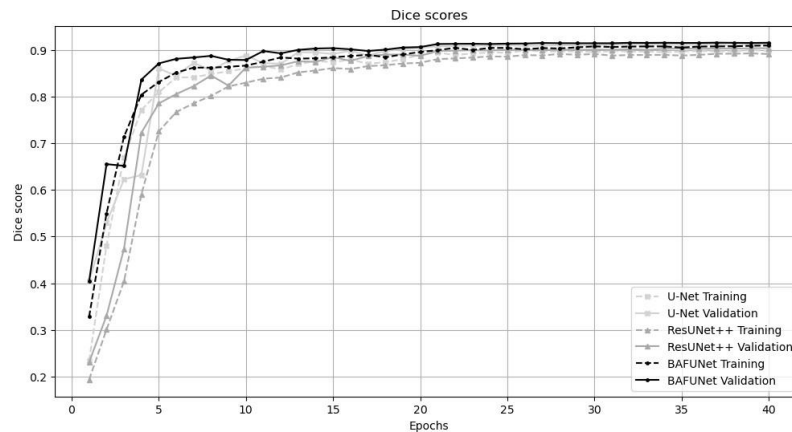


Рисунок 4.2 – Керване навчання, коефіцієнт подібності Дайса

Рисунок 4.3 демонструє сегментацію за допомогою навченої мережі архітектури BAFUNet. Результат є достатньо непоганим, хоча й помітні певні помилки сегментації: відсутність невеликих частинок тіл хребців на 2D зрізах та деякі проблеми з виділенням границь сильно деформованих міжхребцевих дисків.

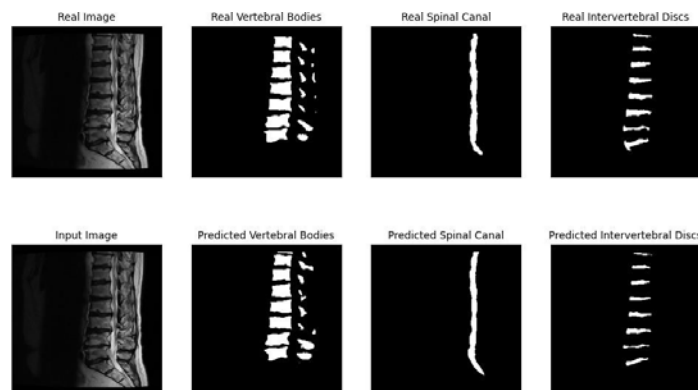


Рисунок 4.3 – Керване навчання, приклад сегментації

Найкращі результати для моделі гібридної архітектури BAFUNet були отримані при 37 епохах, значення функції втрат Жаккара: ≈ 0.157 , коефіцієнт подібності Дайса: ≈ 0.916 . Ці результати є достатньо непоганими, враховуючи відносно невеликий розмір тренувального набору та не дуже високу якість вхідних зображень.

4.2. Попереднє навчання BAFUNet

Для напівкерованого навчання тренувальний набір даних було розділено на марковану (10%, 36 зображень) та немарковану (90%, 324 зображення) вибірки. На етапі попереднього навчання була створена мережа архітектури BAFUNet з випадковою ініціалізацією ваг. Кількість епох та інші параметри відповідають параметрам з підрозділу 4.1, єдина різниця – в розмірі набору даних (в 10 разів менше маркованих даних) та в додатковій аугментації даних шляхом копіювання та вставки сегментів. Зміну значення функції втрат можна побачити на рисунку 4.4.

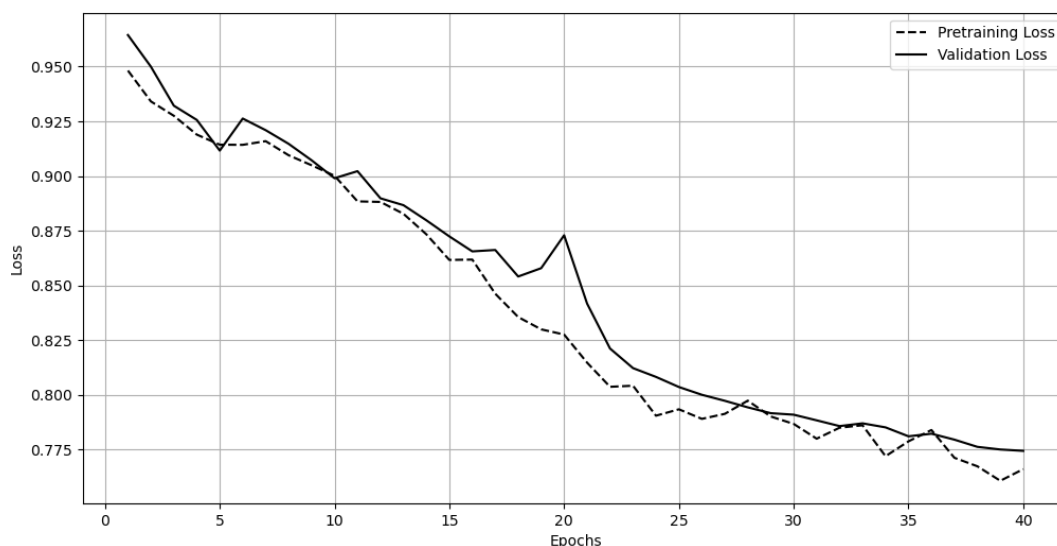


Рисунок 4.4 – Попереднє навчання, значення функції втрат

Зміна коефіцієнту подібності Дайса від епохи демонструється на рисунку 4.5.

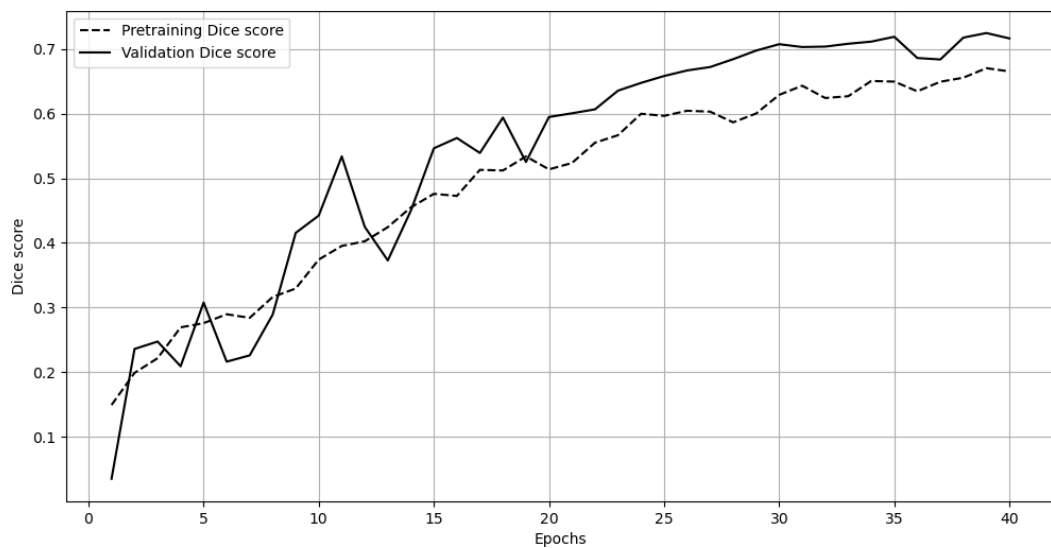


Рисунок 4.5 – Попереднє навчання, коефіцієнт подібності Дайса

Приклад сегментації за допомогою попередньо навченої мережі можна побачити на рисунку 4.6.

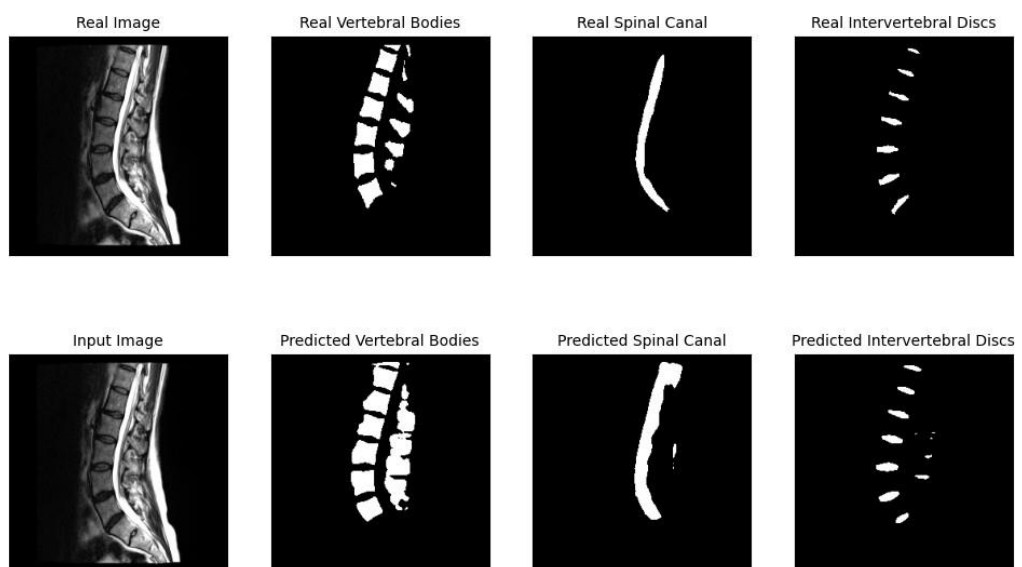


Рисунок 4.6 – Попереднє навчання, приклад сегментації

Як можна побачити з рисунків 4.4-4.6, результат є помітно гіршим, однак це очікувано, бо розмір тренувальної вибірки для етапу попереднього навчання був в 10 разів меншим. Найкращі результати були отримані при 39 епохах, значення функції втрат Жаккара: ≈ 0.775 , коефіцієнт подібності Дайса: ≈ 0.724 .

