ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ ІМЕНІ СЕМЕНА КУЗНЕЦЯ

ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

КАФЕДРА ІНФОРМАЦІЙНИХ СИСТЕМ

Пояснювальна записка

до дипломної роботи

МАГІСТРА

на тему: “Модель сегментації зображень з застосуванням нейронної мережі Mask R-CNN.”

Виконав**:**

студент 2 року навчання

групи 8.04.122.010.20.1

спеціальності 122 "Комп’ютерні науки"

Матюнін Олексій Володимирович

Керівник:

Проф.Каф. Мінухін Сергій Володимирович

Харків – 2021 рік

РЕФЕРАТ

Модель сегментації зображень з застосуванням нейронної мережі Mask R-CNN.

Об'єктом дослідження є розробка нейромережі для сігментації зображень за допомогою Mask r-cnn

Предмет дослідження – нейромережа для сігментації зображень

Мета наукової роботи розробити нейронну мережу для сігментації зображень, також проаналізувати точність роботи мережі

Актуальність теми обумовлена тим, що в даний час машинне навчання знаходиться в процвітанні, могіе послуги і бізнеси починають використовувати штучний інтелект, зокрема сігментацію зображень

Структура та обсяг дипломного проекту: робота складається зі вступу, трьох розділів, висновку, переліку спеціальної літератури (31 позицій), 43 рисунків і додатків; обсяг роботи без додатків і переліку літератури – 78 стр.

Ключові слова: нейронна мережа, машинне навчання, штучний інтелект, розробка програмного забезпечення, сігментація зображень

ABSTRACT

Image segmentation model using the Mask R-CNN neural network.

The object of study is the development of a neural network for image segmentation using Mask r-cnn

The subject of research is a neural network for image segmentation

The purpose of scientific work is to develop a neural network for image segmentation, as well as to analyze the accuracy of the network

The relevance of the topic is due to the fact that currently machine learning is thriving, my services and businesses are beginning to use artificial intelligence, including image segmentation

Structure and scope of the diploma project: the work consists of an introduction, two sections, a conclusion, a list of special literature (31 items), 43 figures and appendices; volume of work without appendices and bibliography – 78 pages

Keywords: neural network, machine learning, artificial intelligence, software development, image segmentation

Зміст

ВСТУП

Перше, що ми бачимо на своєму зображенні це ваше обличчя. Ви можете помітити своє обличчя, оскільки ваш мозок здатний ідентифікувати ваше обличчя та відокремити його від решти зображення (фону).

Але щоб комп'ютер ідентифікував обличчя він може виконувати сегментацію зображень.

Що таке сегментація зображення?

Сегментація зображень — це галузь цифрової обробки зображень, яка зосереджується на поділі зображення на різні частини відповідно до їх особливостей та властивостей. Основна мета сегментації зображень – спростити зображення для полегшення аналізу. Під час сегментації зображення ви поділяєте зображення на різні частини, які мають подібні атрибути. Частини, на які ви поділяєте зображення, називаються об’єктами зображення.

Це перший крок для аналізу зображення. Без виконання сегментації зображення виконання реалізацій комп’ютерного зору було б для вас майже неможливим.

Використовуючи методи сегментації зображення, ви можете розділяти та групувати специфічні пікселі зображення, призначати їм мітки та класифікувати інші пікселі відповідно до цих міток. Ви можете малювати лінії, вказувати межі та відокремлювати окремі об’єкти (важливі компоненти) на зображенні від решти об’єктів (неважливих компонентів).

У машинному навчанні ви можете використовувати мітки, згенеровані в результаті сегментації зображень, для навчання під наглядом і без нагляду. Це дозволить вирішити багато проблем бізнесу.

На прикладі краще зрозуміти, як працює сегментація зображень.

Рис 1.1 Приклад сегментація зображень

Тут ви можете побачити стілець, розміщений посеред дороги. Використовуючи сегментацію зображення, можно відокремити стілець від зображення. Крім того, можно використовувати численні методи сегментації зображень, щоб отримати різні результати. Наприклад, якщо використовувати зображення з кількома стільцями, доведеться використовувати семантичну сегментацію.

З іншого боку, якщо ідентифікувати кожен стілець, присутній на зображенні, як-от наведене нижче, вам доведеться використовувати сегментацію екземплярів:

Рис 1.2 Сегментація екземплярів

Чому потрібна сегментація зображення?

Сегментація зображень є великим аспектом комп’ютерного зору і має багато застосувань у багатьох галузях промисловості. Деякі з помітних областей, де широко використовується сегментація зображень:

1. Розпізнавання обличчя

Технологія розпізнавання облич, наявна у iPhone та передові системи безпеки, використовує сегментацію зображень для ідентифікації обличчя. Він повинен мати можливість ідентифікувати унікальні риси обличчя, щоб будь-яка сторона не могла отримати доступ до вашого телефону або системи.

2. Ідентифікація номерних знаків

Багато світлофорів і камер використовують ідентифікацію номерних знаків для стягнення штрафів і допомоги при обшуку. Технологія ідентифікації номерних знаків дозволяє системі дорожнього руху розпізнавати автомобіль та отримувати інформацію про його власника. Він використовує сегментацію зображення, щоб відокремити номерну табличку та її інформацію від решти об’єктів, присутніх у його баченні. Ця технологія значно спростила процес накладення штрафів для урядів.

3. Пошук на основі зображень

Google та інші пошукові системи, які пропонують засоби пошуку на основі зображень, використовують методи сегментації зображень, щоб ідентифікувати об’єкти, присутні на вашому зображенні, і порівняти їхні результати з відповідними зображеннями, які вони знаходять, щоб отримати результати пошуку.

4. Медична візуалізація

У медичній галузі використовується сегментація зображень, щоб знаходити та ідентифікувати ракові клітини, вимірювати об’єм тканин, запускати віртуальне моделювання хірургічних операцій та здійснювати навігацію в межах операції. Сегментація зображень має багато застосувань у медичній сфері. Це допомагає визначити уражені ділянки та спланувати лікування для них.

Окрім цих додатків, сегментація зображень використовується у виробництві, сільському господарстві, безпеці та багатьох інших секторах. Оскільки технології комп’ютерного зору стають все більш розвиненими, використання методів сегментації зображень буде відповідно зростати.

Наприклад, деякі виробники почали використовувати методи сегментації зображень, щоб знайти неякісні продукти. Тут алгоритм фіксує лише необхідні компоненти з зображення об’єкта та класифікує їх як несправні чи оптимальні. Ця система знижує ризик людських помилок і робить процес тестування більш ефективним для організації.

Поширені реалізації класифікації зображень є в Python, C, C++ і Matlab.

Які існують види сегментації зображень?

Сегментація зображень – це дуже широка тема і має різні способи виконання цього процесу. Можемо класифікувати сегментацію зображень за такими параметрами:

1. Класифікація на основі підходу

У найпростішому сенсі сегментація зображення — це ідентифікація об’єкта. Алгоритм не може класифікувати різні компоненти, не ідентифікувавши спочатку об’єкт. Від простих до складних реалізацій усі сегменти зображень працюють на основі ідентифікації об’єкта.

Отже, ми можемо класифікувати методи сегментації зображень на основі того, як алгоритми ідентифікують об’єкти, тобто збирають подібні пікселі та відокремлюють їх від різнорідних. Існує два підходи до виконання цього завдання:

Регіональний підхід (виявлення схожості)

У цьому методі ви виявляєте схожі пікселі на зображенні відповідно до вибраного порогу, об’єднання регіонів, розширення області та збільшення області. Кластеризація та подібні алгоритми машинного навчання використовують цей метод для виявлення невідомих функцій та атрибутів. Алгоритми класифікації дотримуються цього підходу для виявлення ознак і розділення сегментів зображення відповідно до них.

Підхід на основі меж (виявлення розривів)

Підхід на основі кордонів є протилежністю підходу на основі регіонів для ідентифікації об’єктів. На відміну від визначення на основі регіону, коли ви знаходите пікселі, які мають схожі характеристики, ви знаходите пікселі, які відрізняються один від одного в підході на основі меж. Виявлення точок, Виявлення краю, Виявлення лінії та подібні алгоритми дотримуються цього методу, коли вони виявляють край різнорідних пікселів і відповідно відокремлюють їх від решти зображення.

2. Класифікація на основі техніки

Обидва підходи мають свої відмінні методи сегментації зображень. Використовуються ці методи відповідно до типу зображення, яке хочемо обробити та проаналізувати, і від результатів, які хочемо отримати.

На основі цих параметрів ми можемо розділити алгоритми сегментації зображень на такі категорії:

Конструкційні прийоми

Ці алгоритми вимагають, щоб були структурні дані зображення, яке використовується. Сюди входять пікселі, розподіли, гістограми, щільність пікселів, розподіл кольору та інша відповідна інформація. Потім необхідно мати структурні дані про регіон, який необхідно відокремити від зображення.

Необхідна ця інформація, щоб ваш алгоритм міг визначити регіон. Алгоритми, які ми використовуємо для цих реалізацій, дотримуються підходу на основі регіону.

Стохастичні техніки

Ці алгоритми вимагають інформації про дискретні значення пікселів зображення, а не про структуру необхідної частини зображення. Завдяки цьому вони не вимагають багато інформації для сегментації зображень і корисні, коли вам доводиться працювати з кількома зображеннями. Алгоритми машинного навчання, такі як кластеризація K-середніх і алгоритми ANN, належать до цієї категорії.

Гібридні техніки

Як можна здогадатися з назви, ці алгоритми використовують як стохастичні, так і структурні методи. Це означає, що вони використовують структурну інформацію необхідної області та інформацію про дискретні пікселі всього зображення для виконання сегментації зображення.

Які існують види методів сегментації зображень?

Нижче наведено основні типи методів сегментації зображень:

1. Порогова сегментація
2. Сегментація на основі країв
3. Сегментація на основі регіонів
4. Сегментація вододілу
5. Алгоритми сегментації на основі кластеризації
6. Нейронні мережі для сегментації

Детально наведемо кожну з цих методик, щоб зрозуміти їх властивості, переваги та обмеження:

1. Порогова сегментація

Найпростішим методом сегментації в обробці зображень є пороговий метод. Він ділить пікселі в зображенні, порівнюючи інтенсивність пікселя з заданим значенням (порогом). Це корисно, коли необхідний об’єкт має більшу інтенсивність, ніж фон (непотрібні частини).

Можно вважати порогове значення (T) константою, але воно працюватиме лише в тому випадку, якщо зображення має дуже мало шумів (непотрібна інформація та дані). Можно підтримувати порогове значення постійним або динамічним відповідно до ваших вимог.

Метод порогового значення перетворює зображення у відтінках сірого в двійкове, розділяючи його на два сегменти (обов’язкові та не обов’язкові).

Відповідно до різних порогових значень ми можемо класифікувати порогову сегментацію на такі категорії:

Просте порогове значення

У цьому методі замінюються пікселі зображення білими або чорними. Тепер, якщо інтенсивність пікселя в певній позиції менша за порогове значення, ви б замінили його чорним. З іншого боку, якщо він вищий за поріг, ви б замінили його на білий. Це просте встановлення порогів і особливо підходить для початківців у сегментації зображень.

Бінаризація Оцу

У простому пороговому значенні вибирається постійне порогове значення і використовуєте його для сегментації зображення. Однак як визначити, що вибране вами значення було правильним? Хоча простий метод для цього полягає в тому, щоб перевірити різні значення та вибрати одне, він не найефективніший.

Зображення з гістограмою, яка має два піки: один для переднього плану, а другий для заднього плану. Використовуючи бінаризацію Otsu, ви можете прийняти за порогове значення приблизне значення середини цих піків.

У бінаризації Otsu обчислюється порогове значення з гістограми зображення, якщо зображення є бімодальним.

Цей процес досить популярний для сканування документів, розпізнавання візерунків та видалення непотрібних кольорів із файлу. Однак він має багато обмежень. Ви не можете використовувати його для зображень, які не є бімодальними (зображення, гістограми яких мають кілька піків).

Адаптивне порогове значення

Наявність одного постійного порогового значення може бути невідповідним підходом для кожного зображення. Різні зображення мають різний фон та умови, які впливають на їхні властивості.

Таким чином, замість використання одного постійного порогового значення для сегментації на всьому зображенні, ви можете зберегти змінну порогового значення. У цій техніці зберігається різні порогові значення для різних частин зображення.

Цей метод добре працює з зображеннями з різними умовами освітлення. Вам потрібно буде використовувати алгоритм, який сегментує зображення на менші ділянки та обчислює порогове значення для кожного з них.

2. Сегментація на основі країв

Сегментація на основі країв є однією з найпопулярніших реалізацій сегментації в обробці зображень. Він фокусується на визначенні країв різних об’єктів на зображенні. Це важливий крок, оскільки він допоможе вам знайти особливості різних об’єктів, присутніх на зображенні, оскільки краї містять багато інформації, яку ви можете використовувати.

Виявлення країв є широко популярним, оскільки допомагає видалити небажану та непотрібну інформацію із зображення. Це значно зменшує розмір зображення, полегшуючи його аналіз.

Алгоритми, що використовуються в сегментації на основі країв, визначають краї зображення відповідно до відмінностей у текстурі, контрасті, рівні сірого, кольорі, насиченості та інших властивостях. Ви можете покращити

якість результатів, з’єднавши всі краї в ланцюжки країв, які точніше відповідають межам зображення.

Існує багато доступних методів сегментації на основі країв. Ми можемо розділити їх на дві категорії:

Виявлення краю на основі пошуку

Методи виявлення країв на основі пошуку зосереджені на обчисленні міри міцності краю та пошуку локальних спрямованих максимумів величини градієнта за допомогою обчисленої оцінки локальної орієнтації краю.

Виявлення країв на основі нуля

Методи виявлення країв на основі перетину нуля шукають нульові перетини в похідному виразі, отриманому із зображення, щоб знайти краї.

Як правило, доведеться попередньо обробити зображення, щоб видалити небажаний шум і полегшити виявлення країв. Canny, Prewitt, Deriche і Roberts Cross є одними з найпопулярніших операторів виявлення країв. Вони полегшують виявлення розривів і знаходження країв.

При виявленні на основі країв ваша мета — отримати мінімум часткової сегментації, щоб ви могли згрупувати всі локальні ребра в двійкове зображення. У вашому щойно створеному двійковому зображенні крайові ланцюжки повинні відповідати існуючим компонентам відповідного зображення.

3. Сегментація на основі регіонів

Алгоритми сегментації на основі регіонів поділяють зображення на ділянки з подібними ознаками. Ці області є лише групою пікселів, і алгоритм знаходить ці групи, спочатку визначаючи початкову точку, яка може бути невеликою частиною або великою частиною вхідного зображення.

Після знаходження початкових точок алгоритм сегментації на основі регіону або додасть до них більше пікселів, або зменшить їх, щоб об’єднати їх з іншими початковими точками.

На основі цих двох методів можно класифікувати сегментацію на основі регіонів на такі категорії:

Зростання регіону

У цьому методі починається з невеликого набору пікселів, а потім починається ітеративно об’єднувати більше пікселів відповідно до конкретних умов подібності. Алгоритм збільшення області вибирає довільний початковий піксель на зображенні, порівнює його з сусідніми пікселями і починає збільшувати область, знаходячи відповідність початковій точці.

Коли конкретна область не може розвиватися далі, алгоритм вибере інший початковий піксель, який може не належати жодному існуючому регіону. Одна область може мати занадто багато атрибутів, через що вона займає більшу частину зображення. Щоб уникнути такої помилки, алгоритми вирощування регіонів розвивають кілька регіонів одночасно.

Обов'язково використовувати алгоритми збільшення області для зображень, які мають багато шуму, оскільки шум ускладнить пошук країв або використання порогових алгоритмів.

Розділ та злиття регіонів

Як випливає з назви, сфокусований метод поділу та об’єднання областей виконував би дві дії разом – розділяючи та об’єднуючи частини зображення.

Спочатку зображення розбивається на області, які мають подібні атрибути, і об’єднує сусідні частини, подібні один до одного. При розділенні регіонів алгоритм розглядає все зображення, тоді як при зростанні області алгоритм фокусується на певній точці.

Метод розділення та злиття регіонів дотримується методології «розділяй і владарюй». Він ділить зображення на різні частини, а потім узгоджує їх відповідно до заданих умов. Інша назва алгоритмів, які виконують це завдання, — алгоритми розділеного злиття.

4. Сегментація вододілу

При обробці зображень вододіл — це перетворення зображення у відтінках сірого. Він відноситься до геологічного вододілу або вододілу. Алгоритм вододілу обробляв би зображення так, ніби це була топографічна карта. Він розглядає яскравість пікселя як його висоту і знаходить лінії, які проходять уздовж вершини цих хребтів.

Вододіл має багато технічних визначень і має кілька застосувань. Окрім визначення хребтів пікселів, він зосереджується на визначенні улоговин (протилежних від хребтів) і заливає улоговини маркерами, поки вони не зустрінуться з лініями вододілу, що проходять через хребти.

Оскільки басейни мають багато маркерів, а гребені — ні, зображення розбивається на кілька областей відповідно до «висоти» кожного пікселя.

Метод вододілу перетворює кожне зображення на топографічну карту. Метод сегментації вододілу відображатиме топографію через сірі значення пікселів.

Тепер ландшафт з долинами та хребтами, безумовно, мав би тривимірні аспекти. Вододіл розглядатиме тривимірне відображення зображення і відповідно створюватиме регіони, які називаються «басейнами водозбору».

Він має багато застосувань у медичній галузі, наприклад, МРТ, медичне зображення тощо. Сегментація вододілу є важливою частиною сегментації медичних зображень, тому, якщо ви хочете увійти в цей сектор, вам слід зосередитися на вивченні цього методу для сегментації, зокрема, при обробці зображень.

5. Алгоритми сегментації на основі кластеризації

Алгоритми кластеризації є неконтрольованими алгоритмами, які допомагають знайти приховані дані на зображенні, які можуть бути невидимими для звичайного зору. Ці приховані дані містять таку інформацію, як кластери, структури, затінення тощо.

Як випливає з назви, алгоритм кластеризації розбиває зображення на кластери (неперетинаються групи) пікселів, які мають схожі характеристики. Це розділить елементи даних на кластери, де елементи в кластері більш схожі на елементи, присутні в інших кластерах.

Деякі з популярних алгоритмів кластеризації включають нечіткі c-середніх (FCM), k-середніх та покращені алгоритми k-середніх. При сегментації зображень ви в основному використовували б алгоритм кластеризації k-середніх, оскільки він досить простий та ефективний. З іншого боку, алгоритм FCM поміщає пікселі в різні класи відповідно до їх різного ступеня належності.

Найважливішими алгоритмами кластеризації для сегментації в обробці зображень є:

K-means Clustering

K-means — це простий алгоритм машинного навчання без нагляду. Він класифікує зображення за певною кількістю кластерів. Він починає процес з поділу простору зображення на k пікселів, які представляють k центроїдів групи.

Потім вони відносять кожен об’єкт до групи на основі відстані між ними та центроїдом. Коли алгоритм призначив усі пікселі всім кластерам, він може переміщати та перепризначати центроїди.

Fuzzy C Means

За допомогою методу кластеризації нечітких c-середніх пікселів на зображенні можна об’єднати в кілька кластерів. Це означає, що піксель може належати більш ніж одному кластеру. Однак кожен піксель буде мати різні рівні

подібності з кожним кластером. Алгоритм нечітких с-середніх має функцію оптимізації, яка впливає на точність результатів.

6. Нейронні мережі для сегментації

Можливо, щоб AI виконував більшість ваших завдань, що ви, безумовно, можете зробити за допомогою нейронних мереж для сегментації зображень.

Використовування AI для аналізу зображення та визначення його різних компонентів, таких як обличчя, об’єкти, текст тощо. Згорткові нейронні мережі досить популярні для сегментації зображень, оскільки вони можуть ідентифікувати та обробляти дані зображення дуже швидко й ефективно.

Експерти Facebook AI Research (FAIR) створили архітектуру глибокого навчання під назвою Mask R-CNN, яка може створити піксельну маску для кожного об’єкта, присутній на зображенні. Це розширена версія архітектури виявлення об’єктів Faster R-CNN. Швидший R-CNN використовує дві частини даних для кожного об’єкта на зображенні, координати рамки та клас об’єкта. З Mask R-CNN ви отримуєте додатковий розділ у цьому процесі. Маска R-CNN виводить маску об'єкта після виконання сегментації.

1. АНАЛІЗ МОДЕЛЮВАННЯ СЕГМЕНТАЦІЇ МЕДИЧНИХ ЗОБРАЖЕНЬ ЗАСНОВАННОГО НА МАШИННОМУ НАВЧАННІ

1.1 Різні типи сегментації зображення

Можно широко розділити методи сегментації зображень на два типи. Приклад наведен нижче:

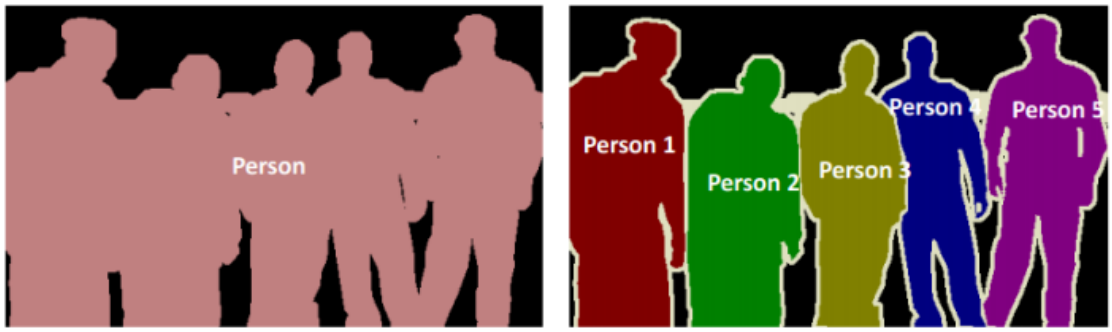

Рис 1.3 Семантична та екземплярна сегментація

Чи можно визначити різницю між цими двома? Обидва зображення використовують сегментацію зображень, щоб ідентифікувати та знайти присутніх людей.

