

Факультет _____ Комп'ютерних наук _____

Кафедра _____ Системотехніки _____

Рівень вищої освіти _____ другий (магістерський) _____

Спеціальність _____ 122 Інформаційні технології проектування _____
(код і повна назва)

Тип програми _____ освітньо-професійна _____

Освітня програма _____ Комп'ютерні науки _____
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

«____» _____ 20__ р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові _____ Семенку Олександр Ігоровичу _____
(прізвище, ім'я, по батькові)

1. Тема роботи: Дослідження методів сегментації зображень для ідентифікації загиблих під час війни

затверджена наказом по університету від 20.11 2023 р. № 1373Ст

2. Термін подання студентом роботи до екзаменаційної комісії: 19.01.2024 р

3. Вихідні дані до роботи: Дослідити методи сегментації зображень комп'ютерної томографії, а саме зробити порівняльний аналіз їх характеристик та провести серію експериментів з метою виявлення найрезультативніших методів. Перелік програмних засобів, що були використані: ОС Microsoft Windows v.11, інтегроване середовище розробки Google Colabotory.

4. Перелік питань, що потрібно опрацювати в роботі: 4.1 Вступ 4.2 Аналіз предметної області 4.2.1 Проблема ідентифікації особистості 4.2.2 Етичність питання 4.3 Особливості застосування методів обробки зображення 4.4 Порівняльний аналіз 4.5 Можливість покращення результату

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій: *Код програми*

6. Консультанти розділів роботи

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підп	дата

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	<i>Отримання завдання на виконання роботи</i>	<i>16.10.2023</i>	
2	<i>Постановка задачі та узгодження з керівником</i>	<i>18-19.10.2023</i>	
3	<i>Опис та аналіз предметної області</i>	<i>25-31.11.2023</i>	
4	<i>Огляд літератури</i>	<i>01-17.11.2023</i>	
5	<i>Математичний опис задачі</i>	<i>18-30.11.2023</i>	
6	<i>Проектування алгоритму сегментації зображення</i>	<i>01-15.12.2023</i>	
7	<i>Проведення експерименту</i>	<i>10-15.12.2023</i>	
8	<i>Збор та аналіз даних експерименту</i>	<i>15-20.12.2023</i>	
9	<i>Оформлення пояснювальної записки</i>	<i>20-30.12.2023</i>	
10	<i>Представлення на рецензування</i>	<i>19.01.2024</i>	

Дата видачі завдання 16.10.2023 р.

Студент _____
(підпис)

Керівник роботи _____ проф Нечипоренко А.С.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка до кваліфікаційної роботи магістра містить: 52 с., 21 рис., 1 додаток, 23 джерела інформації.

U-NET, SEGNET, PYTHON, KERAS, TENSORFLOW, СЕГМЕНТАЦІЯ ЗОБРАЖЕНЬ

Об'єктом дослідження роботи є процес сегментації КТ зображень.

Предметом дослідження роботи є методи сегментації зображень.

Метою кваліфікаційної роботи є дослідження методів сегментації зображень КТ.

Методи дослідження – системний аналіз, теорія обробки зображень, мова програмування Python, бібліотеки Keras, TensorFlow, OpenCV. Методи машинного навчання та сегментації зображення.

У роботі проведено аналіз теоретичних основ сегментації зображень, та визначено параметри, критерії та метрики оцінювання, які є ефективними для сегментації КТ зображень.

Під час дослідження було проведено аналіз існуючих методів сегментації зображень.

Сфера застосування – сегментація медичних зображень, знімків КТ, для подальшої ідентифікації.

ABSTRACT

Master thesis contains: 52 pages, 21 figures, 1 appendice, 23 references.

U-NET, SEGNET, PYTHON, KERAS, TENSORFLOW, SEGMENTATION
CT IMAGE

The object of research is the process of segmentation of CT images.

The subject of research are methods of image segmentation.

The purpose of the thesis is to study CT image segmentation methods.

Research methods – system analysis, theory of image processing, Python programming language, Keras, TensorFlow, OpenCV libraries. Methods of machine learning and image segmentation.

The thesis analyzes the theoretical foundations of image segmentation, and defines parameters, criteria, and metrics for evaluating segmentation methods, which are the most effective

During the research, an analysis of existed methods of image segmentation was carried out.

The field of application is the segmentation of medical images, CT images, for further identification.

ЗМІСТ

Перелік умовних позначень.....	10
Вступ.....	11
1 Аналіз предметної області та постановка задачі	13
1.1 Проблема ідентифікації особистості	13
1.2 Огляд сучасного стану розвитку методів сегментації зображень	14
1.3 Потенціал застосування методів штучного інтелекту для обробки зображень комп'ютерної томографії	17
1.4 Постановка задачі.....	19
2 Аналіз методів сегментації зображень.....	22
2.1 Методи сегментації на основі нейромереж глибинного навчання ...	22
2.2 Аналіз існуючих засобів обробки медичних зображень	23
2.3 Опис експериментальних даних	25
2.4 Етичність питання	26
3 Дослідження методів вирішення задачі	28
3.1 Математичний опис моделі SegNet	28
3.2 Математичний опис моделі U-Net.....	29
3.3 Удосконалення методу функції втрат	31
3.4 Особливості застосування методів обробки зображення.....	32
3.5 Вибір мови програмування.	35
3.6 Вибір бібліотек	37
3.6.1 Бібліотека Keras.....	37
3.6.2 Бібліотека Tensorflow	38
3.7 Запуск моделей	38
4 Опис створеної програми	39
4.2 Модель U-Net.....	39
4.3 Модель SegNet.....	41
4.4 Різниця методів.....	43
4.5 Метрики ефективності для оцінювання результатів сегментації	44

4.6 Результат сегментації.....	47
4.7 Можливість покращення результату.....	49
ВИСНОВКИ	51
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	53
Додаток А КОД ПРОГРАМИ	55

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

КТ – комп'ютерна томографія;

CNN (Convolutional Neural Network) – архітектура нейронної мережі;

GPU (Graphics Processing Unit) – графічний процесор;

TPU (Tensor Processing Unit) — тензорний процесор;

GLCM (Gray Level Cooccurrence Matrix) – Матриця збіжності рівнів сірого.

ВСТУП

У контексті сучасного світу, в якому велика кількість військових конфліктів та надзвичайних ситуацій які за собою залишають трагічні наслідки постає проблема ідентифікації загиблих. Серед існуючих підходів можна відокремити використання сучасних методів розпізнавання об'єктів на зображенні, а саме методів сегментації зображення.

Сегментація зображення в останні роки набула значної популярності в області медичних підходів аналізу. Сучасні тенденції у розвитку комп'ютерного зору та обробки зображень роблять проблему сегментації зображення вкрай актуальною. Зі зростанням обсягу візуальних даних, таких як медичні зображення, потрібна розробка ефективних методів для автоматичного виділення об'єктів та регіонів на зображенні.

В умовах війни методи ідентифікації зображень можуть бути надзвичайно корисними для встановлення особової інформації та допомоги у визначенні осіб, але використання таких методів повинно бути в останню чергу, коли немає можливості ідентифікувати людину методом сепарації ДНК, або інших більш точних методів.

Застосування сегментації зображень має широкий спектр практичних застосувань, включаючи покращення точності діагностики у медичних зображень та підвищення продуктивності систем автоматичного розпізнавання та класифікації, а також покращення безпеки.

В той же час незважаючи на значні досягнення в цій галузі, існуючі методи сегментації стикаються с великою кількістю проблем, такими як обробка зображень з низькою роздільною здатністю, складні умови освітлення та сцени з підвищеним рівнем шумів. Дослідження нових методів стає необхідним для подолання цих викликів та покращення загальної ефективності сегментації зображень.

Таким чином це дослідження спрямоване на дослідження та порівняння методів сегментації зображень з метою виявлення самого ефективного в області

ідентифікації, що є актуальним завданням у контексті сучасних вимог до обробки візуальних даних.

Застосування сучасних технологій для обробки зображень у такій важливій галузі може сприяти вдосконаленню процесу пошуку та розпізнавання осіб, що, в свою чергу допоможе державі та родинам у пошуку та ідентифікації військовослужбовців та цивільних під час конфліктів.

Виходячи з цієї мети були визначені завдання дослідження:

- розгляд існуючих методів сегментації зображень;
- порівняльний аналіз обраних методів сегментації зображення;
- аналіз ефективності імплементації обраного методу.

Об'єктом дослідження є процес сегментації КТ зображень для ідентифікації загиблих під час війни. Цей процес включає в себе збір, обробку та аналіз інформації пов'язаної за особовими даними та зображеннями загиблих.

Предмет дослідження – методи сегментації зображень, які використовуються для аналізу та обробки зображень, для ідентифікації загиблих.

Для досягнення мети роботи використовується комплекс з теоретичних (порівняння, аналіз технічної літератури та результатів дослідження) та емпіричних (експеримент, опис та моделювання) методів.

У даному дослідженні буде проведено огляд існуючих методів сегментації зображень та їх застосувань в області ідентифікації загиблих.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Проблема ідентифікації особистості

У сучасному світі відбувається низка військових конфліктів, які нещадно знищують життя людей та приводять до безлічі жертв. Кожного дня збільшується кількість людей, які вважаються безвісти зниклими, і багато з них стають неможливими до ідентифікації через різноманітні обставини. Серйозною проблемою стає визначення осіб, які загинули під час конфліктів, зокрема тих, чиє тіло сильно пошкоджено.

Для вирішення проблеми ідентифікації осіб, зокрема загиблих, використання методів сегментації зображення може стати ключовим елементом. Ефективний аналіз різних параметрів, таких як анатомічні особливості, дактилоскопія, патологічні зміни та особливі прикмети, може допомогти у точній ідентифікації. Кожен з цих методів має свої унікальні переваги та використовується в залежності від конкретних обставин.

У разі сильних опіків тіла, наприклад, може виникнути виклик у визначенні особи за допомогою традиційної дактилоскопії. В таких випадках аналіз зображень комп'ютерної томографії (КТ) набуває особливого значення. Цей метод дозволяє ідентифікувати особу за унікальними даними, такими як форма черепа, або наявні дефекти, які можуть бути отримані протягом життя.

Застосування аналізу КТ знімків виявляється особливо ефективним у випадках, коли інші методи ідентифікації не ефективні, наприклад, при відсутності близьких родичів для проведення ідентифікації за ДНК чи при складних умовах, які ускладнюють проведення дактилоскопії. Такий підхід може допомогти подолати труднощі ідентифікації в умовах воєнного конфлікту та допомагати встановлювати особи, що втратили своє життя.

1.2 Огляд сучасного стану розвитку методів сегментації зображень

На сьогоднішній день розвиток техніки дозволяє створювати системи які можуть не тільки доповнювати але і в деяких випадках замінювати людину в задачах розпізнавання.

Такі задачі зовсім у різноманітних сферах, як діагностика органів по томографічних знімках, або розпізнавання рукописних текстів та інших прикладних задачах.