4.3. Напівкероване навчання BAFUNet

Під час напівкерованого навчання застосовувалися дві моделі з ваговими коефіцієнтами найкращої моделі з етапу попереднього навчання (підрозділ 4.2). Етап напівкерованого навчання передбачав використання і маркованих, і немаркованих даних, та реалізує метод, описаний в підрозділі 2.2. Функція втрат обчислювалася як сума втрат для маркованої та псевдомаркованої частин, при цьому окрім індексу Жаккара також враховувалася крос-ентропія. Значення коефіцієнту втрат маркованої частини: 1.0, псевдомаркованої частини: 0.5. Значення функції втрат під час тренування можна побачити на рисунку 4.7.

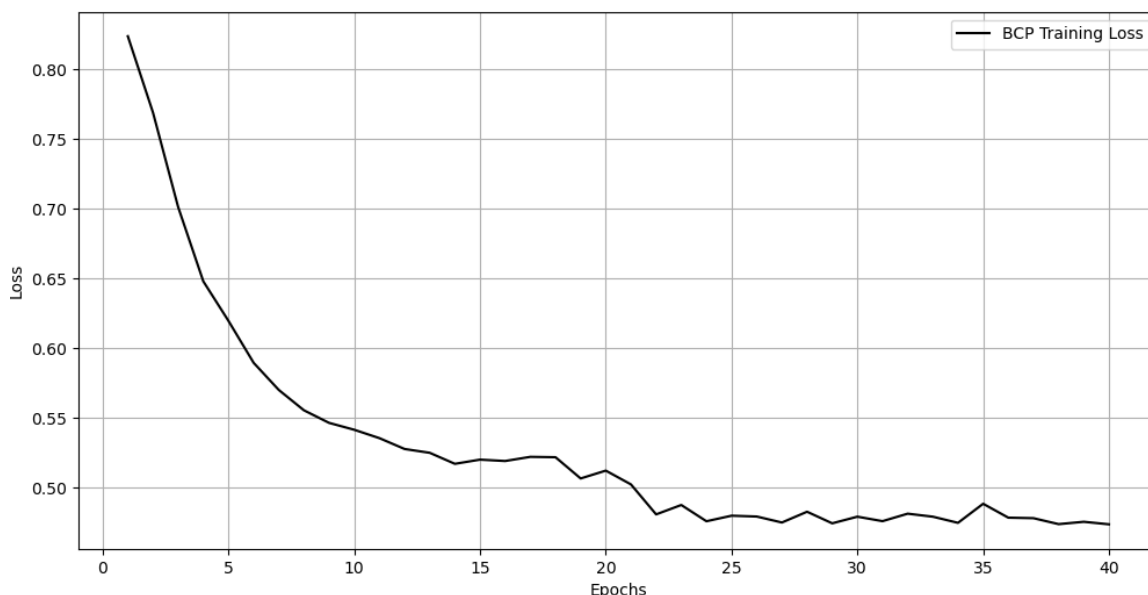


Рисунок 4.7 – Напівкероване навчання, тренування, значення функції втрат

Значення функції втрат на валідаційній вибірці показано на рисунку 4.8.

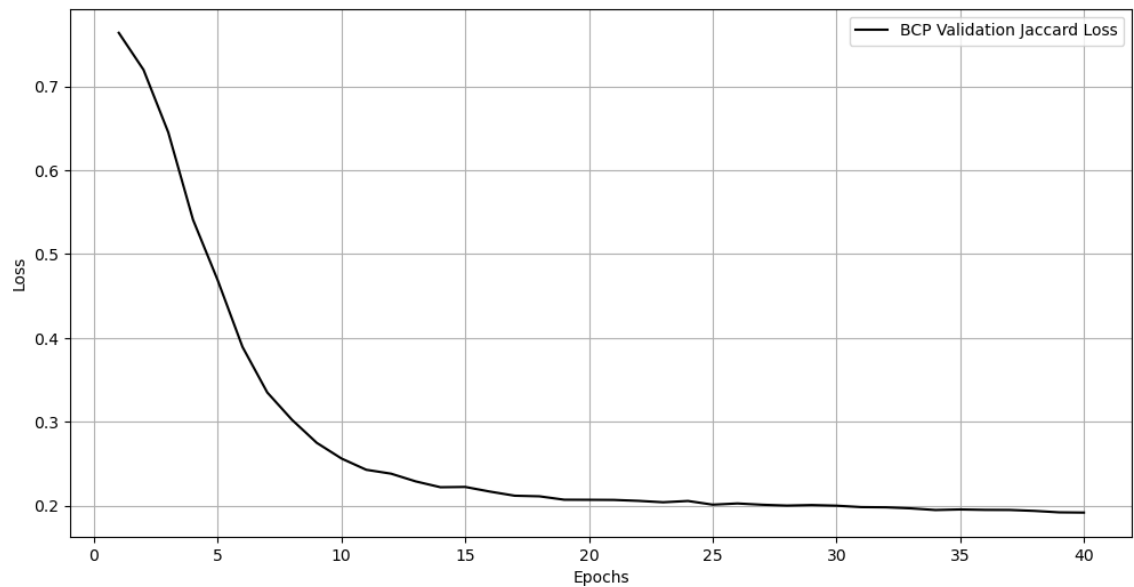


Рисунок 4.8 – Напівкерване навчання, валідація, значення функції втрат

Рисунок 4.9 демонструє залежність коефіцієнту подібності Дайса від епохи при обчисленні цієї метрики на валідаційній вибірці.

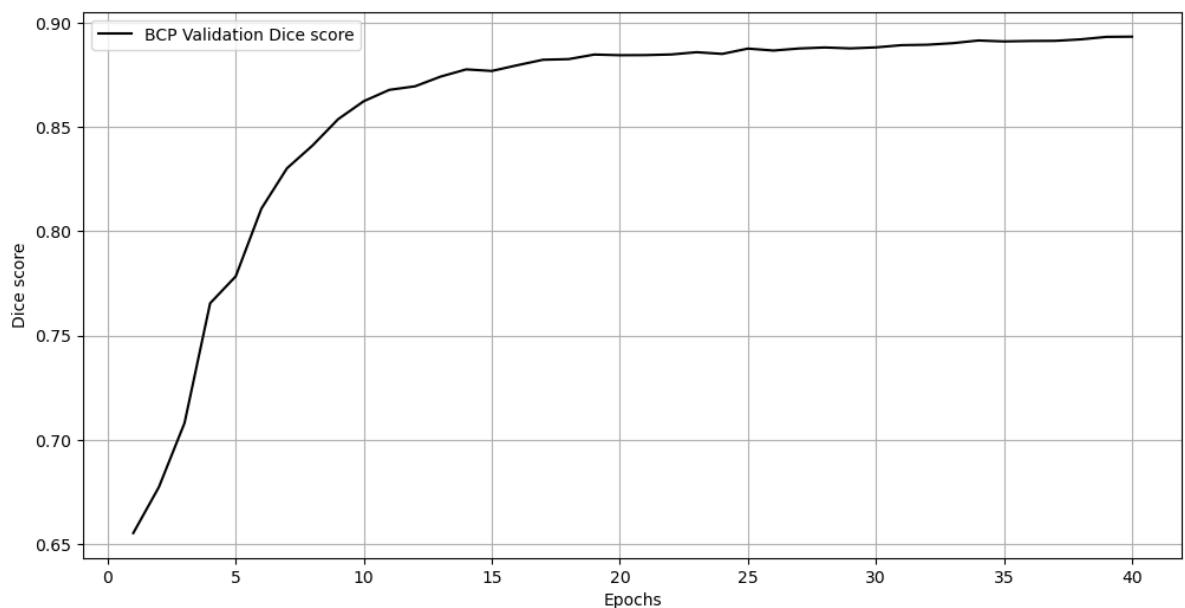


Рисунок 4.9 – Напівкерване навчання, валідація, значення коефіцієнту Дайса

Приклад сегментації за допомогою навченої мережі видно на рисунку 4.10. Результати є схожими з сегментацією з рисунку 4.3, однак тепер нова модель інколи замість пропуску невеликих частинок тіл хребців намагається виділити зайві сегменти.

