

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ХАРКІВСЬКИЙ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

Факультет _____ комп'ютерних наук _____
(повна назва)
Кафедра _____ програмної інженерії _____
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти другий (магістерський)

Дослідження методів сегментації зображень
комп'ютерної томографії для ідентифікації людини
(тема)

Виконав:

Студент 2 курсу групи ІІЗМ-22-2

Новіков Р.К.

(прізвище, ініціали)

Спеціальність 121 – Інженерія програмного забезпечення

(код і повна назва спеціальності)

Тип програми _____

освітньо-наукова

(освітньо-професійна або освітньо-наукова)

Керівник _____

проф. Білоус Н. В.

(посада, прізвище, ініціали)

Допускається до захисту
Зав. Кафедри _____

(підпис)

Дудар З. В.

(прізвище, ініціали)

2024 р.

Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук

Кафедра Програмної інженерії

Рівень вищої освіти другий (магістерський)

Спеціальність 121 - Інженерія програмного забезпечення

(код і повна назва)

Тип програми освітньо-наукова програма

Освітня програма Інженерія програмного забезпечення

(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____

(підпис)

« _____ » _____ 2024 р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові Новікову Руслану Костянтиновичу

(прізвище, ім'я, по батькові)

1. Тема роботи «Дослідження методів сегментації зображень комп'ютерної томографії для ідентифікації людини»

Затверджена наказом по університету від 29.03.2024р. № 250 Ст.

2. Термін подання студентом роботи до екзаменаційної комісії 20.06.2024

3. Вихідні дані до роботи сегментація зображень, комп'ютерний зір, електронні ресурси за обраною тематикою, пояснювальна записка

4. Перелік питань, що потрібно опрацювати в роботі мета роботи, аналіз предметної області, постановка задачі, дослідження методів сегментації зображень, дослідження можливостей використання для ідентифікації людини, аналіз та розмітка знімків комп'ютерної томографії мозку.

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Отримання завдання	01.04.2024	Виконано
1	Аналіз предметної галузі та постановка задачі	02.04– 11.03	Виконано
2	Пошук найкращих методів сегментації	11.04 – 20.04	Виконано
3	Програмна реалізація U-Net	20.04 – 29.04	Виконано
4	Програмна реалізація SegNet	29.04 – 5.05	Виконано
5	Програмна реалізація DeepLab	05.05– 8.05	Виконано
6	Пошук інструменту для порівняння	08.05– 15.05	Виконано
7	Аналіз результатів експериментальних досліджень	15.05– 20.05	Виконано
8	Підготовка пояснювальної записки	20.05– 03.06	Виконано
9	Підготовка презентації та доповіді	03.06– 10.06.	Виконано
10	Перевірка на плагіат	13.06	Виконано
11	Нормоконтроль	14.06	Виконано
12	Рецензування	17.06	Виконано
13	Попередній захист	20.06	Виконано
14	Занесення диплома в електронний архів	22.06	Виконано
15	Допуск до захисту у зав. кафедри	23.06	Виконано

Дата видачі завдання 01 квітня 2024 р.

Студент _____
(підпис)

_____ Новіков Р. К.

Керівник кваліфікаційної роботи _____

_____ проф. Білоус Н. В.

(підпис)

(посада, прізвище, ініціали)

РЕФЕРАТ/ABSTRACT

Робота містить: 53 стор., 27 рис., 1 табл., 6 формул, 20 джерел.

КОМП'ЮТЕРНИЙ ЗІР, СЕГМЕНТАЦІЯ ЗОБРАЖЕНЬ, НЕЙРОННІ МЕРЕЖІ, КОМП'ЮТЕРНА ТОМОГРАФІЯ, ІДЕНТИФІКАЦІЯ.

Об'єкт дослідження даної роботи — зображення комп'ютерної томографії мозку, які відображають внутрішню структуру головного мозку людини. Предмет дослідження — методи сегментації цих зображень, тобто техніки, які дозволяють розподілити зображення на окремі сегменти або області з метою визначення їхніх характеристик та відмінностей.

Метою даної роботи є аналіз існуючих методів сегментації зображень комп'ютерної томографії мозку, визначення їхніх переваг та недоліків, а також знаходження інструменту для порівняння сегментацій.

Результатом цієї роботи стане створення ефективного інструменту для сегментації та подальшої ідентифікації особи на основі порівняння зображень комп'ютерної томографії мозку. Ця система надасть можливість об'єктивно оцінювати ступінь схожості між різними видами сегментації та виявляти ключові аспекти для точної ідентифікації людини.

COMPUTER VISION, IMAGE SEGMENTATION, NEURAL NETWORKS, COMPUTER TOMOGRAPHY, IDENTIFICATION.

The object of this research is the images obtained through computer tomography of the brain, depicting the internal structure of the human brain. The subject of the study is the segmentation methods applied to these images, referring to techniques that enable the division of an image into distinct segments or regions to analyze their characteristics and differences.

The aim of this work is to analyze existing methods of image segmentation in computer tomography brain scans, identify their advantages and drawbacks, and develop a system for comparing segmentations.

The outcome of this work will be the creation of an efficient tool for segmentation

and subsequent identification of individuals based on computer tomography brain images. This system will provide the ability to objectively assess the similarity between different segmentations and identify key aspects for accurate person identification.

Я, Новіков Руслан Костянтинович, студент гр. ПЗМ-22-2, здобувач вищої освіти на другому (магістерському) рівні кафедри «Програмна інженерія», заявляю: моя кваліфікаційна робота на тему «Дослідження методів сегментації зображень комп'ютерної томографії для ідентифікації людини», що буде представлена в екзаменаційну комісію для публічного захисту, виконана самостійно, в ній не містяться елементи плагіату і вона може бути опублікована в електронному архіві відкритого доступу EIAr KhNURE. Усі запозичення з друкованих та електронних джерел мають відповідні посилання.

Я ознайомлений з діючим положенням «Про протидію академічному плагіату в ХНУРЕ», згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування дисциплінарних заходів.

ЗМІСТ

1	Аналіз предметної області.....	8
1.1	Опис предметної області	8
1.2	Огляд сучасних рішень.....	9
1.3	Ручна сегментація	10
1.4	Сегментація на основі інтенсивності	10
1.5	Машинне навчання	12
1.6	Огляд додаткової літератури	15
2	Постановка задачі та опис прийнятих проектних рішень.....	20
2.1	Завдання дослідження.....	20
2.2	Постановка задачі.....	21
2.3	Вибір технологій	23
2.4	Бібліотеки та інструменти.....	23
3	Опис програмної реалізації	25
3.1.	Реалізація U-Net	25
3.2.	Реалізація SegNet.....	34
3.3.	Реалізація DeepLab.....	38
4	Опис експериментальних досліджень.....	41
	Висновки	49
	Перелік джерел посилання	50
	Додаток А Перелік джерел посилання за науковими напрямками керівника та науковців кафедри програмної інженерії	Помилка! Закладку не визначено.
	Додаток Б Звіт результатів перевірки на унікальність тексту в базі ХНУРЕ	Помилка! Закладку не визначено.
	Додаток В Слайди презентації.....	Помилка! Закладку не визначено.
	Додаток Г Апробація результатів роботи.....	Помилка! Закладку не визначено.
	Додаток Д Експертний висновок результатів перевірки кваліфікаційної роботи на відповідність оформлення вимогам ДСТУ 3008:2015	Помилка! Закладку не визначено.

ВСТУП

Комп'ютерний зір, зокрема область обробки зображень, є однією з ключових технологій, яка не тільки знаходить широке застосування у відеоспостереженні, медицині та інших сферах, але й має потенціал вирішення важливих суспільних завдань. Однією з таких завдань є ідентифікація людини за допомогою обробки знімків комп'ютерної томографії мозку.

Метою даної роботи є проведення комплексного аналізу сучасних методів сегментації зображень медичних томограм та їх застосування для ідентифікації особи. Об'єктом дослідження є знімки комп'ютерної томографії мозку, предметом – методи сегментації та їх вплив на ефективність ідентифікації особи.

В контексті сучасних безпекових реалій, де зростає необхідність впізнання людини, яку неможливо чітко ідентифікувати візуально чи за документами, розробка системи ідентифікації на основі комп'ютерної томографії мозку стає вкрай актуальною. Дана робота спрямована на розробку системи, яка, шляхом аналізу та порівняння сегментованих зображень, допоможе в розв'язанні цієї проблеми, надаючи ефективний та точний інструмент для ідентифікації особи на основі медичних зображень.

Важливість даного дослідження полягає в його потенційній здатності допомагати в ситуаціях, коли інші методи ідентифікації є недостатньо ефективними чи неможливими. Робота спрямована на вирішення конкретної проблеми, а саме, надання засобів для ідентифікації осіб, які залишаються невпізнаними за звичайними методами.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Опис предметної області

Сегментація зображення є одним із найважливіших завдань аналізу медичних зображень і часто є першим і найважливішим кроком у багатьох клінічних застосуваннях. У КТ-аналізі головного мозку сегментація зображення зазвичай використовується для вимірювання та візуалізації анатомічних структур мозку, аналізу змін головного мозку, окреслення патологічних ділянок, а також для ідентифікації людини, коли звичайні методи впізнання неможливі. За останні кілька десятиліть у літературі було розроблено та описано в різних методах сегментації різної точності та ступеня складності [1].

За останні кілька десятиліть швидкий розвиток технологій візуалізації мозку відкрив нові горизонти в аналізі та вивченні анатомії та функції мозку. За допомогою комп'ютерної томографії (КТ) було досягнуто величезного прогресу в доступі до травм головного мозку та дослідженні анатомії мозку. Прогрес у КТ-зображенні головного мозку також забезпечив отримання великої кількості даних все більш високої якості. Аналіз цих великих і складних наборів даних КТ став виснажливим і складним завданням для лікарів, яким доводиться вручну отримувати важливу інформацію. Цей ручний аналіз часто займає багато часу та схильний до помилок через різні проблеми людського фактору. Ці труднощі в аналізі даних КТ головного мозку вимагали винаходів комп'ютеризованих методів для покращення діагностики та тестування захворювань. Сьогодні комп'ютеризовані методи сегментації, реєстрації та візуалізації КТ-зображень широко використовуються для допомоги лікарям у якісній діагностиці [2].

Сегментація КТ головного мозку є важливою задачею в багатьох клінічних застосуваннях, оскільки вона впливає на результат усього аналізу. Це пояснюється тим, що різні етапи обробки залежать від точної сегментації анатомічних областей. Наприклад, сегментація КТ зазвичай використовується для вимірювання та візуалізації різних структур мозку, для окреслення уражень, для аналізу розвитку мозку, а також для втручань із зображенням і планування

хірургічних операцій. Це розмаїття програм обробки зображень призвело до розробки різноманітних методів сегментації різної точності та ступеня складності [3].

Необхідно розглянути найпопулярніші методи, які зазвичай використовуються для сегментації КТ головного мозку. Також треба підкреслити відмінності між ними та описати їхні можливості, переваги та обмеження. Щоб проаналізувати складність проблеми сегментації КТ мозку та вирішити її проблеми, спочатку представимо основні поняття сегментації зображення. Це включає визначення 2D зображень, опис проблеми сегментації зображення та його особливості.

1.2 Огляд сучасних рішень

Загалом сегментація КТ не є тривіальним завданням, оскільки отримані КТ-зображення недосконалі та часто спотворені шумом та іншими артефактами зображення. Різноманітність застосувань для обробки зображень призвела до розробки різних методів сегментації зображень [4–6].

Однак більшість методів сегментації, розроблених для одного класу зображень, можна легко застосувати/розширити до іншого класу зображень. Наприклад, теорія розрізів графів, хоча спочатку розроблена для бінарних зображень [7], може бути модифікована та використана для сегментації КТ тканини мозку. Крім того, неконтрольована нечітка кластеризація була успішно застосована в різних областях, таких як дистанційне зондування, геологія, а також медичні, біологічні та молекулярні зображення.

Методи сегментації із застосуванням до КТ головного мозку можна згрупувати наступним чином:

- ручна сегментація;
- методи, засновані на інтенсивності (включаючи порогове значення, зростання області, класифікацію та кластеризацію);
- поверхневі методи (включаючи активні контури та поверхні та багатofазні активні контури);

- методи машинного навчання;
- гібридні методи сегментації.

1.3 Ручна сегментація

Сегментація вручну означає процес, коли людина-оператор (наприклад, лікар-експерт) вручну сегментує та позначає зображення. Ця сегментація зазвичай виконується «зріз за зрізом» для 3D-об'ємних зображень. Ручний метод вважається найточнішим через складність точного та надійного окреслення структур на медичних зображеннях. Труднощі сегментації пов'язані з якістю зображення та артефактами.