При чому під час розпізнавання зображень важливою частиною є сегментація. Сегментація – це процес поділу зображення на деяку кількість сегментів. Основними поняттями пов’язаними із сегментацією зображень є:

- піксель, найменша одиниця растрового зображення, яка має колір або яскравість, кожен піксель приймає участь в сегментації зображення, зазвичай пікселі розділяють на групи, або сегменти, залежно від певних властивостей;

- сегмент, як група пікселей, які мають спільні характеристики чи властивості, та утворюють окрему область на зображенні. Основною ціллю сегментації це виділити ці області, роблячи їх якомога більш однорідними всередині, та відмінними від інших частин зображення;

- сегментація об’єктів це процес виділення окремих об’єктів на зображенні, що забезпечує можливість ізолювати конкретні об’єкти та розрізняти їх один від одного та визначати їхні межі;

- семантична сегментація як вид сегментації, де кожен сегмент призначається конкретній семантичній категорії або класу. Це дозволяє розрізняти та класифікувати області зображення відповідно до їхнього семантичного значення;

- порогова сегментація являє собою метод, де пікселі розділяються на сегменти за допомогою порогового значення, яке визначається певним критерієм, таким як яскравість або колір. Це простий метод, який дозволяє визначити області на зображенні, що відрізняються від заданого порогу.

Метою сегментації є спрощення зображення для подальшого його аналізу. Її використовують для виділення об'єктів та меж на зображенні.

Приклад сегментації зображено на рисунку 1.1.



Рисунок 1.1 – Приклад сегментації зображення

Технологія сегментації зображення здобула велику популярність завдяки своєму потенціалу ідентифікувати об'єкти на зображеннях з великою точністю.

Алгоритми глибокого навчання, які використовуються в більшій частині технологій сегментації зображень здатні адаптуватися до різноманітних завдань, включаючи аналіз даних, класифікацію та регресію.

Технології сегментації зображень досягли значних успіхів та вкрай швидко розвиваються, особливо завдяки застосуванню глибокого навчання.

Основні тенденції розвитку в галузі сегментації зображень можуть включати ряд інноваційних підходів та покращень, які спрямовані на поліпшення точності, ефективності та робочих характеристик методів сегментації. Ось деякі загальні тенденції, які можуть впливати на цю область:

- глибоке навчання для сегментації дозволяє автоматично вивчати різноманітні функції та особливості на зображеннях, що призводить до покращення якості сегментації;

- розвиток методів, які забезпечують точну семантичну сегментацію в режимі реального часу, що важливо для застосувань у сфері комп'ютерного зору, розпізнавання об'єктів та автономних систем;

- зростання застосувань сегментації зображень у медичних дослідженнях та практиці, зокрема в аналізі зображень магнітно-резонансної томографії (MRI) та комп'ютерної томографії (КТ);

- розробка методів, які здатні адаптуватися до різних умов та типів зображень, що робить їх більш гнучкими та універсальними;

- інтеграція інших джерел інформації, таких як геопросторові дані, дані від сенсорів та контекстуальна інформація, для покращення точності сегментації та зниження помилок;

- звернення уваги до етичних питань, пов'язаних із застосуванням сегментації в чутливих сферах, таких як розпізнавання осіб, медична діагностика та обробка зображень в контексті військових досліджень;

- розробка універсальних методів, які можуть вирішувати не тільки задачі сегментації об'єктів, але і завдання, пов'язані з трекінгом, класифікацією та взаємодією об'єктів на зображенні.

Використання глибоких нейронних мереж вкрай добре перетворило ландшафти сегментації. Глибокі моделі можуть ефективно визначати контури та різноманітні області на зображеннях, що дозволяє використовувати її в різноманітних задачах, включаючи медицину.

Однією з найважливіших областей застосування методів сегментації зображень зараз є медицина. Наприклад для виявлення чужорідних об'єктів, ракових клітин, біологічних об'єктів та інших.

Найпопулярнішою технологією сегментації медичних зображень є технологія U-Net, яка була розроблена для сегментації біомедичних зображень. Архітектура була розроблена так, щоб вона могла працювати з мінімальною кількістю прикладів та робила більш точну сегментацію.

Сегментація зображень для ідентифікації загиблих під час війни стає вкрай важливою областю досліджень. Такі технології можуть бути використані у самих невідкладних та тяжких випадках, таких як масові катастрофи, де необхідно ідентифікувати вкрай тяжкі випадки.

1.3 Потенціал застосування методів штучного інтелекту для обробки зображень комп'ютерної томографії

Згідно з статистикою опублікованою МВС[2] що входить до єдиного реєстру осіб, які зникли безвісти в наслідок особливих обставин. На сьогоднішній день фіксується інформація про близько 26 тис. осіб, з яких 11 тис. це цивільні та близько 15 тис. це військовослужбовці. Ця статистика зазначена на рисунку 3.2.

Більшу частину з цих людей вже не ідентифікують через недостатку технологій ідентифікації, але розвиток методів сегментації медичних зображень може вкрай змінити ваги у кращу сторону.

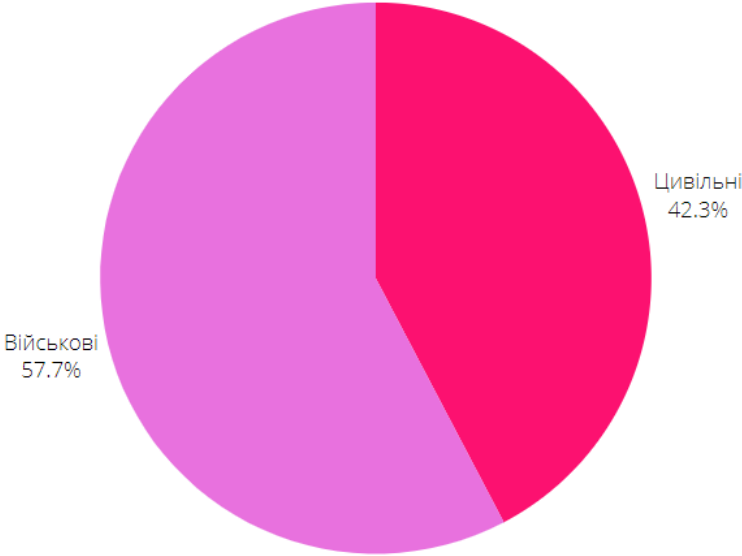


Рисунок 1.2 – Статистика безвісти зниклих

Час йде та технології розвиваються, інвестиції у AI тільки підвищуються, що можна подивитись на рисунку 3.3[3], це говорить про те що, все більша частина компаній зацікавлена у розвитку штучних мереж.

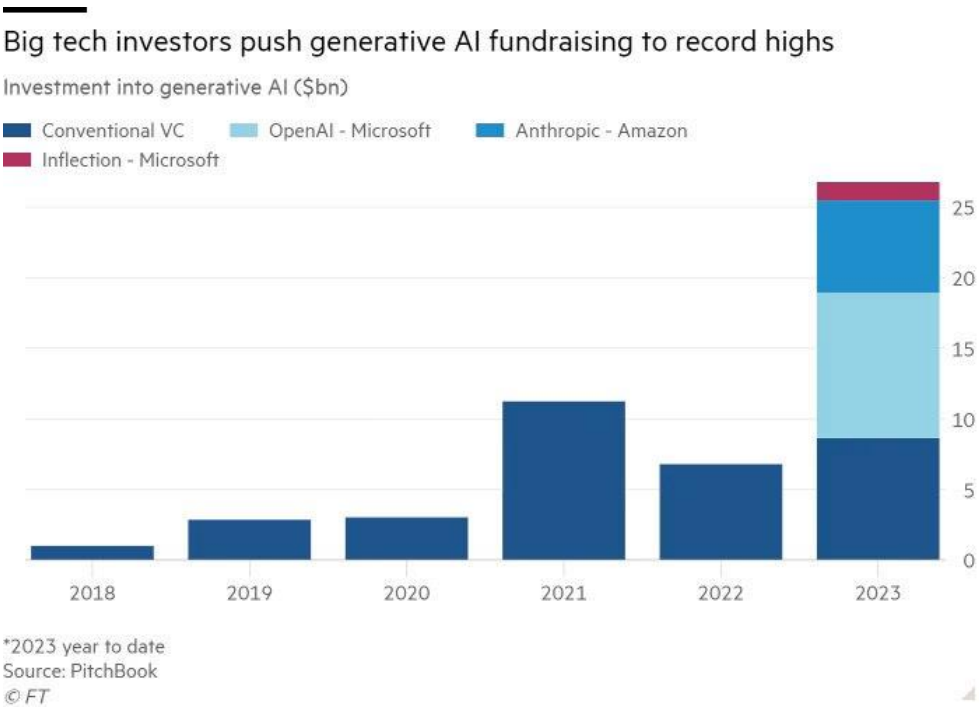


Рисунок 1.3 – Інвестиції у AI

Тобто якщо більша частина медичних даних стає цифровою моделлю, то це дає можливість поступово використовувати ці технології для ідентифікації за різними методами AI.

Якщо розвивати технологічні методи ідентифікації людини за певними ознаками його внутрішніх ознак, то через певний час можлива ситуація повністю прибрати можливість не ідентифікованої загиблої людини.

У використанні такі технології також можуть бути вкрай простими для патологоанатомів, так як при інтеграції мережі штучного інтелекту до лікарень питання швидкої ідентифікації відпадає, тому що потрібно лише відсканувати череп та за досить короткий деякий час отримати результат сегментації та ідентифікації.

При використанні технології сегментації медичних зображень великою кількістю медичних закладів точність ідентифікації також буде тільки зростати, так як нейроні мережі навчаються під час праці, та мають досить гнучку можливість оновлень зі сторони розробника.

1.4 Постановка задачі

Кваліфікаційна робота прагне виявити, оцінити та розвинути методи сегментації зображень для завдання ідентифікації осіб, які загинули під час війни, з метою покращення точності розпізнавання об'єктів, зокрема структур черепу, на зображеннях комп'ютерної томографії(КТ).

Сегментація зображення КТ передбачає розділення зображення на окремі сегменти або регіони з метою визначення та вивчення структури чи інших об'єктів на зображенні. Ця задача є важливою в медичній діагностиці та дослідженнях, де точна інтерпретація анатомічних структур є вирішальною для правильного визначення патологій чи аномалій які можуть сприяти ідентифікації людини, або пошуку захворювання.

Ця тема вкрай актуальна з кількох ключових поглядів. Нижче наведено деякі аспекти, які можна врахувати при обґрунтуванні актуальності цієї теми:

- в умовах воєнного конфлікту часто виникає потреба ідентифікувати та документувати жертви для подальшого повідомлення сім'ям та розслідування подій. Здатність ефективної сегментації зображень загиблих може значно полегшити цей процес, забезпечуючи швидше та точніше визначення осіб;

- дослідження сегментації зображень загиблих може мати практичне застосування в медичній сфері для аналізу травм та пошкоджень. Визначення точних меж тіл або областей на зображеннях може допомогти медичним працівникам при швидкому та ефективному наданні допомоги постраждалим;

- з впровадженням нових технологій, таких як машинне навчання та комп'ютерне бачення, з'являються нові можливості для розробки та вдосконалення методів сегментації зображень. Ваше дослідження може внести важливий внесок у цю область, що в свою чергу сприятиме розвитку сучасних технологій;

- при вирішенні кримінальних справ та розслідуванні військових злочинів, точна ідентифікація та аналіз слідів на зображеннях може бути вирішальною для встановлення фактів і відповідальності;

- забезпечення гідності та поваги до загиблих є важливим етичним аспектом. Розробка ефективних методів сегментації може сприяти тому, щоб зображення були адекватно та шанобливо оброблені.