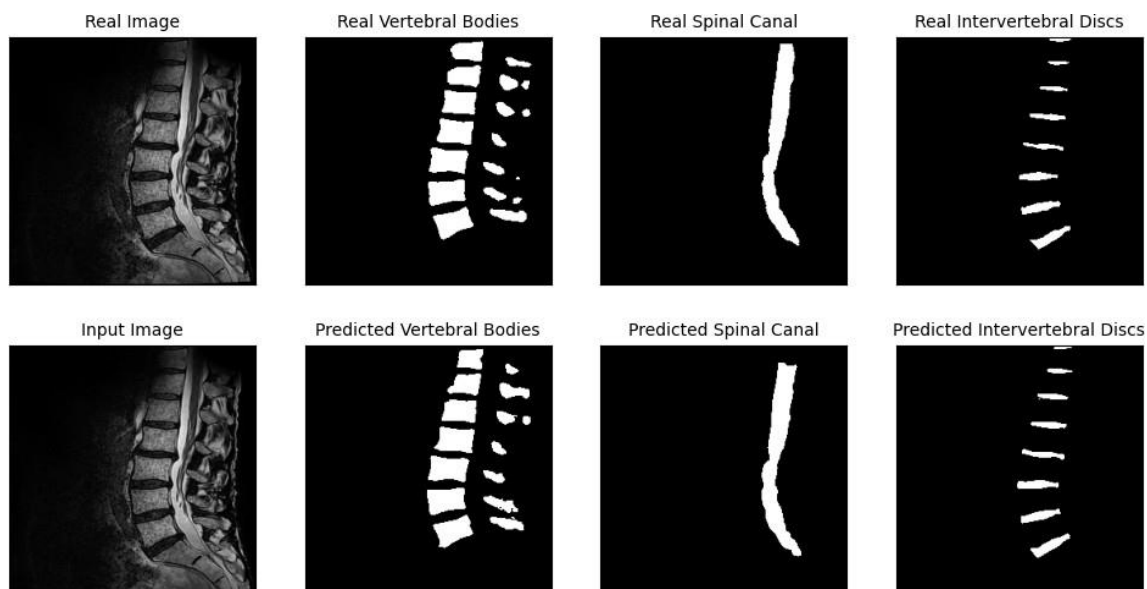


Рисунок 4.10 – Напівкероване навчання, приклад сегментації

Найкращі результати були досягнуті на 40 епосі, функції втрат: ≈ 0.192 , коефіцієнт подібності Дайса: ≈ 0.893 . Отримані метрики для моделей гібридної архітектури BAFUNet на усіх етапах можна побачити в таблиці 4.1.

Таблиця 4.1 – Метрики для експериментального дослідження

<i>Етап</i>	<i>Метрики</i>	
	<i>Коефіцієнт Жаккара</i>	<i>Коефіцієнт Дайса</i>
Кероване навчання	0.843	0.916
Попереднє навчання	0.225	0.724
Напівкероване навчання	0.808	0.893

Таким чином, оглянувши отримані метрики з таблиці 4.1, можна побачити, що за допомогою напівкерованого навчання обраним методом при

10% маркованих даних в наборі вдалося отримати значно кращі результати, ніж при навчанні керованим підходом на цих 10% даних. Як видно, коефіцієнт подібності Дайса збільшився на 0.169, а покращення помітно навіть неозброєним оком, якщо порівнювати рисунок 4.6 та рисунок 4.10. Також варто зазначити, що отримані результати достатньо близькі до результатів, які були отримані при навчанні зі 100% маркованим набором даних: індекс Жаккара відрізняється лише на 0.035, а коефіцієнт подібності Дайса – на 0.023.

4.4. Навчання ResNet

Навчання мереж для класифікації ступенів деформації міжхребцевих дисків виконувалося на невеликих квадратних регіонах інтересу зі стороною в 64 пікселя. Дані готувалися з використанням бібліотеки OpenCV: виділялися контури сегментованих дисків, після чого відповідні контурам регіони вирізалися. За допомогою OpenCV було отримано набір даних, що складався з 2D зображень, де кожне зображення містило лише один міжхребцевий диск. Вхідні дані містили 3 канали: початкове МРТ-зображення, а також 2 канали від модуля сегментації (тіла хребців та міжхребцеві диски), приклад вхідних даних можна побачити на рисунку 4.11.

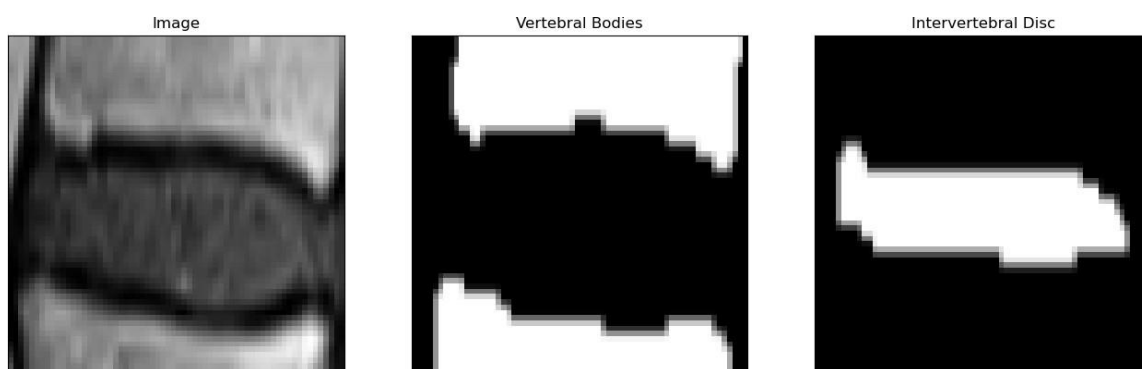


Рисунок 4.11 – Вхідні дані ResNet

Навчання моделей ResNet-18 виконувалося на 45 епохах з використанням фокальної функції втрат FocalLoss зі значенням параметру модуляції $\gamma = 2$ та оптимізатору Adam з початковою швидкістю навчання 0.001 зі зменшенням в 10 разів кожні 15 епох, розмір пакету дорівнював 32. Функція втрат FocalLoss була обрана для покращення навчання в умовах дисбалансу класів. FocalLoss допомагає зосереджувати увагу на важких зразках, що допомагає моделі краще справлятися з класами, представленими в обмеженій кількості. Після кожної епохи зберігалася модель з найвищим значенням точності на тестувальній вибірці, а для найкращої моделі будувалася матриця невідповідностей.

Перший експеримент був спрямований на діагностування ступеня дегенеративних змін міжхребцевих дисків за класифікацією Pfirrmann, що включає здоровий стан та п'ять рівнів дегенеративних змін. На рисунку 4.12 можна побачити графік залежності точності моделі від епохи. На останніх епохах спостерігається стабілізація точності на тестовій вибірці (≈ 0.5), в той же час точність на тренувальній вибірці продовжує рости, що може свідчити про перенавчання моделі.

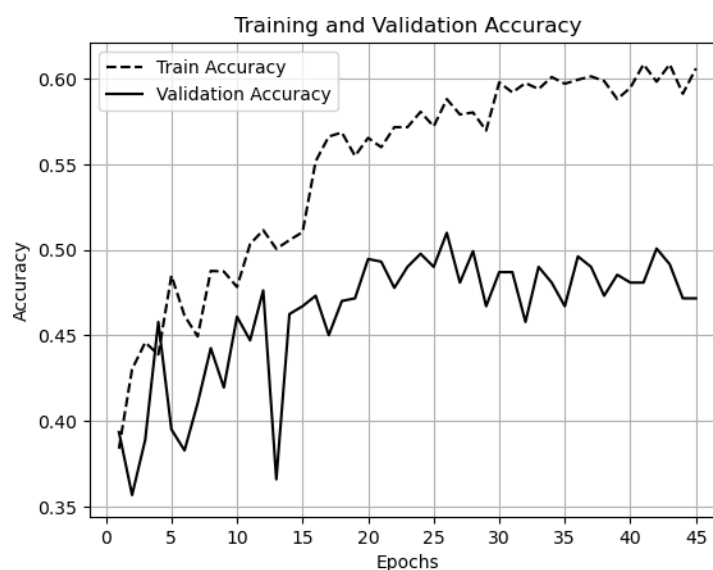


Рисунок 4.12 – Експеримент №1, точність моделей для діагностування

Матриця невідповідностей найкращої моделі зображена на рисунку 4.13. На цій матриці можна побачити цікаву особливість: модель схильна переоцінювати рівні за Pfirrmann, а також має певні проблеми із суміжними класами (наприклад, рівні III-IV та IV-V), що пояснюється подібністю між класами. На тестовій вибірці модель значно краще справлялася з класифікацією V та IV рівнів за Pfirrmann: повнота (recall) для цих рівнів дорівнює 0.79 та 0.56 відповідно. Також можна помітити, що в обраному наборі даних були майже відсутні здорові міжхребцеві диски, що потенційно може викликати проблеми з визначенням здорових дисків на іншому наборі даних.

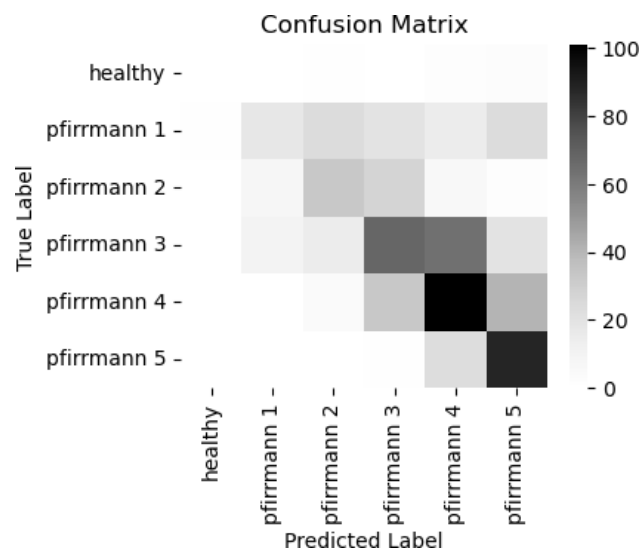


Рисунок 4.13 – Експеримент №1, матриця невідповідностей

У другому експерименті реалізовано власну класифікацію, що поділяє деформацію дисків на три категорії: незначна деформація (рівні за Pfirrmann до II включно), помірна деформація (рівень III) та тяжка деформація (рівні IV–V). Відповідний графік залежності точності моделі від епохи зображено на рисунку 4.14. Як і під час минулого експерименту, на останніх епохах спостерігається перенавчання, точність на тестовій вибірці коливалася біля значення ≈ 0.65 .