Враховуючи вдосконалення, досягнуті за останні роки за допомогою інструментів візуалізації, сегментація вручну стала інтенсивним і трудомістким завданням. Навченому оператору зазвичай доводиться переглядати приблизно вісімдесят зображень розміром 512×512 , фрагмент за фрагментом, щоб виділити контури цільових структур. Це ручне сегментування не тільки виснажливе, але також особливо схильне до помилок, як оцінено різними дослідженнями варіабельності всередині або між операторами [8]. Крім того, результати сегментації вручну часто важко і навіть неможливо відтворити, оскільки навіть досвідчені оператори демонструють значну мінливість щодо свого попереднього окреслення.

1.4 Сегментація на основі інтенсивності

Методи сегментації на основі інтенсивності класифікують окремі пікселі на основі їх інтенсивності. У випадку КТ головного мозку виділяють три основні класи тканин, на основі інтенсивності (див. рис. 1.1). Більш детальна класифікація неможлива, оскільки профілі інтенсивності більш детальних структур мозку збігаються. Навіть поділ трьох основних класів тканин на основі самої інтенсивності вимагає включення інструментів для роботи з артефактами в КТ, такими як неоднорідність інтенсивності, шум і частковий об'єм, а також перекриття інтенсивності мозкової та немозкової тканини (наприклад, шкіра

ГОЛОВИ МАЄ ТАКОЇ Ж ІНТЕНСИВНОСТІ, ЯК І ТКАНИНИ МОЗКУ).

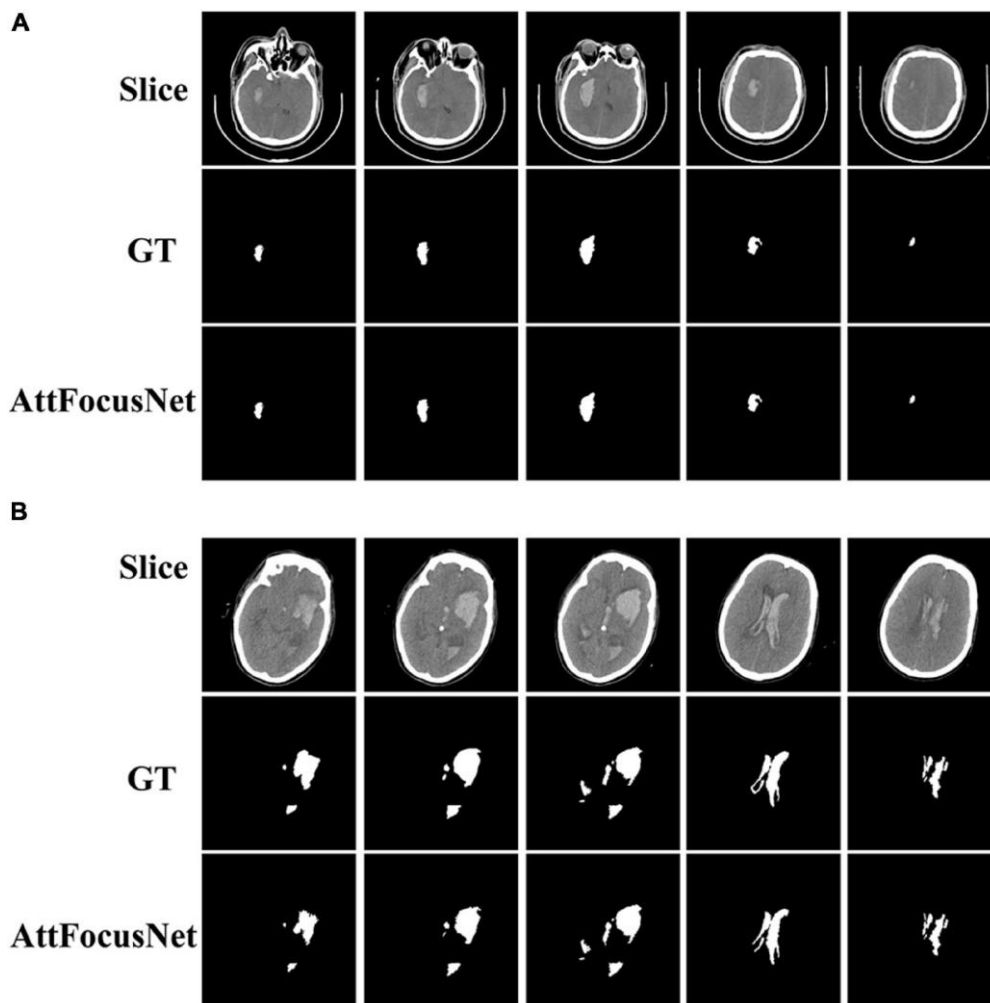


Рисунок 1.1 – Комп'ютерна томографія (КТ) та результати сегментації (за даними [9])

Для класифікації тканин доступні кілька методів, заснованих на інтенсивності. Найпоширенішим методом є використання гістограми інтенсивності всіх вокселів і підгонка функцій Гауса до розподілу. Вокселі (від англ. "voxel" - volume element) - це тривимірні пікселі, основні елементи об'єму в тривимірних зображеннях. Воксель представляє собою маленький кубічний елемент простору, подібний до пікселя в двовимірних зображеннях, але з додатковою третьою координатою для глибини. Таким чином, можна зробити висновок про ймовірність того, що задана інтенсивність відповідає даному типу тканини, і відповідно вокселям призначаються типи тканин. Додаткове включення інформації про сусідство допомагає надавати перевагу просторово

однорідним регіонам у кінцевій сегментації. Це може значно зменшити помилкову класифікацію через випадковий шум на зображенні.

1.5 Машинне навчання

Традиційне машинне навчання. Підходи кластеризації та класифікації, будучи традиційними методами машинного навчання, мотивуються багатовимірним простором ознак, який можна отримати з різних модальностей КТ. Класифікатори навчаються з використанням простору ознак, створеного шляхом комбінування ознак на основі різної інтенсивності та текстури, що представляють відомі класи. Потім виконується передбачення класу, до якого належить цільова структура, шляхом призначення мітки класу, яка найбільше схожа на простір функцій цільової структури. Водночас методи кластеризації — це неконтрольовані методи на основі пікселів, які поділяють непомічені зображення на кластери пікселів із подібними характеристиками без використання навчальних зображень.

Глибоке навчання. Останнім часом метод, заснований на глибокому навчанні, привернув великий інтерес дослідників завдяки його чудовій продуктивності та здатності автоматично отримувати адаптивні функції, які перевершують функції, створені вручну. Крім того, ці функції були вивчені в тенденції до збільшення складності функцій, що призводить до більш надійного вивчення функцій. Протягом останніх кількох років було розроблено більше досліджень із використанням поєднання методу глибокого навчання та нового методу сегментації пухлин головного мозку. У більшості досліджень використовувалися згорткові нейронні мережі через їхню ефективність у виявленні закономірностей на зображенні, зокрема на КТ-зображеннях, і повідомлялося про багатообіцяючі результати.

Методи на основі глибокого навчання з використанням 2D-зображень.

Для глибокого навчання з використанням 2D-зображень потрібні фрагменти зображень мозку або виділені 2D-патчі з 3D-зображень як вхідні дані для 2D-згорткового ядра. Було опубліковано кілька досліджень методу глибокого

навчання з використанням 2D-зображень. Серхіо Перейра та ін. представив каскадні шари, використовуючи малі ядра згортки 3×3 , щоб зменшити переобладнання [9]. Дослідження дозволило сегментувати зображення на чотири області, а саме (i) некроз, (ii) збільшення пухлини, (iii) набряк і (iv) нормальна тканина. Дві архітектури згорткової нейронної мережі (CNN) були навчені та використані в запропонованій роботі для вилучення карт ознак, якими були гліоми низького та високого ступеня злоякісності. Використання невеликих ядер призвело до глибшого проектування архітектури, що зменшило кількість вагових коефіцієнтів у мережі та значно вплинуло на переобладнання. Крім того, сегментація тканини була виконана як завдання на основі патчів у дослідженні, де локальна залежність міток під час класифікації пікселів ігнорувалась (див рис. 1.2).

Іншою помітною моделлю є U-Net, яка нагадує форму «U» під час візуалізації архітектури. Він складається з двох частин: підвищення та зменшення дискретизації, які також називаються скороченням шляху та розширенням відповідно. Важливість U-Net полягає в точності та швидкості, яких він досягає для сегментації зображення шляхом перепрофілювання тих самих карт функцій, які спочатку використовувалися для розширення вектора в повністю сегментоване вихідне зображення. Найпоширенішим використанням архітектури U-Net є сегментація зображень для медичної візуалізації [10].

Маска R-CNN є ефективним і простим методом для створення маски сегментації. Вихідні дані моделі також містять обмежувальні рамки для кожного екземпляра, що робить модель більш гнучкою для використання в різних програмах. Маска R-CNN — двоступенева згорткова нейронна мережа; перший етап — Мережа регіональних пропозицій (RPN), яка, як випливає з назви, пропонує цікаві регіони. Другий етап — це паралельна обробка на основі регіону пошуку обмежувальної рамки екземпляра, класифікації та бінарної маски.

Ще одна модель — це DeepLab, яка має кілька версій.

DeepLabV1: у цій моделі використовуються глибокі згорткові нейронні мережі (DCNN) і повнозв'язані умовні випадкові поля (CRF). У DCNN

використовується атральна звивина – це метод згортки, який дозволяє збільшити область сприйняття без збільшення кількості параметрів або втрати просторової роздільної здатності або розширена звивина, що дозволяє контролювати роздільну здатність для реакції функції. Вихідні дані DCNN передаються до CRF для покращення країв сегментації екземпляра.

DeepLabV2: Atrous Spatial Pyramid Pooling (ASPP) розроблено для сегментації об'єктів у різних масштабах. У цьому випадку CRF отримує вхідні дані від ASPP, і ефективність сегментації та визначення краю збільшується.

DeepLabV3: CRF було видалено, а ASPP оновлено, щоб мати функції зображення під час введення та нормалізації партії.

DeepLabV3+: архітектуру мережі було змінено на структуру кодера-декодера. Мережа кодувальника схожа на попередню версію, а мережа декодера — це прості операції підвищення дискретизації, які відновлюють вихідні межі.

SegNet. Архітектура SegNet використовує енкодер з блоками пулінгу та декодер для відновлення. У процесі кодування важливі фічі передаються за допомогою максимального пулінгу, а декодер реконструює зображення за допомогою отриманих індексів.

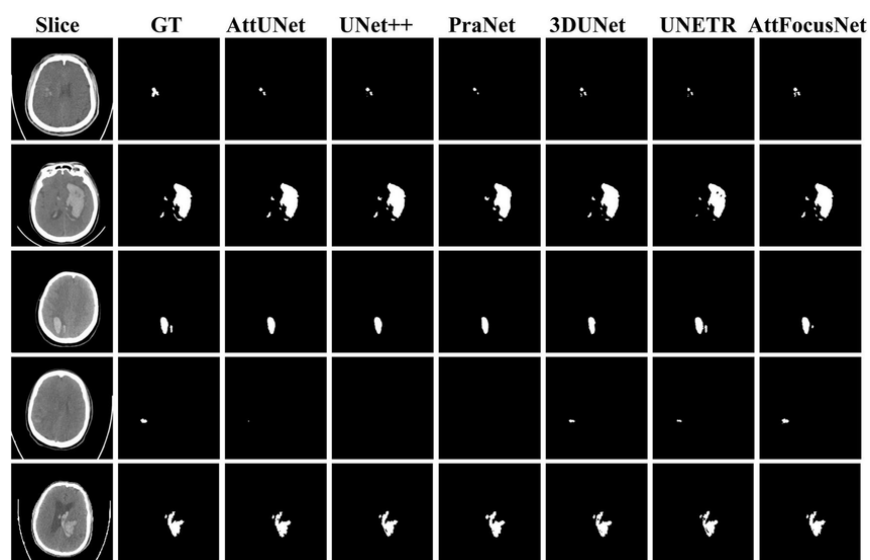


Рисунок 1.2 – Порівняння результатів сегментації різними методами (за даними [10])

1.6 Огляд додаткової літератури

Сучасні досягнення в галузі обробки медичних зображень відкривають широкі можливості для покращення діагностики та лікування захворювань. Одним із ключових напрямків у цьому відношенні є сегментація медичних зображень. Правильна сегментація зображень медичних досліджень, зокрема комп'ютерної томографії (КТ), є важливим етапом у виявленні та аналізі патологій, таких як пухлини, інфаркти, аномалії та інші.