- На зображенні 1 кожен піксель належить до певного класу (фону або людини). Крім того, всі пікселі, що належать до певного класу, представлені одним кольором (фон чорним, а особа рожевим). Це приклад семантичної сегментації
- Зображення 2 також присвоїло певний клас кожному пікселю зображення. Однак різні об’єкти одного класу мають різні кольори (Людина 1 — червоний, Людина 2 — зелений, фон — чорний тощо). Це приклад сегментації екземплярів

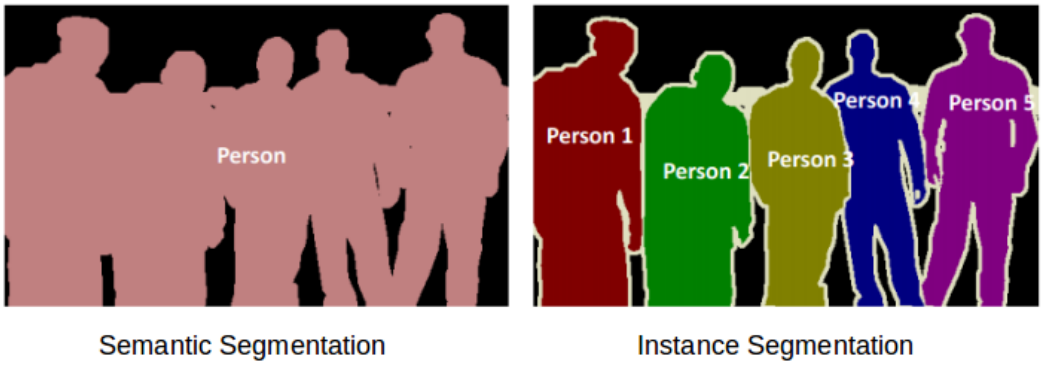

Рис 1.4 Семантична та екземплярна сегментація

Якщо на зображенні 5 людей, семантична сегментація буде зосереджена на класифікації всіх людей як одного екземпляра. З іншого боку, сегментація екземплярів. визначить кожного з цих людей окремо.

1.2 Сегментація на основі регіонів

Одним із простих способів сегментації різних об’єктів може бути використання їх значень пікселів. Важливо зауважити – значення пікселів будуть відрізнятися для об’єктів і фону зображення, якщо між ними є різкий контраст.

У цьому випадку можно встановити порогове значення. Значення пікселів, що падають нижче або вище цього порогу, можна відповідно класифікувати (як об’єкт або фон). Ця техніка відома як порогова сегментація.

Якщо розділити зображення на дві області (об’єкт і фон), визначаємо єдине порогове значення. Це відоме як глобальний поріг.

Якщо є кілька об’єктів разом із фоном, ми повинні визначити кілька порогових значень. Ці пороги спільно відомі як локальний поріг.

Реалізуємо порогове значення. Завантажуемо зображення та запускаемо наведений нижче код.

Спочатку ми імпортуємо необхідні бібліотеки.

```
from skimage.color import rgb2gray
import numpy as np
import cv2
import matplotlib.pyplot as plt
%matplotlib inline
from scipy import ndimage
```

Прочитаємо завантажене зображення та побудуємо його:

```
image = plt.imread('1.jpeg')
image.shape
plt.imshow(image)
```

```
(192, 263, 3)
<matplotlib.image.AxesImage at 0x7fc845de2048>
```

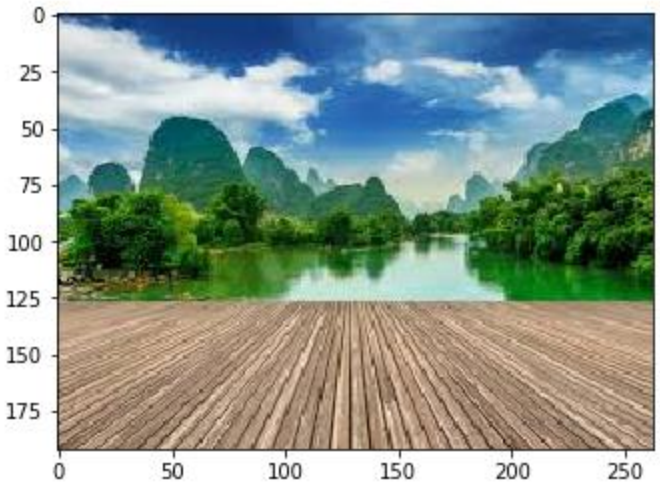

Рис 1.5 Порогова сегментація

Це триканальне зображення (RGB). Потрібно перетворити його в градації сірого, щоб у нас був лише один канал.

```
gray = rgb2gray(image)
plt.imshow(gray, cmap='gray')
```

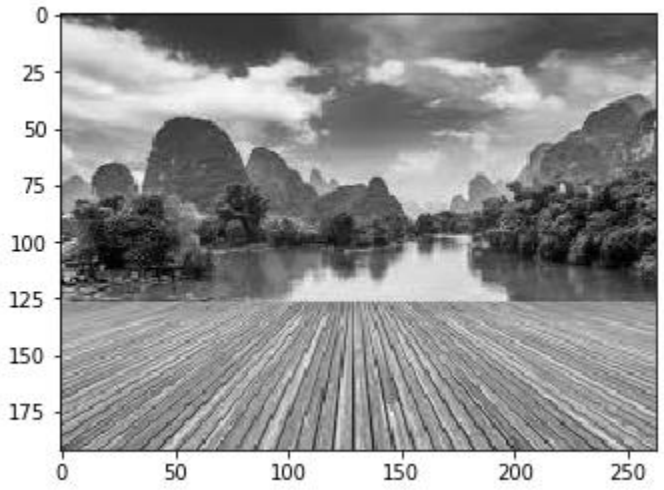

Рис 1.6 Зображення у відтінках сірого

Тепер застосовуємо певний поріг до цього зображення. Цей поріг повинен розділяти зображення на дві частини – передній план і задній план. Перш ніж ми це зробимо, давайте швидко перевіримо форму цього зображення:

```
gray.shape
```

```
(192, 263)
```

Висота і ширина зображення 192 і 263 відповідно. Візьмемо середнє значення пікселів і використаємо його як порогове значення. Якщо значення пікселя перевищує наш поріг, ми можемо сказати, що він належить об’єкту. Якщо значення пікселя менше за порогове значення, воно буде розглядатися як фон.

```
gray_r = gray.reshape(gray.shape[0]*gray.shape[1])
for i in range(gray_r.shape[0]):
    if gray_r[i] > gray_r.mean():
        gray_r[i] = 1
    else:
        gray_r[i] = 0
gray = gray_r.reshape(gray.shape[0],gray.shape[1])
plt.imshow(gray, cmap='gray')
```

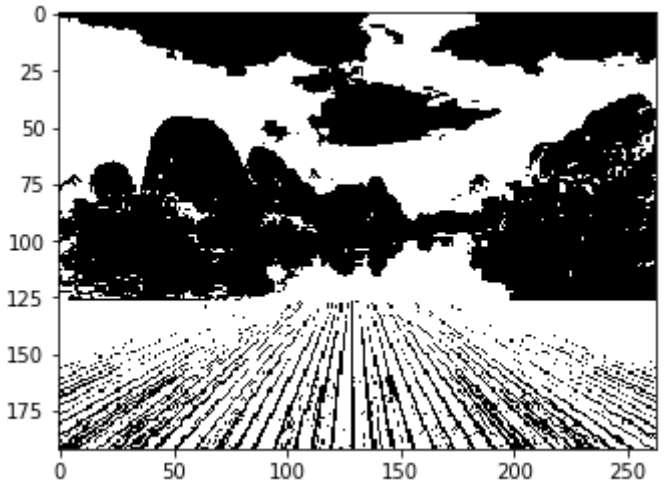

Рис 1.7 Глобальний поріг

Темніша область (чорна) представляє фон, а світліша (біла) область – передній план. Також можемо визначити кілька порогових значень для виявлення кількох об’єктів:

```
gray = rgb2gray(image)
gray_r = gray.reshape(gray.shape[0]*gray.shape[1])
for i in range(gray_r.shape[0]):
    if gray_r[i] > gray_r.mean():
        gray_r[i] = 3
    elif gray_r[i] > 0.5:
        gray_r[i] = 2
    elif gray_r[i] > 0.25:
        gray_r[i] = 1
    else:
        gray_r[i] = 0
gray = gray_r.reshape(gray.shape[0],gray.shape[1])
plt.imshow(gray, cmap='gray')
```

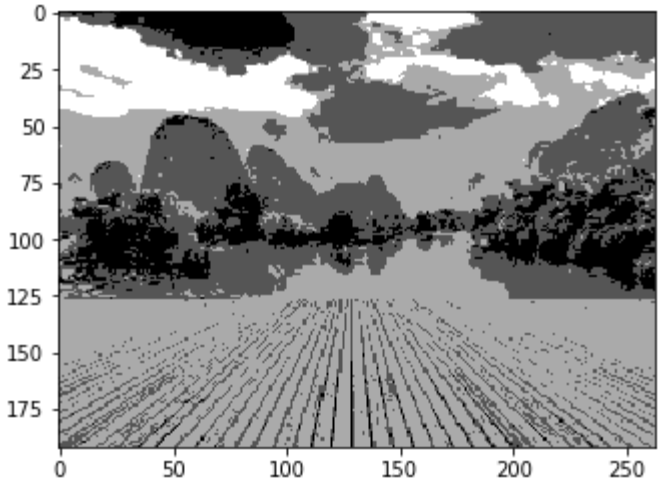

Рис 1.8 Локальний поріг

На зображенні вище є чотири різні сегменти. Можно встановити різні порогові значення та перевірити, як створені сегменти. Деякі з переваг цього методу:

- Розрахунки простіші
- Висока швидкість роботи
- Коли об'єкт і фон мають високий контраст, цей метод працює дуже добре

Але у цього підходу є деякі обмеження. Коли немає значної різниці у відтінках сірого або є перекриття значень пікселів у відтінках сірого, отримати точні сегменти стає дуже важко.

1.3 Сегментація виявлення країв

Що розділяє два предмети на зображенні? Завжди існує межа між двома сусідніми областями з різними значеннями відтінків сірого (значеннями пікселів). Краї можна розглядати як розривні локальні особливості зображення.

Можемо використовувати цей розрив для виявлення країв і, отже, визначення кордону об’єкта. Це допомагає виявляти форми кількох об’єктів, присутніх на даному зображенні. Тепер питання в тому, як ми можемо виявити ці краї? Тут ми можемо використовувати фільтри та згортки. Зверніться до цієї статті, якщо вам потрібно дізнатися про ці поняття.

Наведений нижче візуальний матеріал допоможе вам зрозуміти, як фільтр згортається над зображенням:

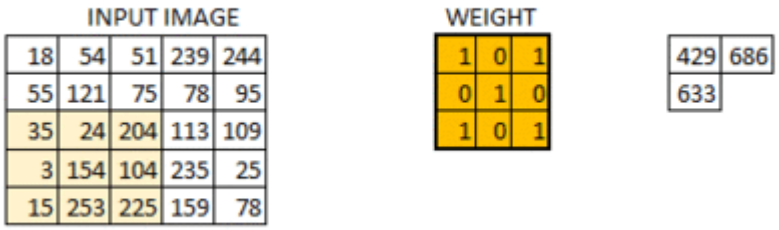

Рис 1.9 Згортка

Ось покроковий процес, як це працює:

- Візьмемо вагову матрицю
- Кладемо його поверх зображення
- Виконаемо поелементне множення та отримайте результат
- Переміщуемо матрицю ваги відповідно до вибраного кроку
- Згортаемо, доки не будуть використані всі пікселі вхідних даних

Значення вагової матриці визначають вихід згортки. Моя порада – це допомагає витягувати функції з введення. Дослідники виявили, що вибір деяких конкретних значень для цих вагових матриць допомагає нам виявляти горизонтальні або вертикальні ребра (або навіть комбінацію горизонтальних і вертикальних країв).

Однією з таких вагових матриць є оператор Sobel. Зазвичай використовується для виявлення країв. У оператора sobel є дві вагові матриці – одна для виявлення горизонтальних країв, а інша для виявлення вертикальних країв.

Sobel filter (horizontal) =

1	2	1
0	0	0
-1	-2	-1

Sobel filter (vertical) =

-1	0	1
-2	0	2
-1	0	1

Виявлення країв працює шляхом згортання цих фільтрів над даним зображенням.

```
image = plt.imread('index.png')
plt.imshow(image)
```

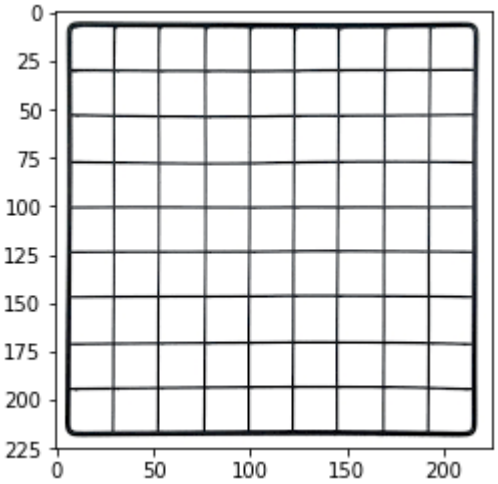

Рис 1.10 Виявлення краю

Перетворимо його в градації сірого і визначимо фільтр Sobel (як горизонтальний, так і вертикальний), який буде згорнутий над цим зображенням:

```
# converting to grayscale
gray = rgb2gray(image)

# defining the sobel filters
sobel_horizontal = np.array([np.array([1, 2, 1]), np.array([0, 0,
0]), np.array([-1, -2, -1])])
print(sobel_horizontal, 'is a kernel for detecting horizontal
edges')

sobel_vertical = np.array([np.array([-1, 0, 1]), np.array([-2, 0,
2]), np.array([-1, 0, 1])])
print(sobel_vertical, 'is a kernel for detecting vertical edges')
```

```
[[ 1  2  1]
 [ 0  0  0]
 [-1 -2 -1]] is a kernel for detecting horizontal edges
[[-1  0  1]
 [-2  0  2]
 [-1  0  1]] is a kernel for detecting vertical edges
```

Рис 1.11 Sobel фильтр

Тепер згорнемо цей фільтр над зображенням за допомогою функції convolve пакета ndimage з scipy.

```
out_h = ndimage.convolve(gray, sobel_horizontal, mode='reflect')
out_v = ndimage.convolve(gray, sobel_vertical, mode='reflect')
# here mode determines how the input array is extended when the
filter overlaps a border.
```

Зобразимо ці результати:

```
plt.imshow(out_h, cmap='gray')
```

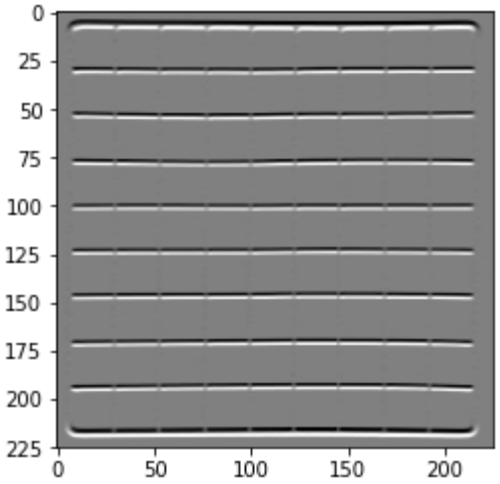

Рис 1.12 Виявлення горизонтальних країв

```
plt.imshow(out_v, cmap='gray')
```

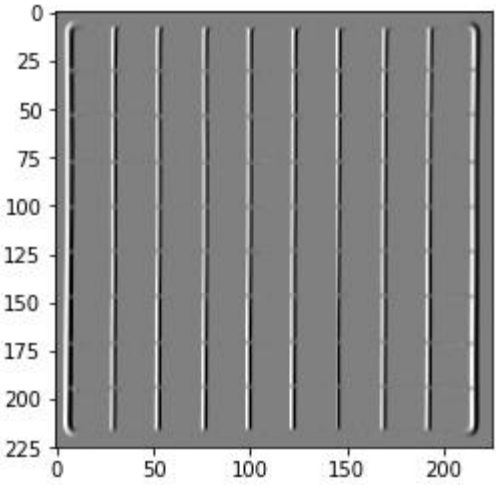

Рис 1.13 Виявлення вертикальних країв

Можливо визначити як горизонтальні, так і вертикальні краї. Є ще один тип фільтра, який може виявляти як горизонтальні, так і вертикальні краї одночасно. Це називається оператором Лапласа:

1 1 1
1 -8 1
1 1 1

Визначимо цей фільтр у Python і згорнемо його на тому самому зображенні:

```
kernel_laplace = np.array([np.array([1, 1, 1]), np.array([1, -8, 1]), np.array([1, 1, 1])])
print(kernel_laplace, 'is a laplacian kernel')
```

```
[[ 1  1  1]
 [ 1 -8  1]
 [ 1  1  1]] is a laplacian kernel
```

Рис 1.14 Фільтр лапласа

Далі згорнемо фільтр і роздрукуемо результат:

```
out_l = ndimage.convolve(gray, kernel_laplace, mode='reflect')
plt.imshow(out_l, cmap='gray')
```

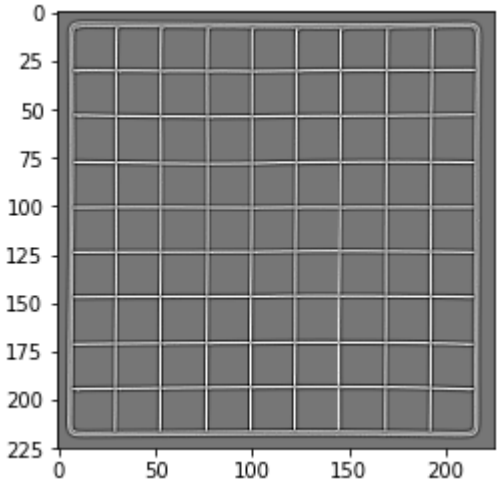

Рис 1.15 Виявлення країв лапласа

Метод виявив як горизонтальні, так і вертикальні краї.

1.4 Сегментація на основі кластеризації

Кластеризація — це завдання поділу сукупності (точок даних) на кілька груп, щоб точки даних в тих самих групах були більш схожими на інші точки даних у тій самій групі, ніж в інших групах. Ці групи відомі як кластери.

Одним з найбільш часто використовуваних алгоритмів кластеризації є k-середніх. Тут k представляє кількість кластерів (не плутати з k-найближчим сусідом). Як працює k-means:

- Спочатку випадковим чином вибераемо k початкових кластерів
- Довільно призначаемо кожну точку даних будь-якому з k кластерів
- Обчисляемо центри цих скупчень
- Обчисляемо відстань усіх точок від центру кожного скупчення
- Залежно від цієї відстані очки перепризначаються до найближчого кластера
- Обчисляемо центр новоутворених скупчень

- Нарешті, повторюємо кроки (4), (5) і (6) до тих пір, поки центр кластерів не зміниться, або ми не досягнемо заданої кількості ітерацій

Ключова перевага використання алгоритму k-середніх полягає в тому, що він простий і легкий для розуміння. Ми відносимо точки до кластерів, які є найближчими до них.

Перевіримо, наскільки добре k-середнє сегментує об'єкти на зображенні.

```
pic = plt.imread('1.jpeg')/255  # dividing by 255 to bring the pixel values between 0 and 1
print(pic.shape)
plt.imshow(pic)
```

```
(192, 263, 3)

<matplotlib.image.AxesImage at 0x7fc845de2048>
```

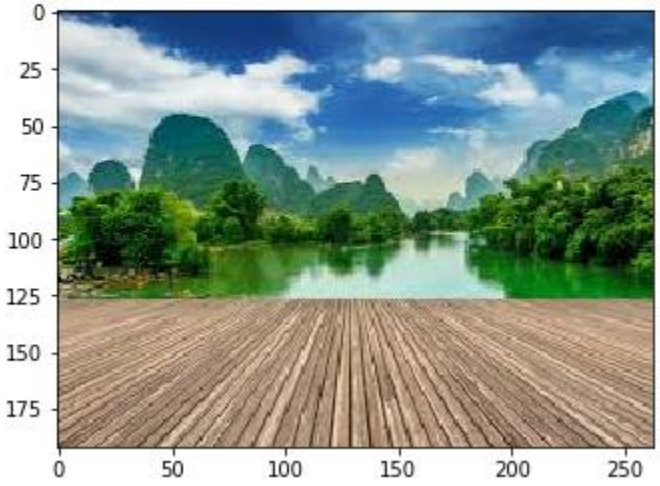

Рис 1.16 Кластеризація

Це тривимірне зображення форми (192, 263, 3). Для кластеризації зображення за допомогою k-середніх нам спочатку потрібно перетворити його в двовимірний масив, форма якого буде (довжина*ширина, канали). У прикладі це буде (192*263, 3).

```
pic_n = pic.reshape(pic.shape[0]*pic.shape[1], pic.shape[2])
pic_n.shape
```

(50496, 3)

Бачимо, що зображення було перетворено в 2-вимірний масив. Далі вставляємо алгоритм k-середніх на цей змінений масив і отримайте кластери. Функція cluster_centers_ k-means поверне центри кластерів, а функція labels_ дасть нам мітку для кожного пікселя (вона скаже нам, який піксель зображення до якого кластера належить).

```
from sklearn.cluster import KMeans
kmeans = KMeans(n_clusters=5, random_state=0).fit(pic_n)
```

```
pic2show = kmeans.cluster_centers_[kmeans.labels_]
```

Вибрано 5 кластерів тепер повернемо кластери до їх початкової форми, тобто тривимірного зображення, і побудуємо результати.

```
cluster_pic = pic2show.reshape(pic.shape[0], pic.shape[1],
pic.shape[2])
plt.imshow(cluster_pic)
```

```
<matplotlib.image.AxesImage at 0x7fc845d40c50>
```

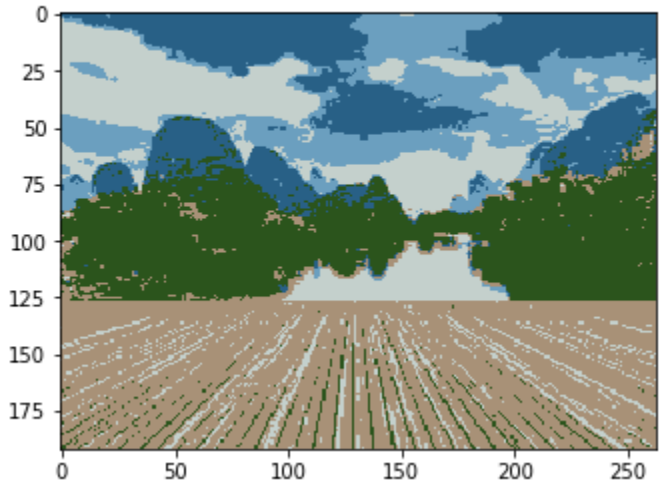

Рис 1.17 Сегментація на основі кластеризації

Можно досить добре сегментувати зображення, використовуючи лише 5 кластерів.

k-means працює дуже добре, коли маємо невеликий набір даних. Він може сегментувати об’єкти на зображенні та давати вражаючі результати. Але алгоритм стикається з перешкодою, коли застосовується до великого набору даних (більша кількість зображень).