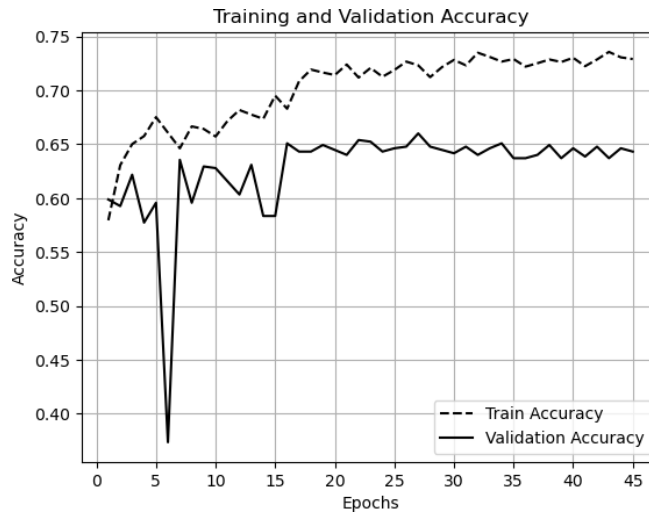


Рисунок 4.14 – Експеримент №2, точність моделей для діагностування

На рисунку 4.15 зображена матриця невідповідностей для цього експерименту. Ця матриця демонструє, що модель значно краще виявляє тяжкі та незначні деформації міжхребцевих дисків, аніж помірні. При цьому отримана модель також схильна переоцінювати категорію деформації, що особливо помітно для помірних та тяжких деформацій.

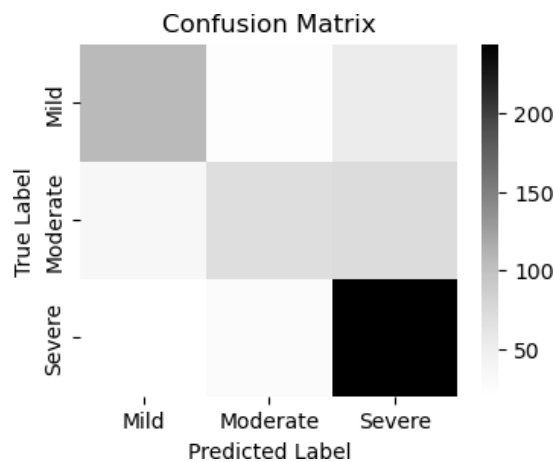


Рисунок 4.15 – Експеримент №2, матриця невідповідностей

В таблиці 4.2 наведені метрики для кожного з класів окремо. З цієї таблиці видно, що модель має високе значення повноти для тяжких деформацій, що може допомогти вчасно виявити такі деформації.

Таблиця 4.2 – Метрики найкращої моделі діагностування

<i>Класи деформацій міжхребцевих дисків</i>	<i>Метрики</i>		
	<i>Влучність (precision)</i>	<i>Повнота (recall)</i>	<i>F₁-міра</i>
Незначна деформація	0.66	0.57	0.61
Помірна деформація	0.57	0.40	0.47
Тяжка деформація	0.66	0.84	0.74

Висновки

В четвертому розділі наведені результати досліджень мереж для сегментації 2D МРТ-зображень хребта та мереж для діагностування деформації міжхребцевих дисків за двома різними класифікаціями. Виявлено, що запропонована гібридна мережа BAFUNet демонструє кращі результати на обраному наборі даних, аніж архітектури U-Net та ResUNet++. При цьому застосування методу напівкерованого навчання Mean Teacher з двонаправленим копіюванням та вставкою при 10% маркованих даних призвело до отримання результатів, майже порівняних з повністю керованим навчанням.

Під час навчання мереж архітектур U-Net, ResUNet++ та BAFUNet розроблена гібридна мережа BAFUNet продемонструвала кращі результати: коефіцієнт подібності Дайса цієї моделі дорівнював 0.916, що приблизно на 0.003 більше за аналоги. При використанні напівкерованого навчання на 10% маркованих даних значення коефіцієнту подібності Дайса зменшилося лише на 0.023 в порівнянні з керованим навчанням на 100% даних. Висока ефективність обраного методу при значно меншому обсязі маркованих даних надає можливість його застосування в умовах, коли більшість даних в отриманому наборі є немаркованими.

Було виконано навчання моделей ResNet-18 діагностування деформації міжхребцевих дисків за класифікацією Pfirrmann та власною класифікацією (незначна, помірна та тяжка деформації). Було виявлено, що введення власної класифікації підвищило точність приблизно на 15%, однак обидві моделі демонстрували труднощі у розрізненні суміжних класів. Найкраща отримана модель демонструє високе значення повноти для тяжких деформацій (0.84), що може допомогти вчасно виявляти ці деформації.

Розділ 5

РОЗРОБКА ПИТАНЬ ОХОРОНИ ПРАЦІ

Даний розділ дипломної роботи виконується за планом, оскільки його метою є розробка програмного забезпечення, яке використовує згорткові нейронні мережі для автоматизованого виявлення ранніх стадій хвороби хребта шляхом аналізу МРТ-зображень. Це передбачає використання наявних приладів та вирішення питань, пов'язаних з організацією проведення досліджень.

Метою цього розділу є аналіз впливу шкідливих факторів та умов праці на виробництві, а також розробка відповідних заходів щодо усунення негативного впливу на персонал.

У роботі було розглянуто МРТ «Toshiba Vantage Titan 1.5T» та його використання для отримання МРТ-зображень, які були використані в дипломній роботі.

5.1. Характеристика приміщення та робочого місця

Перш за все слід зазначити що блок приміщень для МРТ складається з трьох складових:

1. Пультова кімната або операторська – кімната, де знаходиться оператор, який здійснює сканування і візуальний контроль через оглядове вікно.
2. Технічна кімната – в ній розміщуються шафи МРТ, градієнтний і радіочастотні підсилювачі, шафа обробки даних, криокомпресор.
3. Сканерна кімната – приміщення, де встановлюється томограф.

Загальні розміри блоку повинні бути приблизно 45-65м². Нижче наведено характеристики приміщення, специфікація технологічного обладнання та оснащення в якому проводяться дослідження за завданням.

Таблиця 5.1 – Характеристика приміщення

№	Найменування	Основні характеристики	Кількість	Позиція на рисунку
1.	Параметри приміщення: 1) Пультова 2) Технічна 3) Сканерна	5000*2500*3000; S=12,5м ² ; V=37,5м ³ 4000*2500*3000; S=10м ² ; V=30м ³ 5000*6000*3000; S=30м ² ; V=90м ³		-
2.	Кількість працюючих	оператор, рентгенолог	2	-
3.	Штучне освітлення	ЛПБ 80	12	-
4.	Штучна вентиляція	Спеціальний шлюз для охолодження гелію (чиллер), радіочастотний фільтр та повітроводи з немагнітної нержавіючої сталі, загальна вентиляція		-
5.	Опалення	Центральне водяне опалення низького тиску		-
6.	Підлога	Електропровідний матеріал (клітка фарадея), бетон		-
7.	Стіни	Електропровідний матеріал (клітка фарадея), бетон		-
Обладнання і оснащення				
8.	Toshiba Vantage Titan 1.5T	Габарити – 4500 (з ліжком)*2400*2200; Живлення – 3х фазне 380В; Тип – закритий; Вага та тип магніту – 4 тони надпровідний; Кількість каналу прийому і передачі – 16/32; Поле огляду – 55*55*50; Швидкість наростання градієнта – 120 мТ/м/мкс; Макс. Величина градієнтів – 30мТ/м;	1	1
9.	Стіл для пацієнта МРТ апарату Toshiba Vantage Titan 1.5T	Розміри - 2230*680*934; робочий хід – 2190 по горизонталі, 633 по вертикалі; максимальна вага пацієнта – 130 кг	1	7
10.	Крісло робоче Barsky Mesh BM-02	Розміри - 440*420*105; матеріал – корпус – пластик, наповнювач – паралон, обшивка - текстиль	1	2
10.	Стіл	Розміри – 1200*700*700; матеріал – стільниця - дерево (дуб), опори - ДСП	1	3
10.	Шафа	Розміри – 700*500*1600; матеріал – пластик, покриття – полімерно-порошкове	2	4
11.	Пульт управління	Комп'ютер з спеціальним програмним забезпеченням (M-Power, WorkFlow Driven Application (WFDA)) -	1	5

На рисунку 5.1 показано план блоку:

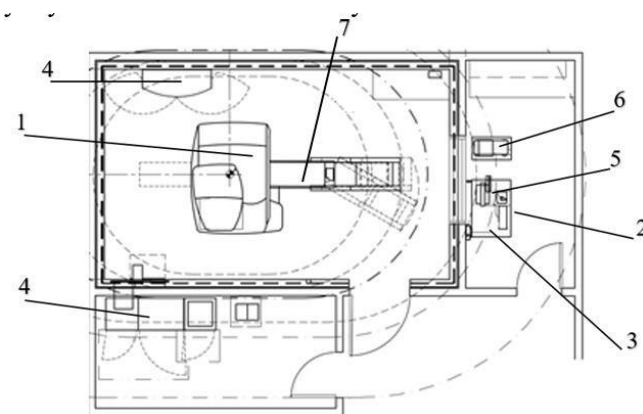


Рисунок 5.1 – План робочого приміщення

Встановлення на відповідність вимогам нормативних документів об'єму і площі приміщення на 1 працівника та місця розташування технологічного обладнання наведено [26]

Базуючись на даних з таблиці 4.2 [26], можна зробити висновок, що кабінет відповідає нормативним параметрам, встановленим для цього типу приміщення. Вказані параметри кабінетів відповідають вимогам технічно-санітарних норм, що забезпечує безпечність та комфортність під час праці.