У цьому розділі проводиться огляд наукових та патентних публікацій, що стосуються методів та технологій сегментації медичних зображень, зокрема використання нейромереж та інших алгоритмів машинного навчання. Проведений аналіз дозволить з'ясувати поточний стан розв'язання проблеми сегментації медичних зображень, ідентифікувати недоліки та потенційні напрямки подальших досліджень.

Для досягнення цієї мети вивчаються результати провідних досліджень, що виконані у галузі медичної обробки зображень, а також враховано інноваційні підходи та технології, які можуть бути застосовані для покращення сегментації медичних зображень. Цей огляд дозволить уточнити сучасний стан галузі та визначити перспективні напрямки подальших досліджень у сфері сегментації медичних зображень.

Стаття «Brain image segmentation of the corpus callosum by combining Bi-Directional Convolutional LSTM and U-Net using multi-slice CT and MRI» пропонує новий підхід до сегментації області інтересу на зображеннях мозку шляхом поєднання Bi-Directional Convolutional LSTM (BDC-LSTM) та U-Net. Дослідження проводиться з використанням мультишарових СТ та MRI зображень [11].

Automation: Стаття розробляє метод, який дозволяє автоматизовано виконувати сегментацію зображень мозку з високою точністю без необхідності втручання оператора або корекції.

Completeness: Дослідження включає ретельний аналіз проблеми сегментації області інтересу на зображеннях мозку та пропонує метод, який враховує різні аспекти проблеми, такі як багатосарові дані.

Accuracy: За результатами експериментів, запропонований метод показав високу точність в сегментації зображень мозку з даними про множинні церебральні інфаркти.

Методологія: Стаття використовує комбінацію підходів, включаючи BDC-LSTM та U-Net, для вирішення проблеми сегментації області інтересу на зображеннях мозку. Дослідження включає аналіз набору даних, використання певних архітектур та експериментальне порівняння результатів.

Fully Convolutional Neural Networks (FCNs): Ці неймережі стали дуже ефективними в сегментації зображень, включаючи біомедичні зображення. Однак, через різноманітність багатьох медичних зображень, пряме розширення цих методів сегментації зображень 2D на 3D може не дати задовільних результатів.

Bi-Directional Convolutional LSTM (BDC-LSTM): Це розширення конволюційних неймереж, яке дозволяє враховувати просторові залежності між шарами та ефективно використовувати кореляційну інформацію між двовимірними зрізами. BDC-LSTM використовується для кореляції інформації між кадрами в 2D сегментації зображень.

U-Net: Ця неймережа використовується для сегментації зображень, зокрема для вирішення проблеми зображень біомедичних даних. U-Net є потужним інструментом для сегментації зображень мозку та інших медичних зображень.

Convolutional Neural Networks (CNNs): CNN використовується для екстракції просторових та часових ознак з медичних даних. У поєднанні з LSTM, CNN може ефективно використовуватися для передбачення різних параметрів та властивостей зображень.

FCNs: Плюси - висока точність сегментації, можливість роботи з більшими об'ємами даних. Мінуси - неефективність в роботі з 3D зображеннями та анізотропними даними.

BDC-LSTM: Плюси - здатність враховувати просторові залежності, кореляційну інформацію між зрізами. Мінуси - висока обчислювальна складність.

U-Net: Плюси - ефективність в сегментації медичних зображень, включаючи зображення мозку. Мінуси - можливість перенавчання при недостатньому обсязі

даних.

CNNs: Плюси - висока ефективність у виявленні просторових та часових ознак у медичних зображеннях. Мінуси - потреба у великій кількості даних для тренування, можливість перенавчання.

В цілому, використані методи та нейромережі є ефективними для сегментації медичних зображень, але потребують ретельного налаштування параметрів та обробки даних, для того, щоб їх можна було використовувати для порівняння двох знімків з метою ідентифікації людини.

Інше дослідження показує наступне. Глибоке навчання стало перспективним підходом до вирішення проблеми корекції поглинання PET зображень (метод образного дослідження, який використовує радіоактивні речовини для вивчення та візуалізації функцій та активності органів та тканин у тілі) мозку на основі сегментації MR, перевершаючи традиційні методи машинного навчання. Наприклад, методи, керовані глибоким навчанням, показують кращу ефективність порівняно з методами машинного навчання, заснованими на Dixon-моделях (клас методів обробки зображень у медичній діагностиці, які використовуються для розділення сигналів від різних тканин у тілі на магнітно-резонансних (MRI) зображеннях), для сегментації м'яких тканин та повітряних просторів.

Повністю згорнуті нейронні мережі (CNN) використовуються для генерації сегментованих зображень MR, використовуючи сегментовані СТ зображення як основу для створення псевдо СТ зображень [12].

Набір даних для образного дослідження складається з 50 пацієнтів, які пройшли СТ та MR зображення після надання інформованої згоди.

Розмір об'єму кожного пацієнта становить $256 \times 256 \times 48$, де 48 - це кількість вибраних зрізів на пацієнта. Велика кількість пікселів створює обмеження для пам'яті графічного процесора (GPU) та впливає на вибір архітектури мережі, такого як вибір кількості фільтрів на кожному згортковому шарі. Для подолання цієї ситуації кожний об'єм розділяється на перекриття фрагментів для зменшення розміру введення та збільшення обсягу навчальних наборів.

Вплив кожного типу блоків виділення вивчається за допомогою розміщення блока

SE у тому ж місці у мережі та порівняння метрик оцінки, які генеруються з матриці плутанини.

У цьому дослідженні представлено вдосконалений підхід для сегментації мозку на три класи тканин за допомогою повністю згорнутої нейронної мережі з виділеними блоками. Ці блоки переналаштовують ознаки, захоплюючи корисні та ті, що розрізняються в каналному та просторовому аспектах в послідовній структурі. Запропонований підхід демонструє ефективність у покращенні сегментації кістки, мінімізуючи складність моделі та потребу в обчислювальних ресурсах. Результати також свідчать про успішність запропонованого методу.

Також актуальне рішення для задачі сегментації представлено в роботі «Numerical Error Analysis For Configurable Cell Segmentation Problem», в якій пропонується розробка гнучкого алгоритму розпізнавання, який можна налаштувати для сегментації ділянок на різних типах зображень [13].

В дослідженні «Configurable Cell Segmentation Solution Using Hough Circles Transform and Watershed Algorithm» пропонується вододільний алгоритм для сегментації клітин та перетворення Хью для кластеризації клітин, що перетинаються. Мета досягається шляхом надання можливості налаштувати алгоритм під властивості конкретного набору даних. Відповідно, результати цієї роботи можуть бути використані для майбутніх досліджень з новими завданнями сегментації клітин. Новизна даної роботи полягає в багатогранності та в можливість оптимізації під різний діапазон завдання сегментації, а також для сегментації зображень комп'ютерної томографії мозку, що актуально для даного дослідження [14].

В роботі «Використання методів глибинного навчання для сегментації пухлин головного мозку на МРТ-зображеннях» представлена каскадна модель трьох згорткових нейронних мереж для обробки мультимодальних МРТ-зображень мозку та розпізнавання і сегментації пухлин головного мозку. Для збільшення точності та достовірності діагностики, класифікації та подальшого лікування пухлини головного мозку автори використовують сучасні технології машинного зору та машинного навчання [15]. Це дослідження описує програмну реалізацію

моделі сегментації пухлин головного мозку, побудованої з використанням фреймворків NiftyNet та TensorFlow, а також бібліотеки PyWavelets на мові програмування Python. Розроблена модель успішно виконує задачу сегментації пухлин на зображеннях МРТ, демонструючи високу точність та ефективність. Актуальність даної роботи полягає в тому, що сегментація пухлин головного мозку є важливим завданням в медичній діагностиці та плануванні лікування. Автоматизовані методи сегментації дозволяють прискорити та покращити процес діагностики. Наробки цієї роботи можна використати в подальшому.

Інше дослідження під назвою «» метою роботи якого є розробка методу ідентифікації та порівняння поз і вправ, виконуваних людиною, який матиме низьку чутливість до помилок у даних. Цей метод використовує їх формальні описи у вигляді кон'юнкцій логічних висловлювань і має працювати незалежно від кута зйомки, під яким було зроблено відео, та пропорцій людини на ньому. Кожне висловлювання описує положення суглобів відносно один одного вздовж однієї з осей. Координати суглобів коригуються з урахуванням довжини кісток, які їх з'єднують, що усуває необхідність обробки викидів та покращує точність визначення положення суглобів [16]. Видалення помилок з даних здійснюється методом усереднення графа вздовж кожної осі на кожному кроці. Для цього послідовні точки групуються так, щоб різниця між максимумом і мінімумом не перевищувала похибку. Метод 3D точок для виявлення людини на відеопотоці, який знаходить контури людини можна використати для пошуку ключових точок на комп'ютерній томографії мозку людини.

2 ПОСТАНОВКА ЗАДАЧІ ТА ОПИС ПРИЙНЯТИХ ПРОЕКТНИХ РІШЕНЬ

2.1 Завдання дослідження

У рамках магістерського дослідження визначено ряд завдань, які будуть розглядатись для досягнення цілей даного дослідження:

1. Аналіз існуючих методів сегментації зображень комп'ютерної томографії: Провести докладний аналіз літератури, що стосується методів сегментації зображень комп'ютерної томографії, з особливим акцентом на їх застосування для ідентифікації людини.

2. Визначення ключових аспектів та обмежень сучасних методів сегментації в комп'ютерній томографії: Розглянути ключові характеристики та обмеження існуючих моделей сегментації, зосереджуючись на їхньому потенціалі для ідентифікації особи.

3. Аналіз критеріїв ефективності методів сегментації в контексті ідентифікації: Переглянути та проаналізувати існуючі критерії, які використовуються для оцінки ефективності методів сегментації в контексті їхнього застосування до задачі ідентифікації особи.

4. Вивчення та вибір набору даних для навчання моделей: Ретельно дослідити доступні набори даних і визначити той, який буде найбільш підходящим для навчання моделей сегментації в контексті ідентифікації особи.

5. Розробка та порівняння різних моделей сегментації: Реалізувати та порівняти різні обрані моделі сегментації для вирішення завдання ідентифікації особи.

6. Тестування моделей на реальних зображеннях з комп'ютерної томографії: Протестувати розроблені моделі на реальних зображеннях комп'ютерної томографії з метою визначення їхньої точності та ефективності в контексті ідентифікації особи.

7. Аналіз впливу факторів на продуктивність моделей: Проаналізувати вплив різних факторів, таких як розмір набору даних та особливості зображень комп'ютерної томографії, на продуктивність розроблених моделей.

8. Розробка рекомендацій для оптимізації та поліпшення моделей: Розробити рекомендації, спрямовані на вдосконалення моделей сегментації з метою поліпшення точності та оптимізації часу обробки для ідентифікації особи.

2.2 Постановка задачі

Вибір та оцінка методів та моделей сегментації зображень є критичним завданням, оскільки від цього залежить точність та швидкість обробки. У цьому розділі ми розглянуто ключові критерії ефективності, які визначають якість та придатність методів сегментації.

1. Точність сегментації:

Одним із головних критеріїв є точність визначення границь об'єктів та їхніх областей. Точність безпосередньо впливає на правильність ідентифікації та виділення цікавих областей на зображенні.

2. Швидкодія та ефективність:

У великих застосуваннях, таких як обробка великих обсягів медичних даних, швидкість обробки має велике значення. Методи та моделі повинні забезпечувати високу швидкість, не пропускаючи при цьому важливі деталі.

3. Стійкість до шуму:

Наявність шуму та коливання освітлення може впливати на якість сегментації. Методи повинні бути стійкими до цих артефактів, забезпечуючи стабільність результатів при різних умовах.

4. Масштабованість:

Деякі застосування можуть вимагати обробки зображень різного масштабу. Методи та моделі повинні бути масштабованими та здатними працювати як з маленькими, так і з великими зображеннями без втрати якості.

7. Гнучкість та універсальність:

З урахуванням широкого спектру застосувань сегментації, методи повинні бути гнучкими та універсальними. Вони повинні забезпечувати можливість використання в різних галузях та для різних видів зображень.

В рамках дослідження використовувався достовірний та валідований

датасет, що включав у себе медичні знімки КТ мозку. Цей датасет був ретельно розмічений та перевірений кваліфікованими спеціалістами в галузі медицини. Забезпечена якість та достовірність даних у датасеті гарантує надійність результатів та об'єктивність подальших висновків дослідження.