Він переглядає всі зразки на кожній ітерації, тому витрачений час занадто великий. Отже, це також надто дорого для реалізації. А оскільки k-середніх є алгоритмом на основі відстані, він застосовний лише до опуклих наборів даних і не підходить для кластеризації неопуклих кластерів.

1.5 Mask R-CNN

Дослідники даних і дослідники з Facebook AI Research (FAIR) розробили архітектуру глибокого навчання під назвою Mask R-CNN, яка може створювати піксельну маску для кожного об’єкта на зображенні.

Mask R-CNN є розширенням популярної архітектури виявлення об'єктів Faster R-CNN. Маска R-CNN додає розгалуження до вже існуючих виходів Faster R-CNN. Метод Faster R-CNN генерує дві речі для кожного об’єкта на зображенні:

- Його клас
- Координати рамки

Маска R-CNN додає до цього третю гілку, яка також виводить маску об'єкта.

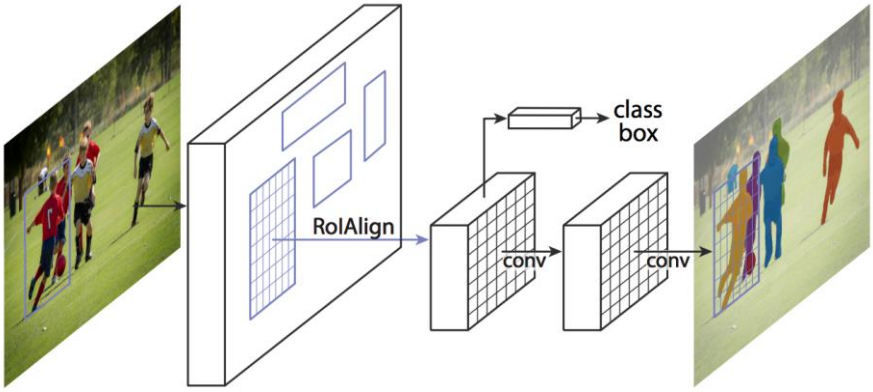

Рис 1.18 Mask R-CNN Source

- Беремо зображення як вхідні дані та передаємо його в ConvNet, яка повертає карту ознак для цього зображення
- Мережа регіональних пропозицій (RPN) застосована до цих карт об’єктів. Це повертає пропозиції об’єктів разом з їх оцінкою об’єктивності
- До цих пропозицій застосовується шар об’єднання ROI, щоб зменшити всі пропозиції до однакового розміру
- Нарешті, пропозиції передаються на повністю підключений шар, щоб класифікувати та вивести обмежувальні рамки для об’єктів. Він також повертає маску для кожної пропозиції

Mask R-CNN – це найсучасніший інструмент для сегментації зображень і працює зі швидкістю 5 кадрів в секунду.

1.6 Резюме методів сегментації зображень

У таблиці нижче підсумуємо різні алгоритми сегментації зображень.

Алгоритм	Опис	Переваги	Обмеження
Сегментація на основі регіонів	Розділяє об’єкти на різні області на основі деяких порогових значень.	а. Прості розрахунки б. Швидка швидкість роботи с. Коли об’єкт і фон мають високий	Якщо немає значної різниці у відтінках сірого або перекриття значень пікселів у відтінках сірого, отримати

		контраст, цей метод працює дуже добре	точні сегменти стає дуже важко.
Сегментація виявлення країв	Використовує розривні локальні особливості зображення для виявлення країв і, отже, визначення меж об'єкта.	Це добре для зображень із кращим контрастом між об'єктами.	Не підходить, якщо на зображенні занадто багато країв і якщо контраст між об'єктами менший.
Сегментація на основі кластеризації	Розбиває пікселі зображення на однорідні кластери.	Дуже добре працює з невеликими наборами даних і створює чудові кластери.	а. Час обчислень занадто великий і дорогий. б. k-means є алгоритмом на основі відстані. Він не підходить для кластеризації неопуклих кластерів.
Mask R-CNN	Надає три вихідні дані для кожного об'єкта на зображенні: його клас, координати рамки та маску об'єкта	а. Простий, гнучкий і загальний підхід б. Це також сучасний стан сегментації зображень	Високий час навчання

Табл 1 Різні алгоритми сегментації зображень

1.7 Сегментація медичних зображень

Сегментація зображень на основі глибокого навчання вже міцно закріпилася як надійний інструмент сегментації зображень. Він широко використовується для відокремлення однорідних областей як перший і важливий компонент діагностики та лікування. У цій статті ми представляємо критичну оцінку популярних методів, які використовують методи глибокого навчання для сегментації медичних зображень. Крім того, ми підсумовуємо найпоширеніші виклики та пропонуємо можливі рішення.

Сегментація медичних зображень, ідентифікація пікселів органів або уражень на фонових медичних зображеннях, таких як зображення КТ або МРТ, є однією з найскладніших завдань в аналізі медичних зображень, яка полягає в наданні важливої інформації про форми та об'єми цих органів. Багато

дослідників запропонували різні автоматизовані системи сегментації, застосовуючи наявні технології. Раніше системи були побудовані на традиційних методах, таких як фільтри виявлення країв і математичні методи. Потім підходи машинного навчання, що витягують створені вручну функції, стали домінуючою технікою протягом тривалого періоду. Розробка та виділення цих функцій завжди були основною проблемою для розробки такої системи, і складність цих підходів розглядалася як значне обмеження для їх розгортання. У 2000-х роках, завдяки вдосконаленню апаратного забезпечення, підходи глибокого навчання з'явилися в світ і почали демонструвати свої значні можливості в задачах обробки зображень. Багатообіцяючі можливості підходів глибокого навчання зробили їх основним варіантом для сегментації зображень, і, зокрема, для сегментації медичних зображень. Особливо в останні кілька років сегментація зображень на основі методів глибокого навчання приділяла велику увагу, і це підкреслює необхідність її всебічного огляду. Наскільки нам відомо, не існує вичерпного огляду спеціально для сегментації медичних зображень із застосуванням методів глибокого навчання. Існує кілька останніх статей опитування щодо сегментації медичних зображень, таких як [49] і [67]. Shen et al. у розглянуто різні види аналізу медичних зображень, але приділено мало уваги технічним аспектам сегментації медичного зображення. У багатьох інших розділах аналізу медичних зображень, таких як класифікація, виявлення та реєстрація, також охоплено, що робить аналіз медичних зображень оглядом, а не специфічним дослідженням сегментації медичних зображень. Через велику область, яку висвітлюється в цій статті, відомості про мережі, можливості та недоліки відсутні.

Підходи/Мережеві структури

Згорткові нейронні мережі (CNN)

CNN — це гілка нейронних мереж і складається зі стеку шарів, кожен з яких виконує певну операцію, наприклад, згортка, об'єднання, обчислення втрат тощо. Кожен проміжний рівень отримує вихід попереднього рівня як вхід (див. рис. 1).). Початковий шар – це вхідний шар, який безпосередньо пов’язаний з вхідним зображенням з кількістю нейронів, що дорівнює кількості пікселів у вхідному зображенні. Наступний набір шарів — це згорткові шари, які представляють результати згортки певної кількості фільтрів із вхідними даними та виконують функцію витягувача ознак. Фільтри, відомі як ядра, мають довільні розміри, визначені дизайнерами, і залежать від розміру ядра. Кожен нейрон реагує лише на певну область попереднього шару, звану рецептивним полем.

Вихід кожного шару згортки розглядається як карта активації, яка підкреслює ефект застосування певного фільтра до входу. За згортковими шарами зазвичай слідують шари активації, щоб застосувати нелінійність до карт активації. Наступний шар може бути шаром об’єднання залежно від дизайну, і це допомагає зменшити розмірність результату згортки. Щоб виконати об’єднання, існує кілька стратегій, наприклад, максимальне об’єднання та середнє об’єднання. Нарешті, високорівневі абстракції витягуються повністю пов’язаними шарами. Вага нейронних зв'язків і ядер постійно оптимізується під час процедури зворотного поширення на етапі навчання.

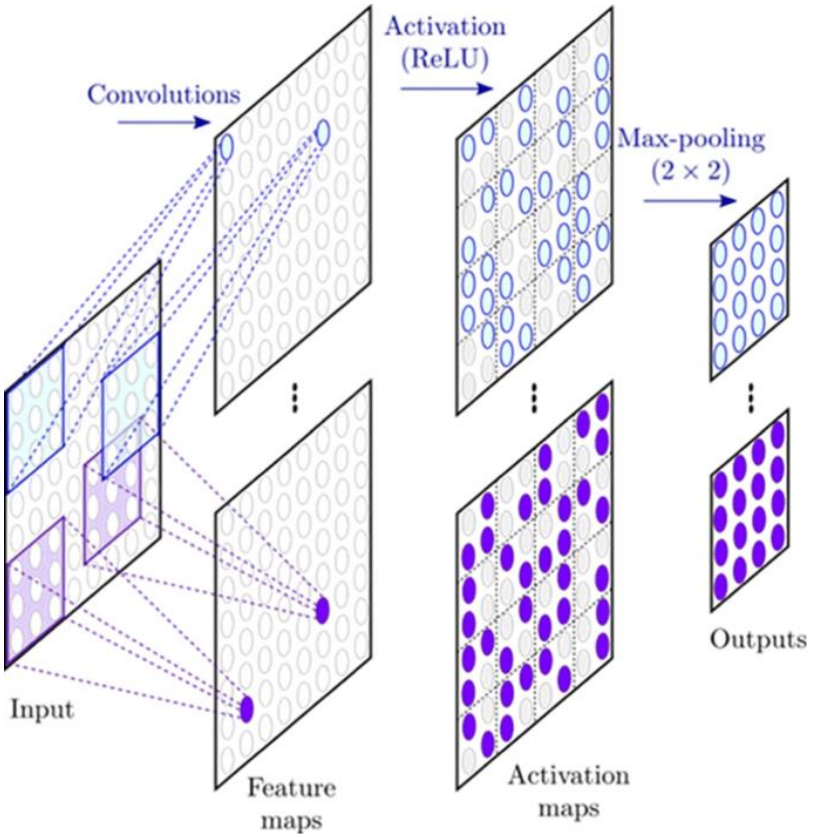

Рис 1.19 Структура CNN

Вищенаведена структура відома як звичайна CNN.

2D CNN

Завдяки перспективним можливостям CNN у виконанні класифікації зображень і розпізнавання образів застосування CNN до сегментації медичних зображень досліджувалося багатьма дослідниками.

Загальна ідея полягає в тому, щоб виконати сегментацію, використовуючи двовимірне вхідне зображення та застосовуючи до нього 2D-фільтри. У дослідженні, проведеному Zhang et al. [89], численні джерела інформації (T1, T2 і FA) у вигляді двовимірних зображень передаються на вхідний рівень CNN у

різних каналах зображення (наприклад, R, G, B), щоб дослідити, чи мультимодальні зображення як вхідні дані покращують результати сегментації. Їхні результати продемонстрували кращу продуктивність, ніж результати, що використовували один метод введення. В іншому експерименті, проведеному Баром та іншими [4], враховується підхід до навчання з перенесенням, а функції низького рівня запозичені з попередньо навченої моделі в Imagenet. Функції високого рівня взяті з PiCoDes [6], а потім усі ці функції об’єднуються разом.

2.5D CNN

2.5D підходи [6] натхненні тим фактом, що 2.5D має багатшу просторову інформацію сусідніх пікселів з меншими обчислювальними витратами, ніж 3D. Як правило, вони включають вилучення трьох ортогональних 2D латок у площинах XY, YZ і XZ відповідно, як показано нижче, з ядрами, які все ще знаходяться в 2D.

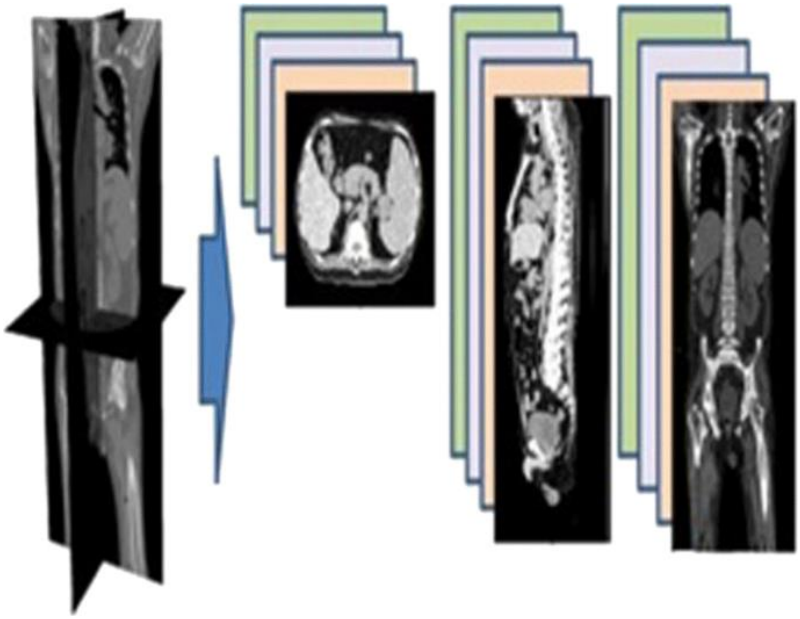

Рис 1.20 Ортогональне представлення 3D об'єму

Автори в [6] застосували цю ідею для сегментації хряща колінного суглоба. У цьому методі було визначено три окремі CNN, кожна з яких отримує набір патчів, витягнутих з кожної ортогональної площини. Відносно низька кількість навчальних вокселів (120 000) і задовільне досягнення коефіцієнта Dice 0,8249 довели, що трипланер CNN може забезпечити баланс між продуктивністю та обчислювальними витратами. У [5] три ортогональні види були об'єднані та оброблені як три канали вхідного зображення.

Moeskops та ін. [4] використовували архітектуру 2.5D для багатозадачної сегментації, щоб оцінити, чи може один проект мережі виконувати сегментацію

кількох органів. Вони ще більше розширили ідею, застосувавши різні методи (наприклад, МРТ головного мозку, МРТ молочної залози та СТА серця) для кожного завдання сегментації. Вибір невеликого розміру ядра 3 × 3 вокселів дозволив їм заглибитися в структуру і спроектувати 25-шарову мережу глибини. Цю конструкцію можна розглядати як дуже глибоку структуру, вперше запропоновану [9]. Здається, остаточні результати відповідають попереднім дослідженням, які демонструють, що одну CNN можна навчити візуалізувати різні анатомії з різними способами.

Підходи 2.5D отримують вигоду від навчання системи двовимірним позначеним даними, які є більш доступними в порівнянні з 3D-даними і мають кращу відповідність до поточного обладнання. Більше того, розкладання об’ємних зображень на набір випадкових двовимірних зображень допомагає полегшити проблему розмірності [24]. Хоча цей підхід здається оптимальною ідеєю з прийнятною продуктивністю (трохи краще, ніж 2D-методи), деякі люди (наприклад, [15]) дотримуються думки, що використання лише трьох ортогональних переглядів із багатьох можливих переглядів 3D-зображення не є оптимальне використання об’ємних медичних даних. Крім того, виконання двовимірних згорток із ізотропним ядром на анізотропних 3D-зображеннях може бути проблематичним, особливо для зображень із значно меншою роздільною здатністю по глибині (вісь Z).

3D CNN

Застосування 2,5D-структури було спробою створити більш багату просторову інформацію. Проте 2.5D-методи все ще обмежені 2D-ядрами, тому вони не можуть застосовувати 3D-фільтри. Використання 3D CNN полягає в тому, щоб отримати більш потужне об’ємне представлення по всіх трьох осях (X, Y і Z). Тривимірна мережа навчена передбачати мітку центрального вокселя відповідно до вмісту навколишніх 3D-латок. Структура мережі загалом схожа на 2D CNN з відмінністю застосування 3D-модулів у кожній необхідної секції, наприклад, у 3D згорткових шарах і шарах 3D підвибірки.

Доступність 3D медичних зображень, а також величезне вдосконалення комп’ютерного обладнання привели до ідеї використання 3D-інформації для сегментації, щоб повністю використовувати переваги просторової інформації. Об’ємні зображення можуть надавати вичерпну інформацію в будь-якому напрямку, а не лише мати один вигляд у 2D підходах і три ортогональних види в 2,5D підходах.

Одна з перших чистих 3D-моделей була введена для сегментації пухлини мозку довільного розміру [7]. Їхню ідею дотримувався Камніцас [14], який розробив багатомасштабну подвійну 3D CNN, в якій існувало два паралельних шляхи з однаковим розміром сприйнятливого поля, а другий шлях отримував латки з підвибіркового представлення зображення. Це дозволило обробити більші області навколо воксела, що принесло користь усій системі з багатомасштабним контекстом. Ця модифікація разом із використанням меншого розміру ядра 3 × 3 забезпечила кращу точність (середній коефіцієнт Dice 0,66). Крім того, було досягнуто меншого часу обробки (3 хвилини для 3D-сканування з чотирма модальностями) порівняно з його оригінальним дизайном.

Щоб вирішити проблему розмірності та скоротити час обробки, Dou et al. у [23] запропонували використовувати набір 3D ядер, які розділяли вагові значення просторово, що допомогло зменшити кількість параметрів.

Щоб сегментувати орган зі складних об’ємних зображень, зазвичай нам потрібна глибока модель для виділення високоінформативних ознак. Але навчання такої глибокої мережі вважається серйозною проблемою для 3D-моделей. У статті «Виклики та найсучасніші рішення» ми детально розглянемо це питання та підсумуємо деякі з доступних ефективних рішень.

Для шару підвибірки введено об’єднання 3D max, яке фільтрує максимальну реакцію в невеликій кубічній околиці, щоб стабілізувати вивчені характеристики проти локального перекладу в 3D просторі. Це допомогло досягти набагато більшої швидкості зближення порівняно з чистою 3D CNN завдяки застосуванню масок згортки з таким же розміром вхідного об’єму. У [4] Kleesiek et al. виконував складне завдання виявлення кордонів мозку за допомогою 3D CNN. Вони застосували двійкову сегментацію, використовуючи порогову функцію відсікання, і зіставили вихідні дані з потрібними мітками, і досягли майже 6% покращення порівняно з іншими звичайними методами. 3D-сприйнятливе поле моделі Kleesiek здатне витягувати більш дискримінаційну інформацію в порівнянні з 2D і 2.5, оскільки ядра засвоїли більш точні та більш організовані орієнтовані шаблони як обсяг. Це добре для сегментації великих органів, які мають більше об’ємної інформації, ніж маленькі органи, які містяться в дуже кількох фрагментах зображення.

Повністю згортка мережа (FCN)

У повністю згортковій мережі (FCN), розробленій Long et al. [10], останній повністю зв'язаний шар замінено на повністю згорткий шар (див. рис. 1.18). Це значне поліпшення дозволяє мережі мати щільне попіксельне передбачення. Щоб

досягти кращої продуктивності локалізації, карти активації високої роздільної здатності поєднуються з виходами з підвищеною дискретизацією і передаються на шари згортки для отримання більш точного результату.

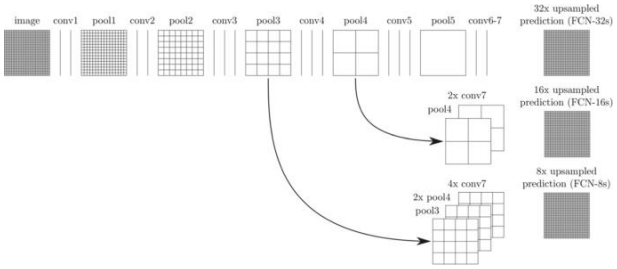

Рис 1.21 Структура FCN

Це покращення дає змогу FCN мати попіксельні прогнози з повнорозмірного зображення замість передбачення по ланцюгах, а також може виконувати передбачення для всього зображення лише за один прохід вперед.

Той самий експеримент, що й у [8], був проведений Nie et al. але із застосуванням FCN [5]. Оскільки в обох експериментах були використані однакові методи та однаковий набір даних, результат чітко показав перевагу FCN над CNN, досягнувши середнього коефіцієнта Dice 0,885 порівняно з 0,864.

FCN для багатоорганної сегментації

Багатоорганна сегментація спрямована на сегментацію кількох органів одночасно, що широко використовується для сегментації органів черевної порожнини [27]. Чжоу та ін. [9] використовували FCN у 2,5D підході для сегментації 19 органів на 3D-зображеннях КТ. У цьому дослідженні був використаний підхід навчання від пікселя до мітки з використанням 2D-зрізів тривимірного об’єму [9]. Для кожного двовимірного виду розрізу було розроблено один окремий FCN (всього три FCN). Зрештою, результати сегментації кожного пікселя були об’єднані з результатами інших FCN для створення кінцевого результату сегментації. Ця техніка дала більшу точність для великих органів, таких як печінка (значення Dice 0,937), але дала меншу точність при роботі з меншими органами, наприклад, підшлунковою залозою (значення Dice 0,553). FCN також використовувався для багатоорганної сегментації з тривимірних зображень [31]. Автори в [6] застосували ієрархічну стратегію від грубого до тонкого, що значно покращило результати сегментації малих органів.

Каскадний FCN (CFCN)

Основна ідея каскадного FCN полягає в тому, щоб укласти серію FCN таким чином, щоб кожна модель використовувала контекстні ознаки, витягнуті картою прогнозування попередньої моделі. Для цього рішенням є застосування паралельного FCN, що може збільшити складність моделі та витрати на

обчислення. Запропонована простіша конструкція полягає в об’єднанні FCN у каскадний спосіб, коли перший FCN сегментує зображення на ROI для другого FCN, де виконується сегментація ураження. Перевага використання такої конструкції полягає в тому, що для кожного етапу можна застосовувати окремі набори фільтрів, а отже, якість сегментації може значно підвищитися. Аналогічно, у, Wu et al. досліджували каскадний FCN для збільшення потенціалу FCN у виявленні кордонів плода на ультразвукових зображеннях. Результати показали кращу продуктивність порівняно з іншими методами уточнення кордонів для ультразвукової сегментації плода.