5.2. Оцінка небезпечних і шкідливих виробничих факторів

Відповідно до ДСТ 12.0.003-2014 [27], всі небезпечні та шкідливі чинники класифікуються на 4 групи. В таблиці 5.2 наведені результати з оцінки присутності чи відсутності цих чинників при виконанні роботи з тестування.

Таблиця 5.5 – Небезпечні та шкідливі виробничі фактори

Група небезпечних факторів	Перелік основних чинників
Фізичні	Магнітне поле; електронезбезпека
Хімічні	Контрастні речовини
Біологічні	Відсутні
Психофізіологічні	Клаустрофобія

Результати таблиці 5.2 показують фізичні, хімічні та біологічні небезпечні та шкідливі фактори. Найбільш вплив з яких становить магнітне

поле та електронезбезпека, саме вони будуть розглянуті в роботі.

Магнітне поле

Принцип роботи магнітно-резонансного томографа полягає в активації енергії протонів водню, що є у всіх тканинах людського організму. Під дією магнітного поля томографа енергію протонів водню можна зафіксувати. Інформація про джерела магнітного поля та пов'язані з ним небезпеки наведено

Таблиця 5.3 – Реальні і нормативні значення показників

№	Фактор небезпеки	Реальне значення	Нормативне значення
1.	Магнітне поле	1,5 Т	0,2-3 Т

Для мінімізації шкідливих наслідків у таблиці 5.4 наведено заходи та засоби захисту від дії магнітного поля.

Таблиця 5.4 – Засоби і заходи захисту від впливу магнітного поля

№	Група номенклатурних заходів з ОП	Вид заходу	Критерій вибору
1.	Технічні заходи	Екранування приміщення, використання екранованих кабелів	Захист від магнітного поля
2.	Організаційні заходи	Заборона наявності металевих предметів у зоні МРТ	Усвідомлення принципів безпеки
3.	Режимні заходи	Використання спеціальних протоколів безпеки для пацієнтів і персоналу	Усвідомлення принципів безпеки
4.	Експлуатаційні заходи	Регулярна перевірка технічного стану МРТ, проведення профілактичних обслуговувань	Контроль працездатності приладів

Електронезбезпека

Кабінет магнітно-резонансної томографії містить прилади, які споживають електроенергію. В таблиці 5.5 наведені фактори та наслідки електронезбезпеки.

Таблиця 5.5. – Реальні і нормативні значення показників

№	Найменування електроприладу	Реальне значення	Нормативне значення
1.	Постійний струм	350 мА	до 500 мА
2.	Змінний струм в мережевих кабелях живлення	11-14 А	до 16 А

Таблиця 5.6 – Засоби і заходи захисту від електронебезпеки

№	Група номенклатурних заходів з ОП	Вид заходу	Критерій вибору
1.	Технічні заходи	Заземлення та використання плавких запобіжників	Електроізоляція
2.	Організаційні заходи	Інструктаж з питань електробезпеки	Навчання з питань безпеки при експлуатації обладнання
3.	Режимні заходи	Використання спеціальних протоколів безпеки для пацієнтів і персоналу	Усвідомлення принципів безпеки
4.	Експлуатаційні заходи	Своєчасна заміна деталей техніки, що вийшли з ладу чи пошкоджені, перевірка електричних апаратів за допомогою мегомметра не менше одного разу на рік	Контроль працездатності приладів, контроль показників стрибків струму

Висновки

В даному розділі було проаналізовано умови праці в приміщення для МРТ, згідно розрахунків вони задовольняють нормативні значення. Також розглянуто основні види небезпек такі як електронебезпека та дія магнітного поля. Для кожної з небезпек розроблено ряд заходів для зменшення ризику їх виникнення. Кабінет МРТ відповідає вимогам електро- та пожежної безпеки.

ВИСНОВКИ

У результаті виконання дипломної роботи було досліджено й концептуально описано принципи створення біомедичної системи комп'ютерного аналізу медичних зображень гібридного томографа. В умовах стрімкого розвитку медичних технологій та зростання обсягів діагностичних даних автоматизовані системи набувають особливої важливості для підвищення ефективності роботи медичного персоналу та якості надання медичних послуг.

Застосування гібридних методів візуалізації, таких як ПЕТ/КТ та ПЕТ/МРТ, забезпечує об'єднання анатомічної й функціональної інформації, що суттєво підвищує точність діагностики. Однак ефективне використання таких систем можливе лише за умови застосування потужних алгоритмів обробки, аналізу та візуалізації зображень. У межах роботи було розглянуто можливості використання сучасних методів комп'ютерного зору, машинного навчання та нейронних мереж, що дозволяють покращити точність і швидкість виявлення патологічних змін.

Особливу увагу приділено технічним аспектам формування медичних зображень, таким як просторова роздільна здатність, параметри вокселів, контрастність, узгодженість даних між різними модальностями. Надано приклади їхнього впливу на якість діагностики та можливості інтеграції в клінічну практику.

Отримані результати свідчать про актуальність і перспективність впровадження систем комп'ютерного аналізу медичних зображень у практику сучасної медицини. Розроблена концепція може слугувати основою для подальшого створення прикладного програмного забезпечення та систем підтримки прийняття клінічних рішень у галузі біомедичної інженерії.

Метою роботи була розробка моделі гібридної штучної нейронної мережі для сегментації МРТ-зображень хребта та її навчання з використанням методів напівкерованого навчання. Були виконані: огляд архітектур нейронних мереж для сегментації та методів напівкерованого навчання, розгляд модифікацій архітектури U-Net і методу напівкерованого навчання

Mean Teacher з двонаправленим копіюванням та вставкою, розроблена власна гібридна архітектура BAFUNet для сегментації МРТ-зображень хребта, а також проведені експериментальні дослідження з аналізом результатів, що підтверджується низкою діаграм, графіками з результатами, таблицею порівняння метрик а також написаним вихідним кодом. Крім цього було розроблено прототип модульної інтелектуальної медичної системи діагностики деформації міжхребцевих дисків.

В якості вхідних даних був обраний набір даних з сегментованими МРТ-зображеннями хребта в форматі даних МНА. Була проведена попередня обробка даних, навчання моделей, а також їхнє використання із заміром метрик. Результати порівняння моделей були занесені в таблиці та відображені у вигляді графіків. Було виявлено, що розроблена гібридна мережа BAFUNet продемонструвала кращі за інші мережі результати: коефіцієнт подібності Дайса приблизно на 0.003 перевищував аналоги. Також було виявлено, що модель, навчена шляхом напівкерovanого навчання, демонструє значно кращі результати за модель, навчену лише на невеликій (маркованій) частині набору, однак при цьому результати є трохи гіршими, ніж при навчанні моделі повністю керованим шляхом на 100% розмічених даних. Виявлено, що на обраному наборі даних різниця між напівкерованою моделлю із 10% маркованих даних та повністю керованою є незначною: індекс Жаккара відрізняється лише на 0.035, а коефіцієнт подібності Дайса – на 0.023. Отже, отримані результати свідчать про те, що метод напівкерovanого навчання Mean Teacher з двонаправленим копіюванням та вставкою в поєднанні з архітектурою BAFUNet можуть бути вдало застосовані для сегментації медичних зображень при малій кількості маркованих даних.

Окрім мережі для сегментації МРТ-зображень хребта також було виконано навчання моделей ResNet-18 для діагностування деформації міжхребцевих дисків. Виявлено, що введення власної класифікації деформації міжхребцевих дисків підвищило точність приблизно на 15% в порівнянні з класифікацією Pfirrmann, при цьому найкраща модель демонструвала високе

значення повноти для тяжких деформацій (≈ 0.84). Таким чином, в результаті проведення експериментів були отримані навчені мережі архітектур BAFUNet та ResNet-18, які можуть бути використані в якості модулів сегментації та діагностування інтелектуальної медичної системи діагностики деформації міжхребцевих дисків. Ці результати можуть бути використані в медичній діагностиці захворювань хребта за МРТ-зображеннями.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Нейронні мережі – шляхдо глибинного навчання – Codeguida: веб-сайт. URL: <https://codeguida.com/post/739> (дата звернення: 18.05.2023).
2. Татарин М. М., ЯкимовичГ. М., ПанкевичН. А. Хвороба Альцгеймера (Бібліографічний покажчик): Івано-Франківськ 2021. 15 с.
3. Олещенко Л.М. Машинне навчання: навч. посіб. / Київ. нац. ун-т ім. Ігоря Сікорського. Київ: 2022. 92 с.
4. Метод опорних векторів –Wikipedia: веб-сайт. URL:https://upload.wikimedia.org/wikipedia/commons/thumb/7/72/SVM_margin.png/1024px-SVM_margin.png(дата звернення: 23.05.2023).
5. Згорткові нейронні мережі – YouTube: веб-сайт. URL:https://www.youtube.com/watch?v=Qkq0VLuw5Vg&list=PL_ug4sOLLpQiHmRGMki4ID6By9nPzAH0A&index=23 (дата звернення: 23.05.2023).
6. Тимченко А.О, Скачков В.О. Функція активації нейрона ReLu: Житомирський державний технологічний університет. Житомир: 2018. 1 с.
7. Навчання штучної нейронної мережі методом зворотного розповсюдження помилки – Studfile:веб-сайт. URL:<https://studfile.net/preview/5461803/> (дата звернення: 23.05.2023).
8. М.А. Новотарський, Б.Б. Нестеренко. Штучні нейронні мережі: 9. Обчислення: Інститут математики НАН України. Київ: 2004. 207 с.
10. Зінченко О. В., Звенігородський О. С., Кисіль Т. М. Згорткові нейронні мережі для вирішення задач комп'ютерного зору: Державний університет телекомунікацій. Київ: 2022. 73 с.
11. Класифікація зображень за допомогою ЗНМ – Codeguida: веб-сайт. URL: <https://codeguida.com/post/1400>(дата звернення: 25.05.2023).
- 12.Ramzan F, Khan MUG, Rehmat A, et al. A Deep Learning Approach for Automated Diagnosis and Multi-Class Classification of Alzheimer's Disease Stages

Using Resting-State fMRI and Residual Neural Networks. *Journal of Medical Systems*. 2019 Dec;44(2):37. DOI: 10.1007/s10916-019-1475-2. PMID: 31853655.