Важливо відзначити, що основною метою даного дослідження є робота з вже розміченим та перевіреним датасетом (див. приклад на рис. 1.3). Це дозволяє уникнути можливих неточностей у процесі розмітки та забезпечує правдивість результатів. Використання передових методів сегментації на вже готовому датасеті сприяє точнішим та ефективним вивченням властивостей об'єктів на медичних зображеннях КТ мозку.



Рисунок 2.1 – Приклад КТ мозку з перевіреного датасету (рисунок надано медичним працівником)

2.3 Вибір технологій

Для дослідження методів сегментації зображень було обрано мову програмування Python. Ця мова є дуже популярною в області обробки зображень та машинного навчання завдяки своїй простоті, зручності та великій кількості бібліотек, що підтримують ці завдання. Використання Python дозволяє швидко розробляти та тестувати різні моделі, а також легко інтегрувати їх з іншими інструментами для аналізу даних.

Вибір зупинено на моделях нейромереж U-Net, SegNet та DeepLab для порівняння методів сегментації зображень. Ці моделі були обрані через їхню високу популярність і ефективність у розв'язанні задач сегментації зображень. Вони забезпечують хорошу точність і мають численні реалізації у відкритих бібліотеках, що полегшує їх використання та модифікацію.

2.4 Бібліотеки та інструменти

- TensorFlow: Відкрита бібліотека для машинного навчання, розроблена компанією Google. Використовується для розробки та тренування моделей глибокого навчання. TensorFlow підтримує різні архітектури нейронних мереж, включаючи U-Net, SegNet та DeepLab.
- Keras: Високорівнева API для машинного навчання, що працює поверх TensorFlow. Keras забезпечує простий і зрозумілий інтерфейс для створення та тренування нейронних мереж, що значно полегшує розробку моделей.
- OpenCV: Бібліотека для обробки зображень і комп'ютерного зору. Вона використовується для попередньої обробки зображень, таких як зміна розміру, нормалізація та аугментація зображень перед подачею їх до нейронної мережі.
- NumPy: Бібліотека для роботи з багатовимірними масивами і матрицями, а також для виконання різних математичних операцій. NumPy широко використовується для маніпуляції даними та їх підготовки перед обробкою в нейронних мережах.
- Matplotlib: Бібліотека для візуалізації даних. Вона використовується для

побудови графіків та візуалізації результатів сегментації зображень, що дозволяє наочно оцінити продуктивність моделей.

- PyCharm: Інтегроване середовище розробки (IDE) для Python. PyCharm забезпечує зручний інтерфейс для написання, тестування та налагодження коду, що значно спрощує процес розробки та експериментування з різними моделями сегментації.

Ці бібліотеки та інструменти були обрані через їхню високу ефективність, популярність у науковому середовищі та зручність використання. Вони забезпечують потужний набір функцій для розробки та оцінки моделей сегментації зображень, що є ключовим для успішного виконання даного дослідження.

3 ОПИС ПРОГРАМНОЇ РЕАЛІЗАЦІЇ

В даному розділі представлені ключові технічні особливості, архітектура та структура коду головних компонентів системи. Також описуються основні методи та функції, що застосовуються в дослідженні, та практичне використання зазначених бібліотек.

3.1. Реалізація U-Net

Загальний вигляд архітектури проекту для проведення дослідження представлено на рисунку 3.1.

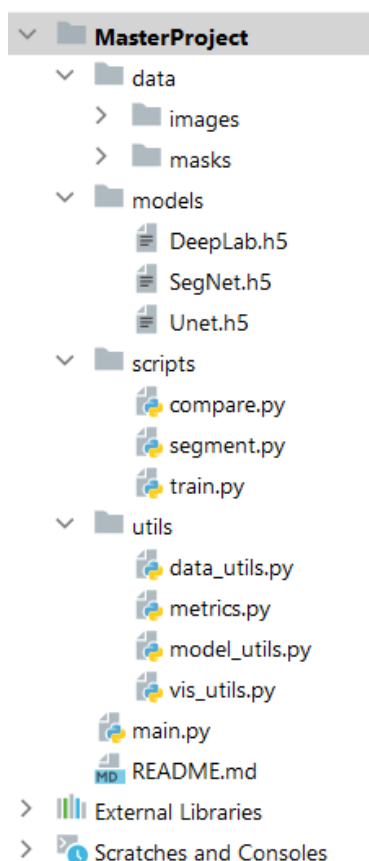


Рисунок 3.1 – Архітектура проекту (рисунок створено самостійно)

Одним із інструментів сегментації зображень, що розглядаються в даному дослідженні є архітектура U-Net — це широко використовувана архітектура глибокого навчання, яка вперше була представлена в статті «U-Net: згорткові мережі для сегментації біомедичних зображень». Основною метою цієї архітектури

було вирішення проблеми обмежених анотованих даних у галузі медицини. Ця мережа розроблена для ефективного використання меншого обсягу даних, зберігаючи швидкість і точність [17].

Архітектура U-Net унікальна тим, що складається з скорочувального шляху та розширювального шляху. Шлях скорочення містить шари кодувальника, які фіксують контекстну інформацію та зменшують просторову роздільну здатність вхідних даних, тоді як розширений шлях містить шари декодера, які декодують закодовані дані та використовують інформацію зі шляху скорочення через з'єднання пропуску для створення карти сегментації.

Контрактний шлях в U-Net відповідає за визначення відповідних функцій у вхідному зображенні. Рівні кодера виконують згорткові операції, які зменшують просторову роздільну здатність карт функцій, одночасно збільшуючи їхню глибину, таким чином фіксуючи все більш абстрактні представлення вхідних даних. Цей шлях скорочення подібний до шарів прямого зв'язку в інших згорткових нейронних мережах. З іншого боку, розширений шлях працює над декодуванням закодованих даних і визначенням місцезнаходження функцій, зберігаючи просторову роздільну здатність вхідних даних. Рівні декодера в розширеному шляху підвищують дискретизацію карт функцій, одночасно виконуючи згорткові операції. Пропускання з'єднань зі звужувального шляху допомагає зберегти просторову інформацію, втрачену на звужувальному шляху, що допомагає рівням декодера точніше визначати об'єкти.

На схематичному рисунку 3.2 показано, як мережа U-Net перетворює вхідне зображення у відтінках сірого розміром $572 \times 572 \times 1$ у двійкову сегментовану вихідну карту розміром $388 \times 388 \times 2$. Можна помітити, що вихідний розмір менший за вхідний, оскільки не використовується відступи. Однак, якщо ми використовуємо відступи, ми можемо зберегти розмір введення. Під час згортання вхідне зображення поступово зменшується у висоту та ширину, але збільшується у кількості каналів. Це збільшення каналів дозволяє мережі охоплювати функції високого рівня, коли вона просувається по шляху. У вузькому місці виконується остання операція згортання для створення карти об'єктів у формі $30 \times 30 \times 1024$.

Розширений шлях потім бере карту функцій із вузького місця та перетворює її назад на зображення того самого розміру, що й початковий вхід. Це робиться за допомогою шарів підвищення дискретизації, які збільшують просторову роздільну здатність карти функцій, одночасно зменшуючи кількість каналів. Пропуск з'єднань із шляху скорочення використовується, щоб допомогти рівням декодера знайти та уточнити особливості в зображенні. Нарешті, кожен піксель у вихідному зображенні представляє мітку, яка відповідає певному об'єкту або класу у вхідному зображенні. У цьому випадку вихідна карта є бінарною картою сегментації, де кожен піксель представляє область переднього плану або фону.

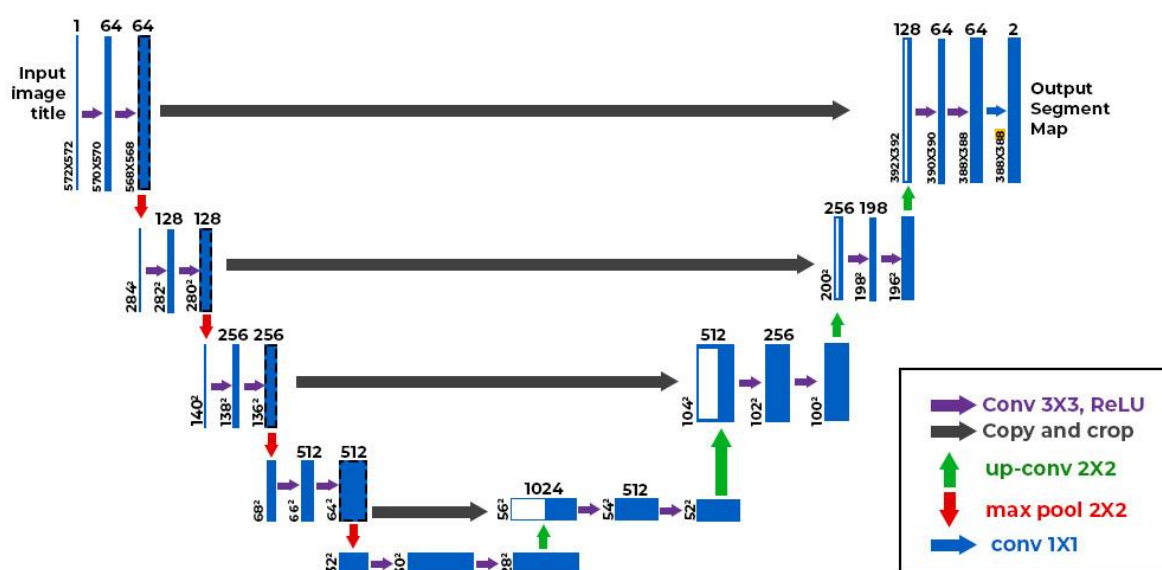


Рисунок 3.2 – Архітектура U-Net (за даними[17])

Програмна система що розробляється для проведення дослідження працюватиме із зображеннями розміром 512x512. Основний код роботи з моделлю представлено на рисунку 3.3. Спочатку імпортуються необхідні бібліотеки для роботи з даними, нейромережею і попередньої обробкою зображень. Функція `UNET_MODEL` визначає архітектуру моделі U-Net із заданими параметрами вхідного зображення. В архітектурі використовуються згорткові та пулінгові шари для отримання ознак та апсемплінгові шари для відновлення просторової роздільної здатності.

```

1  import numpy as np
2  import tensorflow as tf
3  from tensorflow.keras import layers, models, Input
4  from PIL import Image
5  import os
6  import cv2
7  from sklearn.model_selection import train_test_split
8
9  @tf.function
10 def unet_model(input_size=(512, 512, 3)):
11     inputs = Input(input_size)
12
13     conv1 = layers.Conv2D(64, 3, activation='relu', padding='same')(inputs)
14     conv1 = layers.Conv2D(64, 3, activation='relu', padding='same')(conv1)
15     pool1 = layers.MaxPooling2D(pool_size=(2, 2))(conv1)
16
17     conv2 = layers.Conv2D(128, 3, activation='relu', padding='same')(pool1)
18     conv2 = layers.Conv2D(128, 3, activation='relu', padding='same')(conv2)
19     pool2 = layers.MaxPooling2D(pool_size=(2, 2))(conv2)
20
21     conv3 = layers.Conv2D(256, 3, activation='relu', padding='same')(pool2)
22     conv3 = layers.Conv2D(256, 3, activation='relu', padding='same')(conv3)
23     pool3 = layers.MaxPooling2D(pool_size=(2, 2))(conv3)
24
25     conv4 = layers.Conv2D(512, 3, activation='relu', padding='same')(pool3)
26     conv4 = layers.Conv2D(512, 3, activation='relu', padding='same')(conv4)
27     drop4 = layers.Dropout(0.5)(conv4)
28     pool4 = layers.MaxPooling2D(pool_size=(2, 2))(drop4)
29
30     conv5 = layers.Conv2D(1024, 3, activation='relu', padding='same')(pool4)
31     conv5 = layers.Conv2D(1024, 3, activation='relu', padding='same')(conv5)
32     drop5 = layers.Dropout(0.5)(conv5)

```

Рисунок 3.3 – Функція моделі U-Net (рисунок створено самостійно)

Кожний рядок коду визначає шар нейромережі. Вхідний шар (inputs) представляє собою тензор (багатовимірний масив чисел, який є основною структурою даних у бібліотеках машинного навчання) із заданим розміром.

Згорткові шари (conv1, conv2, conv3, ...) виконують згортку з зображенням вхідного шару, використовуючи фільтри для вилучення ознак.

Пулінгові шари (pool1, pool2, pool3, ...) застосовуються після кожного блоку згорткових шарів, який зменшує розмірність простору ознак, зберігаючи найважливіші ознаки. Шар із Dropout (drop4, drop5): Шари з Dropout (Dropout) застосовуються для регуляризації моделі та запобігання перенавчанню.