Багатопотоковий FCN

Вхідні зображення часто відрізняються модальністю (багатомодальні методи) та роздільною здатністю (багатомасштабні методи). Багатопотоковий дизайн може дозволити системі скористатися перевагами кількох форм зображення з одного органу. Багатопотокова техніка була застосована до 3D FCN, щоб максимізувати використання контекстної інформації з різною роздільною здатністю зображення, одночасно застосовуючи багатомодальну техніку, яка покращила стійкість системи проти широкого спектру форм органів. і структура. На відміну від [8], де вміщували кілька джерел, об'єднували вихід кожної модальності в кінці шляху кодера, два класифікатори з пониженою вибіркою були введені в мережу для використання контекстної інформації та сегментування на кількох вихідних рівнях.

Проблема FCN полягає в тому, що сприйнятливий розмір фіксований, тому якщо розмір об’єкта змінюється, FCN намагається виявити їх усі. Одним з рішень є багатомасштабні мережі [22], де розміри вхідних зображень були змінені та подані в мережу. Багатомасштабні методи можуть подолати проблему фіксованого рецептивного розміру в FCN. Однак спільний доступ до параметрів однієї мережі на зображенні зі зміненим розміром не є дуже ефективним способом, оскільки для обробки об’єкта різного масштабу потрібні різні параметри. Як ще одне рішення для сприйнятливого поля фіксованого розміру, для зображень з розміром, більшим за поле зору, FCN можна застосувати у вигляді ковзного вікна по всьому зображенню [12].

FCN, натренований на цілих 3D-зображеннях, має високий класовий дисбаланс між переднім і заднім планом, що призвело до неточної сегментації дрібних органів [9]. Одним з можливих рішень для пом’якшення цієї проблеми є застосування двоетапної сегментації в ієрархічній манері, коли другий етап використовує вихідні дані першого етапу, зосереджуючись більше на

прикордонних областях [6]. У деяких моделях використовуються багатопотокові методики для виявлення кількох органів.

U-Net

2D U-Net

Однією з найвідоміших структур для сегментації медичних зображень є U-Net, спочатку запропонована Ronneberger et al. [6] з використанням концепції деконволюції, введеної в [8]. Ця модель побудована на основі елегантної архітектури FCN. Окрім збільшення глибини мережі до 19 шарів, U-Net має переваги від покращеної конструкції пропуску з’єднань між різними етапами мережі [15]. Він використовує деякі модифікації, щоб подолати компроміс між локалізацією та використанням контексту. Цей компроміс зростає, оскільки великі латки вимагають більшої кількості шарів об’єднання і, отже, зменшить точність локалізації. З іншого боку, латки невеликого розміру можуть спостерігати лише невеликий контекст введення. Запропонована структура складається з двох шляхів аналізу та синтезу. Шлях аналізу відповідає структурі CNN. Шлях синтезу, широко відомий як фаза розширення, складається з шару підвищення дискретизації, за яким слідує шар деконволюції. Найважливішою властивістю U-Net є швидкі з'єднання між шарами з однаковою роздільною здатністю в шляху аналізу до шляху розширення. Ці з’єднання забезпечують основні функції високої роздільної здатності для шарів деконволюції.

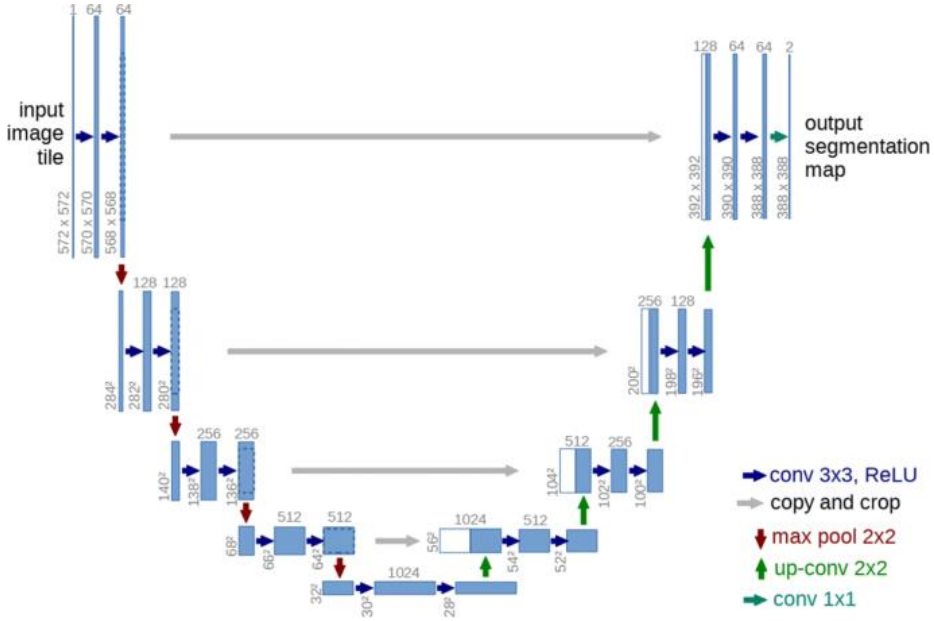

Рис 1.22 Структура U-net

Ця нова структура привернула велику увагу в сегментації медичних зображень і на основі якої було розроблено багато варіацій. Наприклад, Гордієнко та ін. досліджували сегментацію легенів при рентгенівському скануванні за допомогою структурної мережі U-Net. Отримані результати продемонстрували, що U-Net здатна швидко та точно сегментувати зображення. У тому ж дослідженні запропонована модель була протестована на одному ЦП і порівняна з кількома ЦП і ГП, щоб оцінити вплив апаратного забезпечення на продуктивність моделі. Продемонстровані результати показали прискорення в 3 і 9,5 рази відповідно.

DCAN – це ще одна модель, яка застосовувала багаторівневу контекстну інформацію та отримала переваги від допоміжного класифікатора на вершині U-Net. Їхня конструкція показала 0,8001 точності сегментації на сегментації залози, що майже на 2% вище, ніж оригінальний U-Net [2] за менший час 1,5 с на тестове зображення. Підвищена точність пояснюється можливістю структури DCAN боротися з помилками сегментації об’єкта торкання.

3D U-Net

У спробі розширити структуру U-Net більшою просторовою інформацією, Cicek et al. розробив 3D модель U-Net. Запропонована модель була здатна генерувати щільну об’ємну сегментацію з деяких 2D анотованих фрагментів. Мережа змогла виконувати як анотації нових зразків з розріджених, так і ущільнення розріджених анотованих зразків. Вся робота мережі перероблена, щоб мати можливість виконувати тривимірну операцію. Середнє значення IoU (тобто Intersection over Union) 0,863 продемонструвало, що мережа змогла знайти весь тривимірний об’єм з кількох анотованих фрагментів за допомогою зваженої функції втрат softmax.

У 3D U-Net використовували для виявлення судинних кордонів. Початкова модель цього дослідження була названа HED (Holistic Edge Detection), що представляла собою 2D CNN. Оскільки HED страждав від поганої локалізації дрібних судинних об’єктів, автори модифікували мережу, додавши шлях розширення до її структури, і успішно подолали цей недолік. На кожній стадії фази розширення використовували шар змішування та два шари згортки. Структура шару змішування подібна до шару скорочення в GooLeNet, але з іншим використанням та ініціалізацією.

Вони розділили розширювальну частину мережі на три рівні: низький, середній і вищий. На нижньому та середньому рівнях додаються блоки деконволюції, щоб масштабувати зображення до тієї самої роздільної здатності

входу. Отже, окрім сегментованого виходу верхнього рівня (останнього рівня), мережа має ще два виходи сегментації з однаковою роздільною здатністю для покращення кінцевих результатів сегментації.

Як один з недоліків 3D U-Net, розмір вхідного зображення встановлено на 248 × 244 × 64 і не може бути розширений через обмеження пам'яті. Тому вхідний розмір ROI не має достатньої роздільної здатності, щоб представити анатомічну структуру на всьому зображенні. Цю проблему можна вирішити, розділивши вхідний об'єм на кілька партій і використавши їх для навчання та тестування.

V-Net

Ймовірно, одним з найвідоміших виведень U-Net є V-Net, запропонована Milletari et al. Вони застосували згортки на шляху стискання мережі, як для виділення функцій, так і для зменшення роздільної здатності, вибравши відповідний розмір ядра та крок (розмір ядра становить 2 × 2 × 2, а крок – 2). Згортки служать для об'єднання з перевагою меншого обсягу пам'яті, оскільки, на відміну від шарів об'єднання, перемикачі, які відображають вихідний рівень об'єднання назад на вхід, не потрібно зберігати для зворотного поширення. Це схоже на деконволюцію програми замість об'єднання в пул. Фаза розширення витягуватиме об'єкти та розширює об'єднану карту об'єктів із низькою роздільною здатністю та в кінцевому підсумку створить двоканальну об'ємну сегментацію на останньому згортковому шарі. Потім вихідні дані перетворюються на імовірнісну карту сегментації та переходять до воксельної softmax для сегментації фону та переднього плану. V-Net використовувався з більшим сприйнятливим полем (охоплює 50–100% вхідного зображення) і багатомасштабним (чотири різні роздільної здатності) і забезпечував на 12% вищий коефіцієнт Dice порівняно з оригінальною V-Net.

Техніки навчання мережі

Під глибоким наглядом

Основна ідея глибокого нагляду полягає в тому, щоб забезпечити пряме спостереження за прихованими шарами та поширити його на нижні шари, а не просто робити це на вихідному шарі. Ця ідея була реалізована в немедичних цілях шляхом додавання супутньої цільової функції до прихованих шарів. Також у GoogLeNet було здійснено нагляд для двох прихованих шарів 22-рівневої мережі.

Dou et al. у застосовуваних підходах із глибоким наглядом для сегментації об'ємів 3D КТ печінки. Це було досягнуто шляхом підвищення дискретизації функцій нижнього та середнього рівня за допомогою шарів деконволюції та

застосування шару softmax для ущільнення результатів класифікації. Їх представлені результати показують не тільки кращу конвергенцію, але й меншу похибку навчання та перевірки.

У подібному підході було введено три класифікатори, щоб класифікувати вихідні характеристики середнього рівня із стискаючої частини структури, подібної до U-Net. Засекречені виходи були використані як регулятор на етапі навчання. Багаторівнева контекстна інформація в мережі допомогла покращити можливості локалізації та дискримінації. Крім того, допоміжні класифікатори посилили зворотний потік поширення градієнта на етапі навчання.

Слабо контролюється

Існуючі контрольовані підходи для автоматизованої сегментації медичних зображень вимагають анотації на рівні пікселів (рівень вокселів у випадку 3D), яка не завжди доступна в різних випадках. Також робити таку анотацію буде дуже втомливо і дорого. У загальній обробці зображень ця проблема полегшилася за допомогою аутсорсингу послуг маркування, таких як Amazon MTurk, які, очевидно, не можна застосувати до медичних зображень. Крім того, використання даних із зображенням, наприклад, із двійковою міткою, яка показує наявність або відсутність шаблону, є новим підходом до вирішення цієї проблеми.

Ця ідея була реалізована шляхом використання «міток точок», які, по суті, є розташуванням одного пікселя, що вказує на наявність вузла, щоб зменшити залежність системи до повністю анотованих зображень. Вони зайняли позицію цього пікселя, витягли навколишній об’єм і використали його як позитивний зразок для навчання, використовуючи статистичну інформацію про вузлики. Наприклад, зазвичай вузлики представлені у вигляді 3–7 послідовних зрізів і мають різну ширину від 3 до 28 пікселів. Метод досяг прийнятної чутливості 80% зі слабко міченими зразками.

Feng та ін. [25] використовували CNN для повністю автоматизованої сегментації легеневих вузлів у слабо позначених даних. Їх метод заснований на висновку [9], який продемонстрував здатність CNN ідентифікувати дискримінаційні регіони. Відповідно, вони використовували класифікацію CNN для виявлення зрізів, що містять вузлики, і в той же час вони використовували ознаки дискримінаційної області для виділення дискримінаційних областей із зрізу, які називають картою активації вузлів (NAM). Більше того, була введена мульти-GAP CNN, щоб скористатися перевагами NAM з більш дрібних шарів з більш високою просторовою роздільною здатністю, так само, як ідея.

Представлений результат 0,55 бала Dice був близьким, але менш точним порівняно з підходами під повним контролем. Очікувалась перевага методів із глибоким контролем, оскільки вони використовують анотацію на рівні пікселів, і це надає важливу інформацію для роботи з різними шаблонами інтенсивності, особливо на краях. Проте запропонований метод допомагає витягувати ділянки, що містять вузлики, більш автоматично в порівнянні з [2], який був більш встановлений на жорстких припущеннях, отриманих зі статистичної інформації про розмір і форму вузлика.

Передача навчання

Передача навчання визначається як здатність системи розпізнавати та використовувати знання, засвоєні в попередній області джерела, для нового завдання.

Трансферне навчання можна здійснити за допомогою двох підходів, тобто як тонке налаштування мережі, попередньо натренованої на загальних зображеннях [30], і тонке налаштування мережі, попередньо натренованої на медичних зображеннях, для іншого цільового органу або завдання. Було доведено, що навчання з перенесенням має кращу продуктивність, коли завдання вихідної та цільової мережі більш подібні, але навіть передача ваги далеко віддалених завдань була кращою, ніж випадкова ініціалізація. Вагові коефіцієнти беруться із загальної мережі (VGG16), а потім тонко налаштовуються на сегментації пренатального зображення в ультразвуку. Аналогічно, оригінальні ваги були взяті з віддаленого застосування та застосовані для виявлення поліпів. Тому авторам довелося тонко налаштувати всі шари. Вони помітили збільшення чутливості на 25% шляхом точного налаштування всіх шарів порівняно з останнім шаром. Проте були деякі експерименти, які навчалися з нуля, що також дало кращі результати в порівнянні з точним налаштуванням попередньо навченої мережі.

Трансферне навчання може здійснюватися на трьох основних рівнях: (1) повна адаптація мережі, яка полягає в ініціалізації ваг за допомогою попередньо підготовленої мережі (а не випадкової ініціалізації), але їх оновлення всіх під час навчання [9]. (2) Часткова мережева адаптація, яка полягає в ініціалізації параметра мережі з попередньо навченої мережі, але заморожування ваг для перших кількох шарів і оновлення останніх шарів під час навчання [11, 29]. (3) Нульова адаптація, яка полягає в ініціалізації ваг для всієї мережі з попередньо навченої моделі і не змінюється взагалі. Як правило, підхід нульової адаптації з іншої медичної мережі не рекомендується через величезну різницю у

зовнішньому вигляді органу (цілі). Особливо не рекомендується, якщо джерела були навчені на загальних зображеннях. Крім того, об’єкти на біомедичних зображеннях можуть мати дуже різний зовнішній вигляд і розмір, тому передача навчання з моделей із величезними варіаціями зовнішнього вигляду органів може не зменшити результат сегментації.

Структура мережі

Однак вибір підходу також залежить від структури мережі. Для більш дрібних мереж повна адаптація забезпечує кращу продуктивність, але в більш глибоких структурах частково адаптивні підходи зменшать час зближення та обчислювальне навантаження.

Орган і модальність

Інший важливий елемент у трансферному навчанні – це орган-мішень та спосіб його візуалізації. Наприклад, вони застосували повну передачу ваги для T1 МРТ і часткову передачу для модальності T2. Результати роботи показують, що підхід повної адаптації має кращу середню оцінку Dice (ADS) порівняно з нульовою та частковою адаптацією при ультразвуковій сегментації нирок, оскільки модальність має багато шумів, а також орган має величезну різницю зовнішнього вигляду.

Розмір набору даних

Розмір цільового набору даних також є рольовим параметром, який визначає рівень навчання з передачі. Якщо цільовий набір даних невеликий, а кількість параметрів велика (більш глибокі мережі), повна адаптація може призвести до переобладнання. Таким чином, часткова адаптація є кращим вибором. З іншого боку, якщо розмір цільового набору даних відносно більший, проблема переобладнання не виникне, і повна адаптація може працювати нормально. Таджбахш та ін. у оцінено вплив розміру набору даних на підхід повної адаптації. Результати показують підвищення чутливості на 10% (з 62 до 72%) за рахунок збільшення набору даних з чверті до повного розміру набору навчальних даних.

Проблеми та сучасні рішення

Обмежені анотовані дані

Методи глибокого навчання значно покращили точність сегментації завдяки їх здатності працювати зі складними умовами. Щоб отримати цю можливість, мережі зазвичай вимагають великої кількості анотованих зразків для виконання навчального завдання. Збір такого величезного набору даних

анотованих випадків при обробці медичних зображень часто є дуже складним завданням, а виконання анотації для нових зображень також буде дуже втомливим і дорогим. Для вирішення цієї проблеми широко використовується кілька підходів. Таблиця 2 підсумовує деякі з широко використовуваних наборів даних різної сегментації органів.

Збільшення даних

Найпоширенішим методом збільшення розміру навчального набору даних є збільшення даних, яке є застосуванням набору афінного перетворення, наприклад, перевернути, поворот, дзеркало, до зразків, а також збільшення значень кольору (сірого). У немедичному експерименті оцінюється ефективність збільшення даних, і результати показують, що традиційні методи збільшення здатні підвищити продуктивність до семи відсотків.

Передача навчання

Іншим рішенням для вирішення цієї проблеми є передача навчання з успішних моделей, реалізованих у тій же області (або навіть в інших областях). У порівнянні зі збільшенням даних, навчання з перенесенням є більш конкретним рішенням, яке залежить від багатьох параметрів, як пояснюється в розділі «Навчання з перенесенням».

Patch-Wise навчання

У цій стратегії зображення розбивається на кілька ланок, які можуть бути як перекриваються, так і випадковими. Випадкове виправлення може призвести до більшої дисперсії серед латок і кращої конвергенції, особливо в 3D-випадках, коли N випадкового перегляду об’єму, що цікавить (VOI) береться як навчальна вибірка[65] (якщо N = 3, це 2,5D підхід) [2]. Тим не менш, випадкове виправлення має проблему дисбалансу класів і нижчу точність у порівнянні з патчами, що перекриваються. Таким чином, він не рекомендується для сегментації дрібних органів. Патчі, що перекриваються, показали більш високу точність, але інтенсивні обчислення [23]. Продуктивність відносно залежить від перекриття патчів і розміру міні-латок.

Слабко контрольоване навчання

Методи навчання без нагляду також використовувалися для вилучення більш надійних даних із слабо позначених даних, а потім використання вилучених анотованих даних для навчання мережі, що розглядається як гібридний підхід для вирішення цієї проблеми [2].

Розріджена анотація

Оскільки повне анотування даних не завжди можливо, особливо в 3D-випадках, часто нам доводиться використовувати дані з розрідженими анотаціями. Застосування функцій зважених втрат, де ваги для немаркованих даних встановлені на нуль, є ключем до навчання лише з позначених пікселів у рідко анотованих обсягах [17].

Ефективний негативний набір

Ще одна проблема, яку необхідно подолати, — зібрати відповідний набір негативних зразків. Щоб підвищити здатність мережі розрізняти хибнопозитивні випадки, негативний набір повинен містити випадки, схожі на вузли, але не позитивні. Наприклад, автори в [2] вибрали випадкові зразки всередині легенів зі шкалою Хаунсфілда від 400 до 500. Цей діапазон HU містить зразки, подібні до вузликів, які є негативними. Сорок відсотків зібраних зразків за допомогою цього підходу використовуються як позитивні зразки, а решта – для негативного набору.

Класовий дисбаланс

Дуже часто в обробці медичних зображень анатомія, що цікавить, займає лише дуже малу частину зображення. Отже, більшість вилучених плям належить до фонової області, тоді як ці невеликі органи (аномалії) мають більшу важливість. Навчання мережі з такими даними часто призводить до того, що навчена мережа зміщується у задній план і потрапляє в пастку локальних мінімумів.

Популярним рішенням цієї проблеми є повторне зважування зразка, коли під час навчання до плям переднього плану застосовується більша вага [6]. Розроблено автоматичну модифікацію повторного зважування зразка з використанням шару втрат Dice та коефіцієнта Dice [4]. Проте ефективність у боротьбі з крайнім класовим дисбалансом обмежена. Тренування на основі латок у поєднанні з вибором патчів можуть допомогти вирішити проблему дисбалансу класів [18]. По суті, під час створення навчального набору механізм керування може бути налаштований на збалансовану кількість патчів на задньому та передньому плані.

Інший підхід до вирішення цієї проблеми — вибіркова втрата, при якій втрати не будуть розраховані для всього зображення, а лише деякі випадкові пікселі (області) будуть вибрані для обчислення втрат [6]. Випадковість відбору кандидатів для оцінки збитків є основним недоліком цього методу, який може вплинути на точність розрахунку збитків.

Проблеми з навчанням глибоких моделей

Переобладнання

Переобладнання відбувається, коли модель може фіксувати закономірності та закономірності в навчальному наборі з достатньо високою точністю порівняно з необробленими екземплярами проблеми [30]. Як правило, головною причиною переобладнання є малий розмір навчального набору даних. Тому будь-яке рішення, яке може збільшити розмір даних («Обмежені анотовані дані»), також може допомогти у боротьбі з проблемою переобладнання.

Наприклад, було доведено, що створення кількох переглядів латки (збільшення), а не одного перегляду, має позитивний ефект у переобладнанні [25]. Іншим методом обробки переобладнання є застосування «випадання» під час процесу навчання, щоб відкинути вихід випадкового набору нейронів на кожній ітерації з повністю підключених шарів. Аналогічно, дроп-коннект, який, як було доведено, допомагає вирішити проблему переобладнання.

Час навчання

Скорочення часу навчання та швидша конвергенція є основною темою багатьох досліджень. Одним із попередніх рішень цієї проблеми є застосування шарів об’єднання, які можуть зменшити розмірність параметрів [23]. Останні рішення на основі об’єднання використовують згортку з кроком, що має той самий ефект, але полегшує мережу. Пакетна нормалізація, що відноситься до центрування значень пікселів навколо 0 шляхом віднімання їх на середнє зображення[4], також відома як ефективний ключ для швидшої конвергенції[5, 17]. Пакетна нормалізація є більш кращим підходом для покращення конвергенції мережі, оскільки, як повідомляється, не має негативного впливу на продуктивність, тоді як методи об’єднання та зменшення вибірки призвели до втрати корисної інформації.

Зникнення градієнта

Доведено, що більш глибокі мережі мають кращу продуктивність, але вони борються з проблемою вибуху або повного зникнення поширюваного сигналу (градієнта) [25], іншими словами, остаточні втрати не можуть бути ефективно поширені назад до неглибоких шарів. Ця проблема є більш серйозною в 3D-моделях.