13.K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, 2016, pp. 770-778, doi: 10.1109/CVPR.2016.90.

14.Ta, N., Chen, H., Liu, X. *et al.* LET-Net: locally enhanced transformer network for medical image segmentation. *Multimedia Systems* **29**, 3847–3861 (2023). <https://doi.org/10.1007/s00530-023-01165-z>

15. Suri, A., Jones, B. C., Ng, G., Anabaraonye, N., Beyrer, P., Domi, A., ... & Rajapakse, C. S. (2021). A deep learning system for automated, multi-modality 2D segmentation of vertebral bodies and intervertebral discs. *Bone*, 149, 115972.

16. Gao, S., Zhang, Z., Ma, J., Li, Z., & Zhang, S. (2023, October). Correlation-Aware Mutual Learning for Semi-supervised Medical Image Segmentation. *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 98-108.

17. Wei, Q., Yu, L., Li, X., Shao, W., Xie, C., Xing, L., & Zhou, Y. (2023, October). Consistency-Guided Meta-learning for Bootstrapping Semi-supervised Medical Image Segmentation. *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 183-193.

18. Bai, Y., Chen, D., Li, Q., Shen, W., & Wang, Y. (2023). Bidirectional copy-paste for semi-supervised medical image segmentation. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 11514- 11524.

19. Khalil, Y.A., Becherucci, E.A., Kirschke, J.S. *et al.* (2022). Multi-scanner and multi-modal lumbar vertebral body and intervertebral disc segmentation database. *Sci Data* 9, 97.

20. Chu, C., Belavý, D. L., Armbrecht, G., Bansmann, M., Felsenberg, D., & Zheng, G. (2015). Fully automatic localization and segmentation of 3D vertebral bodies from CT/MR images via a learning-based method. *PLoS One*, 10(11), e0143327.

21. van der Graaf, J. W., van Hooff, M. L., Buckens, C. F., Rutten, M., van Susante,

J. L., Kroeze, R. J., ... & Lessmann, N. (2024). Lumbar spine segmentation in MR images: a dataset and a public benchmark. *Scientific Data*, 11(1), 264.

22. Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference*, Munich, Germany, October 5-9, 2015, proceedings, part III 18 (pp. 234-241). Springer International Publishing.

23. Azad, R., Aghdam, E. K., Rauland, A., Jia, Y., Avval, A. H., Bozorgpour, A., & Merhof, D. (2024). Medical image segmentation review: The success of U-Net. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

24. Buda, M., Saha, A., & Mazurowski, M. A. (2019). Association of genomic subtypes of lower-grade gliomas with shape features automatically extracted by a deep learning algorithm. *Computers in biology and medicine*, 109, 218- 225.

25. Tarvainen, A., & Valpola, H. (2017). Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *Advances in neural information processing systems*, 30.

26. Zhang, Z., Wu, C., Coleman, S., & Kerr, D. (2020). DENSE-INception U-net for medical image segmentation. *Computer methods and programs in biomedicine*, 192, 105395.

27. Li, Z., Lyu, H., & Wang, J. (2024, February). FusionU-Net: U-Net with Enhanced Skip Connection for Pathology Image Segmentation. *In Asian Conference on Machine Learning* (pp. 694-706). PMLR.

28. Wang, Z., Zou, Y., & Liu, P. X. (2021). Hybrid dilation and attention residual U-Net for medical image segmentation. *Computers in biology and medicine*, 134, 104449.

29. Jha, D., Smedsrud, P. H., Riegler, M. A., Johansen, D., De Lange, T., Halvorsen, P., & Johansen, H. D. (2019, December). ResUNet++: An advanced architecture for medical image segmentation. *In 2019 IEEE international symposium on multimedia (ISM)* (pp. 225-2255). IEEE.

30. *Signal, Image and Video Processing*, 16(6), 1607-1614.

31. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *In Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778).
32. Ramzan, F., Khan, M. U. G., Rehmat, A., Iqbal, S., Saba, T., Rehman, A., & Mehmood, Z. (2020). A deep learning approach for automated diagnosis and multi-class classification of Alzheimer's disease stages using resting-state fMRI and residual neural networks. *Journal of medical systems*, 44, 1-16.
33. Документація по Jupyter Notebook [Електронний ресурс] – Режим доступу: <https://jupyter-notebook.readthedocs.io/> (дата звернення: 26.05.2024).
34. Exposing Computer Generated Images by Eye's Region Classification via Transfer Learning of VGG19 CNN –Ieexplore:веб-сайт. URL: <https://ieeexplore.ieee.org/abstract/document/8260745>(дата звернення: 26.05.2023).
35. Detecting Affect States Using VGG16, ResNet50 and SE-ResNet50 Networks – Link.Springer: веб-сайт. URL: <https://link.springer.com/article/10.1007/s42979-020-0114-9>(дата звернення: 26.05.2023).
36. Закон України «Про небезпечні та шкідливі виробничі фактори. Класифікація»/ Законодавство України про охорону праці. – К. Нова редакція 2002 р.

ДОДАТОК А

ОПИС ПРЕДМЕТНОГО СЕРЕДОВИЩА

Робота виконувалася на MacBook Pro з чипом Apple M3 Pro (12 CPU / 18 GPU) та 36 ГБ об'єднаної пам'яті, в інтерактивному середовищі Jupyter Notebook з використанням мови програмування Python версії 3.11.7 та наступними бібліотеками:

- PyTorch версії 2.4.1;
- torchvision версії 0.19.1;
- NumPy версії 1.26.4;
- Pandas версії 2.1.4;
- scikit-learn версії 1.4.2;
- Matplotlib версії 3.8.0;
- OpenCV (cv2) версії 4.9.0;
- Segmentation Models версії 0.3.3;
- Tiger версії 6.3.0;
- Albumentations версії 1.4.4.

В якості початкового набору даних виступають дані з набору з сегментованими МРТ-зображеннями хребта SPIDER (<https://spider.grand-challenge.org/>). Формат даних на вході: МНА, марковані 3D МРТ-зображення.

ДОДАТОК Б

ЛІСТИНГ ПРОГРАМИ

```
import pandas as pd

overview = pd.read_csv('~\Projects/KPI/datasets/spider/overview.csv')
gradings =
pd.read_csv('~\Projects/KPI/datasets/spider/radiological_gradings.csv')
import os, random

images_path = os.path.join(os.path.expanduser('~'),
"Projects/KPI/datasets/spider/images")

masks_path = os.path.join(os.path.expanduser('~'),
"Projects/KPI/datasets/spider/masks")

random_file_name = random.choice(os.listdir(images_path))
from tiger.io import read_image

file_name = random_file_name

image, header = read_image(os.path.join(images_path, file_name))
mask, mask_header = read_image(os.path.join(masks_path,
file_name)) from tiger.resampling import reorient_image,
align_mask_with_image image, header = reorient_image(image,
header)

mask, _ = align_mask_with_image(mask, mask_header, image.shape, header)
from tiger.screenshots import ScreenshotGenerator, ScreenshotCollage
generator_kwargs = {"axis": 0, "pixel_spacing": (1, 1)}

generator = ScreenshotCollage([
    ScreenshotGenerator(coordinate=0.45, **generator_kwargs),
    ScreenshotGenerator(coordinate=0.5, **generator_kwargs),
    ScreenshotGenerator(coordinate=0.55, **generator_kwargs)
])

screenshot = generator(image, mask, spacing=header['spacing'])
display(screenshot)

import numpy as np

import matplotlib.pyplot as plt
def visualize(**images):
    n = len(images)
    plt.figure(figsize=(16, 5))
```

```

for i, (name, image) in enumerate(images.items()):
    plt.subplot(1, n, i + 1)

    plt.xticks([])

    plt.yticks([])

    plt.title(' '.join(name.split('_')).title())
    plt.imshow(image, cmap='gray')

plt.show()

images_2d_path =
os.path.join(os.path.expanduser('~'),
"Projects/KPI/datasets/spider/2D/images")
masks_2d_path =
os.path.join(os.path.expanduser('~'),
"Projects/KPI/datasets/spider/2D/masks")

import cv2

with os.scandir(images_2d_path) as
    it: if any(it):

        print('2D images directory is not empty, skipping slices
generation')

    else:

        print('Creating 2D slices:')
        names_3d =
os.listdir(images_path) for
image_id in names_3d:

            image_3d_fp = os.path.join(images_path,
image_id) mask_3d_fp = os.path.join(masks_path,
image_id) image_3d, header =
read_image(image_3d_fp) mask_3d, mask_header =
read_image(mask_3d_fp)

            image_3d, header = reorient_image(image_3d, header)
mask_3d, _ = align_mask_with_image(mask_3d,
mask_header,

image_3d.shape, header)

            max_slices_count = image_3d.shape[0]

            for slice_index in
                range(max_slices_count): image =
                    np.rot90(image_3d[slice_index]) mask