У контексті даної роботи, "сегментація" відноситься до процесу виділення та визначення границь об'єктів на зображеннях. У даному випадку, завдання полягає в дослідженні методів сегментації зображень осіб, які стали жертвами воєнних подій.

Об'єктом дослідження є процес сегментації КТ зображень для ідентифікації загиблих під час війни. Цей процес включає в себе збір, обробку та аналіз інформації пов'язаної за особовими даними та зображеннями загиблих.

Предмет дослідження – методи сегментації зображень, які використовуються для аналізу та обробки зображень для ідентифікації загиблих.

Основна мета – покращити точність розпізнавання структур черепу на КТ-знімках. Це може мати важливе значення в медичній сфері, де точність сегментації може визначати ефективність та правильність проведення лікування.

Процес дослідження може включати в себе розробку алгоритмів або моделей машинного навчання для автоматичної сегментації, експерименти з різними архітектурами, методами обробки зображень, а також оцінку результатів за допомогою метрик ефективності. Окрім того, детальний аналіз медичних вимірювань та характеристик сегментації для забезпечення коректності та надійності розроблених методів.

Таким чином у роботі необхідно вирішити такі завдання дослідження:

- розгляд існуючих методів сегментації зображень;
- порівняльний аналіз обраних методів сегментації зображення;
- аналіз ефективності імплементації обраного методу.

2 АНАЛІЗ МЕТОДІВ СЕГМЕНТАЦІЇ ЗОБРАЖЕНЬ

2.1 Методи сегментації на основі неймереж глибинного навчання

В ході аналізу було використано такі джерела наукової інформації:

– Digital Image Processing[4], яка вивчає теорію та методи обробки цифрових зображень, автори якої розглядають основні принципи та методи обробки зображень, починаючи з перетворень, фільтрації та аналізу образів. Книга присвячена методам фільтрації та відновлення зображення, включаючи лінійні та не лінійні фільтри. Автори розглядають застосування обробки зображень у різних галузях, включаючи медицину, телекомунікації та CV;

– Computer Vision: Algorithms and Applications[5], яка розкриває тему методів виділення ознак з зображень, це включає техніки, такі як виявлення границь, кутів, дискрипторів та інші методи. Також це видання зачіпає методи машинного навчання в галузі CV, такі як класифікація, кластеризація та навчання з вчителем. Загальний підхід автора спрямований на практичне застосування навичок.

Відповідно до офіційних ресурсів U-Net, SegNet, які було обрано для реалізації методів сегментації як одні з кращих, що вказано на рисунку 3.1 [6].

Segmentation Model	Base Model	Accuracy	Precision	Recall	F1-score	FPS
SegNet	VGG 16	0.9841	0.9923	0.9689	0.9791	11
U-Net		0.9814	0.9962	0.9587	0.9752	11
SegNet	Resnet-50	0.9923	0.9984	0.9833	0.9906	11
U-Net		0.9924	0.9978	0.9836	0.9904	10
SegNet	MobileNet	0.9922	0.9969	0.9838	0.9900	18
U-Net		0.9921	0.9959	0.9844	0.9899	17

Рисунок 2.1 – Аналіз моделей

Для розробки ще мою увагу привернула англomовна публікація по методу SegNet[7], стаття по U-Net[8] та ще одна стаття на тему виявлення об'єктів[9], які дають уявлення про працю методів сегментації зображення на практиці, та схематичних прикладах. Зі статей ми розуміємо, що сегментація зображення це розділення зображення на деяку кількість сегментів для виділення об'єктів за допомогою енкодера та декодера.

2.2 Аналіз існуючих засобів обробки медичних зображень

В ході пошуку аналогів по заданим критеріям було знайдено велику кількість реалізованих проектів, а саме:

- 3D Slicer;
- ITK-SNAP;
- SimpleITK;
- DeepLabCut;
- DeepMedic.

3D Slicer[10] – це вільна та відкрита платформа для обробки, візуалізації та аналізу медичних та біомедичних зображень. Вона має ряд інструментів для сегментації, включаючи графічний інтерфейс користувача для легкої роботи з різноманітними областями тіла.

ITK-SNAP[11] – це програма для обробки медичних зображень, призначена для сегментації та візуалізації різних анатомічних структур. Розроблена на основі бібліотеки ITK, ця програма стала популярною в області медичної обробки зображень завдяки своїм функціональним можливостям та зручному інтерфейсу користувача. ITK-SNAP надає потужні інструменти для сегментації медичних зображень, користувачі можуть власноруч визначати контури або використовувати автоматичні методи сегментації під час або після виконання повної сегментації.

SimpleITK[12] – це простий інтерфейс для бібліотеки ITK, призначений для роботи з медичними зображеннями. Цей інтерфейс спрощує використання

потужних інструментів ІТК для обробки зображень, роблячи його більш доступним для користувачів з різним рівнем підготовки та досвіду в області програмування. Цей інтерфейс доступний для використання з різними мовами програмування, такими як Python, Java, C#, R та Tcl, це дає можливість вибрати найзручнішу мову для використання цього інтерфейсу у проектах. Інтерфейс використовує високо рівневі операції для зручної роботи з медичними зображеннями, такі як реєстрація, сегментація, фільтрація та обробка зображень, це дозволяє вам швидко та ефективно використовувати основні функції обробки зображень без глибокого розуміння всіх деталей.

DeepLabCut[13] – це бібліотека та програмне забезпечення для відстеження та аналізу поведінки тварин, що використовує техніки глибокого навчання. Розроблена лабораторією Макса Планка для медичних досліджень у співпраці з лабораторією Олександри Робінсон в Університеті Амстердаму. DeepLabCut дозволяє створювати точні моделі відстеження для об'єктів зображення або відео, працівник анотує кадри, вказуючи положення об'єктів, а DeepLabCut використовує ці дані для навчання моделі глибокого навчання. Проект є повністю відкритим, що означає, що можна адаптувати код до своїх потреб опираючись на докладну документацію, а також приклади використання бібліотеки.

DeepMedic[14] – це глибокий навчальний алгоритм для автоматичної сегментації медичних зображень, розроблений командою DeepMind, яка є частиною компанії Alphabet Inc. Цей алгоритм використовує глибокі згорткові нейронні мережі(CNN) для точної і автоматичної сегментації різних структур та патологій на медичних зображеннях. Цей алгоритм має можливість сегментації тривимірних медичних зображень. Це особливо важливо для завдань пов'язаних з обробкою об'ємних даних, наприклад, зі зображеннями мозку, серця чи інших органів. Також цей алгоритм може сегментувати відразу декілька класів анатомічних або патологічних областей на медичних зображеннях, що дозволяє враховувати різні структури та патології одночасно. Він має відкритий вихідний код, що дозволяє спільноті дослідників та розробників вносити свої внески,

розширювати функціональність та використовувати його для власних досліджень.

2.3 Опис експериментальних даних

Для вирішення задачі сегментації зображення знімків КТ було використано датасет з медичних зображень 52 знімка КТ різних людей з великою роздільною здатністю розміром 512x512 для вивчення деталей та малих структур. Кожне зображення має відповідні анотації та підписи, які визначають області інтересу та границі сегментів.

На рисунку 2.2 можна подивитись приклад КТ зображення

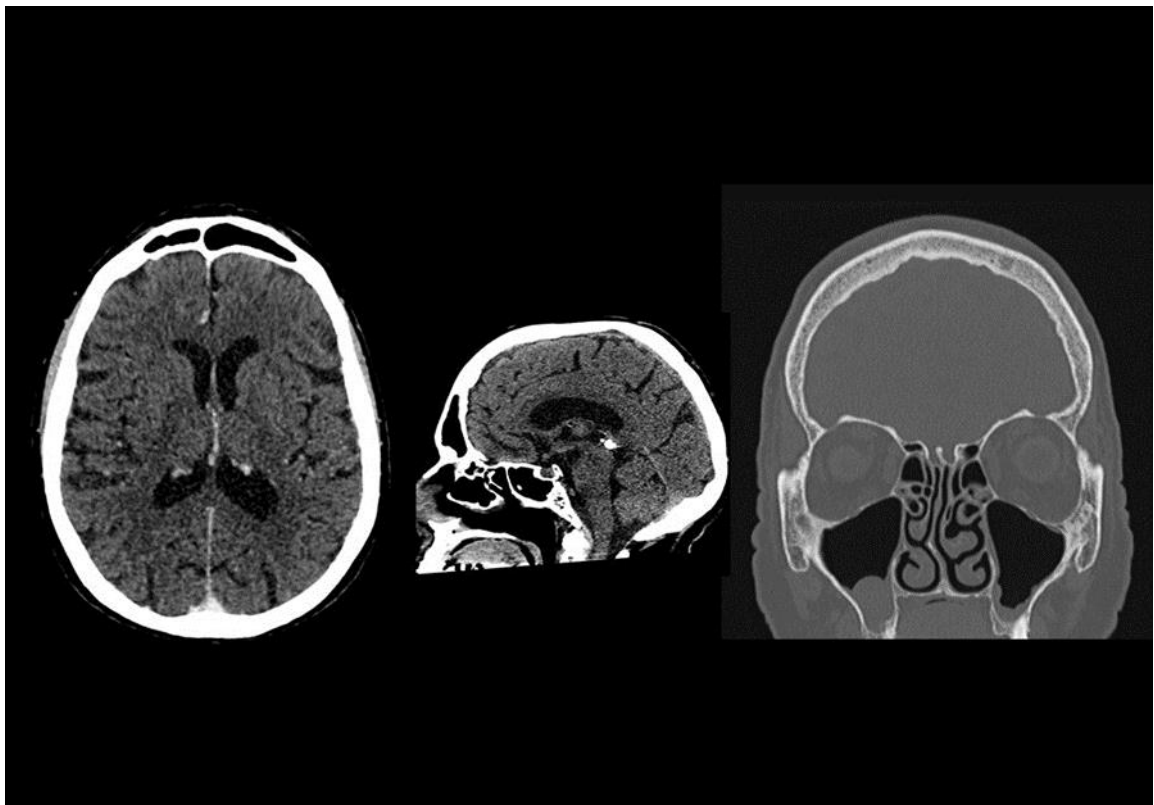


Рисунок 2.2 – Приклад зображення КТ

Вхідні дані мають ключове значення для розробки та оцінки ефективності моделей сегментації, через це не було використано синтетичних даних, які генеруються за допомогою програмно-апаратних інструментів.

На рисунку 2.3 можна побачити приклад вхідних даних.



Рисунок 2.3 – Приклад вхідних даних

У представленому датасеті було проведено розмітку медичним експертом.

2.4 Етичність питання

Питання етичності використання методів сегментації для ідентифікації загиблих пов'язане з рядом факторів, які варто враховувати.

Конфіденційність це ключовий аспект етичного використання технологій, особливо в ситуаціях які пов'язані з загибеллю чи катастрофою. При використанні методів сегментації важливо зберігати отримані дані у безпечному середовищі та забезпечити, щоб вони не потрапили у руки несанкціонованих осіб.

Приватність відноситься до права особи на власну індивідуальність та особисте життя, що значить, що збирання, аналіз та використання великої кількості візуальних даних може потенційно порушувати їх приватність, важливо дотримуватись принципів обмеження обробки особистих даних та забезпечення контролю над цими даними власника.