Загальне рішення для зникнення градієнта полягає в тому, щоб мати підходи з глибоким контролем, за яких вихідні дані проміжних прихованих шарів будуть масштабуватися за допомогою деконволюції та передаватися до softmax,

щоб отримати від них передбачення. Допоміжні втрати разом із вихідною втратою прихованого шару поєднуються з посиленням градієнта [23].

У підходах із навчанням «з нуля» ретельна ініціалізація ваги також покращує ефект у зникненні градієнта, як показано в [21], де вага ядер ініціалізувалася шляхом вибірки з нормального розподілу.

Зовнішній вигляд органу

Неоднорідний вигляд цільового органу є однією з великих проблем у сегментації медичних зображень. Орган-мішень або ураження можуть сильно відрізнятися за розміром, формою та місцем розташування від пацієнта до пацієнта [27]. Збільшення глибини мережі повідомляється як ефективне рішення.

Неоднозначна межа з обмеженим контрастом між цільовими органами та сусідніми тканинами є відомою проблемою візуалізації. Зазвичай це викликано коефіцієнтом ослаблення на КТ та часом релаксації при МРТ [23]. Підходи, засновані на мультимодальності, можуть вирішити цю проблему [5]. Крім того, відомо, що інформація про суперпіксель корисна для сегментації перекриття або органів на межі [2]. Іншим успішним підходом для дотику до об’єктів того ж класу є застосування функції зваженої втрати з більшою вагою, виділеною для розділення фонових міток між органами дотику [12].

3D-завдання

Усі вище згадані проблеми під час навчання можуть бути набагато серйознішими при роботі з об’ємними даними через низьку дисперсію між цільовим та сусідніми вокселами, більшу кількість параметрів, а також обмежені об’ємні навчальні дані. Наявність обчислювального висновку відома як проблема, яка перешкоджає використанню тривимірних підходів. Виявляється, що застосування щільного висновку значно скорочує час висновку приблизно до хвилини для одного сканування мозку. Виконання стратегії виключення для видалення областей, які навряд чи містять цільовий орган, може ефективно зменшити простір пошуку та привести до швидшого висновку [2].

2. МОДЕЛЮВАННЯ ТА СТВОРЕННЯ РІШЕННЯ ДЛЯ СЕГМЕНТАЦІЇ МЕДИЧНИХ ЗОБРАЖЕНЬ НА БАЗІ MASK R-CNN

2.1 Що таке згортка нейронна мережа (CNN)?

Згортка нейронна мережа (CNN) — це тип штучної нейронної мережі, яка використовується для розпізнавання та обробки зображень, оптимізована для обробки піксельних даних. Таким чином, згорткові нейронні мережі є фундаментальними і основними будівельними блоками для задачі комп'ютерного зору сегментації зображень (сегментація CNN).

Архітектура згорткової нейронної мережі складається з трьох основних шарів:

- Згортковий шар: цей шар допомагає абстрагувати вхідне зображення як карту об'єктів за допомогою фільтрів і ядер.
- Об'єднаний шар: цей шар допомагає зменшити вибірку карт об'єктів, підсумовуючи наявність об'єктів у виправленнях карти об'єктів.
- Повністю підключений шар: Повністю з'єднані шари з'єднують кожен нейрон в одному шарі з кожним нейроном в іншому шарі.

Об'єднання шарів CNN дає змогу розробленій нейронній мережі навчитися ідентифікувати та розпізнавати об'єкт, що цікавить, на зображенні. Прості згорткові нейронні мережі створені для класифікації зображень і виявлення об'єктів з одним об'єктом на зображенні.

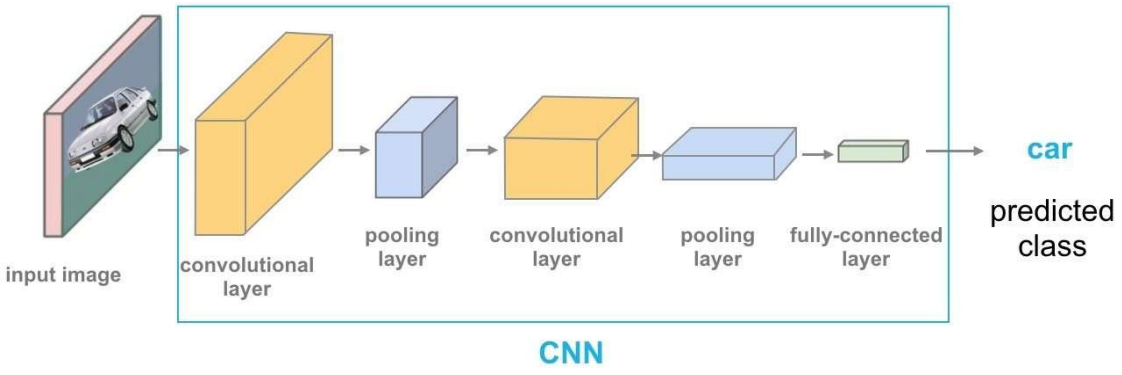

Рис 2.1 Концепція архітектури CNN: як працює згортка нейронна мережа.

У більш складній ситуації з кількома об'єктами на зображенні проста архітектура CNN не є оптимальною. Для таких ситуацій Mask R-CNN — це найсучасніша архітектура, яка базується на R-CNN (також згадується як RCNN).

Що таке R-CNN?

R-CNN або RCNN, означає згорткову нейронну мережу на основі регіонів, це тип моделі машинного навчання, який використовується для завдань комп'ютерного зору, зокрема для виявлення об'єктів.

Як працює R-CNN?

На наступному зображенні зображено концепцію CNN на основі регіонів (R-CNN). Цей підхід використовує обмежувальні рамки між областями об'єкта, які потім оцінюють згорткові мережі незалежно від усіх регіонів інтересу (ROI), щоб класифікувати кілька областей зображення до запропонованого класу.

Архітектура RCNN була розроблена для вирішення завдань виявлення зображень. Крім того, архітектура R-CNN є основою Mask R-CNN, і вона була вдосконалена до того, що ми знаємо як Faster R-CNN.

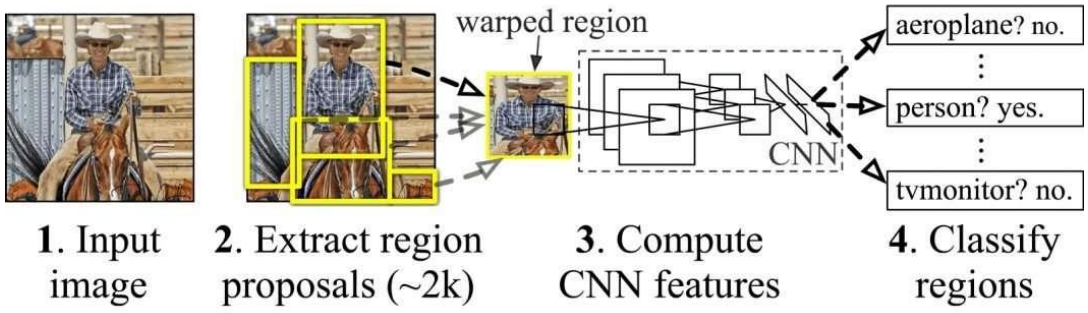

Рис 2.2 - Концепція R-CNN – регіональні згорткові мережі

2.2 Архітектура мережі Mask R-CNN

Сегментація екземплярів

Сегментація екземплярів об'єктів – це останній підхід, який дає нам найкраще з обох світів. Він об'єднує завдання виявлення об'єктів, метою якого є виявлення класу об'єкта разом із передбаченням обмежувальної рамки в зображенні та завданням семантичної сегментації, яка класифікує кожен піксель за попередньо визначеними категоріями. Таким чином, він дозволяє нам виявляти об'єкти на зображенні, точно сегментуючи маска для кожного екземпляра об'єкта.

Сегментація екземплярів дозволяє нам вирішувати такі проблеми, як виявлення пошкоджень, де важливо знати ступінь пошкодження. Інший випадок використання – у випадку самокерованих автомобілів, де важливо знати положення кожного автомобіля на сцені. Створення площі будівлі для кожної окремої будівлі є популярною проблемою в галузі ГІС. Тому дає нам перевагу використовувати модель Mask R-CNN для вирішення таких проблем реального життя.

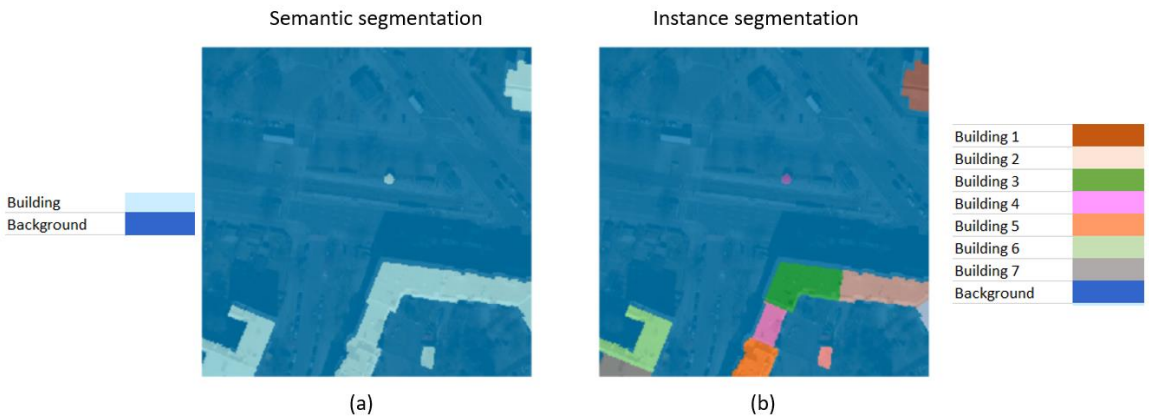

Рис 2.3 Типи сегментації

Зображення (a) має два типи пікселів. Один належить об'єкту (Будівлі), а інший — фону. Важко підрахувати кількість будівель, присутніх на зображенні. На зображенні (b) кожна будівля ідентифікована як окрема сутність, отже, долається обмеження семантичної сегментації.

Архітектура маски R-CNN

Mask R-CNN — це найсучасніша модель, наприклад, сегментація, розроблена на основі Faster R-CNN. Faster R-CNN — це згортка нейронна мережа на основі регіонів [2], яка повертає обмежувальні прямокутники для кожного об’єкта та його мітку класу з оцінкою впевненості.

Щоб зрозуміти Mask R-CNN, давайте спочатку обговоримо архітектуру Faster R-CNN, яка працює в два етапи:

Етап 1: Перший етап складається з двох мереж, магістральної (ResNet, VGG, Inception тощо) та мережі пропозицій регіону. Ці мережі запускаються один раз для кожного зображення, щоб надати набір пропозицій регіону. Пропозиції регіонів – це регіони на карті об’єктів, які містять об’єкт.

Етап 2: На другому етапі мережа прогнозує обмежувальні рамки та клас об'єкта для кожного із запропонованих регіонів, отриманих на етапі 1. Кожна запропонована область може мати різний розмір, тоді як повністю підключені шари в мережах завжди потребують вектора фіксованого розміру для прогнозування. Розмір цих запропонованих регіонів фіксується за допомогою пулу RoI (який дуже схожий на MaxPooling) або методу RoIAlign.

Розглянемо приклад використання методу виявлення слідів побудови.

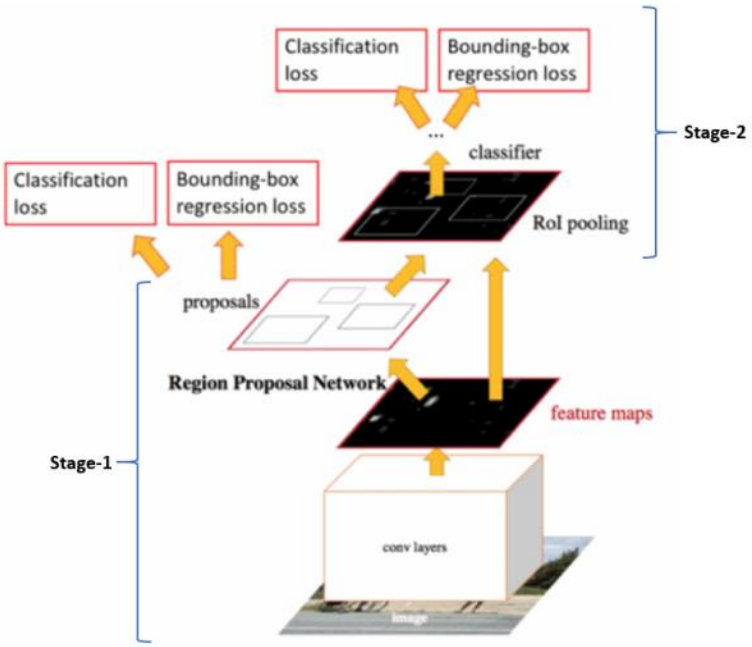

Рис 2.4 Faster R-CNN – це єдина уніфікована мережа для виявлення об’єктів

Faster R-CNN прогнозує клас об’єктів і обмежувальні рамки. Mask R-CNN є розширенням Faster R-CNN з додатковою гілкою для прогнозування масок сегментації для кожного регіону інтересу (RoI).

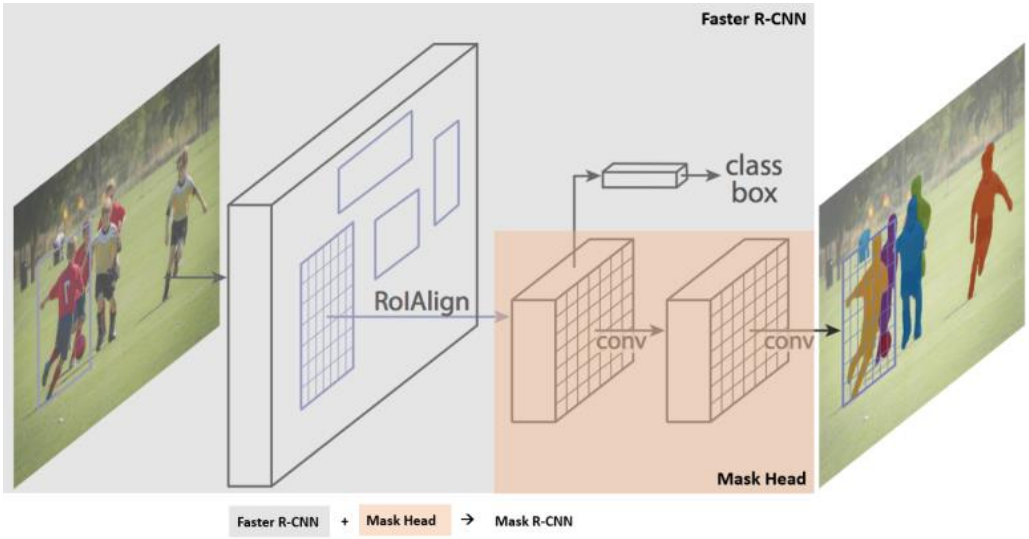

Рис 2.5 Структура Mask R-CNN, наприклад сегментація

На другому етапі Faster R-CNN пул RoI замінюється на RoIAlign, який допомагає зберегти просторову інформацію, яка вирівнюється у випадку пулу RoI. RoIAlign використовує двійкову інтерполяцію для створення карти об’єктів фіксованого розміру, напр. 7 x 7.

Вихідні дані з шару RoIAlign потім надходять у головку маски, яка складається з двох шарів згортки. Він генерує маску для кожного RoI, таким чином сегментуючи зображення у вигляді пікселя до пікселя.

Покращення PointRend

Сегментаційні моделі можуть мати тенденцію генерувати занадто гладкі межі, які можуть бути не точними для об’єктів або сцен з неправильними межами. Щоб отримати чіткі межі сегментації, модуль нейронної мережі рендеринга на основі точки під назвою PointRend був доданий як доповнення до існуючої моделі. Цей модуль використовує методологію з класичної комп’ютерної графіки та дає перспективу візуалізації задачі сегментації. Моделі сегментації зображень часто передбачають мітки на звичайній сітці з низькою роздільною здатністю, наприклад, 1/8 вхідного. Ці моделі використовують інтерполяцію, щоб збільшити прогнози до початкової роздільної здатності. На відміну від цього, PointRend використовує ітераційний алгоритм поділу для підвищення масштабу прогнозів шляхом прогнозування міток точок у вибраних місцях навченою маленькою нейронною мережею. Цей метод забезпечує ефективний вихід з високою роздільною здатністю.

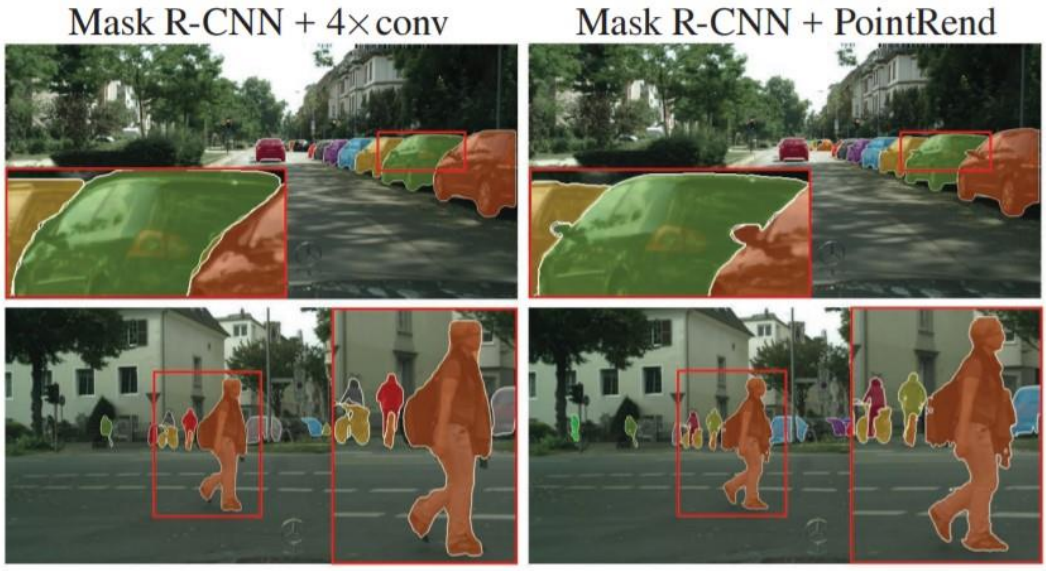

Рис 2.6 Покращення PointRend (праворуч) над оригінальною моделлю сегментації (ліворуч)

2.3 Архітектура нейронної мережі U-Net

Архітектура U-Net, вперше опублікована в 2015 році, стала революцією в області глибокого навчання. Архітектура з великим відривом перемогла на Міжнародному симпозіумі з біомедичної візуалізації (ISBI) у 2015 році у багатьох

категоріях. Деякі з їхніх робіт включають сегментацію нейронних структур у електронно-мікроскопічні стеки та зображення мікроскопії прохідного світла.

Завдяки цій архітектурі U-Net сегментацію зображень розміром 512X512 можна обчислити за допомогою сучасного графічного процесора за невеликі проміжки часу. Варіантів і модифікацій цієї архітектури було багато завдяки її феноменальному успіху. Деякі з них включають LadderNet, U-Net з увагою, повторювану та залишкову згортку U-Net (R2-UNet) і U-Net із залишковими блоками або блоками з щільними зв'язками.

Хоча U-Net є значним досягненням у сфері глибокого навчання, не менш важливо розуміти попередні методи, які використовувалися для вирішення такого роду подібних завдань. Одним із основних прикладів, який завершується, був підхід з розсувними вікнами, який з великим відривом переміг у конкурсі сегментації EM на ISBI у 2012 році. Підхід ковзного вікна зміг створити широкий набір зразків патчів, крім вихідного набору навчальних даних.

Цей результат відбувся тому, що він використовував метод налаштування мережі архітектури розсувних вікон, створюючи мітку класу кожного пікселя як окремі одиниці, забезпечуючи локальну область (патч) навколо цього пікселя. Ще одним досягненням цієї архітектури було те, що вона могла досить легко локалізуватися на будь-якому наборі навчальних даних для відповідних завдань.

Однак підхід з розсувними вікнами зазнав двох основних недоліків, яким протистояла архітектура U-Net. Оскільки кожен піксель розглядався окремо, то отримані латки ми перекривали багато. Таким чином, було створено велику кількість загальної надмірності. Іншим обмеженням було те, що загальна процедура навчання була досить повільною та забирала багато часу та ресурсів. Доцільність роботи мережі викликає сумніви з наступних причин.

U-Net — це елегантна архітектура, яка вирішує більшість проблем, що виникають. Для цього підходу використовується концепція повністю згорткових мереж. Мета U-Net полягає в тому, щоб охопити як особливості контексту, так і локалізацію. Цей процес успішно завершується типом побудованої архітектури. Основна ідея реалізації полягає у використанні послідовних стискаючих шарів, за якими відразу йдуть оператори підвищення дискретизації для досягнення більш високої роздільної здатності на вхідних зображеннях.

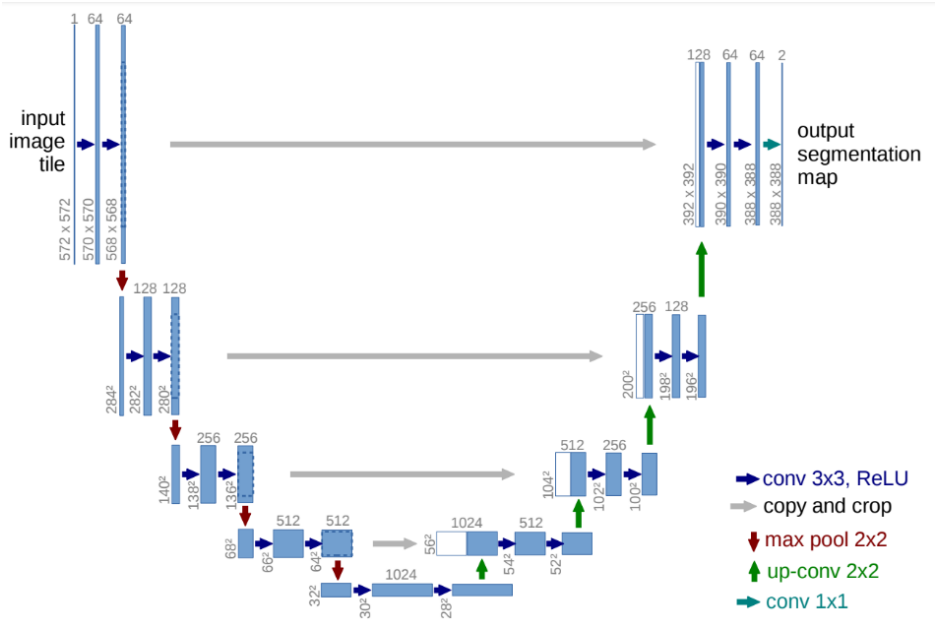

Рис 2.7 U-Net architecture

Якщо коротко поглянути на архітектуру, показану на зображенні, ми можемо помітити, чому її, ймовірно, називають архітектурою U-Net. Форма сформованої архітектури має форму «U», звідси й наступна назва. Просто поглянувши на структуру та численні елементи, залучені в процес побудови цієї архітектури, ми можемо зрозуміти, що побудована мережа є повністю згортковою мережею. Вони не використовували жодних інших шарів, таких як щільні або сплощені або інші подібні шари. Візуальне уявлення показує початковий шлях скорочення, за яким слід розширюється шлях.