```

```

        = np.rot90(mask_3d[slice_index]) if
        not np.any(mask):

            continue

        mask[np.logical_or(np.logical_and(mask > 0, mask < 100),
np.logical_and(mask > 100, mask < 200))] = 1

        mask[mask == 100] = 2

        mask[np.logical_and(mask > 200, mask < 300)] = 3

        image = np.expand_dims(((image - image.min()) * (255 /
(image.max() - image.min()))), axis=-1).astype('uint8')

        mask =
        mask.astype('uint8')
        out_file_name =

f'{os.path.splitext(image_id)[0]}_slice_{slice_index}.png'
        cv2.imwrite(os.path.join(images_2d_path, out_file_name),
image)
names_3d = os.listdir(images_path)
images_centers = {}

for image_id in names_3d:

    image_3d_fp = os.path.join(images_path,
image_id)
    mask_3d_fp = os.path.join(masks_path,
image_id)
    image_3d, header =
read_image(image_3d_fp)
    mask_3d, mask_header =
read_image(mask_3d_fp)

    image_3d, header = reorient_image(image_3d, header)
    mask_3d, _ = align_mask_with_image(mask_3d,
mask_header,

image_3d.shape, header)

    images_centers[os.path.splitext(image_id)[0]] = image_3d.shape[0] //

from torch.utils.data import DataLoader

from torch.utils.data import Dataset as
BaseDataset
class Dataset(BaseDataset):

    LABELLED_CLASSES = ['vertebral bodies', 'spinal canal',
'intervertebral discs']

    ALL_CLASSES = ['unlabelled'] + LABELLED_CLASSES

    def __init__(self, images_dir, masks_dir, ids=None, classes=None,

```

```

augmentation=None, preprocessing=None):
    self.ids = os.listdir(images_dir) if ids is None else list(ids)
    self.images_fps = [os.path.join(images_dir, image_id) for
image_id in self.ids]

    self.masks_fps = [os.path.join(masks_dir, image_id) for
image_id in self.ids]

    self.class_values = [self.ALL_CLASSES.index(cls.lower()) for
cls in classes]

    self.augmentation = augmentation
    self.preprocessing =
preprocessing

def __getitem__(self, i):
    image = cv2.imread(self.images_fps[i], 0)
    mask = cv2.imread(self.masks_fps[i], 0)
    image = np.expand_dims(image, -1)

    masks = [(mask == v) for v in
self.class_values] mask = np.stack(masks,
axis=-1).astype('float') if self.augmentation:
        sample = self.augmentation(image=image, mask=mask)
        image, mask = sample['image'], sample['mask']

    if self.preprocessing:
        sample = self.preprocessing(image=image,
mask=mask) image, mask = sample['image'],
sample['mask']

    return image,
mask def __len
(self):
    return len(self.ids)

dataset = Dataset(images_2d_path, masks_2d_path,
classes=Dataset.LABELLED_CLASSES)

print('Total images: ', len(dataset))
image, mask = dataset[4]
visualize(image=image,
mask=mask.squeeze()) import
matplotlib.pyplot as plt

```

```

def visualize_grid(nrows=1, ncols=None, figsize_width=8,
column_height=2.5, **images):

    n =
    len(images) if
    ncols is None:

        ncols = n // nrows + (n % nrows > 0)
    plt.figure(figsize=(figsize_width, column_height * nrows))
    for i, (name, image) in enumerate(images.items()):

        plt.subplot(nrows, ncols, i + 1)
        plt.xticks([])

        plt.yticks([])

        plt.title(' '.join(name.split('_')).title(), fontsize=10)
        plt.imshow(image, cmap='gray')

    plt.show()

image, mask = dataset[4]
visualize_grid(

    image=image,
    vertebral_bodies=mask.squeeze()[ :, :, 0 ],
    spinal_canal=mask.squeeze()[ :, :, 1 ],
    intervertebral_discs=mask.squeeze()[ :, :, 2 ],

    figsize_width=4,
    nrows=2,

)

import albumentations as albu
def
get_training_augmentation():

    train_transform = [
        albu.HorizontalFlip(p=0.5),

        albu.ShiftScaleRotate(scale_limit=0.25, rotate_limit=15,
shift_limit=0.1, p=1, border_mode=0),

        albu.PadIfNeeded(min_height=512, min_width=512,
always_apply=True, border_mode=0),

        albu.Resize(height=256, width=256, always_apply=True),
        albu.RandomCrop(height=224, width=224, always_apply=True),
        albu.GaussNoise(p=0.2),

```

```

        albu.OneOf([albu.CLAHE(p=1), albu.RandomBrightnessContrast(p=1),
albu.RandomGamma(p=1)], p=0.9),

        albu.OneOf([albu.Blur(blur_limit=3, p=1),
albu.MotionBlur(blur_limit=3, p=1)], p=0.9),

    ]

    return albu.Compose(train_transform)

def get_nondeforming_training_augmentation():
    train_transform = [

        albu.HorizontalFlip(p=0.5),
        albu.PadIfNeeded(min_height=512, min_width=512,

always_apply=True, border_mode=0),

        alb.Resize(height=256, width=256, always_apply=True),
        albu.RandomCrop(height=224, width=224, always_apply=True),
        albu.GaussNoise(p=0.2),

        albu.OneOf([albu.CLAHE(p=1), albu.RandomBrightnessContrast(p=1),
albu.RandomGamma(p=1)], p=0.9),

        albu.OneOf([albu.Blur(blur_limit=3, p=1),
albu.MotionBlur(blur_limit=3, p=1)], p=0.9),

    ]

    return
albu.Compose(train_transform) def
get_validation_augmentation():

    test_transform = [

        albu.PadIfNeeded(min_height=512, min_width=512, border_mode=0),
        albu.Resize(height=256, width=256)

    ]

    return albu.Compose(test_transform)
def to_tensor(x, **kwargs):

    return x.transpose(2, 0, 1).astype('float32')
def get_preprocessing(preprocessing_fn=None):

    _transform = [

        albu.Lambda(image=preprocessing_fn) if preprocessing_fn is not
None else None,

        albu.Lambda(image=to_tensor, mask=to_tensor),

```

```

    ]
    _transform = [fn for fn in _transform if fn is not
None] return albu.Compose(_transform)
augmented_dataset = Dataset(
    mages_2d_path,
    masks_2d_path,
    augmentation=get_training_augmentation(),
    classes=Dataset.LABELLED_CLASSES,
)
for i in range(3):
    image, mask = augmented_dataset[1]
    visualize(image=image, mask=mask.squeeze())
import pandas as pd
overview = pd.read_csv('~\Projects/KPI/datasets/spider/overview.csv')
training_file_names = set(overview[overview["subset"] ==
"training"]["new_file_name"].tolist())
validation_file_names = set(overview[overview["subset"] ==
"validation"]["new_file_name"].tolist())
training_centers_file_names =
[f'{prefix}_slice_{images_centers[prefix]}.png' for prefix
in training_file_names]
validation_centers_file_names =
[f'{prefix}_slice_{images_centers[prefix]}.png' for prefix in
validation_file_names]
all_slices = os.listdir(images_2d_path)
use_only_centers = True
if use_only_centers:
    training_file_names = training_centers_file_names
    validation_file_names = validation_centers_file_names
else:
    training_file_names = [x for scan_id in training_file_names for x in
all_slices if x.startswith(scan_id)]
    validation_file_names = [x for scan_id in validation_file_names for
x in all_slices if x.startswith(scan_id)]

```

```

import random
random.shuffle(training_file_names)
random.shuffle(validation_file_names
)

print('Training size:', len(training_file_names), 'Validation size:',
len(validation_file_names))

import torch
import numpy as
np

import segmentation_models_pytorch as
smp CLASSES = Dataset.LABELLED_CLASSES

DEVICE = 'cuda' if torch.cuda.is_available() else ('mps'
if torch.backends.mps.is_available() else 'cpu')
preprocessing_fn = None

train_dataset = Dataset(
    images_2d_path,
    masks_2d_path,
    ids=training_file_names,

    augmentation=get_training_augmentation(),
    preprocessing=get_preprocessing(preprocessing_fn),
    classes=CLASSES,
)

valid_dataset = Dataset(images_2d_path, masks_2d_path,
ids=validation_file_names,

    augmentation=get_validation_augmentation(),
    preprocessing=get_preprocessing(preprocessing_fn),
    classes=CLASSES,
)

train_loader = DataLoader(train_dataset, batch_size=8,
shuffle=True) valid_loader = DataLoader(valid_dataset, batch_size=1,
shuffle=False) from segmentation_models_pytorch import utils

def train_and_validate_simple_model(model, epochs=40,
learning_rate=0.001, show_graphs=True):

    loss = utils.losses.JaccardLoss()
    metrics =
[smp.utils.metrics.Fscore()]

```

```

optimizer = torch.optim.Adam([dict(params=model.parameters(),
lr=learning_rate)])

train_epoch =
    smp.utils.train.TrainEpoch( model,

    loss=loss,
    metrics=metrics,
    optimizer=optimizer,
    device=DEVICE,
    verbose=True,

)

valid_epoch =
    smp.utils.train.ValidEpoch( model,

    loss=loss,
    metrics=metrics,
    device=DEVICE,
    verbose=True,

)

train_history =
[] valid_history
= [] max_score =
0

best_model_epoch = 0

for i in range(1, epochs + 1):
    print('\nEpoch:
    {}'.format(i))

    train_logs =
    train_epoch.run(train_loader)
    train_history.append(train_logs)
    valid_logs =
    valid_epoch.run(valid_loader)
    valid_history.append(valid_logs)

    if          max_score          <
        valid_logs['fscore']:
        max_score          =
        valid_logs['fscore']
        torch.save(model,