Фактично порушувати конфіденційність та приватність даних заборонено етичністю питання, за виключенням дозволу або згоди на використання інформації, тобто перед тим, як використовувати інформацію для сегментації людина повинна надати відповідний дозвіл, щоб стати об'єктом дослідження. Забезпечення дозволу є важливим кроком у створенні довіри між тим хто використовує технології сегментації для ідентифікації і самими особами, чи їхніми законними представниками. Таку згоду людина може дати до моменту смерті, для того щоб занести свої дані до бази, та у подальшому мати можливість бути ідентифікованою за даними технологіями.

3 ДОСЛІДЖЕННЯ МЕТОДІВ ВИРІШЕННЯ ЗАДАЧІ

3.1 Математичний опис моделі SegNet

Модель SegNet[15] є глибокою нейронною мережею для сегментації зображень, тобто для призначення класу кожному пікселю на зображенні. Вона була розроблена для вирішення задач семантичної сегментації, де метою є визначення класу об'єкта для кожного пікселя на зображенні. Основною ідеєю SegNet є використання архітектури енкодера та декодера для вилучення та відновлення інформації зображення. Розглянемо основні компоненти моделі:

- енкодер використовується для витягування та розпізнавання важливих функцій на вхідному зображенні. Він складається з кількох згорткових шарів та шарів пулінгу, які допомагають зменшити розмір зображення та виділити його ключові ознаки. У випадку SegNet це може бути модифікована архітектура конволюційної нейронної мережі CNN, наприклад така як VGG16;

- піддружні шари використовуються під час проходження сигналу через енкодер для операції зменшення розмірності, такі як MaxPooling, для зменшення просторових розмірів ознакових карт та підвищення їхньої абстракції;

- декодер відповідає за відновлення розмірності та просторового розташування сегментаційних ознак. Декодер збільшує проміжне кодове представлення до вихідного розміру зображення та відновлює сегментацію шляхом присвоєння класів кожному пікселю.;

- Апсемплінг використовується для збільшення розмірності ознакових карт, що отримані в енкодері. Це може бути здійснене, наприклад, за допомогою шарів умовного апсемплінгу або транспонованих конволюцій;

- функція втрат визначає наскільки великою є різниця між прогнозованими сегментаційними масками та справжніми масками на зображеннях.

Усі ці математичні операції і компоненти допомагають SegNet навчатись розпізнавати та сегментувати об'єкти на зображеннях під час процесу навчання,

коли мережа пристосовує свої ваги так, щоб максимізувати точність передбачень.

Візуальна архітектура SegNet на рисунку 3.1.

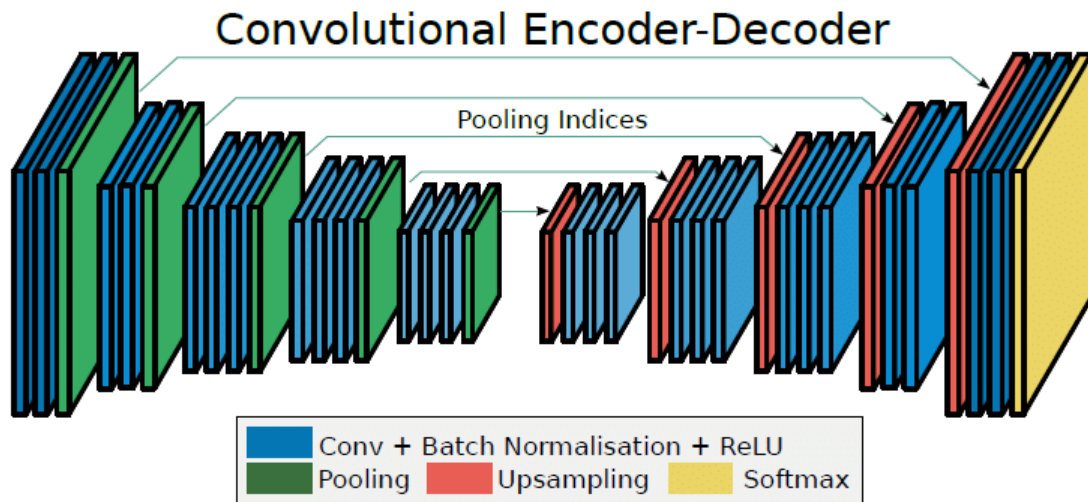


Рисунок 3.1 – Архітектура SegNet

3.2 Математичний опис моделі U-Net

Модель U-Net[16] – це архітектура глибокого навчання, розроблена для сегментації зображень. Ця архітектура має форму "U" і включає дві основні частини: енкодер та декодер. Давайте розглянемо математичний опис моделі U-Net.

Енкодер:

Згорткові шари:

– кожен згортковий шар визначається функцією згортки, наприклад, з використанням ядра W , та функцією активації, такою як ReLU;

– математично, для шару i :

$$H_i = \text{ReLU}(W_i * H_{i-1} - 1b_i), \quad (3.1)$$

де H_{i-1} – вихід попереднього шару.

Пулінгові шари:

– шари пулінгу використовують операцію пулінгу (наприклад, максимальне пулінгу) для зменшення розмірів зображення;

– математично, для пулінгового шару i :

$$H_i = \text{MaxPooling}(H_{i-1}) \quad (3.2)$$

З'єднувальний шар:

Після енкодера додається з'єднувальний шар, який з'єднує вихід останнього шару енкодера з вхідним декодера.

Декодер:

Транспоновані згорткові шари:

– Шари транспонованої згортки використовують транспоновану операцію згортки для збільшення розмірів зображення;

– Математично, для шару i :

$$H_i = \text{ReLU}(W_i * H_{i-1} + b_i) \quad (3.3)$$

де H_{i-1} – вихід попереднього шару.

Фінальний шар:

Фінальний шар використовується для отримання карти сегментації.

У всіх математичних виразах важливо враховувати ваги W та зсуви b , які піддаються оптимізації під час навчання.

Таким чином повний математичний опис може виглядати таким чином:

$$N = \text{FinalLayer}(\text{Decoder}(\text{Concatenate}(\text{Encoder}(\text{Input}))))$$

Де Encoder , Decoder та FinalLayer представляють відповідно енкодер, декодер та фінальний шар.

Візуальна архітектура U-Net на рисунку 2.3.

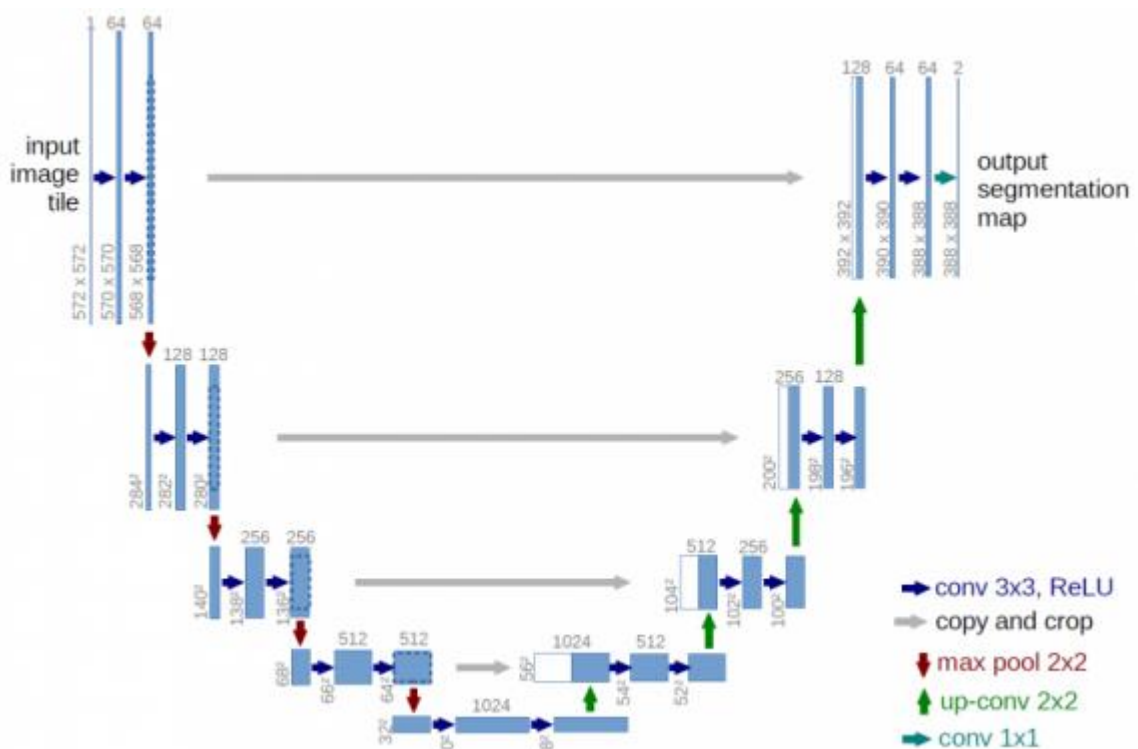


Рисунок 3.2 – Архітектура U-Net

3.3 Удосконалення методу функції втрат

Однією з популярних функцій втрат для задачі сегментації є функція перехресної ентропії (Cross-Entropy Loss). Проте, в залежності від вашої конкретної задачі та характеристик даних, може варто вдосконалити цю функцію втрат. Ось кілька можливих удосконалень:

Збалансована функція втрат:

Додаємо ваги класам, які можуть мати різний вплив на результат:

$$Loss = -\frac{1}{N} \sum_{i=1}^N (W_{pos} y_i \log(\hat{y}_i) + w_{neg} (1 - y_i) \log(1 - \hat{y}_i)) \quad (3.4)$$

Де:

N - кількість пікселів у зображенні,

y_i - істинна мітка для пікселя i ,

\hat{y}_i - передбачена ймовірність для пікселя i ,

W_{pos} та W_{neg} - ваги для позитивних та негативних класів відповідно.

Додавання додаткових термінів:

Можна додати додаткові терміни до функції втрат для врахування додаткової інформації або зменшення проблем, таких як класифікація невідомих класів або надмірне відхилення від границь об'єктів:

$$Loss = CrossLoss - \lambda_1 \sum_{i=1}^N IoU(y_i \hat{y}_i) - \lambda_2 \sum_{i=1}^N BLoss(y_i \hat{y}_i) \quad (3.5)$$

Де:

λ_1 та λ_2 - коефіцієнти для регулювання ваги додаткових термінів,

IoU - Індекс схожості Жаккара для вимірювання відповідності областей,

$BLoss$ - функція втрат для вимірювання точності границь об'єктів.

Модифікація збіжності:

Модифікуємо функцію втрат для забезпечення кращої збіжності, особливо при роботі з несиметричними даними:

$$Loss = CrossLoss + \beta \sum_{i=1}^N (1 - \exp(-\frac{1}{2\sigma^2} (d_i - \mu)^2)) \quad (3.6)$$

Де

β - коефіцієнт для регулювання ваги додаткового терміну,

d_i - відстань від передбаченої границі до істинної границі для пікселя i ,

μ та σ - параметри для контролю форми функції.

3.4 Особливості застосування методів обробки зображення

Для підвищення якості сегментації зображення часто використовують різні способи обробки зображень. До таких відносяться як звичайні операції зміни розміру або кольорової палітри, так і застосування більш важких фільтрів.