Архітектура показує, що вхідне зображення передається через модель, а потім за ним слідує пара згорткових шарів з функцією активації ReLU. Ми можемо помітити, що розмір зображення зменшується з 572X572 до 570X570 і, нарешті, до 568X568. Причина такого скорочення полягає в тому, що вони використовували згортки без доповнення (визначили згортки як «дійсні»), що призводить до зменшення загальної розмірності. Крім блоків Convolution, ми також помічаємо, що у нас є блок кодера з лівого боку, а потім блок декодера з правого боку.

Блок кодера має постійне зменшення розміру зображення за допомогою максимального об’єднання шарів кроків 2. Ми також маємо повторювані згорткові шари зі збільшенням кількості фільтрів в архітектурі кодера. Як тільки ми досягаємо аспекту декодера, ми помічаємо, що кількість фільтрів у згорткових шарах починає зменшуватися разом із поступовим підвищенням дискретизації в

наступних шарах аж до самого верхнього. Ми також помічаємо, що використання з'єднань пропуску, які з'єднують попередні виходи з шарами в блоках декодера.

Це з’єднання пропуску є життєво важливою концепцією для збереження втрат від попередніх шарів, щоб вони сильніше відображалися на загальних значеннях. Крім того, науково доведено, що вони дають кращі результати та призводять до швидшої конвергенції моделі. В останньому блоці згортки ми маємо пару шарів згортки, за якими слідує останній шар згортки. Цей шар має фільтр 2 з відповідною функцією для відображення результату. Цей останній шар можна змінити відповідно до бажаної мети проекту, який ви намагаєтеся виконати.

2.4 Архітектура мережі Fast R-CNN

Намагаючись зрозуміти ключові концепції та віхи виявлення об’єктів за допомогою глибокого навчання, попередня стаття про регіони з CNN висвітлювала особливості та деталі навчання кожного етапу в одній із перших в історії мереж детекторів із основою глибокого навчання. У цій статті розглянуто деталі його популярного наступника, Fast R-CNN. З огляду на технологічний прогрес і збільшення обчислювальної потужності, які ми спостерігали з 2014 року[19], коли була опублікована стаття R-CNN, нам легко побачити очевидні підводні камені R-CNN, які автори виявили тоді:

- Багатоетапне, дороге навчання: окремі навчальні процеси, необхідні для всіх етапів мережі — тонка настройка CNN щодо пропозицій об’єктів, вивчення SVM для класифікації вектора ознак кожної пропозиції від CNN і вивчення регресора обмежувальної рамки до тонка настройка пропозицій об’єктів (докладнішу інформацію див. у розділі Регіони з CNN) виявляється тягарем з точки зору часу, обчислень та ресурсів. Наприклад, для навчання SVM нам знадобляться функції тисяч можливих пропозицій регіонів, які будуть записані на диск з попереднього етапу.
- Повільний час тестування: враховуючи цей багатоступеневий конвеєр, виявлення за допомогою простої мережі VGG як магістральної CNN займає 47 секунд на зображення.

Деталі архітектури:

Щоб краще зрозуміти, як і чому Fast R-CNN покращив ефективність і продуктивність мереж R-CNN і SPP, давайте спочатку розглянемо його архітектуру.

- Швидкий R-CNN складається з CNN (зазвичай попередньо навчений для завдання класифікації ImageNet), його остаточний рівень об’єднання замінений шаром «об’єднання ROI», а його останній рівень FC замінено двома гілками

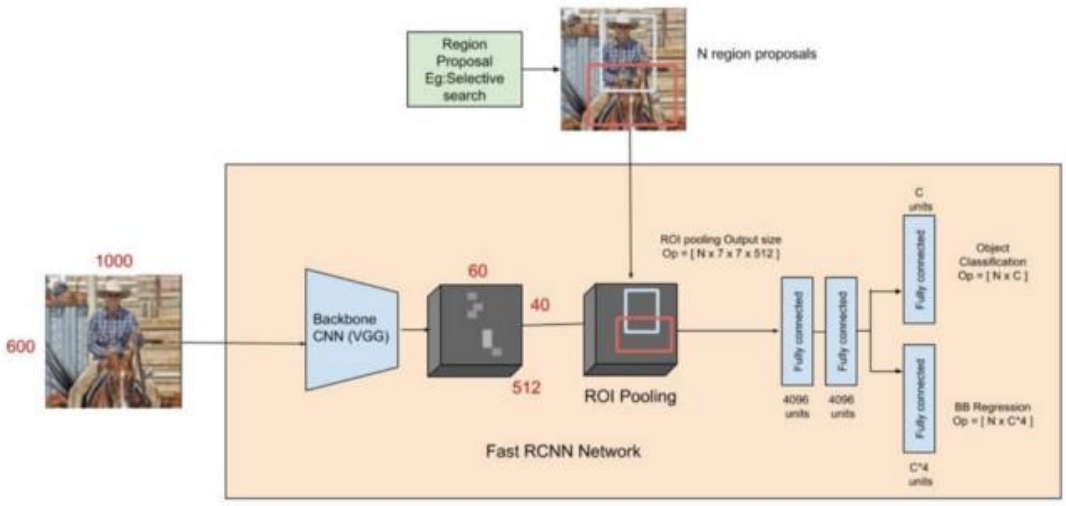

Рис 2.8 The Fast R-CNN pipeline

- Все зображення подається в магістраль CNN, і одержуються ознаки з останнього шару згортки. Залежно від використовуваної базової CNN вихідні карти функцій набагато менші за розмір вихідного зображення. Це залежить від кроку магістралі CNN, яка зазвичай становить 16 у випадку магістралі VGG.
- Тим часом вікна пропозицій об’єкта отримуються за допомогою алгоритму пропозиції регіону, такого як вибірковий пошук[4]. Як пояснюється в розділі «Регіони з CNN», пропозиції об’єктів — це прямокутні області на зображенні, які означають наявність об’єкта.
- Частина карти основних функцій, яка належить цьому вікну, потім подається на шар ROI Pooling.

Навчання:

- Під час навчання кожен міні-пакет складається з N=2 зображень. Міні-пакет складається з 64 ROI з кожного зображення.
- До рівня об’єднання ROI все зображення проходить через CNN. Таким чином, усі ROI з одного зображення спільно використовують обчислення та пам’ять у прямому і зворотному напрямку через CNN (перед шаром об’єднання ROI).

- Стратегія вибірки (вибірка всієї рентабельності інвестицій у пакеті з дуже кількох зображень) дозволяє навчати весь модуль — генерацію ознак, класифікацію та регресію — разом, на відміну від мереж SPP. У мережах SPP більшість вхідних ROI надходять з різних зображень, тому процес навчання для виявлення лише точно налаштовує повністю підключені шари після створення об’єктів, оскільки неможливо оновити ваги перед рівнем SPP (деякі з ROI можуть мати дуже великі рецептивні поля на вихідному зображенні). Фактичні характеристики все ще надходять із попередньо підготовленої мережі, яка була навчена для класифікації. Це обмежує точність мереж SPP.
- Пакетний розмір мережі дуже малий для CNN до рівня об’єднання ROI (розмір пакету = 2), але набагато більший (розмір пакету = 128) для наступних шарів softmax та регресії.

Результати

- На момент публікації на VOC12, VOC10 (з додатковими даними) і VOC07 Fast R-CNN досягла найсучаснішої продуктивності.
- Швидкий R-CNN обробляє зображення в 45 разів швидше, ніж R-CNN, під час тестування і в 9 разів швидше під час навчання. Він також навчається в 2,7 рази швидше і запускає тестові зображення в 7 разів швидше, ніж SPP-Net. При подальшому використанні усіченого SVD час виявлення мережі зменшується більш ніж на 30% за рахунок зниження mAP лише на 0,3.
- Інші внески в статті показують, що для більших мереж також важлива тонка настройка рівнів конверсії (а не тільки рівнів FC, як SPPnets) для досягнення значно кращого mAP. Для менших мереж це покращення може бути не таким великим.
- Багатозадачне тренування не тільки легше, але й покращує продуктивність, оскільки завдання впливають один на одного під час навчання. Отже, навчання різних етапів мережі в цілому покращує її спільну представницьку владу (основну систему CNN)

3. АНАЛІЗ ПРОВЕДЕНИХ ЕКСПЕРИМЕНТІВ ТА ОТРИМАНИХ РЕЗУЛЬТАТІВ

3.1. Розроблення плану проведення експериментів

В якості експериментального випробування необхідно було провести кілька експериментів із різними датасетами.

Підготовка даних для навчання.

Для більш точних результатів для проведення експерименті було вирішено обрати набір даних – BraTS (Brain Tumor Segmentation).

Більше прикладів з датасету можна побачити на рис Б.1 - Б.4

BraTS завжди зосереджувався на оцінці найсучасніших методів сегментації пухлин головного мозку при мультимодальному магнітно-резонансному томографі (МРТ). BraTS 2020 використовує багатоінституційні передопераційні МРТ-сканування і в першу чергу зосереджується на сегментації по суті гетерогенних (за зовнішнім виглядом, формою та гістологією) пухлин головного мозку, а саме гліом. Крім того, щоб визначити клінічну значимість цього завдання сегментації, BraTS'20 також зосереджується на прогнозуванні загальної виживаності пацієнтів і має намір оцінити алгоритмічну невизначеність сегментації пухлини.

В якості даних навчання, валідації та тестування для останнього року надається багато багатозасобових рутинних клінічно отриманих передопераційних мультимодальних МРТ-сканувань гліобластоми (ГБМ/ГРГ) та гліоми нижчого класу (ЛГГ) з патологічно підтвердженим діагнозом та наявною ОС. Зокрема, набори даних, використані в цьому році, були оновлені, починаючи з BraTS'19, з більш рутинними клінічно отриманими 3T мультимодальними МРТ-скануваннями, із супроводжуючими мітками, сертифікованими експертною комісією нейрорадіологами.

Усі мультимодальні сканування BraTS доступні у вигляді файлів NIfTI (.nii.gz) і описують а) native (T1) і б) постконтрастний T1-зважений (T1Gd), в) зважений T2 (T2) і d) флюїд T2 Ослаблені обсяги відновлення інверсії (T2-FLAIR) та отримані за допомогою різних клінічних протоколів та різних сканерів із кількох (n = 19) установ, згаданих тут як автори даних.

Усі набори даних візуалізації були сегментовані вручну від одного до чотирьох оцінювачів, дотримуючись того самого протоколу анотації, і їхні анотації були схвалені досвідченими нейрорадіологами. Анотації включають пухлину, що посилює GD (ET — мітка 4), перитуморальний набряк (ED —

мітка 2) та некротичне та не посилене ядро пухлини (NCR/NET — мітка 1), як описано в доповіді BraTS 2012-2013 року в MIT та останні[підсумкових документ BraTS. Надані дані розповсюджуються після їх попередньої обробки, тобто спільно реєструються в одному анатомічному шаблоні, інтерполюються до тієї ж роздільної здатності (1 мм^3) і позбавляються черепа.

Для отримання доступу до вказаного датасету необхідно пройти деяку процедуру, після якої будуть надані посилання на завантаження датасету.

Початком процедури є реєстрація на офіційному сайті Центру біомедичних обчислень та аналітики оброки зображень (https://ipp.cbica.upenn.edu/). Процес реєстрації профілю зображений на рисунку 3.2. Після заповнення цієї форми через потрібен деякий час щоб профіль було одобренно до створення. На період отримання доступу, на сайті можна було отримати доступ до датасетів останніх трьох років (з 2018 по 2020). Для кожного з них необхідно заповнити форму запиту, яка зображена на рисунку 3.1.

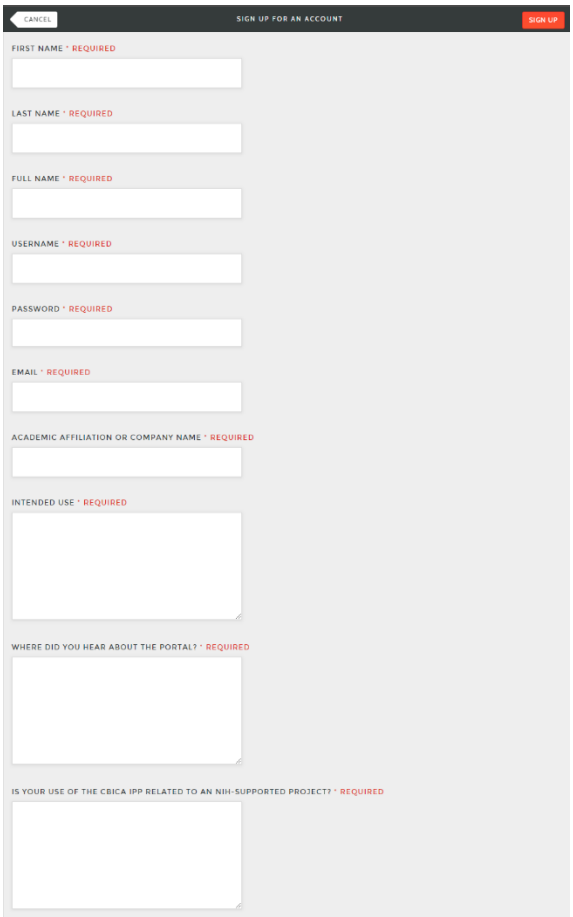

Рис 3.1 Реєстрація профілю для отримання доступу до BraTS

Після заповнення форми також потрібен деякий час після чого в

обліковому записі на 48 годин будуть доступні посилання на архіви з датасетами для тренування та для валідації сегментації.

Для експерименту було обрано датасет BraTS’20, який містить 369 наборі даних у форматі NIfTi.

Інформаційна технологія нейровізуалізації (NIfTI) - це відкритий формат файлу, який зазвичай використовується для зберігання даних візуалізації мозку, отриманих за допомогою методів магнітно -резонансної томографії.

У форматі ni NIfTI fti перші три виміри зарезервовані для визначення трьох просторових вимірів — x, y та z , а четвертий вимір зарезервовано для визначення часових точок — t. Решта розмірів, від п’ятого до сьомого, призначені для інших цілей. П’ятий вимір, однак, все ще може мати деякі заздалегідь визначені види використання, наприклад, для зберігання специфічних для вокселів параметрів розподілу або для зберігання даних на основі векторів.

Так як розроблена програма для проведення експериментів розрахована для обробки зображень у форматі .jpg та .png, то наборы данных NIfTi необхідно було конвертувати до цих форматів. Вивчивши структуру формату NIfTy, візуалізація якого приведена на рисунку 3.10 у спрощеному вигляді. Для цього було розроленно скрипт, який розкладує формат NIfTy на файли .png.

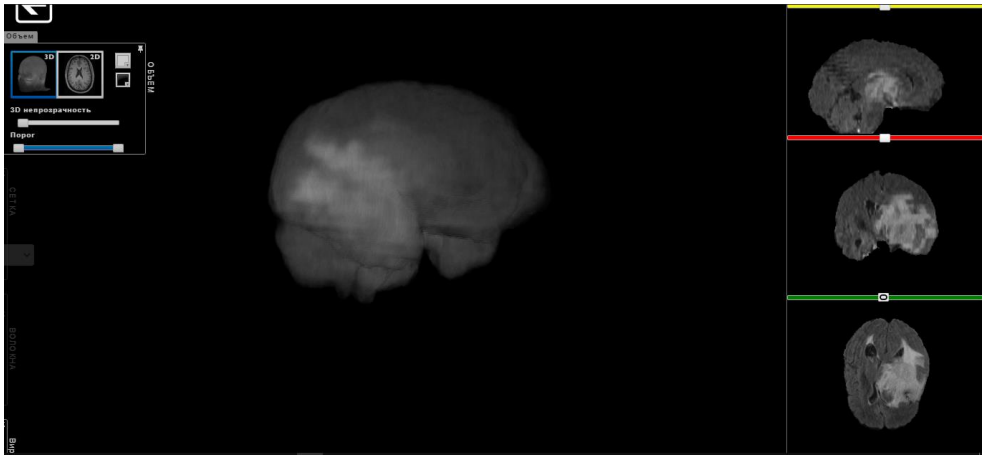

Рис 3.2 BraTS’20 набір Flair

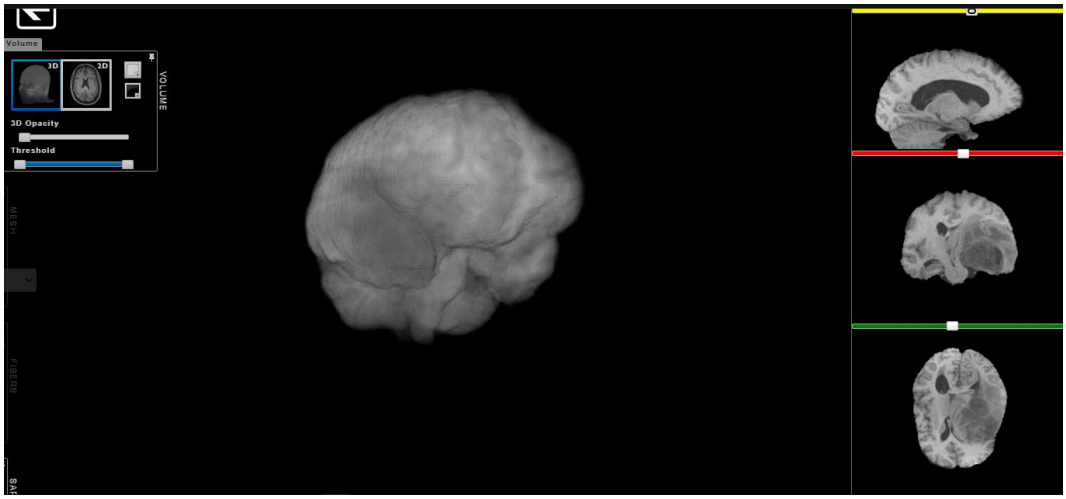

Рис 3.3 BraTS’20 набір T1

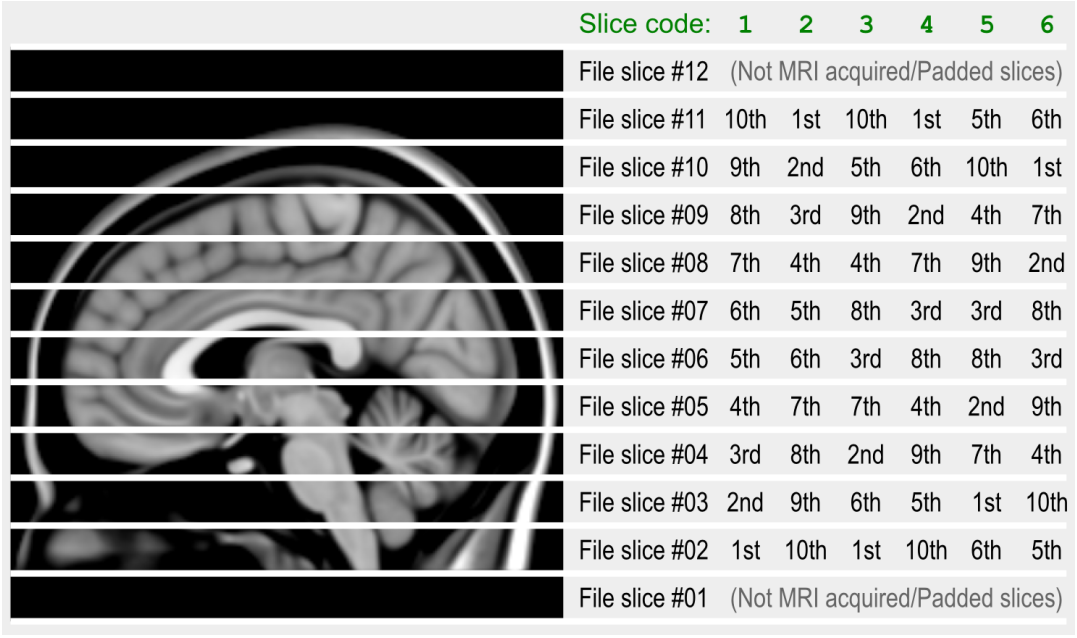

Рис 3.4 Спрощений приклад формату NIfTy

Для проведення експерименту було обрано Flair формат з BraTS’20 датасету. Датасет нараховує 369 наборів файлів для кожного типу МРТ. Після виконання скрипта на одному із набору Flair файлу NIfTy отримано приблизно 150 зображень. На всі 369 наборів отримано приблизно 57 тисяч зображень розміром 240x240.

Для проведення дослідження на BraTS’2020 з отриманного набору зображень було обрано 10 тысяч зображень (майже 64 повних NIfTy файлы) для train набору та 5 тысяч - для val.

Налаштування робочого оточення

Апаратна конфігурація робочих станцій ідентичне та приведене нижче:

1) CPU: Core i5 5600 CPU @ 3.2GHz
    a) Кількість ядер: 4, без HyperThreading
2) RAM: 8 GB DDR4
3) HDD: 500 GB
    a) Інтерфейс: SATA
4) Операційна система: Ubuntu Linux 20.04

Оскільки навчання на процесорі могло зайняти кілька місяців безперебійного роботи ПК було прийнято рішення використовувати графічну карту. На Пк була встановлена дескретная GPU Nvidia GeForce 920м. Програма для навчання Cuda може працювати тільки з відеокартами Nvidia тому була встановлена остання версія бібліотеки.