Шар Upsampling (up6, ...) збільшує розмірність простору ознак з метою відновлення просторової інформації.

Об'єднання ознак (merge6, ...) – для поліпшення якості прогнозів об'єднуються ознаки з нижчих рівнів за допомогою операції конкатенації.

Вихідний шар (conv10) – останній згортковий шар застосовується для отримання остаточних передбачень на основі виділених ознак.

Далі необхідно завантажити дані – за це відповідає функція load_data, яка завантажує зображення та їх маски із зазначених директорій. Зображення завантажуються з використанням OpenCV, змінюються на вказаний розмір, а потім нормалізуються (див. рис. 3.4).

```

62 def load_data(image_folder, mask_folder, image_size=(512, 512)):
63     images = []
64     masks = []
65
66     image_files = os.listdir(image_folder)
67     for image_file in image_files:
68         image_path = os.path.join(image_folder, image_file)
69         mask_path = os.path.join(mask_folder, image_file)
70
71         image = Image.open(image_path)
72         image = image.convert("RGB")
73         image = np.array(image)
74
75         mask = Image.open(mask_path)
76         mask = np.array(mask)
77
78         image = cv2.resize(image, image_size)
79         mask = cv2.resize(mask, image_size)
80
81         images.append(image)
82         masks.append(mask)
83
84     images = np.array(images)
85     masks = np.array(masks)
86
87     images = images / 255.0
88     masks = masks / 255.0
89     masks = np.expand_dims(masks, axis=-1)
90
91     return images, masks

```

Рисунок 3.4 – Функція завантаження даних (рисунок створено самостійно)

Скрипт `train.py` створює модель нейромережі `unet` для навчання на валідованому наборі даних, що представляє собою набір зображень КТ мозку людини та розмічену маску зображення. Спочатку виконується накладання маски у `json`-форматі на зображення `png`, отримуємо масив сегментованих зображень для навчання моделі.

В наступній частині коду визначається точка входу для навчання моделі. Завантажуються дані, розділяються на навчальну та тестову вибірки, створюється модель, навчається на даних і зберігається для подальшого використання (див. рис. 3.5).

```

94 ▶ if __name__ == "__main__":
95     image_folder = 'images'
96     mask_folder = 'masks'
97     X, Y = load_data(image_folder, mask_folder)
98     X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2, random_state=42)
99
100    model = unet_model()
101    model.summary()
102
103    model.fit(X_train, Y_train, batch_size=8, epochs=50, verbose=1, validation_data=(X_test, Y_test))
104
105    model.save('unet_model.h5')
106
107    test_image_path = 'path/to/new_image.png'
108    test_image = Image.open(test_image_path)
109    test_image = test_image.resize((512, 512))
110    test_image = np.array(test_image)
111    test_image = np.expand_dims(test_image, axis=0) / 255.0
112
113    model = tf.keras.models.load_model('unet_model.h5')
114    prediction = model.predict(test_image)
115    prediction = (prediction > 0.5).astype(np.uint8)
116
117    # Save the predicted mask
118    predicted_mask = prediction[0, :, :, 0] * 255
119    predicted_mask = Image.fromarray(predicted_mask)
120    predicted_mask.save('predicted_mask.png')

```

Рисунок 3.5 – Функція навчання моделі (рисунок створено самостійно)

Під час навчання моделі виводяться такі показники (див рис. 3.6):

- епохи навчання (`epoch`) – один повний прохід по всьому набору даних для навчання, номер поточної епохи та загальна їхня кількість;
- втрати (`loss`) на навчальному та валідаційному наборі даних після

кожної епохи;

- точність на навчальному та валідаційному наборі даних після кожної епохи.

```

conv2d_19 (Conv2D) | (None, 512, 512, 64) | 32,832 | up_sampling2d_3[0][0]
concatenate_3 (Concatenate) | (None, 512, 512, 128) | 0 | conv2d_1[0][0], conv2d_19[0][0]
conv2d_20 (Conv2D) | (None, 512, 512, 64) | 73,792 | concatenate_3[0][0]
conv2d_21 (Conv2D) | (None, 512, 512, 64) | 36,928 | conv2d_20[0][0]
conv2d_22 (Conv2D) | (None, 512, 512, 2) | 1,154 | conv2d_21[0][0]
conv2d_23 (Conv2D) | (None, 512, 512, 1) | 3 | conv2d_22[0][0]

Total params: 31,032,837 (118.38 MB)
Trainable params: 31,032,837 (118.38 MB)
Non-trainable params: 0 (0.00 B)
Epoch 1/50
5/5 ██████████ 375s 73s/step - accuracy: 0.6514 - loss: 0.6951 - val_accuracy: 0.9724 - val_loss:
Epoch 2/50
5/5 ██████████ 348s 69s/step - accuracy: 0.9752 - loss: 0.6929 - val_accuracy: 0.9729 - val_loss:
Epoch 3/50
4/5 ██████████ 1:27 88s/step - accuracy: 0.9764 - loss: 0.6926

```

Рисунок 3.6 – Процес навчання моделі (рисунок створено самостійно)

Наступним етапом є отримання файлу .h5. Файл із розширенням .h5 (HDF5, Hierarchical Data Format version 5) використовується для збереження даних в ієрархічній структурі. У контексті машинного навчання і, зокрема, TensorFlow/Keras, файл .h5 використовується для збереження навченої моделі нейронної мережі. Збереження моделі у форматі .h5 має кілька переваг, включаючи можливість зберегти архітектуру моделі, її ваги та конфігурацію, а також параметри навчання. Зберігається структура моделі, тобто інформація про шари, їх типи та зв'язки між ними, це дозволяє відновити модель без необхідності створювати знову її архітектуру з коду. Ваги (параметри) моделі, які були навчені, зберігаються у файлі, це дозволяє використовувати навчену модель без необхідності наново проходити процес навчання. Якщо модель була навчена, параметри навчання, такі як оптимізатор, функція втрат та метрики також зберігаються, що дозволяє продовжити навчання моделі з того місця, де його було

зупинено. Зберігається поточний стан оптимізатора, що дає змогу продовжити навчання моделі без втрати інформації про те, як змінювалися ваги в попередніх епохах.

Наступний крок – тестова вибірка даних. Візьмемо тестове зображення (рис. 3.7) та дамо на вхід нейромережі.

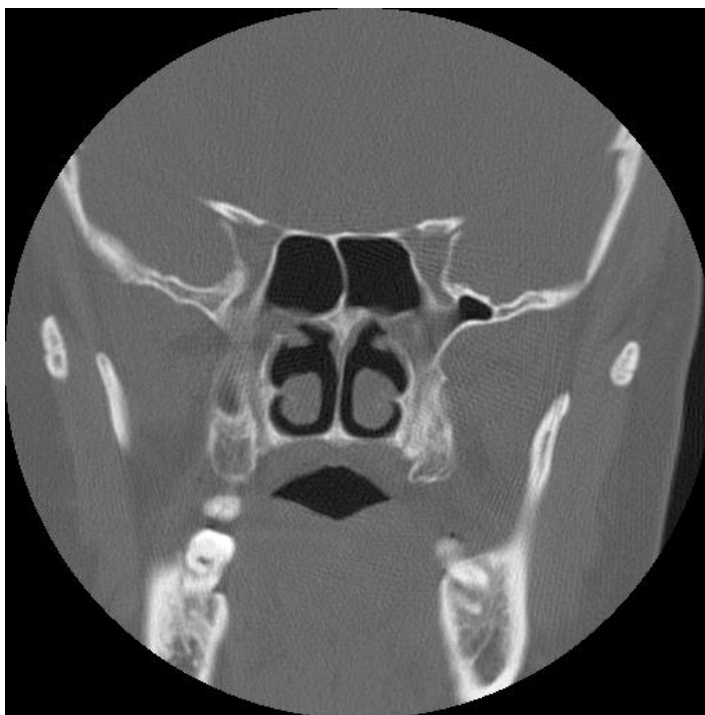


Рисунок 3.7 – Тестове зображення КТ мозку людини (рисунок надано медичним працівником)

Після цього починається виконання скрипта для передбачення сегментаційних масок на нових зображеннях, використовуючи раніше натреновану модель U-Net з великою кількістю параметрів. Основна мета полягає у застосуванні сегментаційної моделі до нових зображень і збереження отриманих передбачених масок (див. рис. 3.8).

Model: "functional_1"

Layer (type)	Output Shape	Param #	Connected to
input_layer (InputLayer)	(None, 512, 512, 3)	0	-
conv2d (Conv2D)	(None, 512, 512, 64)	1,792	input_layer[0][0]
conv2d_1 (Conv2D)	(None, 512, 512, 64)	36,928	conv2d[0][0]
max_pooling2d (MaxPooling2D)	(None, 256, 256, 64)	0	conv2d_1[0][0]
conv2d_2 (Conv2D)	(None, 256, 256, 128)	73,856	max_pooling2d[0][0]
conv2d_3 (Conv2D)	(None, 256, 256, 128)	147,584	conv2d_2[0][0]
max_pooling2d_1 (MaxPooling2D)	(None, 128, 128, 128)	0	conv2d_3[0][0]
conv2d_4 (Conv2D)	(None, 128, 128, 256)	295,168	max_pooling2d_1[0][0]
conv2d_5 (Conv2D)	(None, 128, 128, 256)	590,080	conv2d_4[0][0]
max_pooling2d_2 (MaxPooling2D)	(None, 64, 64, 256)	0	conv2d_5[0][0]
conv2d_6 (Conv2D)	(None, 64, 64, 512)	1,180,160	max_pooling2d_2[0][0]
conv2d_7 (Conv2D)	(None, 64, 64, 512)	2,359,808	conv2d_6[0][0]
dropout (Dropout)	(None, 64, 64, 512)	0	conv2d_7[0][0]
max_pooling2d_3 (MaxPooling2D)	(None, 32, 32, 512)	0	dropout[0][0]
conv2d_8 (Conv2D)	(None, 32, 32, 1024)	4,719,616	max_pooling2d_3[0][0]

Рисунок 3.8 – Параметри моделі нейромережі (рисунок створено самостійно)

Виведені параметри в консолі показують структуру нейронної мережі U-Net. Це зручний спосіб побачити, як побудована модель, скільки параметрів в ній використовується і як вони з'єднані.

Під час тренування та використання нейронних мереж, у консолі можна побачити інформацію про кількість параметрів моделі. Ця інформація дає зрозуміти обсяг і складність моделі, а також пам'ять, необхідну для її зберігання та тренування. Загальна кількість параметрів визначає, скільки інформації може вивчити модель, а треновані параметри показують, які саме параметри будуть змінюватися під час процесу навчання (див. рис 3.9).

сегментації DeepLab, потім порівняти їх і визначити найефективніший.

Для того, щоб налаштувати модель для тренування, потрібно розглянути архітектуру SegNet (див. рис. 3.11).

SegNet – це модель семантичної сегментації, складається з мережі кодувальника, відповідної мережі декодера, за якою слідує рівень класифікації по пікселям. Роль мережі декодера полягає у відображенні карт функцій кодера з низькою роздільною здатністю на карти функцій із повною роздільною здатністю вхідного сигналу для класифікації по пікселях. Новизна SegNet полягає в тому, як декодер підвищує дискретизацію карт вхідних функцій із нижчою роздільною здатністю. Зокрема, декодер використовує індекси об'єднання, обчислені на кроці максимального об'єднання відповідного кодера для виконання нелінійного підвищення дискретизації [18].

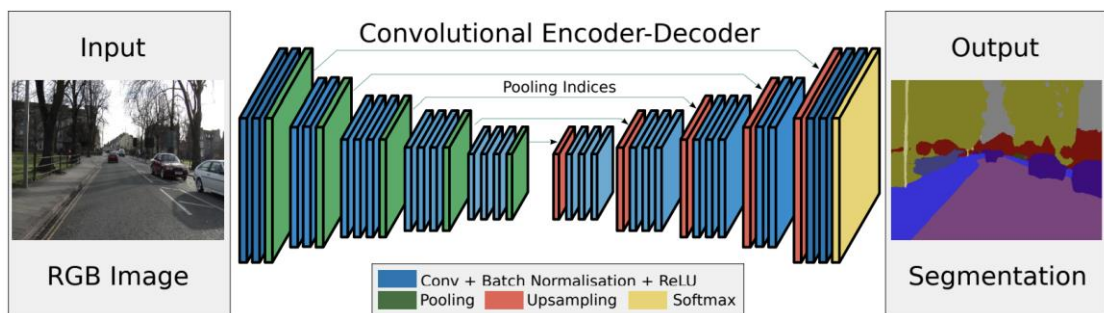


Рисунок 3.11 – Архітектура SegNet (за даними[18])

SegNet має мережу кодувальника та відповідний декодер, а потім останній попіксельний рівень класифікації. Мережа кодера складається з 13 згорткових шарів, призначених для класифікації об'єкта. Можна ініціалізувати процес навчання з ваг, навчених для класифікації на великих наборах даних. Можна також відкинути з'єднані шари на користь збереження карти функцій з вищою роздільною здатністю на найглибшому виході кодера. Це також зменшує кількість параметрів у кодувальнику SegNet мережі значно (від 134 млн. до 14,7 млн.). Остаточний результат декодера подається на мультиклас soft-max для створення ймовірностей класу для кожного пікселя незалежно.