Прикладом обробки зображення можливо використовувати декілька методів текстурного аналізу: статистичний – матриця збіжності рівнів сірого, не лінійний геометричний фільтр Кримменса та простий фільтр нерізкого маскування, які часто використовуються для обробки зображень. Основною задачею останніх двох методів є підвищення якості зображення.

Матриця збіжності рівнів сірого(Gray Level Cooccurrence Matrix – GLCM) – один з самих ранніх методів отримання текстурних ознак, якій був запропонований Хараліком, Шанмаганом та Дінстейном в 1971 році[17, 18]. Основою даного метода є припущення, що інформація про текстуру зображення полягає в просторовому відношенні відтінків сірого, що містяться у зображенні друг до друга.

Кожен елемент матриці GLCM вираховується по формулі:

$$G_{(\Delta x, \Delta y)}(i, j) = \sum_{y=0}^{h-\Delta y} \sum_{x=0}^{w-\Delta x} F, \quad (3.7)$$

$$F = \begin{cases} 1, & \text{if } I(x, y) = i \text{ та } I(x + \Delta x, y + \Delta y) = j, \text{ to} \\ 0, & \end{cases}$$

Де i, j – яскравість пікселя, $(\Delta x, \Delta y)$ – задана околиця, $I(x, y)$ – яскравість пікселя з координатами, (x, y) , w, h – ширина та висота вихідного зображення.

В залежності від значень $(\Delta x, \Delta y)$ можливо отримати різні матриці збіжності, вибираючи для порівняння яскравість сусідніх пікселів під різними кутами. Це означає, що шляхи обчислення збіжності можуть варіюватися в залежності від обраного кута огляду, що в свою чергу впливає на отримані результати. Такий підхід дозволяє аналізувати зміни яскравості пікселів з різних

точок зору, розширюючи можливості алгоритмів обробки зображень та виокремлення деталей зображення.

Матрицю сусідніх пікселів зображено на рисунку 3.3.



Рисунок 3.3 – Матриця сусідніх пікселів

Зазвичай використовуються матриці $G_{(1,0)}$, $G_{(1,1)}$, $G_{(0,1)}$, $G_{(-1,-1)}$.

Отриманні матриці нормуються за формулою:

$$P(\Delta x, \Delta y) = W G_{(\Delta x, \Delta y)}(i, j),$$

$$W = \frac{1}{(w - \Delta x)(h - \Delta y)}. \quad (3.8)$$

В результаті чого виходить чотири матриці. Можемо сказати, що отримані матриці представляють собою оцінки сумісних щільностей розподілу ймовірностей того, що два пікселя з яскравістю i та j будуть сусідами. Отримані матриці використовуються для обчислення характеристик Хараліка, таких як: контраст, відмінність, однорідність та інших.

Геометричний фільтр Кримменса розроблений Т.Р. Кримменсом заснований для застосування до зображення ітеративного алгоритма. Він був розроблений для видалення з радіолокаційних зображень шумів типу «сіль-перець»[19].

Можна сказати, що це одновимірний алгоритм, який застосовується послідовно у чотирьох різних напрямках на двомірному зображенні: горизонтальному, вертикальному та двох діагональних напрямках. Значення інтенсивності у центрі кожного вікна розміром 3x3 порівнюється з найближчим оточуючими його пікселями. Залежно від результатів порівняння, значення інтенсивності центрального пікселя збільшується, або зменшується.

Розглянемо алгоритм фільтру для вертикального напрямлення. Нехай на рисунку 3.4 зображена матриця зображення.

a	e	i	m	q
b	f	j	n	r
c	g	k	o	s
d	h	l	p	t

Рисунок 3.4 – Приклад застосування геометричного методу

Алгоритм для пікселя b має вигляд для кожної ітерації:

1. Налаштування для темних пікселів:

$$\begin{aligned}
 &a. \text{if } a \geq b + 2 \text{ then } b = b + 1, \\
 &b. \text{if } a > b \text{ and } b \leq a \text{ then } b = b + 1, \\
 &c. c > b \text{ and } b \leq a \text{ then } b = b + 1, \\
 &d. c \geq b + 2 \text{ then } b = b + 1.
 \end{aligned} \tag{3.9}$$

2. Налаштування для світлих пікселів:

$$\begin{aligned}
 &a. \text{if } a \leq b - 2 \text{ then } b = b - 1, \\
 &b. \text{if } a < b \text{ and } b \geq a \text{ then } b = b - 1, \\
 &c. c < b \text{ and } b \geq a \text{ then } b = b - 1, \\
 &d. c \leq b - 2 \text{ then } b = b - 1.
 \end{aligned} \tag{3.10}$$

В результаті чого різниця між значеннями інтенсивності сусідніх пікселів зменшується, а зображення згладжується. Рівень згладжування залежить від кількості ітерацій.

Також розглянемо простий фільтр нерізкого маскування, який є методом, що використовується для підвищення різкості зображення[20]. Основа ідея в використанні розмитого зображення для створення маски, яка згодом об'єднується з вихідним зображенням. В результаті виходить більш різке зображення чим оригінал.

Для побудови такого зображення використовується формула:

$$f_{sharp}(x, y) = f(x, y) + kg(x, y) \quad (3.11)$$

$$g(x, y) = f(x, y) - f_{smooth}(x, y) \quad (3.12)$$

Де $f(x, y)$ – вихідне зображення, $f_{smooth}(x, y)$ – вихідне зображення з фільтром по згладжуванню (по Гауссу), $g(x, y)$ – зображення краю, k – константа.

3.5 Вибір мови програмування.

Python – це високорівнева мова програмування загального призначення з динамічною типізацією та автоматичним управлінням пам'яттю. Мова є цілком об'єктно-орієнтовною. Мова орієнтована на продуктивність розробки та якість коду. Синтаксис мови є мінімалістичним за рахунок чого рідко виникає необхідність звертатися до документації. Сама мова є інтерпретованою і використовується в тому числі для написання скриптів. Різницю зображено на рисунку 2.1[21]. Із недоліків мови те що вона має дещо нижчу швидкість та більш високе споживання пам'яті програм порівняно з аналогічними програмами написаними на компільованих мовах, таких як C, C++, C#.

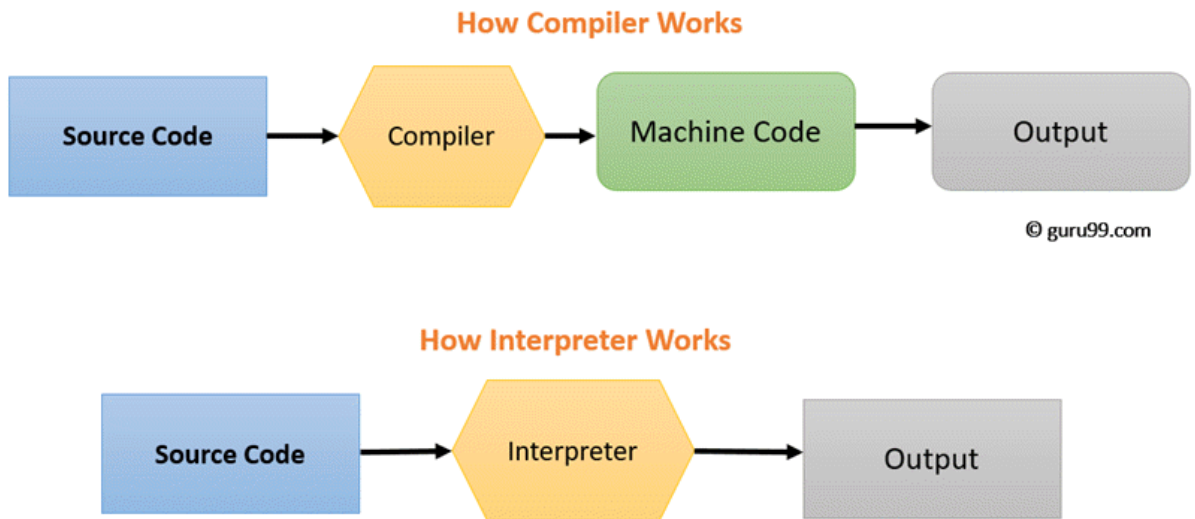


Рисунок 3.5 – Різниця між інтерпретатором та компілятором

Також із причин обрати Python:

По-перше Python має широкий вибір бібліотек та фреймворків, в екосистемі мови існує велика кількість інструментів для розробки машинного навчання та нейронних мереж які ми будемо використовувати в ході дослідження.

По-друге мова має велику активну спільноту розробників, яка надає підтримку, допомогу та розвиток багатьох інструментів для машинного навчання та нейронних мереж.

По-третє Python має високу продуктивність, простота та наявність стандартів дають програмістам можливість писати менше строк коду для виконання більшої кількості задач.

В четверте ця мова має високу портативність, програми написані на Python можливо запускати на різних операційних системах без змін коду(якщо немає системно-залежних функцій)

Також, Python має безліч додаткових бібліотек для обробки, сортування даних та візуалізації результатів та іншого пов'язаного з роботою в галузі машинного навчання.

Але є й недоліки, серед вже наведених є збільшена потреба до пам'яті, так як інтерпретатор Python-а використовує додаткові ресурси для того щоб управляти динамічною типізацією та автоматично видаляю «мусор», така поведінка може привести до підвищеної потреби пам'яті.

3.6 Вибір бібліотек

3.6.1 Бібліотека Keras

Keras[22] - це високорівнева бібліотека для машинного навчання, яка надає зручний і модульний інтерфейс для побудови, тренування та експериментів з нейронними мережами. Важливою особливістю Keras є те, що вона побудована поверх фреймворків більш низького рівня, таких як Tensorflow або Theano, що дозволяє легко використовувати потужність цих фреймворків, при цьому надаючи зручний інтерфейс для користувачів.

Основні характеристики Keras включають у себе модульність, що дозволяє легко збирати нейронні мережі з різноманітними шарами. Модульність Keras дозволяє додавати, видаляти або змінювати шари, що робить її ефективним інструментом для експериментів і розробки нових архітектур нейронних мереж.

Крім того, Keras володіє легкістю використання, що робить її доступною для початківців у машинному навчанні. Інтуїтивний і зрозумілий інтерфейс спрощує процес створення, тренування та оцінювання моделей.

TensorBoard - це інструмент для візуалізації процесу тренування моделей машинного навчання. Інтеграція Keras з TensorBoard дозволяє вам відстежувати метрики, аналізувати графи нейронних мереж та вивчати інші аспекти тренування за допомогою графіків і графічного відображення. Це стає корисним інструментом для відлагодження та оптимізації моделей.

3.6.2 Бібліотека Tensorflow

Tensorflow[23] – це відкрита бібліотека для машинного навчання та глибого навчання. Ця бібліотека була розроблена командою Google та широко використовується для розробки та навчання нейронних мереж. Однією з ключових особливостей Tensorflow є підтримка статичних та динамічних обчислювальних графів.

Tensorflow підтримує як статичні так і динамічні обчислювальні графи. У статичному режимі граф визначається та оптимізується до початку виконання, що може призводити до покращення продуктивності. У динамічному режимі граф будується та оптимізується під час виконання програми, надаючи більшу гнучкість та зручність для прототипування.

Tensorflow може активно використовувати GPU та TPU для прискорення обчислень.