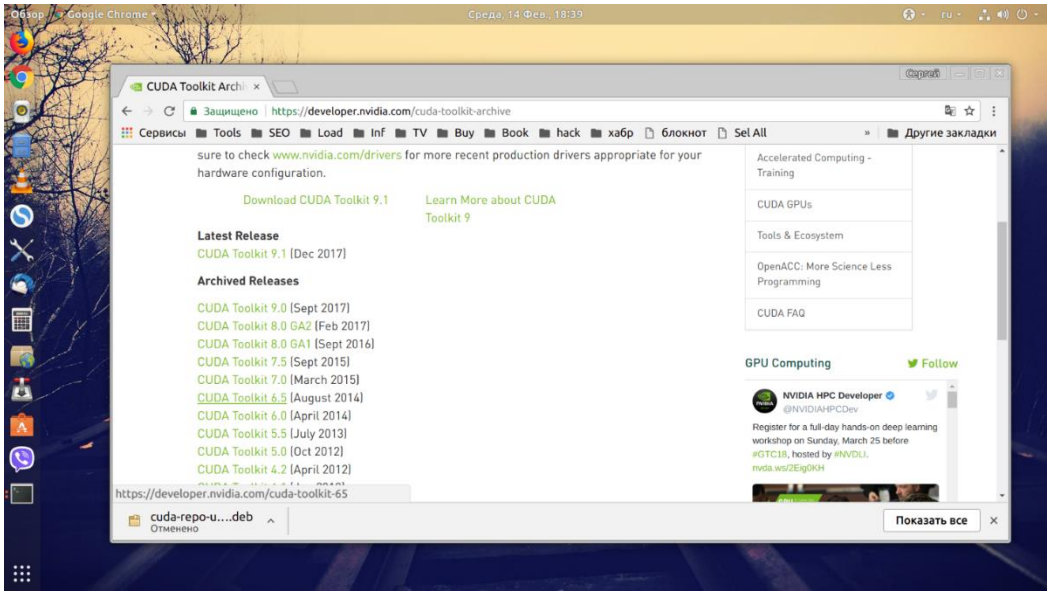

Рис 3.5 Сайт Nvidia Cuda Toolkit для завантаження Cuda

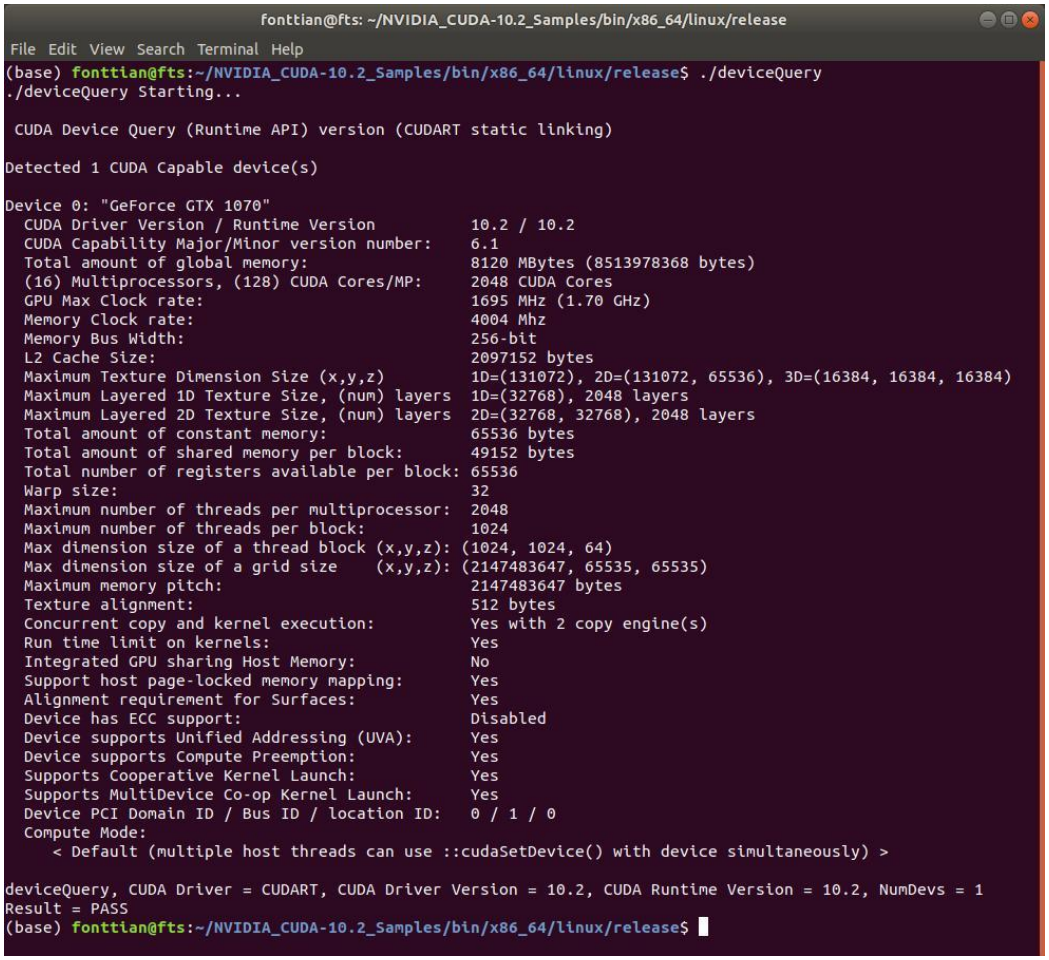

Рис 3.6 Установка Cuda 9.0

Також в код програми була додана перевірка, в разі виявлення Cuda на комп'ютері, вона починала навчання з її використанням

```
device = torch.device("cuda:0" if torch.cuda.is_available() else "cpu")
```

Рис 3.7 Перевірка на наявність Cuda на ПК

Однак після довгих спроб встановлення різних додаткових пакетів, була отримана помилка, що вимагало використання більш старої версії cuda, але оскільки підтримка була неможлива, було ухвалено рішення продовжити роботу з процесором.

```
Python 3.7.5 (default, Oct 31 2019, 15:18:51) [MSC v.1916 64 bit (AMD64)] :: Anaconda, Inc. on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import torch
>>> torch.cuda.is_available()
True
>>> torch.tensor([1.0, 2.0])
tensor([1., 2.])
>>> torch.tensor([1.0, 2.0]).cuda()
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  File "C:\Users\balci\Anaconda3\envs\pytorch\lib\site-packages\torch\tensor.py", line 130, in __repr__
    return torch._tensor_str._str(self)
  File "C:\Users\balci\Anaconda3\envs\pytorch\lib\site-packages\torch\_tensor_str.py", line 311, in _str
    tensor_str = _tensor_str(self, indent)
  File "C:\Users\balci\Anaconda3\envs\pytorch\lib\site-packages\torch\_tensor_str.py", line 209, in _tensor
    formatter = _Formatter(get_summarized_data(self) if summarize else self)
  File "C:\Users\balci\Anaconda3\envs\pytorch\lib\site-packages\torch\_tensor_str.py", line 87, in __init__
    nonzero_finite_vals = torch.masked_select(tensor_view, torch.isfinite(tensor_view) & tensor_view.ne(0))
  File "C:\Users\balci\Anaconda3\envs\pytorch\lib\site-packages\torch\functional.py", line 227, in isfinite
    return (tensor == tensor) & (tensor.abs() != inf)
RuntimeError: CUDA error: no kernel image is available for execution on the device
>>>
```

Рис 3.8 Помилка під час запуску Cuda

3.2. Проведення експериментів

В основі розробленого ML сервісу знаходиться модель глибокого машинного навчання, заснована на свёрточних (convolution) нейронних мережах. Сервіс приймає на вхід знімки з МРТ і генерує продубльоване зображення зі знайденою зоною.

Відображення результатів по коефіцієнтам:

```
korchuni@korchuni-Inspiron-5558:~/Documents/diploma/segmenttumor_v3/v3$ python3
t.py
2021-11-11 00:49:04.523048: W tensorflow/stream_executor/platform/default/dso_lo
ader.cc:64] Could not load dynamic library 'libcudart.so.11.0'; dlerror: libcuda
rt.so.11.0: cannot open shared object file: No such file or directory; LD_LIBRAR
Y_PATH: /usr/local/cuda-6.5/lib64:
2021-11-11 00:49:04.523089: I tensorflow/stream_executor/cuda/cudart_stub.cc:29]
 Ignore above cudart dlerror if you do not have a GPU set up on your machine.
        0: iou = 0.05372882748574076; dice = 8.787373039107246e-08;
        1: iou = 0.13376259818709235; dice = 1.1287466861418753e-06;
        2: iou = 0.14032006643785364; dice = 5.477233999498706e-06;
        3: iou = 0.2634907518959468; dice = 1.6353420737883316e-05;
        4: iou = 0.11177606865768559; dice = 1.6728614759441964e-05;
result 0: iou = 0.14061566253286384; dice = 7.955177982671386e-06;

        0: iou = 0.04975954748831651; dice = 7.23342817389546e-05;
        1: iou = 0.020807836301589226; dice = 5.883749740030111e-05;
        2: iou = 0.13216167905438184; dice = 0.00044365860971081844;
        3: iou = 0.23658536585365852; dice = 0.0006989008145111065;
        4: iou = 0.10819327731092437; dice = 0.0019329985663324662;
```

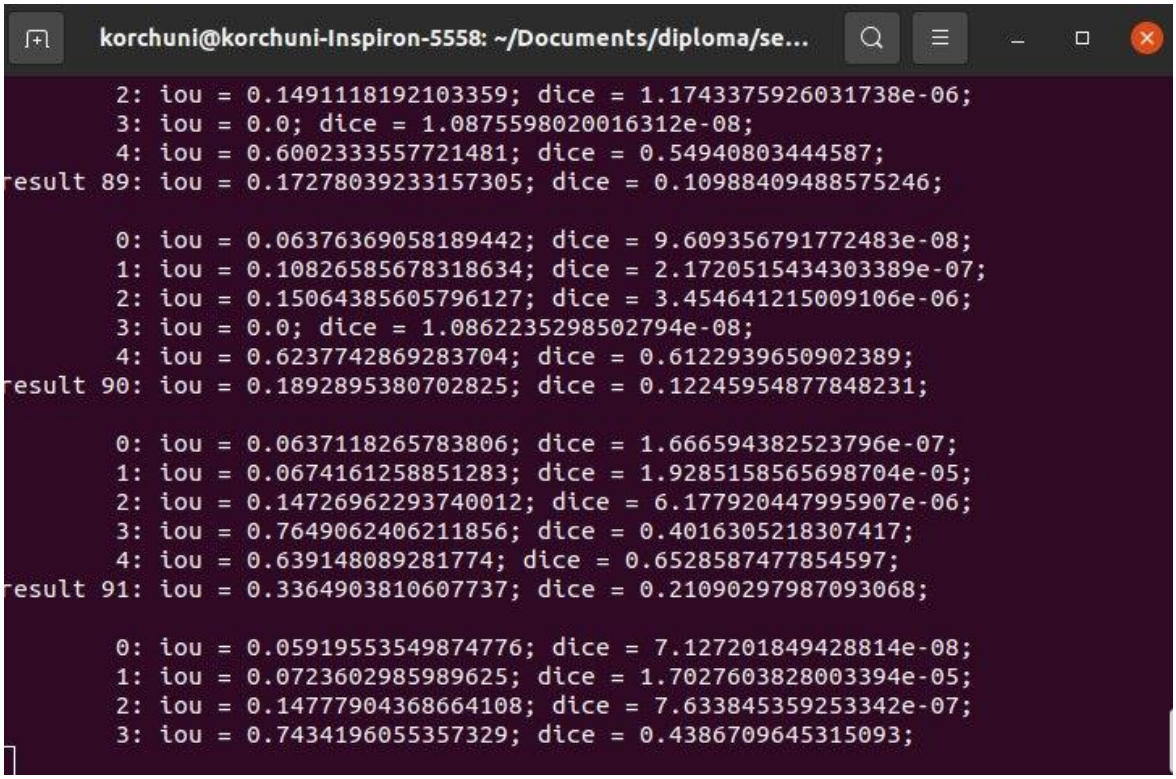

Рис 3.9-3.10 Результат порівняння коефіцієнтів IoU, Dice у процесі навчання

Після процесу навчання ми можемо отримувати результат по зображенням. В результаті це виглядає так:

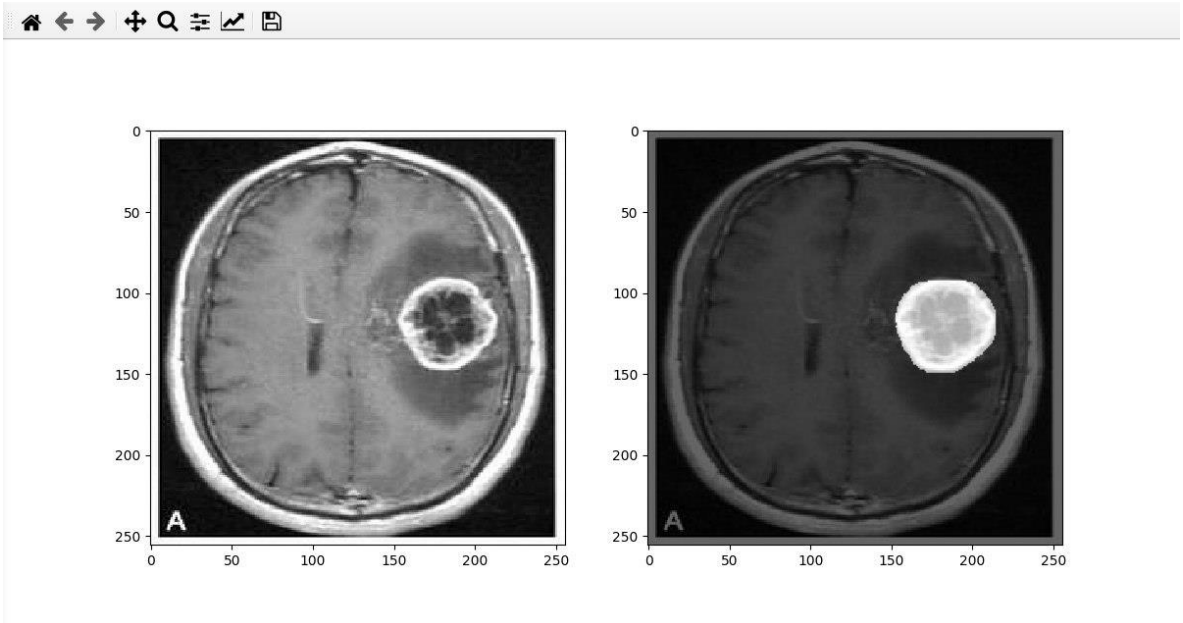

Рис 3.11 Приклад глибокого машинного навчання

Кілька слів про те, які інструменти були використані нами для створення рішення.

- Anaconda - зручна система управління пакетами для мов Python і R.
- Tensorflow - відкрита програмна бібліотека для машинного навчання, розроблена компанією Google.
- Keras - надбудова над фреймворками Deeplearning4j, TensorFlow і Theano.
- OpenCV - бібліотека алгоритмів комп'ютерного зору, обробки зображень та чисельних алгоритмів загального призначення з відкритим кодом.
- Flask - фреймворк для створення веб-додатків на мові програмування Python.

Матеріалів про принципи роботи, видах і завданнях свёрточних нейронних мереж, в тому числі і російською мовою, вже більш ніж достатньо. Зупинимося лише на одній конкретній реалізації, Mask-RCNN - архітектури для локалізації та виділення контурів об'єктів на зображеннях. Є й інші відмінні рішення зі своїми достоїнствами і недоліками, наприклад, UNet, але кращої якості вдалося домогтися саме на Mask-RCNN.

Принцип її роботи полягає у виділенні на зображенні невеликих областей, для кожної з яких проводиться оцінка ймовірності наявності в цій області цільового об'єкта. R-CNN відмінно справлялася з поставленим завданням, проте швидкість її роботи залишала бажати кращого. Закономірним розвитком стали мережі Fast R-CNN і Faster R-CNN, які отримали поліпшення в алгоритмі обходу зображень, що дозволило значно підвищити швидкодію. На виході в Faster R-CNN з'являється розмітка прямокутним виділенням, що позначає межі об'єкта, що не завжди достатньо для вирішення завдання.

У Mask R-CNN додається ще й попіксельне накладення маски, що дозволяє отримати точне обрис об'єкта.

Наочно boundary box і маски можна побачити на результаті роботи моделі (включений фільтр по мінімальної площі будівлі):

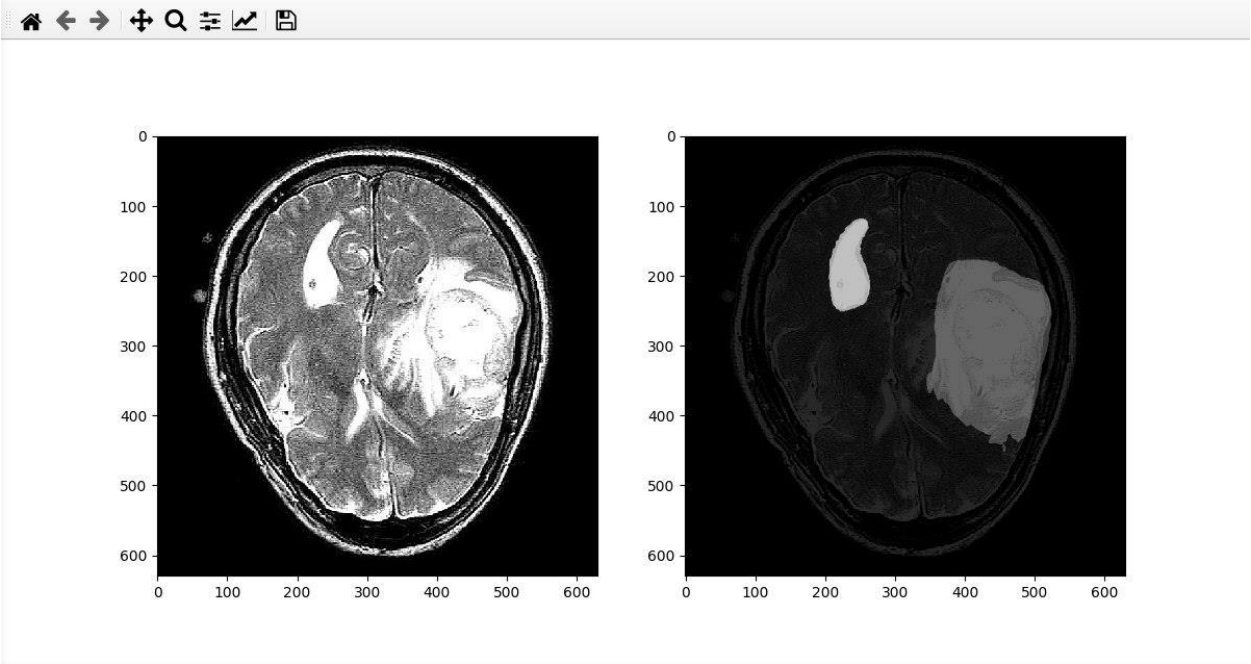

Рис 3.12 Попіксельне накладення маски

Умовно можна виділити 4 етапи в роботі цієї мережі:

- стандартне для всіх згортальних нейронних мереж виділення ознак на зображенні, таких як лінії, вигини, контрастні кордону та інші;
- Region Proposal Network (RPN) сканує невеликі фрагменти зображення, іменовані anchors (якоря) і визначає, чи містить цей якір ознаки, характерні для цільового класу;
- Region of Interest Classification and Bounding Box. На цьому етапі мережу, грунтуючись на результатах попереднього етапу намагається виділити на фотографії великі області, імовірно містять цільовий об'єкт;
- Segmentation Masks. На цьому етапі з області, отриманої при накладенні boundary box, виходить маска шуканого об'єкта.

Крім усього, мережа виявилася дуже гнучкою в налаштуванні, і нам вдалося перебудувати її на обробку зображень з додатковими інформаційними шарами. Більше прикладів фотограмметрії можна побачити на рис А.1 - А.5

3.3 Аналіз одержаних результатів

Метрики, що застосовуються для оцінки якості моделі

Tensorboard

Відстеження процесу навчання моделей зручно контролювати за допомогою Tensorboard - зручного інструменту для контролю метрик, що дозволяє в режимі реального часу отримувати дані про якість моделі і зіставляти їх з іншими моделями.

Створення звітів (summary) у форматі TensorBoard відбувається на момент конструювання графа обчислень.

Як вже сказано нагорі, кожний наш звіт ми збираємо у summary. Це контейнер, в якому зберігається масив value, кожен з яких представляє якусь подію, яку ми хочемо візуалізувати.

Summary надалі буде збережено у файл на файловій системі, де його і прочитає TensorBoard.

Таким чином, нам необхідно створити FileWriter, вказавши граф, який ми хочемо візуалізувати та створити Summary, в який ми складатимемо наші значення.

```
        let summary = Summary(scope: scope)
        let fileWriter = try FileWriter(folder: writerURL, identifier: "iMac", graph:
graph)
```

Запустивши програму та оновивши сторінку, ми вже можемо бачити граф, який ми створили у коді. Він буде інтерактивним, так що по ньому можна переміщатися.

Далі, ми хочемо бачити зміну певної скалярної величини в часі, наприклад значення функції втрат (loss function or cost function) та accuracy нашої нейронної мережі. Для цього додаємо виходи наших операцій у summary:

```
        try summary.scalar(output: accuracy, key: "scalar-accuracy")
        try summary.scalar(output: cross_entropy, key: "scalar-loss")
```

Таким чином, після кожного кроку обчислень нашої сесії TensorFlow автоматично віднімає значення наших операцій і передасть їх на вхід

результуючого Summary, який ми збережемо в FileWriter (як це зробити я опишу нижче).

IoU індекс

Intersection-Over-Union (IoU), також відомий як індекс Жаккарда, є однією з найбільш часто використовуваних метрик у семантичній сегментаці. IoU — це дуже простий показник, який надзвичайно ефективний.

IoU — це область перекриття між прогнозованою сегментацією та основною істиною, поділена на площу об'єднання між передбачуваною сегментацією та основною істиною, як показано на зображенні зліва. Цей показник коливається в діапазоні від 0–1 (0–100%), при цьому 0 означає відсутність перекриття, а 1 означає сегментацію, яка повністю перекривається.

Для бінарної (два класи) або багатокласової сегментації середнє значення IoU зображення розраховується шляхом отримання IoU кожного класу та їх усереднення. (У коді це реалізовано дещо інакше).