На рисунку 3.12 наведено код, який реалізує модель сегментації SegNet на основі фреймворку PyTorch. Конструктор класу SegNet має такі параметри як кількість класів для сегментації, кількість каналів вхідного зображення, а також попередньо навчена модель VGG16. VGG16 має глибоку архітектуру, що складається з 16 послідовних згорткових шарів. Наприкінці мережі розміщені повністю підключені шари для класифікації зображень.

```

9 class SegNet(BaseModel):
10 def __init__(self, num_classes, in_channels=3, pretrained=True, freeze_bn=False, **_):
11     super(SegNet, self).__init__()
12     vgg_bn = models.vgg16_bn(pretrained=pretrained)
13     encoder = list(vgg_bn.features.children())
14
15     if in_channels != 3:
16         encoder[0] = nn.Conv2d(in_channels, 64, kernel_size=3, stride=1, padding=1)
17
18     self.stage1_encoder = nn.Sequential(*encoder[:6])
19     self.stage2_encoder = nn.Sequential(*encoder[7:13])
20     self.stage3_encoder = nn.Sequential(*encoder[14:23])
21     self.stage4_encoder = nn.Sequential(*encoder[24:33])
22     self.stage5_encoder = nn.Sequential(*encoder[34:-1])
23     self.pool = nn.MaxPool2d(kernel_size=2, stride=2, return_indices=True)
24
25     decoder = encoder
26     decoder = [i for i in list(reversed(decoder)) if not isinstance(i, nn.MaxPool2d)]
27     decoder[-1] = nn.Conv2d(64, 64, kernel_size=3, stride=1, padding=1)
28     decoder = [item for i in range(0, len(decoder), 3) for item in decoder[i:i+3][::-1]]
29     for i, module in enumerate(decoder):
30         if isinstance(module, nn.Conv2d):
31             if module.in_channels != module.out_channels:
32                 decoder[i+1] = nn.BatchNorm2d(module.in_channels)
33                 decoder[i] = nn.Conv2d(module.out_channels, module.in_channels, kernel_size=
34
35     self.stage1_decoder = nn.Sequential(*decoder[0:9])
36     self.stage2_decoder = nn.Sequential(*decoder[9:18])
37     self.stage3_decoder = nn.Sequential(*decoder[18:27])
38     self.stage4_decoder = nn.Sequential(*decoder[27:33])
39     self.stage5_decoder = nn.Sequential(*decoder[33:],

```

Рисунок 3.12 – Клас SegNet (рисунок створено самостійно)

SegNet як модель сегментації з використанням попередньо навченої моделі VGG16 для витягнення ознак та послідовних згорткових шарів з декодером для відновлення просторової інформації та отримання карти сегментації.

Дану модель також необхідно натренувати (див. рис. 3.13).

```

19 def main(config, resume):
20     train_logger = Logger()
21     train_loader = get_instance(dataloaders, 'train_loader', config)
22     val_loader = get_instance(dataloaders, 'val_loader', config)
23
24     model = get_instance(models, 'arch', config, train_loader.dataset.num_classes)
25     print(f'\n{model}\n')
26
27     loss = getattr(losses, config['loss'])(ignore_index = config['ignore_index'])
28
29     trainer = Trainer(
30         model=model,
31         loss=loss,
32         resume=resume,
33         config=config,
34         train_loader=train_loader,
35         val_loader=val_loader,
36         train_logger=train_logger)
37     trainer.train()

```

Рисунок 3.13 – Клас SegNet (рисунок створено самостійно)

Фрагмент код на рисунку 3.13 є точкою входу для тренування моделі. Він завантажує конфігурації, тобто виконує парсинг аргументів командного рядка для отримання шляху до конфігураційного файлу та інших налаштувань, завантажує конфігурацію з JSON-файлу. Виконує підготовку даних, створює модель, встановлює функції втрат (loss) – функція цілі або функція ризику, використовується для вимірювання того, наскільки добре модель прогнозує дані.

Ідея проста функції проста: модель робить прогноз для даних, функція втрат порівнює прогноз з дійсними значеннями та обчислює числове значення, що представляє помилку прогнозу. Мета алгоритму машинного навчання полягає в мінімізації цієї функції втрат, тобто в покращенні точності прогнозів моделі.

Завдяки цій нейромережі та натренованій моделі SegNet можна отримувати сегментовані зображення з автоматично передбаченими сегментаційними масками для нових зображень (див. рис. 3.14).

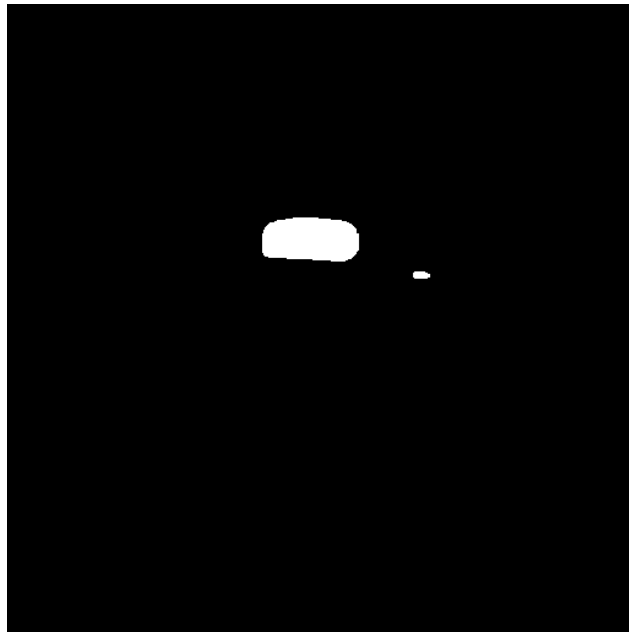


Рисунок 3.14 – Результат сегментації тестового зображення за допомогою SegNet (рисунок створено нейромережею)

3.3. Реалізація DeepLab

Виконання таких самих кроків для DeepLab. Це сімейство архітектур згорткових нейронних мереж (CNN), призначених для семантичної сегментації в комп'ютерному зорі. Ці моделі відомі своєю здатністю захоплювати дрібні деталі та виконувати семантичну сегментацію зображень із високою роздільною здатністю. Архітектура DeepLab зазнала кілька ітерацій, кожна з яких була вдосконалена для досягнення кращих результатів у різних завданнях комп'ютерного зору [19].

Архітектуру моделей DeepLab можна узагальнити наступним чином (див. рис 3.15):

- Мережа Backbone: модель починається з мережі backbone (наприклад, ResNet або MobileNet), яка витягує карти ознак з вхідного зображення.
- Атріальні згортки: карти ознак обробляються через атріальні згортки з різними коефіцієнтами для захоплення контексту на різних масштабах.
- Атріальне просторове пірамідалне підсумовування (ASPP): ASPP застосовується до карт ознак для захоплення контексту на різних масштабах.

Включає паралельні атріальні згортки з різними коефіцієнтами.

- Upsampling: Остаточні карти ознак збільшуються до відповідності оригінальній роздільній здатності зображення.
- Кінцева сегментація: Збільшені карти ознак використовуються для створення кінцевої піксельної карти сегментації.

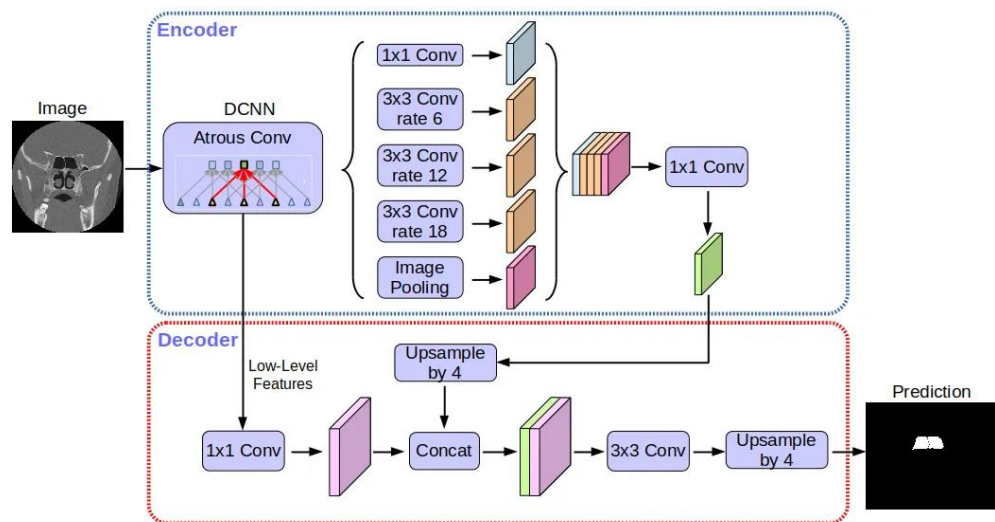


Рисунок 3.15 – Архітектура Deeplab (за даними [19])

Програмний код для Deeplab наведено на рисунку 3.16. На рисунку можна бачити, що створено окремий клас, у якому визначені методи для ініціалізації моделі, завантаження попередньо навчених ваг, а також налаштування втрат та оптимізатора.

Під час створення екземпляра класу Deeplab задаються різні параметри, такі як кількість класів, використання підготовлених моделей, а також налаштування для нормалізації пакетів. У класі також визначені плейсхолдери для вхідних даних, міток, цільових розмірів зображення, швидкості навчання та вагового згасання.

Клас ініціалізує сеанс TensorFlow і зберігає графік обчислень у логах для подальшого аналізу. Якщо модель створюється для навчання, вона також ініціалізує файловий записувач для збереження логів навчання.

```

8 class DeepLab(object):
9
10 def __init__(self, base_architecture, training=True, num_classes=21, ignore_label=255,
11
12     self.is_training = tf.placeholder(tf.bool, None, name='is_training')
13     self.num_classes = num_classes
14     self.ignore_label = ignore_label
15     self.inputs_shape = [None, None, None, 3]
16     self.labels_shape = [None, None, None, 1]
17     self.training = training
18     self.inputs = tf.placeholder(tf.float32, shape=self.inputs_shape, name='inputs')
19     self.labels = tf.placeholder(tf.uint8, shape=self.labels_shape, name='labels')
20
21     self.target_height = tf.placeholder(tf.int32, None, name='target_image_height')
22     self.target_width = tf.placeholder(tf.int32, None, name='target_image_width')
23
24     self.weight_decay = tf.placeholder(tf.float32, None, name='weight_decay')
25     self.regularizer = tf.contrib.layers.l2_regularizer(scale=self.weight_decay)
26     self.batch_norm_momentum = batch_norm_momentum
27
28     self.feature_map = self.backbone_initializer(base_architecture)
29     if pre_trained_model:
30         self.initialize_backbone_from_pretrained_weights(pre_trained_model)
31     self.outputs = self.model_initializer()
32
33     self.learning_rate = tf.placeholder(tf.float32, None, name='learning_rate')
34     self.loss = self.loss_initializer()
35     self.optimizer = self.optimizer_initializer()
36
37     # Initialize tensorflow session
38     self.saver = tf.train.Saver()
39     self.sess = tf.Session()
40     self.sess.run(tf.global_variables_initializer())

```

Рисунок 3.16 – Клас DeepLab (рисунок створено самостійно)

Результат навчання моделі наведено на рисунку 3.17.