```

```

f'./best_{model.__class__.__name__}_model.pth')
    best_model_epoch = i

    print(f'Model saved, fscore:
    {max_score}') if i % 20 == 0:

        optimizer.param_groups[0]['lr'
] = optimizer.param_groups[0]['lr'] / 10

        print(f'Decrease learning rate to
{optimizer.param_groups[0]["lr"]}'
) if show_graphs:

    plt.figure(figsize=(12, 6))
    plt.title(f'{model.__class__.__name__}
model')

    plt.plot(range(1, len(train_history) + 1), [h['jaccard_loss']
for h in train_history], label='Training Loss')

    plt.plot(range(1, len(train_history) + 1), [h['jaccard_loss']
for h in valid_history], label='Validation Loss')

    plt.xlabel('Epochs'
)
    plt.ylabel('Loss')
    plt.legend()
    plt.show()

    plt.figure(figsize=(12, 6))
    plt.title(f'{model.__class__.__name__}
model')

    plt.plot(range(1, len(train_history) + 1), [h['fscore'] for h in
train_history], label='Training Dice score')

    plt.plot(range(1, len(train_history) + 1), [h['fscore'] for h in
valid_history], label='Validation Dice score')

    plt.xlabel('Epochs')
    plt.ylabel('Dice score')
    plt.legend()

    plt.show()
return {

    "train_history": train_history,
    "valid_history": valid_history,
    "max_score": max_score,

```

```

        "best_model_epoch":
        best_model_epoch, "model": model,
    }

# U-Net with batch normalization from brain-segmentation-pytorch
repository1

def create_unet_model(classes=CLASSES, init_features=32):
    return torch.hub.load(

        'mateuszbuda/brain-segmentation-pytorch',
        'unet',

        in_channels=1,
        out_channels=len(CLASSES),
        init_features=init_features,
        pretrained=False

    )

unet_info = train_and_validate_simple_model(create_unet_model())
import torch.nn as nn

import torch.nn.functional as F

# FusionUNet code below is copied from FusionU-Net
repository2 def get_activation(activation_type):

    activation_type = activation_type.lower()

1URL: https://github.com/mateuszbuda/brain-segmentation-pytorch
2URL: https://github.com/Zongyi-Lee/FusionU-Net

    if hasattr(nn, activation_type):

        return getattr(nn,
            activation_type)() else:

        return nn.ReLU()

def _make_nConv(in_channels, out_channels, nb_Conv, activation='ReLU'):
    layers = []

    layers.append(ConvBatchNorm(in_channels, out_channels,
        activation)) for _ in range(nb_Conv - 1):

        layers.append(ConvBatchNorm(out_channels, out_channels,
            activation))

```

```

    return nn.Sequential(*layers)
class ConvBatchNorm(nn.Module):
    def __init__(self, in_channels, out_channels, activation='ReLU'):
        super(ConvBatchNorm, self).__init__()
        self.conv = nn.Conv2d(in_channels, out_channels,
                               kernel_size=3, padding=1)
        self.norm = nn.BatchNorm2d(out_channels)
        self.activation = get_activation(activation)

    def forward(self, x):
        out =
        self.conv(x) out =
        self.norm(out)

        return
self.activation(out) class
DownBlock(nn.Module):
    def __init__(self, in_channels, out_channels, nb_Conv,
activation='ReLU'):
        super(DownBlock, self).__init__()
        self.maxpool = nn.MaxPool2d(2)

        self.nConvs = _make_nConv(in_channels, out_channels, nb_Conv,
activation)

    def forward(self, x):
        out = self.maxpool(x)
        return
        self.nConvs(out)

class UpBlock_attention(nn.Module):
    def __init__(self, in_channels, activation='ReLU'):
        super().__init__()
out_channels, nb_Conv,

        self.up = nn.Upsample(scale_factor=2)

        self.coatt = CCA(F_g=in_channels//2, F_x=in_channels//2)
        self.nConvs = _make_nConv(in_channels, out_channels, nb_Conv,
activation)

```

```

def forward(self, x,
            skip_x): up = self.up(x)

            skip_x_att = self.coatt(g=up,
            x=skip_x) x = torch.cat([skip_x_att,
            up], dim=1) return self.nConvs(x)

class Flatten(nn.Module):
    def forward(self, x):

        return x.view(x.size(0), -
1) class CCA(nn.Module):

    def __init__(self, F_g, F_x):
        super().__init__()

        self.mlp_x =
            nn.Sequential(
                Flatten(),
                nn.Linear(F_x, F_x))

        self.mlp_g =
            nn.Sequential(
                Flatten(),
                nn.Linear(F_g, F_x))

        self.relu = nn.ReLU(inplace=True)
    def forward(self, g, x):

        avg_pool_x = F.avg_pool2d( x, (x.size(2), x.size(3)),
stride=(x.size(2), x.size(3)))

        channel_att_x = self.mlp_x(avg_pool_x)

        avg_pool_g = F.avg_pool2d( g, (g.size(2), g.size(3)),
stride=(g.size(2), g.size(3)))

        channel_att_g = self.mlp_g(avg_pool_g)
        channel_att_sum = (channel_att_x +
        channel_att_g)/2.0 scale =

torch.sigmoid(channel_att_sum).unsqueeze(2).unsqueeze(3).expand_as(x)
        x_after_channel = x * scale

        out =
        self.relu(x_after_channel)
        return out

class ECA(nn.Module):

```

```

def __init__(self, channel, k_size=3):
    super(ECA, self).__init__()
    self.avg_pool =
        nn.AdaptiveAvgPool2d(1)

    self.conv = nn.Conv1d(1, 1, kernel_size=k_size, padding=(k_size
- 1) // 2, bias=False)

    self.sigmoid =
nn.Sigmoid()
def
forward(self, x):

    y = self.avg_pool(x)

    y = self.conv(y.squeeze(-1).transpose(-1, -2)).transpose(-1,
- 2).unsqueeze(-1)

    y = self.sigmoid(y)
    return x *
        y.expand_as(x)

class FuseBlock(nn.Module):

    def __init__(self, base_channels):
        super(FuseBlock, self).__init__()
        self.norm1 =
            nn.BatchNorm2d(base_channels)

        self.norm2 = nn.BatchNorm2d(base_channels *
2)
        self.norm3 = nn.BatchNorm2d(base_channels
*
4)
        self.norm4 =
            nn.BatchNorm2d(base_channels * 8)

        self.up3 = UpFuseBlock(base_channels=base_channels * 4)
        self.up2 = UpFuseBlock(base_channels=base_channels * 2)
        self.up1 = UpFuseBlock(base_channels=base_channels)
        self.down1 = DownFuseBlock(base_channels=base_channels)
        self.down2 = DownFuseBlock(base_channels=base_channels *
2)
        self.down3 =
            DownFuseBlock(base_channels=base_channels * 4)

    def forward(self, fp1, fp2, fp3,
fp4):
        fp4 = self.norm4(fp4)

        fp3 =
            self.norm3(fp3)
        fp2 =
            self.norm2(fp2)
        fp1 =
            self.norm1(fp1)

```

```

    fp2 = self.down1(fp1,
    fp2) fp3 =
    self.down2(fp2, fp3) fp4
    = self.down3(fp3, fp4)
    fp3 = self.up3(fp3, fp4)
    fp2 = self.up2(fp2, fp3)
    fp1 = self.up1(fp1, fp2)
    return fp1, fp2, fp3,
    fp4

```

```
def
```

```

    reshape_downsample(x)
    : b, c, h, w =
    x.shape

    ret = torch.zeros_like(x)

    ret = ret.reshape(b, c * 4, h // 2, -
    1) ret[:, 0::4, :, :] = x[:, :, 0::2,
    0::2]

    ret[:, 1::4, :, :] = x[:, :, 0::2, 1::2]
    ret[:, 2::4, :, :] = x[:, :, 1::2, 0::2]
    ret[:, 3::4, :, :] = x[:, :, 1::2, 1::2]

    return ret

```

```
def
```

```

    reshape_upsample(x
    ): b, c, h, w =
    x.shape

    assert c % 4 == 0, 'number of channels must be multiple of 4'
    ret = torch.zeros_like(x)

    ret = ret.reshape(b, c // 4, h * 2, w *
    2) ret[:, :, 0::2, 0::2] = x[:, 0::4, :,
    :]

    ret[:, :, 0::2, 1::2] = x[:, 1::4, :, :]
    ret[:, :, 1::2, 0::2] = x[:, 2::4, :, :]
    ret[:, :, 1::2, 1::2] = x[:, 3::4, :, :]

    return ret

```

```
class DownFuseBlock(nn.Module):
```

```

def __init__(self, base_channels, dropout_rate=0.1):
    super(DownFuseBlock, self).__init__
    () self.eca = ECA(base_channels *
2) self.down = reshape_downsample

    self.conv1 = nn.Conv2d(base_channels * 4, base_channels * 2, 3,
1, 1, groups=base_channels)

    self.norm1 = nn.BatchNorm2d(base_channels * 2)

    self.fuse_conv = ConvBatchNorm(base_channels * 2, base_channels
*
2)

    self.relu = nn.ReLU()
def forward(self, fp1,
fp2):

    down = self.down(fp1)
    down =
self.conv1(down)

    down = self.relu(self.norm1(down))

    fp2 = self.fuse_conv(fp2 *0.75 + down * 0.25) +
fp2
    fp2 = self.eca(fp2)

    return fp2

class UpFuseBlock(nn.Module):

    def __init__(self, base_channels, dropout_rate=0.1):

        super(UpFuseBlock, self).__init__
        () self.eca = ECA(base_channels)
        self.up = reshape_upsample

        self.conv1 = nn.Conv2d(base_channels // 2, base_channels,
kernel_size=3, stride=1, padding=1, groups=base_channels//2)

```