Tensorflow має інтеграцію з TensorBoard. TensorBoard – це інструмент для візуалізації графів, метрик тренування та інших параметрів.

3.7 Запуск моделей

Програмування та запуск моделей проводиться через безкоштовний хостинг Google Colaboratory. Потрібно зареєструватися, створити віртуальну машину та налаштувати її, після чого завантажити `.ipynb` файл та провести навчання на віртуальному сервері гугл.

4 ОПИС СТВОРЕНОЇ ПРОГРАМИ

4.1 Створення коду

Як було показано вище для розробки цих методів сегментації зображення обрано мову програмування Python, як кращу для такого плану задач, разом з нею буде використано бібліотеки Keras та TensorFlow.

Першочерговою задачею є побудова правильних моделей U-Net та SegNet для навчання, ці методи повинні правильно сегментувати області на існуючому датасеті. TensorFlow як основний фреймворк для реалізації та тренування моделей. Keras, який є інтерфейсом вищого рівня для TensorFlow, дозволив зручно визначати та конфігурувати архітектури U-Net та SegNet.

Після написання правильних моделей потрібно навчити ці моделі на існуючому датасеті зображень КТ черепу, та порівняти їх за вище зазначеними критеріями оцінки.

4.2 Модель U-Net

У даній моделі присутні такі елементи:

- InputLayer, який є вхідним шаром Model_in та приймає зображення з формою (h, w, channels), де h та w – розміри зображення, а channels – кількість кольорових каналів(RGB);
- Down Convolutions, це енкодер, кожен блок Down Convolution складається з двох загорткових слоїв Conv2D та функцією активації ReLU, шаром BatchNormalization та пулінгом MaxPooling2D. Цей етап має чотири рівні conv1, conv2, conv3 та conv4;
- Bottleneck Region це окремий рівень conv5 який не має пулінгу, він має два загорткових шара та являє собою пляшкову шийку мережі;

– Up Convolution, це декодер, кожен блок якого складається з шара BatchNormalization та Conv2DTranspose для збільшення розміру зображення(деконволюції) та двох згорткових шарів Conv2D. Цей етап має чотири рівні conv6, conv7, conv8 та conv9;

– Final 1x1 Convolution являє собою вихідний шар conv10 та виконує функцію фінального згорткового шару з ядром 1x1 та функцією активації sigmoid. Цей етап використовується для отримання бінарної маски сегментації.

Модель U-Net яку було реалізовано представляє собою енкодер-декодерну архітектуру код якої зображено на рисунку 4.1.

```
# down convolutions
conv1 = BatchNormalization(scale=True)(model_in)
conv1 = Conv2D(64, (3, 3), activation='relu',kernel_initializer="he_normal", padding="same")(conv1)
conv1 = Conv2D(64, (3, 3), activation='relu',kernel_initializer="he_normal", padding="same")(conv1)
down1 = MaxPooling2D((2, 2), strides=2)(conv1)

conv2 = BatchNormalization(scale=True)(down1)
conv2 = Conv2D(128, (3, 3), activation='relu',kernel_initializer="he_normal", padding="same")(conv2)
conv2 = Conv2D(128, (3, 3), activation='relu',kernel_initializer="he_normal", padding="same")(conv2)
down2 = MaxPooling2D((2, 2), strides=2)(conv2)

conv3 = BatchNormalization(scale=True)(down2)
conv3 = Conv2D(256, (3, 3), activation='relu',kernel_initializer="he_normal", padding="same")(conv3)
conv3 = Conv2D(256, (3, 3), activation='relu',kernel_initializer="he_normal", padding="same")(conv3)
down3 = MaxPooling2D((2, 2), strides=2)(conv3)

conv4 = BatchNormalization(scale=True)(down3)
conv4 = Conv2D(512, (3, 3), activation='relu',kernel_initializer="he_normal", padding="same")(conv4)
conv4 = Conv2D(512, (3, 3), activation='relu',kernel_initializer="he_normal", padding="same")(conv4)
down4 = MaxPooling2D((2, 2), strides=2)(conv4)

# bottleneck region
conv5 = BatchNormalization(scale=True)(down4)
conv5 = Conv2D(1024, (3, 3), activation='relu',kernel_initializer="he_normal", padding="same")(conv5)
conv5 = Conv2D(1024, (3, 3), activation='relu',kernel_initializer="he_normal", padding="same")(conv5)

# up convolutions
conv6 = BatchNormalization(scale=True)(conv5)
conv6 = Conv2DTranspose(512, (2, 2), activation='relu',strides=(2,2),kernel_initializer="he_normal", padding="same")(conv6)
concat1 = concatenate([conv4, conv6], axis=3)
conv6 = Conv2D(512, (3, 3), activation='relu',kernel_initializer="he_normal", padding="same")(concat1)
conv6 = Conv2D(512, (3, 3), activation='relu',kernel_initializer="he_normal", padding="same")(conv6)

conv7 = BatchNormalization(scale=True)(conv6)
conv7 = Conv2DTranspose(256, (2, 2), activation='relu',strides=2,kernel_initializer="he_normal", padding="same")(conv7)
concat2 = concatenate([conv3, conv7], axis=3)
conv7 = Conv2D(256, (3, 3), activation='relu',kernel_initializer="he_normal", padding="same")(concat2)
conv7 = Conv2D(256, (3, 3), activation='relu',kernel_initializer="he_normal", padding="same")(conv7)

conv8 = BatchNormalization(scale=True)(conv7)
conv8 = Conv2DTranspose(128, (2, 2), activation='relu',strides=2,kernel_initializer="he_normal", padding="same")(conv8)
concat3 = concatenate([conv2, conv8], axis=3)
conv8 = Conv2D(128, (3, 3), activation='relu',kernel_initializer="he_normal", padding="same")(concat3)
conv8 = Conv2D(128, (3, 3), activation='relu',kernel_initializer="he_normal", padding="same")(conv8)

conv9 = BatchNormalization(scale=True)(conv8)
conv9 = Conv2DTranspose(128, (2, 2), activation='relu',strides=2,kernel_initializer="he_normal", padding="same")(conv9)
concat4 = concatenate([conv1, conv9], axis=3)
conv9 = Conv2D(64, (3, 3), activation='relu',kernel_initializer="he_normal", padding="same")(concat4)
conv9 = Conv2D(64, (3, 3), activation='relu',kernel_initializer="he_normal", padding="same")(conv9)

# final 1x1 convolution
conv10 = Conv2D(1,kernel_size=(1,1),activation="sigmoid",kernel_initializer="he_normal", padding="same")(conv9)
```

Рисунок 4.1 – Реалізація моделі U-Net

Даний архітектурний підхід широко використовується у завданнях сегментації зображень, де основною метою є виділення окремих об'єктів на вхідних зображеннях. Структура цієї моделі може бути описана як U-подібна, що вказує на те, що вона включає в себе компоненти енкодера, пляшкову шийку та декодера.

Енкодер відповідає за аналіз та взяття вхідного зображення, конвертуючи його у багато розмірний вектор. Цей вектор представляє важливі ознаки та інформацію, які модель використовує для подальшого розуміння контексту зображення. Пляшкова шийка служить як звуження для отриманого вектора, спрощуючи його представлення, але зберігаючи ключові деталі.

Декодер відповідає за відтворення структури та деталей вихідного зображення на основі обрізаного та абстрактного вектора, отриманого від пляшкової шийки. Такий підхід дозволяє моделі ефективно виділяти та відтворювати об'єкти на зображенні, забезпечуючи точну та високоякісну сегментацію.

4.3 Модель SegNet

У даній моделі присутні такі елементи:

- InputLayer, який приймає зображення з трьома кольоровими каналами;
- Encoder у якому шари Conv2D які використовуються для виконання згортки з ядром 3x3 та функцією ReLU та параметром same для збереження розміру. У даній часті маємо два послідовних згорткових шару для кожного з трьох рівней енкодеру(conv1, conv2, conv3). Шар MaxPooling2D використовується для зменшення розміру зображення для кожного рівня згортки;
- Decoder має шари UpSampling2D які використовуються для збільшення розміру зображення після кожного рівня декодера. Кожен шар Conv2D виконує функцію декодування(зворотна згортка) після кожного UpSampling2D. Для

кожного з рівнів декодера використовується два згорткових шара, а саме deconv1, deconv2 та deconv3;

– Output Layer це вихідний шар який виконує фінальну згортку з ядром 1x1 та функцію активації sigmoid.

Загальний зміст полягає у створенні повнозгорткової нейронної мережі для виконання завдання сегментації зображень.

Данна реалізація архітектурної моделі SegNet з використанням бібліотеки Keras та TensorFlow. Реалізацію цієї моделі можна подивитись на рисунку 4.2.

```
def SegNet(input_shape=(640, 640, 3), num_classes=1):
    inputs = Input(shape=input_shape)

    # Encoder
    conv1 = Conv2D(64, (3, 3), activation='relu', padding='same')(inputs)
    conv1 = Conv2D(64, (3, 3), activation='relu', padding='same')(conv1)
    pool1 = MaxPooling2D(pool_size=(2, 2))(conv1)

    conv2 = Conv2D(128, (3, 3), activation='relu', padding='same')(pool1)
    conv2 = Conv2D(128, (3, 3), activation='relu', padding='same')(conv2)
    pool2 = MaxPooling2D(pool_size=(2, 2))(conv2)

    conv3 = Conv2D(256, (3, 3), activation='relu', padding='same')(pool2)
    conv3 = Conv2D(256, (3, 3), activation='relu', padding='same')(conv3)
    pool3 = MaxPooling2D(pool_size=(2, 2))(conv3)

    # Decoder
    up3 = UpSampling2D(size=(2, 2))(pool3)
    deconv3 = Conv2D(256, (3, 3), activation='relu', padding='same')(up3)
    deconv3 = Conv2D(256, (3, 3), activation='relu', padding='same')(deconv3)

    up2 = UpSampling2D(size=(2, 2))(deconv3)
    deconv2 = Conv2D(128, (3, 3), activation='relu', padding='same')(up2)
    deconv2 = Conv2D(128, (3, 3), activation='relu', padding='same')(deconv2)

    up1 = UpSampling2D(size=(2, 2))(deconv2)
    deconv1 = Conv2D(64, (3, 3), activation='relu', padding='same')(up1)
    deconv1 = Conv2D(64, (3, 3), activation='relu', padding='same')(deconv1)

    outputs = Conv2D(num_classes, (1, 1), activation='sigmoid')(deconv1)

    model = Model(inputs=inputs, outputs=outputs)
    return model
```

Рисунок 4.2 – Реалізація моделі SegNet

4.4 Різниця методів

Обидва підходи U-Net та SegNet є архітектурами згорткових нейронних мереж, розробленими для розв'язання завдань сегментації зображень.

Основні відмінності між підходами:

1. Архітектурна структура:

– SegNet: Має класичну енкодер-декодерну архітектуру. У енкодері зображення послідовно стискається за допомогою згорткових шарів та пулінгу, а потім у декодері відновлюється до початкового розміру за допомогою операцій збільшення розміру та додаткових згорткових шарів;

– U-Net: Також має енкодер-декодерну структуру, але з додаванням "пропускних" з'єднань між відповідними шарами енкодера та декодера. Ці з'єднання допомагають передавати інформацію про високорівневі ознаки безпосередньо з енкодера в декодер, що може бути корисним для точної сегментації об'єктів.