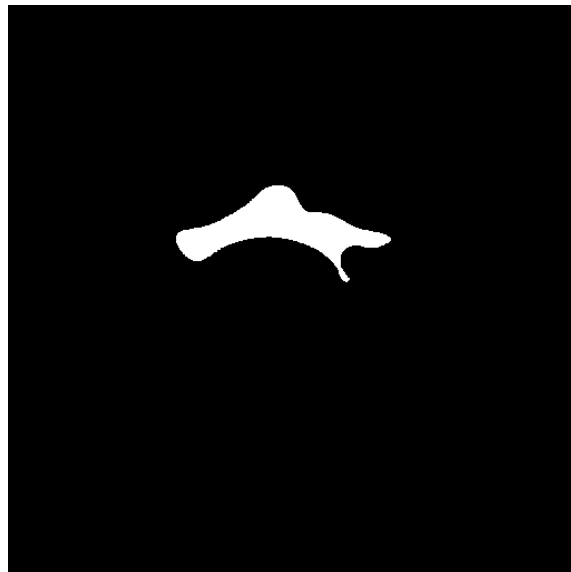


Рисунок 3.17 – Результат сегментації тестового зображення за допомогою DeepLab (рисунок створено нейромережею)

4 ОПИС ЕКСПЕРИМЕНТАЛЬНИХ ДОСЛІДЖЕНЬ

В даному розділі проводиться аналіз отриманих результатів сегментації зображень, порівнюються моделі нейромереж та встановлюється найефективніший метод сегментації.

Для порівняння отриманих результатів слід обрати найкращі метрики для порівняння якості роботи моделей нейромереж, однією з таких є IOU.

IOU (Intersection over Union) — це термін, який використовується для опису ступеня перекриття двох прямокутників чи контурів. Чим більше область перекриття, тим більше IOU. IOU в основному використовується в програмах, пов'язаних із виявленням об'єктів, де ми навчаємо модель виводити контури або рамку, яка ідеально підходить навколо об'єкта.

IoU можна визначити також як відношення площі перетину двох прямокутників чи контурів до їхніх сумарних площ. Основна обмежувальна рамка істини і очікувана обмежувальна рамка охоплюють область об'єднання, яка є знаменником. Ми обчислюємо перекриття між обмежувальною рамкою та прогнозованою обмежувальною рамкою в чисельнику. Математично це записується так:

$$IoU = \frac{A \cap B}{A \cup B} \quad (4.1)$$

де $A \cap B$ — площа перетину між двома множинами об'єктів A і B (області, які відповідають сегментованим областям на зображенні);

$A \cup B$ — площа об'єднання між двома множинами об'єктів A і B . Це область, яка охоплює як сегментовані області A , так і B .

В даному випадку з сегментованими зображеннями A — це істинна маска (true mask), яка містить правильні мітки класів для кожного пікселя на зображенні, які сегментовані, B - це передбачена маска (predicted mask) від кожної з моделей (U-Net, SegNet, DeepLab).

Друга метрика, яку було обрано для експерименту, називається Dice Similarity Coefficient (або Dice Index) - це метрика, яка використовується для вимірювання схожості між двома множинами. У контексті сегментації зображень

вона також застосовується для порівняння схожості між передбаченими масками (прогнозованими моделями) та істинною маскою (true mask).

Формула для обчислення Dice Similarity Coefficient для двох масок A і B має вигляд:

$$DSC = \frac{2 \times |A \cap B|}{|A| \cup |B|} \quad (4.2)$$

де $A \cap B$ - перетин множин A і B, тобто кількість пікселів, які одночасно входять і в передбачену маску, і в істинну маску;

$|A|$ - кількість пікселів в масці A;

$|B|$ - кількість пікселів в масці B.

Значення DSC лежить в діапазоні від 0 до 1, де 1 відповідає повному збігу між масками A і B (ідеальна схожість), а 0 означає відсутність будь-якого збігу.

Третя метрика - це Precision (точність), яка вимірює, наскільки точно модель ідентифікує позитивні приклади серед тих, які вона визначила як позитивні. У контексті сегментації зображень precision обчислюється для кожного класу окремо і потім усереднюється за всіма класами. Формула для обчислення precision має вигляд:

$$Precision = \frac{TP}{TP + FP} \quad (4.3)$$

де TP (True Positives) - кількість правильно визначених позитивних прикладів (відповідних пікселів в передбаченій масці);

FP (False Positives) - кількість неправильно визначених позитивних прикладів (пікселів, які модель вважає позитивними, але вони не належать істинній масці).

Precision відображає частку правильно визначених позитивних прикладів серед усіх прикладів, які модель вважає позитивними. Високе значення precision означає, що модель має мало FP, тобто вона майже не помиляється в тому, що вона визначає як позитивне.

Ще одна метрика яка необхідна для аналізу проведеного експерименту - це Recall (чутливість), яка вимірює, наскільки добре модель визначає всі можливі позитивні приклади. У контексті сегментації зображень recall також обчислюється для кожного класу окремо і потім усереднюється. Формула для обчислення recall

має вигляд:

$$Recall = \frac{TP}{TP + FP} \quad (4.4)$$

де FN (False Negatives) - це кількість тих позитивних прикладів (пікселів), які насправді є позитивними (належать до істинної маски), але модель їх не визначила як позитивні. Іншими словами, це ті випадки, коли модель не виявила те, що повинна була виявити.

Recall (чутливість) показує, наскільки добре модель виявляє всі позитивні приклади. Високе значення recall означає, що модель має мало False Negatives, тобто вона правильно виявляє майже всі позитивні приклади.

Програмна реалізація обчислення і візуалізації метрик виглядає наступним чином (див. рис. 4.1, 4.2).

```

1  import numpy as np
2  import matplotlib.pyplot as plt
3  from PIL import Image
4  from sklearn.metrics import jaccard_score, f1_score, precision_score, recall_score
5
6
7  def load_image_as_array(image_path):
8      image = Image.open(image_path).convert('L')
9      return np.array(image)
10
11 def calculate_metrics(true_mask, pred_mask):
12     true_mask = true_mask.flatten()
13     pred_mask = pred_mask.flatten()
14
15     iou = jaccard_score(true_mask, pred_mask, average='macro')
16     dice = f1_score(true_mask, pred_mask, average='macro')
17     precision = precision_score(true_mask, pred_mask, average='macro', zero_division=1)
18     recall = recall_score(true_mask, pred_mask, average='macro')
19
20     return iou, dice, precision, recall

```

Рисунок 4.1 – Функція обчислення метрик (рисунок створено самостійно)

```

41 def plot_segmented_images(images, titles, metrics=None):
42     fig, axes = plt.subplots(1, len(images), figsize=(20, 10))
43     for ax, img, title in zip(axes, images, titles):
44         ax.imshow(img, cmap='gray')
45         ax.set_title(title)
46         ax.axis('off')
47
48         if metrics and title != 'True Mask':
49             iou = metrics['IoU'][titles.index(title) - 1]
50             contour = np.zeros_like(img)
51             contour[img != 0] = 1
52             ax.contour(contour, colors='red', linewidths=1)
53
54     plt.show()

```

Рисунок 4.2 – Функція візуалізації метрик (рисунок створено самостійно)

В результаті маємо наочний результат обчислення метрик (див. рис. 4.3, 4.4).

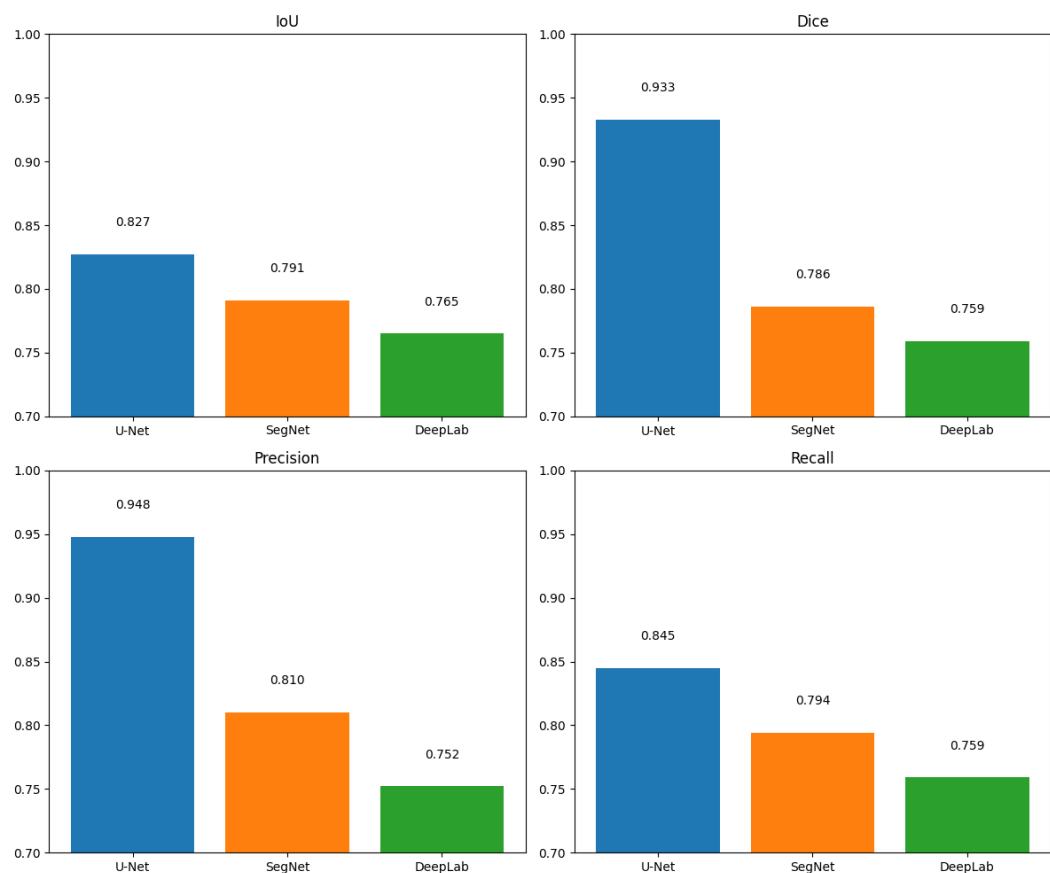


Рисунок 4.3 – Гістограма обчислених метрик (рисунок створено самостійно)

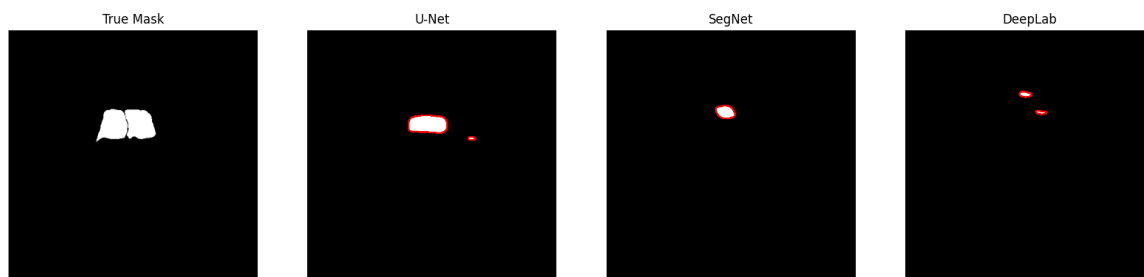


Рисунок 4.4 – Порівняння сегментованих масок зображень з візуалізацією метрики IOU (рисунок створено самостійно)

Аналізуючи результати сегментації трьох моделей (U-Net, SegNet, DeepLab), можна зробити такі висновки на основі гістограм метрик: IoU, Dice, Precision та Recall:

- IoU: модель U-Net має найвищий показник IoU – 82%, що свідчить про найкращу здатність моделі знаходити пересічення між передбаченою маскою та істинною маскою;
- Dice: модель U-Net також має найвищий показник Dice, що підтверджує її перевагу в точності сегментації, на баланс між точністю та повнотою;
- Precision: U-Net має найвищий показник точності (Precision), що означає, що вона має меншу кількість хибнопозитивних результатів;
- Recall: U-Net знову має найвищий показник Recall, що означає, що вона краще знаходить істинно позитивні приклади.

Таблиця 4.1 – Порівняльна таблиця моделей за метриками (створено самостійно)

Модель/ метрика	IoU	Dice	Precision	Recall
U-Net	0.827	0.933	0.948	0.845
SegNet	0.791	0.786	0.810	0.794
DeepLab	0.765	0.759	0.752	0.759

Тепер, коли обчислено числові оцінки якості сегментації, які базуються на

порівнянні передбачуваних результатів моделі з правильними або очікуваними результатами, потрібно вирішити ще одну задачу – оцінити співвідношення подібності між двома зображеннями, щоб ідентифікувати таким чином людину по знімку комп'ютерної томографії мозку, що є метою даної роботи.

Для цього можна використати такі методи, як Structural Similarity Index (SSI) та Peak Signal-to-Noise Ratio (PSNR).

Structural Similarity Index (SSI) – цей індекс оцінює подібність між двома зображеннями, враховуючи як локальні, так і глобальні властивості. Він генерує значення між -1 і 1, де 1 означає ідеальну подібність, 0 - жодної подібності, а значення менше 0 - навіть антикореляцію.