2. Функція активації на виході:

– SegNet: Використовує сигмоїдну функцію активації на виході, що робить її підходящою для завдань бінарної сегментації, наприклад, виділення об'єктів на зображенні;

– U-Net: Також використовує сигмоїдну функцію активації, що дозволяє застосовувати модель для завдань бінарної сегментації.

3. Параметри та ініціалізація ваг:

– SegNet: У наведеному коді використовуються згорткові шари із функцією активації ReLU та ініціалізацією ваг `he_normal`. Блоки енкодера та декодера мають схожі параметри;

– U-Net: Також використовує згорткові шари з функцією активації ReLU та ініціалізацією ваг `he_normal`. У коді U-Net також використовуються шари `BatchNormalization`, а в декодері застосовуються операції деконволюції (`Conv2DTranspose`), а також пропускаючи з'єднання для передачі інформації.

4. Розміри та кількість шарів:

- SegNet: Має три рівні енкодера та три рівні декодера, створюючи структуру у вигляді "U";
- U-Net: Має чотири рівні енкодера та чотири рівні декодера, включаючи бутликове горло (як у наведеному коді). Включає пропускаючи з'єднання для більш ефективного використання ознак.

4.5 Метрики ефективності для оцінювання результатів сегментації

Для порівняння було використано IoU, Dice Coefficient та Pixel-wise Accuracy для кожного тестового зразка.

Цей код призначений для оцінки якості сегментації моделі для тестового набору за допомогою цих метрик.

Основні кроки коду для оцінки:

- Код для отримання метрики IoU розглянемо на рисунку 4.3.

```
# Define a function to calculate Intersection over Union (IoU)
def calculate_iou(y_true, y_pred):
    intersection = np.logical_and(y_true, y_pred)
    union = np.logical_or(y_true, y_pred)
    iou_score = np.sum(intersection) / np.sum(union)
    return iou_score

# Calculate IoU for each test sample
iou_scores = []
for i in range(len(val_masks)):
    iou = calculate_iou(val_masks[i], predicted_masks[i] > 0.5) # Using threshold for predicted masks
    iou_scores.append(iou)

average_iou = np.mean(iou_scores)
print("Average IoU:", average_iou)
```

Рисунок 4.3 – Реалізація метрики IoU

Функція `calculate_iou` розраховує метрику `ioU` для двох масок `y_true` (дійсна маска) та `y_pred` (прогнозована маска). Це вимірює ступінь перекриття між областями, які визначаються цими двома масками. Цю функцію у циклі

використано для обчислення IoU для кожної пари дійсної прогнозованої маски, а потім розраховується середнє значення IoU для всього тестового набору;

- Код для отримання метрики Dice Coefficient розглянемо на рисунку 4.4.

```
# Define a function to calculate Dice coefficient
def calculate_dice_coefficient(y_true, y_pred):
    intersection = np.sum(y_true * y_pred)
    dice_coefficient = (2.0 * intersection) / (np.sum(y_true) + np.sum(y_pred))
    return dice_coefficient

# Calculate Dice coefficient for each test sample
dice_scores = []
for i in range(len(val_masks)):
    dice = calculate_dice_coefficient(val_masks[i], predicted_masks[i] > 0.5)
    dice_scores.append(dice)

average_dice = np.mean(dice_scores)
print("Average Dice Coefficient:", average_dice)
```

Рисунок 4.4 – Реалізація метрики Dice Coefficient

Функція `calculate_dice_coefficient` розраховує коефіцієнт Дайса для дійсної та прогнозованої маски. Ця метрика вимірює схожість між двома масками. Після чого цю функцію використовую для отримання коефіцієнта Дайса для кожного тестового зразка та отримання середнього значення для всього тестового набору;

- Код для отримання метрики Pixel-wise розглянемо на рисунку 4.5.

```
# Define a function to calculate pixel-wise accuracy
def calculate_pixel_accuracy(y_true, y_pred):
    correct_pixels = np.sum(y_true == (y_pred > 0.5))
    total_pixels = y_true.size
    accuracy = correct_pixels / total_pixels
    return accuracy

# Calculate pixel-wise accuracy for each test sample
pixel_accuracy_scores = []
for i in range(len(val_masks)):
    accuracy = calculate_pixel_accuracy(val_masks[i], predicted_masks[i])
    pixel_accuracy_scores.append(accuracy)

average_pixel_accuracy = np.mean(pixel_accuracy_scores)
print("Average Pixel-wise Accuracy:", average_pixel_accuracy)
```

Рисунок 4.5 – Реалізація метрики Pixel-wise

Функція `calculate_pixel_accuracy` розраховує точність для кожного пікселя на основі порівняння дійсної та прогнозованої масок. Дана функція після цього використовується для обрахування точності для кожного тестового зразка.

Для порівняння методів сегментації були використані ці методи, результати яких для U-Net та SegNet. Результати цих метрик:

1. Average IoU:

– U-Net: 0.0186;

– SegNet: 0.0221;

– Високий індекс об'єднання вказує на добре об'єднані області сегментації.

Результати у метода SegNet трохи вищі ніж у U-Net, що свідчить про деяку перевагу цього методу у порівнянні з U-Net.

2. Average Dice Coefficient:

– U-Net: 23.18;

– SegNet: 27.28;

– Коефіцієнт Дайса також вказує на схожість між прогнозами і справжніми мітками. У SegNet вищий коефіцієнт Дайса, що свідчить про кращу схожість.

3. Average Pixel-wise Accuracy:

– U-Net: 616.21;

– SegNet: 616.23;

– Середня піксельна точність вказує на загальну точність класифікації пікселів. Результати метрики майже однакові для обох методів.

Візуальну інфографіку метрик відображено на рисунку 4.6.

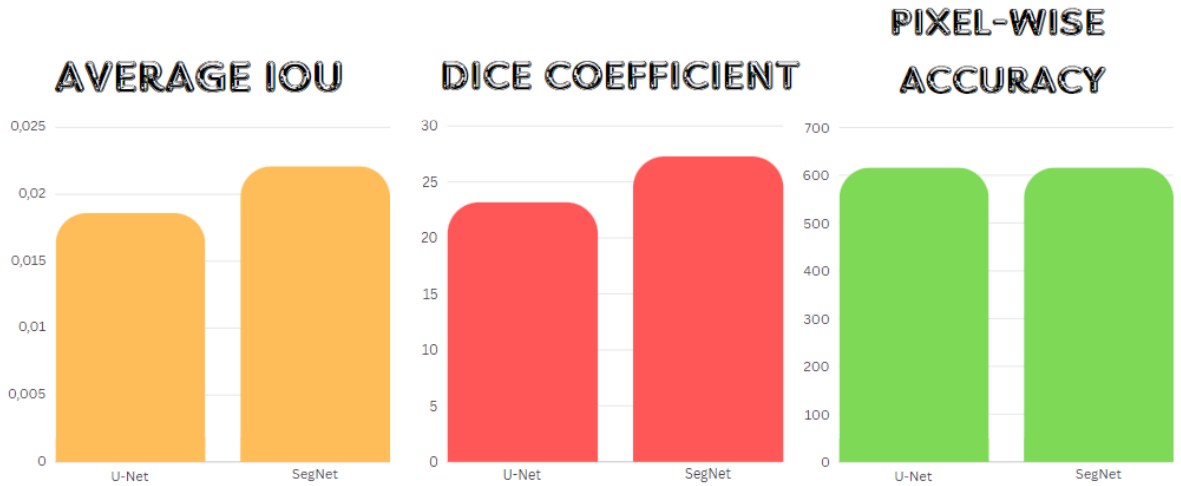


Рисунок 4.6 – Метрики оцінювання

SegNet показує деяка покращення у порівнянні з U-Net з точки зору середнього індексу об'єднання та коефіцієнта Дайса. Однак важливо розуміти, що вибір метрик повинен враховувати конкретні задачі, та вимогу до застосування.

4.6 Результат сегментації

Приклад добре сегментованого зображення методом U-Net на рисунку 4.7. Контури сегментації чіткі та зрозумілі, форма сегментації також хороша.

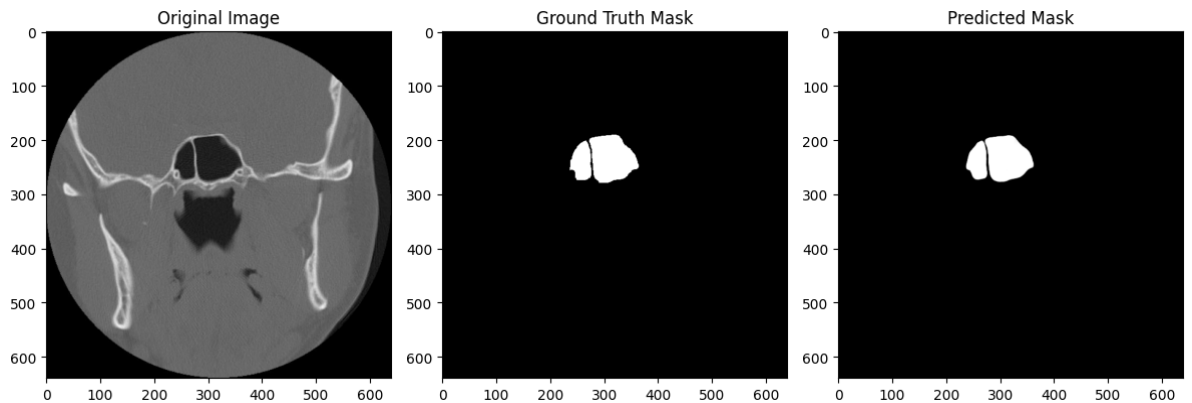


Рисунок 4.7 – Сегментоване зображення методом U-Net

На рисунку 4.8 можемо побачити досить погану сегментацію методом U-Net, місцями контури не чіткі, та у місці де повинне бути повне покриття маскою воно відсутнє.

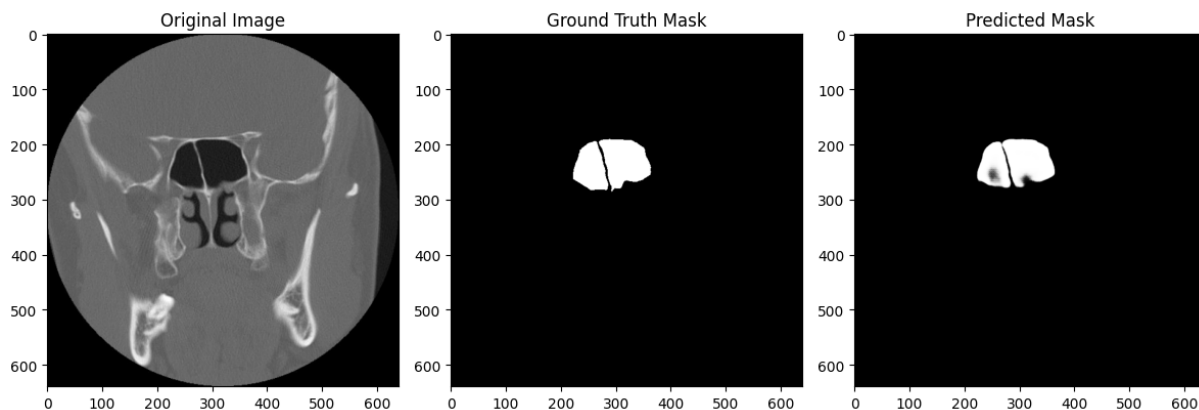


Рисунок 4.8 – Сегментоване зображення методом U-Net

Отже U-Net досить добре сегментував дані зображення, розглянемо ті ж самі КТ знімки які сегментував метод SegNet.

На рисунку 4.9 можемо побачити сегментоване зображення методом SegNet, яке на рисунку 4.7 було сегментовано методом U-Net.

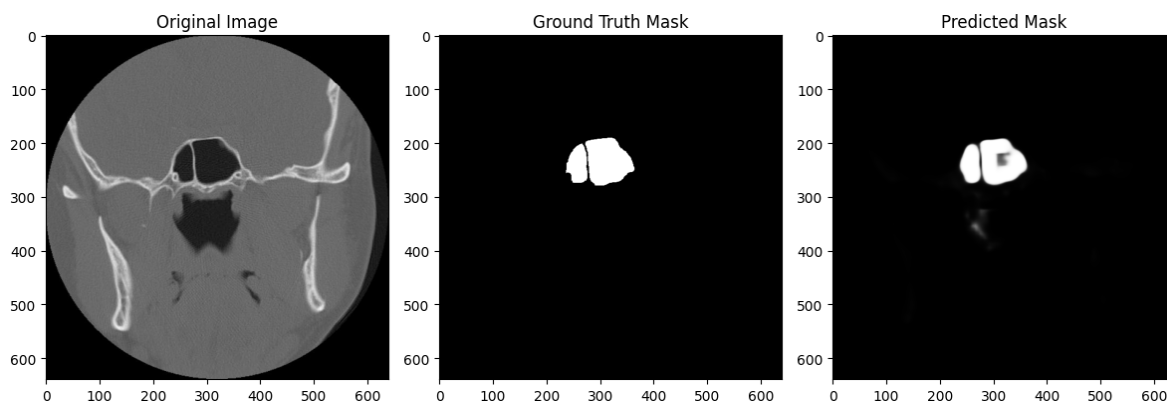


Рисунок 4.9 – Сегментоване зображення методом SegNet

Як ми бачимо на рисунку цей знімок КТ модель SegNet сегментовано гірше, на сегментації помітні артефакти, не чіткі лінії та інше.

На рисунку 4.10 розглянемо сегментацію SegNet, яке було вище сегментовано U-Net.

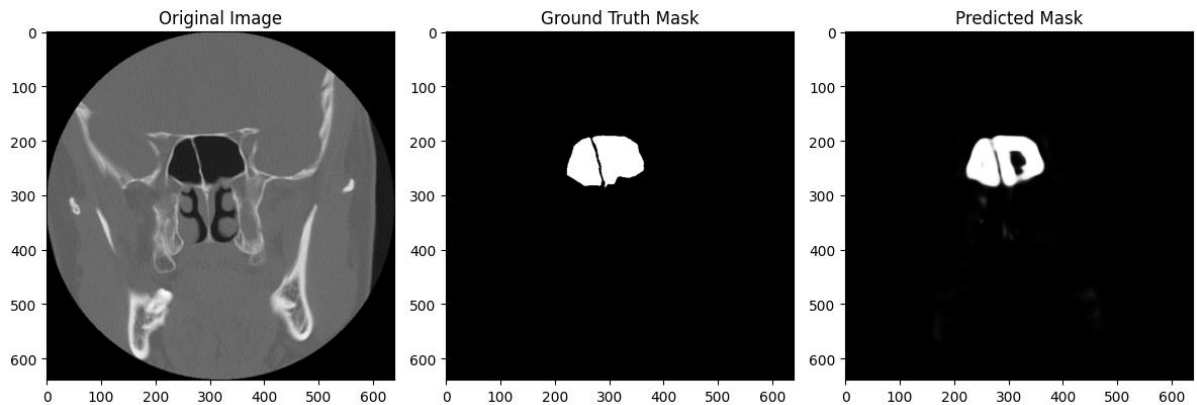


Рисунок 4.10 – Сегментоване зображення методом SegNet

Отже на рисунку також ми бачимо артефакти на сегментованій частині в порівнянні з маскою.

4.7 Можливість покращення результату

За для покращення результатів сегментації зображення потрібно проведення більшої кількості експериментів, а саме:

- Застосування різноманітних технік аугментації, таких як повороти, масштабування, відображення та зміна контрастності, це дозволяє покращити узагальнення моделі та зменшити перенавчання;
- Експерименти з оптимізацією параметрів, таких як learning rate, batch size та ерор навчання для досягнення оптимальних результатів;
- Додавання більшої кількості шарів до мережі може покращити її здатність вивчати складні ієрархії ознак. Однак такі глибокі моделі можуть вимагати більше обчислювальних ресурсів;

- Використання більш складних блоків кодування та декодування може бути корисним, щоб модель краще витягувала та відновлювала ознаки;
- Вибір різних функцій втрат, таких як Dice Loss або Jaccard Index також може сприяти кращій збіжності моделі.

ВИСНОВКИ

В результаті виконання кваліфікаційної роботи були досліджені методи сегментації зображень для задачі ідентифікації загиблих.

У висновках варто підкреслити, що робота зосереджується на дослідженні та застосуванні методів сегментації зображень для ефективної ідентифікації загиблих. Для досягнення цієї мети було використані дві передові архітектури глибокого навчання U-Net та SegNet.

Результати дослідження підтверджують, що обидві архітектури успішно впорались з завданням сегментації зображень для ідентифікації загиблих. Вони демонструють високу точність та швидкість обробки зображень, що робить їх ефективними в умовах надзвичайних ситуацій.

Повнота виконання завдання підтверджується детальним аналізом отриманих показників метрик, які відображають ключові аспекти обох архітектурних стилів. Моделі успішно інтегрують різні вимірювання, такі як продуктивність, масштабованість, вартість впровадження та експлуатації, надаючи комплексний погляд на переваги та недоліки кожного методу.

У рамках цієї кваліфікаційної роботи були досліджені методи та алгоритми для сегментації зображень, та реалізовано дві архітектури для сегментації зображень, а саме U-Net та SegNet. Основною перевагою цих архітектур є простота у використанні, що робить їх доступною для широкого кола фахівців. Реалізації архітектур була розроблена таким чином, щоб забезпечити високу точність, простоту у використанні а також продуктивність.

Крім того, досягнення високої точності моделі відкриває шлях до подальших досліджень та удосконалення та є значним кроком у розвитку напрямку сегментації зображень у медичній сфері.

До того ж застосування цих архітектур має потенціал покращити ефективність рятувальних операцій та ідентифікації загиблих у різних надзвичайних ситуаціях. Дані результати можуть слугувати основою для

впровадження в практику сучасних методів обробки зображень у сфері надзвичайних ситуацій та рятувальних операцій.

Тема дослідження методів сегментації зображень відповідає основним і науковим напрямкам кафедри системотехніки, які включають системний аналіз, моделювання та проектування. Ця робота розширює існуючі знання в області обробки зображень та містить рекомендації щодо імплементації методів на основі нейромереж глибокого навчання.

Досліджені моделі можуть бути використані для навчальних цілей, а також і в якості основи для подальших досліджень, спрямованих на розв'язання задач обробки зображень.

Таке використання може допомогти у підготовці фахівців, які зможуть вдало поєднати теоретичні знання з практичним досвідом, забезпечуючи глибоке розуміння сучасних вимог до обробки зображень та проектування архітектур для сегментації зображень.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Методичні вказівки до організації виконання та захисту атестаційної роботи на здобуття першого (бакалаврського) рівня вищої освіти для студентів усіх форм навчання спеціальності 151 – «Автоматизація та комп'ютерноінтегровані технології» за освітньою програмою «Системна інженерія» / Упорядники: І.В. Гребеннік, В.Г. Іванов, Б.О. Колесник, А.С. Нечипоренко, П.Е. Ситнікова, О.С.Чорна – Харків: ХНУРЕ, 2019. – 58 с.
2. Зниклі безвісти. URL: <https://suspilne.media/587417-mvs-v-ukraini-oficijno-26-tisac-zniklih-bezvisti-11-tisac-iz-nih-civilni/> (дата звернення: 28.11.2023).
3. Штучний інтелект (світовий ринок). URL: <https://www.ft.com/content/c6b47d24-b435-4f41-b197-2d826cse9532> (дата звернення: 30.11.2023).
4. Gonzalez R. C., Woods R. E. Digital Image Processing. New Jersey, 2008, 750 p.
5. Richard Szeliski. Computer Vision: Algorithms and Applications. 2010, 655 p.
6. Deep Learning-Based Water Seegmentation for Autonomous Surface Vssel. URL: https://www.researchgate.net/publication/343451541_Deep_Learning-Based_Water_Segmentation_for_Autonomous_Surface_Vessel (дата звернення: 17.11.2023).
7. Semantic Segmmmentation – SegNet. URL: <https://medium.com/@abhishekkakiak/semantic-segmentation-segnet-a54af19b6d6> (дата звернення: 17.11.2023).
8. Understanding Semantic Segmentation with U-Net. URL: <https://towardsdatascience.com/understanding-semantic-segmentation-with-unet-6be4f42d4b47> (дата звернення: 18.11.2023).
9. Overview of Object Detection Algorithms Using Convolutional Neural Networks. URL:

- <https://www.scirp.org/journal/paperinformation?paperid=115011> (дата звернення: 20.11.2023).
10. 3D Slicer. URL: <https://www.slicer.org/> (дата звернення: 20.11.2023).
 11. ITK-SNAP. URL: <http://www.itksnap.org/pmwiki/pmwiki.php> (дата звернення: 24.11.2023).
 12. SimpleITK. URL: <https://simpleitk.org/> (дата звернення: 24.11.2023).
 13. DeepLabCut. URL: <https://www.mackenziemathislab.org/deeplabcut> (дата звернення: 25.11.2023).
 14. DeepMedic. URL: <https://deepmedic.org/> (дата звернення: 24.11.2023).
 15. SegNet. URL: <https://arxiv.org/abs/1511.00561>
 16. U-Net. URL: <https://neurohive.io/ru/vidy-nejrosetej/u-net-image-segmentation/> (дата звернення: 15.11.2023).
 17. Haralick R.M., Statistical and structural approaches to texture. Proceedings of the IEEE. 1979. Vol. 67, №5, P. 786-804.
 18. Crimmins T.R. Geometric filter for reducing speckle. Optical Engineering. 1986. Vol. 25, №5, P. 651–654.
 19. Haralick R.M., K. Shanmugan, I. Dinstein. Textural Features for Image Classification. IEEE Transactions on systems, man and cybernetics. 1979. №6, P. 610-621.
 20. Unsharp masking. URL: https://en.wikipedia.org/wiki/Unsharp_masking (дата звернення: 17.11.2023).
 21. Compiler vs Interpreter. URL: <https://www.guru99.com/difference-compiler-vs-interpreter.html> (дата звернення: 15.11.2023).
 22. Keras. URL: <https://keras.io/> (дата звернення: 15.11.2023).
 23. Tensorflow. URL: <https://www.tensorflow.org/?hl=en> (дата звернення: 15.11.2023).