Peak Signal-to-Noise Ratio (PSNR) – показник вимірює якість відновленого зображення відносно оригінального шляхом порівняння сигналу (зображення) і шуму (похибок). PSNR виражається в децибелах (dB) і обчислюється на основі середньоквадратичної помилки між пікселями двох зображень. Високе значення PSNR вказує на високу подібність між зображеннями. Формула 4.5 пояснює ці показники математично.

Нехай X і Y – це два зображення, а μ_X, μ_Y – їх середні значення, σ_X^2, σ_Y^2 – дисперсії, σ_{XY} – коваріація між X і Y .

SSIM визначається наступною формулою:

$$SSIM(X, Y) = \frac{(2\mu_X\mu_Y + c_1)(2\sigma_{XY} + c_2)}{(\mu_X^2 + \mu_Y^2 + c_1)(\sigma_X^2 + \sigma_Y^2 + c_2)} \quad (4.5)$$

де μ_X, μ_Y – середні значення зображень X і Y ;

σ_X^2, σ_Y^2 – дисперсії зображень X і Y ;

σ_{XY} – коваріація, тобто міра статистичного взаємозв'язку між двома випадковими величинами між X і Y ;

c_1, c_2 - константи для стабілізації ділення (зазвичай дуже малі значення для запобігання діленню на нуль).

SSIM набуває значень від -1 до 1, де 1 відповідає повній схожості між двома зображеннями.

PSNR вимірює якість відновлення або компресії зображення. Визначається

наступною формулою:

$$PSNR(X, Y) = 10 \times \log_{10} \left(\frac{\max_I^2}{MSE} \right) \quad (4.6)$$

де \max_I - максимальне можливе значення пікселя (зазвичай 255 для 8-бітних зображень);

MSE - середньоквадратична помилка між зображеннями X і Y.

Далі реалізуємо обчислення показників програмно (див. рис. 4.5)

```

1  import numpy as np
2  import matplotlib.pyplot as plt
3  from skimage.metrics import structural_similarity as ssim
4  from skimage.metrics import peak_signal_noise_ratio as psnr
5  from PIL import Image
6
7  def load_image_as_array(image_path):
8      image = Image.open(image_path).convert('L')
9      return np.array(image)
10
11 def calculate_ssim(image1, image2):
12     return ssim(image1, image2)
13
14 def calculate_psnr(image1, image2):
15     return psnr(image1, image2, data_range=image1.max() - image1.min())
16

```

Рисунок 4.5 – Програмна реалізація показників SSI та PSNR (рисунок створено самостійно)

Популярність даних показників для оцінки зображень дуже висока, тому використано вже існуючу бібліотеку `skimage.metrics`. Результат візуалізації числових метрик представлено на рисунку 4.6.

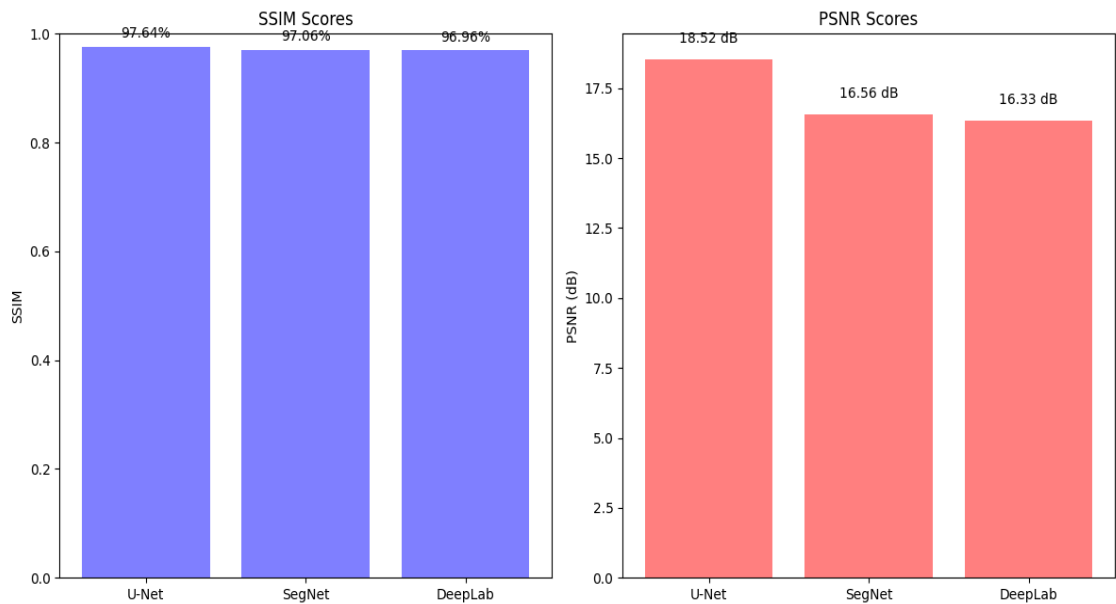


Рисунок 4.6 – Візуалізація показників порівняння передбачень з оригінальною маскою (рисунок створено самостійно)

На рисунку 4.6 видно для U-Net SSIM складає 97.64%. Це означає, що між справжньою маскою і маскою, що передбачається U-Net, спостерігається дуже висока структурна схожість. Для SegNet SSIM дорівнює 97.06%. Також дуже високе значення, що свідчить про дуже непогану точність, але трохи нижче, ніж у U-Net. Для DeepLab SSIM складає 96.96%. Незначний спад порівняно з попередніми двома моделями, але все ще високий рівень схожості.

PSNR: U-Net також має найвищий показник, що означає кращу збереженість якості зображення після обробки. SegNet та DeepLab показують меншу якість відтворення, що може вказувати на меншу точність у відтворенні деталей, що дуже важливо для ідентифікації людини.

Виходячи з отриманих результатів, можна сказати, що модель U-Net, отримавши найкращий результат за різними метриками та показниками, є найкращим вибором серед нейромереж для сегментації та отримання масок зображень комп'ютерної томографії мозку для ідентифікації людини. Числові показники дають змогу оцінити наскільки добре модель справляється зі своєю задачею.

ВИСНОВКИ

Під час цього дослідження проведено детальний аналіз різних методів та моделей для сегментації зображень мозку за допомогою комп'ютерної томографії (КТ). Основною метою було визначення ефективних підходів для ідентифікації особи на основі обробки медичних зображень. Дослідження охопило аналіз таких методів, як порогова сегментація, регіональна сегментація, методи активних контурів та глибоке навчання з використанням нейронних мереж, зокрема архітектур U-Net, SegNet та DeepLab. Кожен метод було оцінено з урахуванням їхніх переваг, недоліків та можливих областей застосування. Досліджено також критерії ефективності моделей, зокрема структурну схожість (SSIM) та пікове відношення сигнал-шум (PSNR), які дозволяють оцінити точність відтворення та збереженість якості зображень після обробки.

За отриманими результатами модель U-Net показала найвищі значення SSIM (97.64%) і PSNR (18.52 dB), що свідчить про високу структурну схожість та збереженість деталей на зразок. SegNet, зі значеннями SSIM 97.06% і PSNR 16.56 dB, демонструє добрі показники, але трохи нижчі порівняно з U-Net. DeepLab, з SSIM 96.96% і PSNR 16.33 dB, показав незначний спад в якості порівняно з іншими моделями. Отже, з врахуванням числових метрик та аналізу результатів можна визначити модель U-Net як найбільш ефективну для задач сегментації зображень КТ мозку для ідентифікації людини.

Цей підхід до аналізу і порівняння моделей сегментації зображень є важливим для забезпечення точності і надійності алгоритмів у практичних застосуваннях та подальшому розвитку медичних інформаційних систем. Отриманий звіт включає детальний аналіз різних аспектів сегментації медичних зображень, а також визначає критерії ефективності, які можуть бути використані для порівняння та вибору методів у майбутніх дослідженнях та практичних застосуваннях в галузі машинного навчання та комп'ютерного зору.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Pham D. L., Xu C., Prince J. L. Current methods in medical image segmentation. *Annual Review of Biomedical Engineering*. 2000;2(2000):315–337. doi: 10.1146/annurev.bioeng.2.1.315.
2. Ashburner J., Friston K. J. Unified segmentation. *NeuroImage*. 2005;26(3):839–851. doi: 10.1016/j.neuroimage.2005.02.018.
3. MRI Segmentation of the Human Brain: Challenges, Methods, and Applications. URL: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4402572/> (дата звернення: 03.03.2024).
4. Pal N. R., Pal S. K. A review on image segmentation techniques. *Pattern Recognition*. 1993;26(9):1277–1294. doi: 10.1016/0031-3203(93)90135-J.
5. Chan T. F., Vese L. A. Active contours without edges. *IEEE Transactions on Image Processing*. 2001;10(2):266–277. doi: 10.1109/83.902291
6. Boykov Y., Funka-Lea G. Graph cuts and efficient N-D image segmentation. *International Journal of Computer Vision*. 2006;70(2):109–131. doi: 10.1007/s11263-006-7934-5.
7. Greig D. M., Porteous B. T., Seheult A. H. Exact maximum a posteriori estimation for binary images. *Journal of the Royal Statistical Society B*. 1989;51(2):271–279.
8. Collier D. C., Burnett S. S. C., Amin M., et al. Assessment of consistency in contouring of normal-tissue anatomic structures. *Journal of Applied Clinical Medical Physics*. 2003;4(1):17–24.
9. Qi Peng, Xingcai Chen, Chao Zhang, Wenyan Li. Deep learning-based computed tomography image segmentation and volume measurement of intracerebral hemorrhage. DOI:10.3389/fnins.2022.965680.
10. Pereira S., Pinto J.A., Alves V., Silva C. Brain tumor segmentation using convolutional neural networks in MRI images. *IEEE Trans. Med. Imaging*. 2016;35:1240–1251. doi: 10.1109/TMI.2016.2538465.
11. Image segmentation detailed overview. URL: <https://www.superannotate.com/blog/image-segmentation-for-machine-learning> (дата звернення: 15.03.2024).
12. Brain image segmentation of the corpus callosum by combining Bi-Directional Convolutional LSTM and U-Net using multi-slice CT and MRI. URL:

<https://www.sciencedirect.com/science/article/abs/pii/S0169260723002675> (дата звернення: 17.03.2024).

13. Bilous, N., O. Hramm, I. Ahekan, A. T. Khudhair, L. Illyashenko, A. Nerukh. Numerical Error Analysis For Configurable Cell Segmentation Problem /ESKİŞEHİR TECHNICAL UNIVERSITY JOURNAL OF SCIENCE AND TECHNOLOGY A-APPLIED SCIENCES AND ENGINEERING, 2019, Vol: 20, pp. 193 — 205.

14. Hramm O, Bilous N, Ahekan I. Configurable Cell Segmentation Solution Using Hough Circles Transform and Watershed Algorithm. In: 2019 IEEE 8th International Conference on Advanced Optoelectronics and Lasers (CAOL) CAOL 2019 September 6-8, 2019, Sozopol, Bulgaria, USA: IEEE, 602-605. <https://ieeexplore.ieee.org/xpl/conhome/9002628/proceeding>.

15. Natalia Bilous, Oleksii Komarov. Використання методів глибинного навчання для сегментації пухлин головного мозку на МРТ-зображеннях. Інформаційні системи та технології ICT-2018 - матеріали 7-ї Міжнародної науково-технічної конференції, 10-15 вересня 2018 Коблеве-Харків, Україна.

16. N. Bilous, O. Svidin, I. Ahekan, and V. Malko, “A skeleton-based method for exercise recognition based on 3D coordinates of human joints,” IJ-AI, vol. 13, no. 2, p. 1805, Jun. 2024, doi: 10.11591/ijai.v13.i2.pp1805-1816.

17. U-Net Architecture Explained. URL: <https://www.geeksforgeeks.org/u-net-architecture-explained/> (дата звернення: 10.06.2024).

18. Vijay Badrinarayanan, Alex Kendall, Roberto Cipolla. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. URL: <https://arxiv.org/abs/1511.00561> (дата звернення: 11.06.2024).

19. Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, Alan L. Yuille. Semantic Image Segmentation with Deep Convolutional Nets and Fully Connected CRFs. arXiv:1412.7062. URL: <https://arxiv.org/abs/1412.7062v4> (дата звернення: 11.06.2024).

20. Zhang, L., Zhang, L., Mou, X., & Zhang, D. (2011). Image Quality Assessment: From Error Visibility to Structural Similarity. IEEE Transactions on Image Processing, 13(4), 600-612. doi: 10.1109/TIP.2009.2031883.