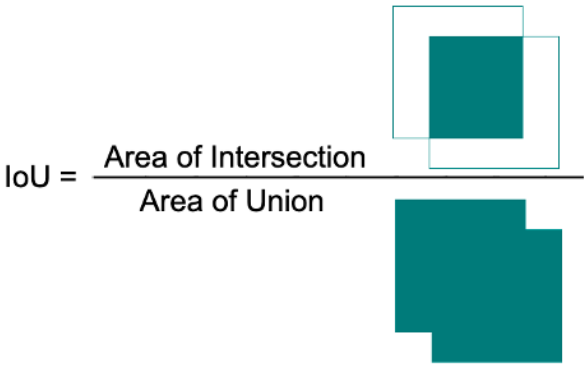

Рис 3.13 Tensorboard IoU

Приклад коду для обчислення IoU з використанням бібліотеки geometry.shapely:

```
from shapely.geometry import Polygon

true_polygon = Polygon([(2, 2), (2, 6), (5, 6), (5, 2)])
predicted_polygon = Polygon([(3, 3), (3, 7), (6, 7), (6, 3)])
print(true_polygon.intersection(predicted_polygon).area / true_polygon.union
(predicted_polygon).area)

>>> 0.3333333333333333
```

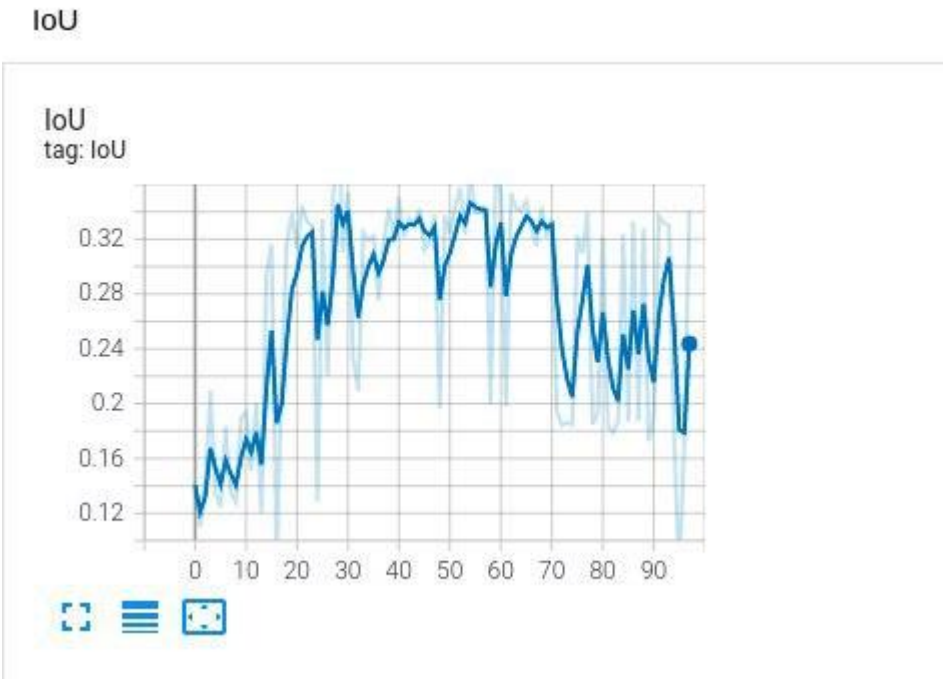

Рис 3.14 Tensorboard IoU

Dice коефіцієнт

Простіше кажучи, коефіцієнт кубика дорівнює 2 * площі перекриття, поділеній на загальну кількість пікселів в обох зображеннях.

Коефіцієнт Dice дуже схожий на IoU. Вони позитивно корелюють, тобто якщо один скаже, що модель A краща за модель B у сегментації зображення, то інший скаже те саме. Як і IoU, обидва вони коливаються від 0 до 1, причому 1 означає найбільшу подібність між прогнозованим і істиною.

Реалізація коефіцієнта Dice:

```
def dice_coef(y_true, y_pred, smooth=1):
  intersection = K.sum(y_true * y_pred, axis=[1,2,3])
  union = K.sum(y_true, axis=[1,2,3]) + K.sum(y_pred, axis=[1,2,3])
  dice = K.mean((2. * intersection + smooth)/(union + smooth), axis=0)
  return dice
```

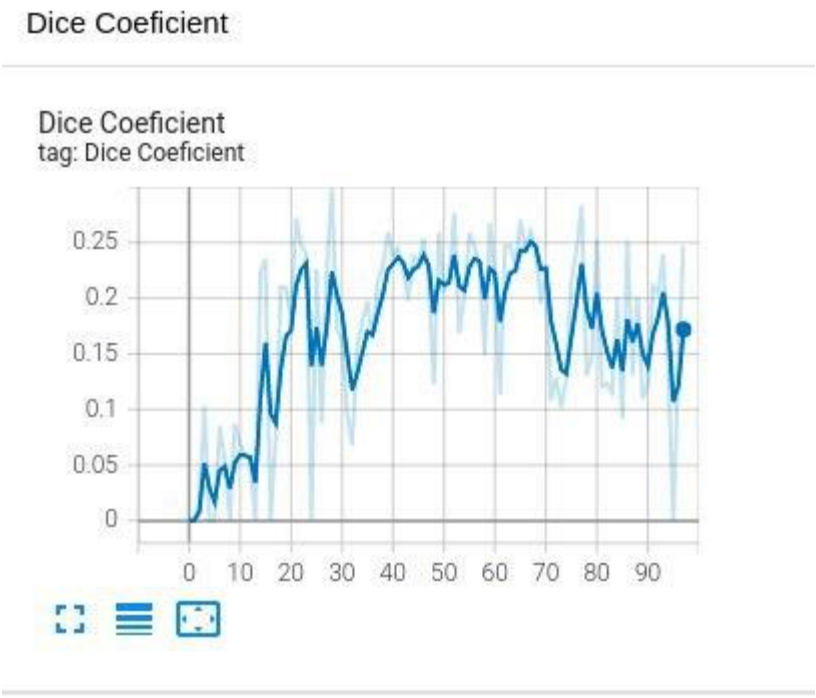

Рис 3.15 Tensorboard Dice

Train/Loss коефіцієнт

Для оптимізації алгоритму машинного навчання використовується функція втрат. Втрата розраховується на основі навчання та перевірки, а її інтерпретація базується на тому, наскільки добре модель працює в цих двох наборах. Це сума помилок, допущених для кожного прикладу в наборах навчання або перевірки. Значення втрат означає, наскільки погано чи добре модель веде себе після кожної ітерації оптимізації.

Показник точності використовується для вимірювання продуктивності алгоритму інтерпретованим способом. Точність моделі зазвичай визначається за параметрами моделі і розраховується у відсотках. Це міра того, наскільки точний прогноз вашої моделі порівняно з істинними даними.

Train/Loss - основний показник ефективності мережі.

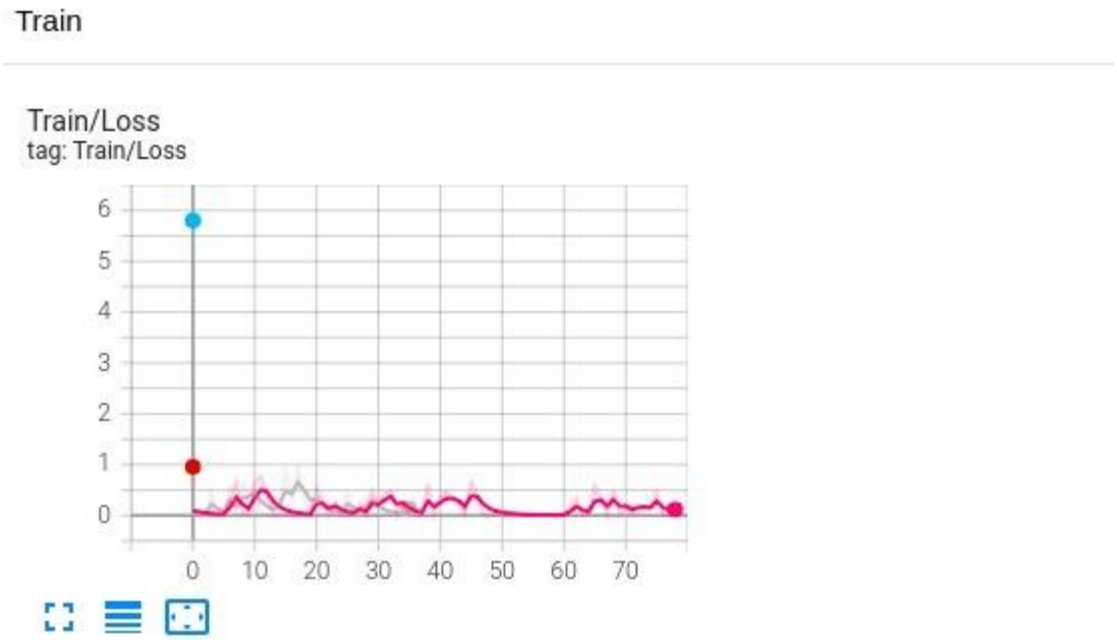

Рис 3.16 Tensorboard Train/loss

3.4 Порівняння метрик Mask R-CNN та Faster R-CNN

Порівняння за кількома параметрами, а саме IoU, Train/Loss з рівною кількістю епох.

Оскільки експеримент проводився на однакових машинах, було виділено близько 120 годин для зняття метрик і результатів.

Результат за кількістю епох Mask R-CNN = 97 епох, за такий самий час результат Faster R-CNN склав 260 епох.

Нижче наведено порівняння роботи обох програм за коефіцієнтом IoU.

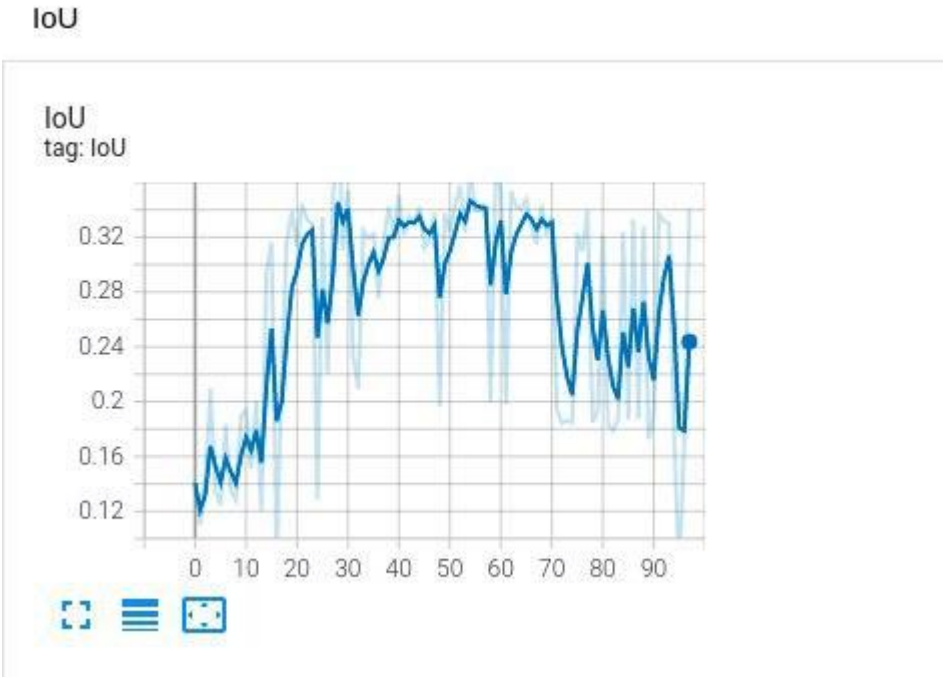

Рис 3.17 Результат метрики IoU Mask R-CNN

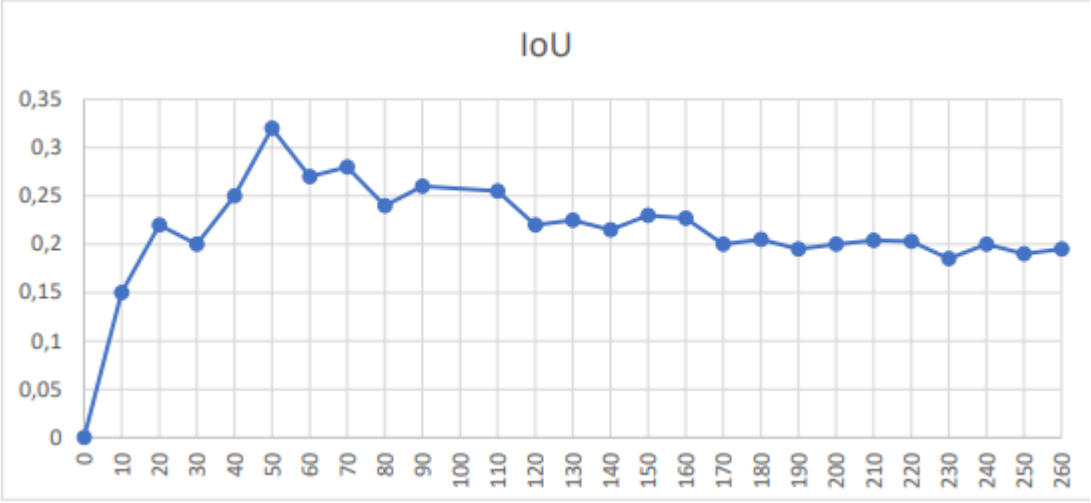

Рис 3.18 Результат метрики IoU Faster R-CNN

За прикладами вище можна зрозуміти, що обидві програми працюють приблизно однаково, але при цьому видно, що пікові значення Mask R-CNN більше ніж Faster R-CNN, а саме 0.34 у Mask R-CNN та 0.32 у Faster R-CNN. В даному випадку ці відмінності незначні, але можливо при тестуванні на кількох тисячах епох, результат Mask R-CNN виявився кращим.

Наступний показник у порівнянні мереж це коефіцієнт Train/Loss:

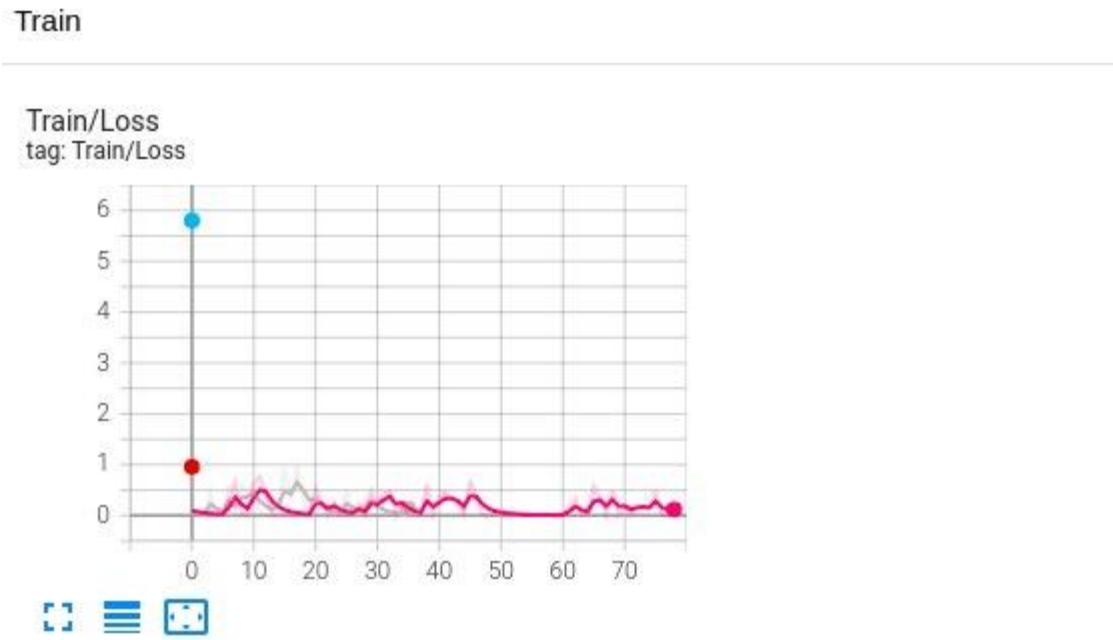

Рис 3.19 Результат метрики Train/Loss Mask R-CNN

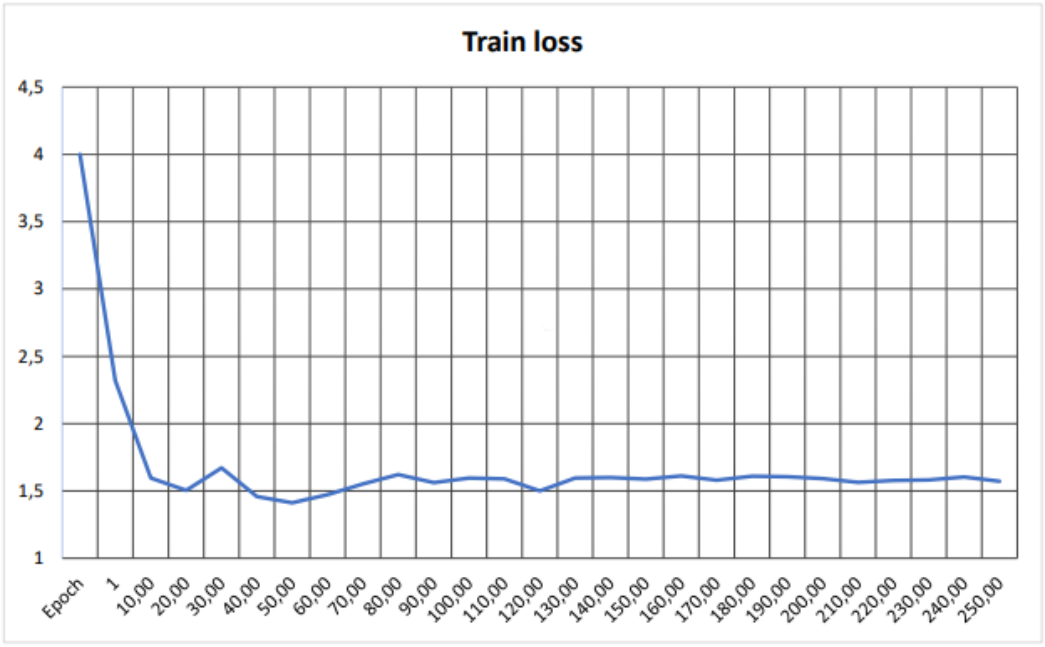

Рис 3.20 Результат метрики Train/Loss Faster R-CNN

ВИСНОВКИ

В ході дипломної роботи була проведена розробка і аналіз технології MASK R-CNN, в ході якої було представлено порівняння декількох популярних платформ для побудови ефективного навчання нейронних мереж.

Як дані для навчання використовувався набір даних Bratz в якому представлені зображення головного мозку, що містять ракові пухлини в 3D.

Технологія MASK R-CNN дозволяє реалізувати ефективне рішення для сегментації даних зображень і надає результат точності більше 80%.

Також було проведено порівняння роботи Mask R-CNN та Faster R-CNN, внаслідок чого можна зробити деякі висновки:

- Mask R-CNN потрібно набагато більше часу для обробки даних, в результаті експерименту було зрозуміло, що Faster R-CNN швидше майже в 3 рази
- Точність мереж майже ідентична, але в пікових показниках Mask R-CNN демонстрував кращі результати.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Artificial neural network [Електронний ресурс] - Режим доступу: https://en.wikipedia.org/wiki/Artificial_neural_network
2. "A gradient method for optimizing multi-stage allocation processes". Proceedings of the Harvard Univ. Symposium on digital computers and their applications. April 1961.
3. ^ Cireşan, Dan Claudiu; Meier, Ueli; Gambardella, Luca Maria; Schmidhuber, Jürgen (21 September 2010). "Deep, Big, Simple Neural Nets for Handwritten Digit Recognition". Neural Computation.
4. ^ Dreyfus, Stuart (1973). "The computational solution of optimal control problems with time lag". IEEE Transactions on Automatic Control.
5. Dreyfus, Stuart E. (1 September 1990). "Artificial neural networks, back propagation, and the Kelley-Bryson gradient procedure". Journal of Guidance, Control, and Dynamics.
6. Dominik Scherer, Andreas C. Müller, and Sven Behnke: "Evaluation of Pooling Operations in Convolutional Architectures for Object Recognition," In 20th International Conference Artificial Neural Networks (ICANN), pp. 92–101
7. Jump up to:a b c Schmidhuber, J. (2015). "Deep Learning in Neural Networks: An Overview". Neural Networks.
8. Jump up to:a b c Schmidhuber, Jürgen (2015). "Deep Learning". Scholarpedia.
9. J. Weng, N. Ahuja and T. S. Huang, "Cresceptron: a self-organizing neural network which grows adaptively," Proc. International Joint Conference on Neural Networks, Baltimore, Maryland, vol I, pp. 576–581, June, 1992.
10. J. Weng, N. Ahuja and T. S. Huang, "Learning recognition and segmentation of 3-D objects from 2-D images," Proc. 4th International Conf. Computer Vision, Berlin, Germany, pp. 121–128, May, 1993.
11. J. Weng, N. Ahuja and T. S. Huang, "Learning recognition and segmentation using the Cresceptron," International Journal of Computer Vision, vol. 25, no. 2, pp. 105–139, Nov. 1997.
12. J. Schmidhuber., "Learning complex, extended sequences using the principle of history compression," Neural Computation, 4, pp. 234–242, 1992.
13. Kelley, Henry J. (1960). "Gradient theory of optimal flight paths". ARS Journal.

14. Kleene, S.C. (1956). "Representation of Events in Nerve Nets and Finite Automata". Annals of Mathematics Studies (34). Princeton University Press. pp. 3–41. Retrieved 17 June 2017.

15. Ivakhnenko, A. G. (1973). Cybernetic Predicting Devices. CCM Information Corporation.

16. Ivakhnenko, A. G.; Grigor′evich Lapa, Valentin (1967). Cybernetics and forecasting techniques. American Elsevier Pub. Co.

17. Hebb, Donald (1949). The Organization of Behavior. New York: Wiley.

18. Farley, B.G.; W.A. Clark (1954). "Simulation of Self-Organizing Systems by Digital Computer". IRE Transactions on Information Theory

19. Minsky, Marvin; Papert, Seymour (1969). Perceptrons: An Introduction to Computational Geometry. MIT Press.

20. Mizutani, E.; Dreyfus, S.E.; Nishio, K. (2000). "On derivation of MLP backpropagation from the Kelley-Bryson optimal-control gradient formula and its application". Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. IJCNN 2000. Neural Computing: New Challenges and Perspectives for the New Millennium.

21. McCulloch, Warren; Walter Pitts (1943). "A Logical Calculus of Ideas Immanent in Nervous Activity". [Електронний ресурс] – Режим доступу:https://link.springer.com/article/10.1007%2FBF02478259

22. Linnainmaa, Seppo (1970). The representation of the cumulative rounding error of an algorithm as a Taylor expansion of the local rounding errors (Masters) (in Finnish). University of Helsinki.

23. Linnainmaa, Seppo (1976). "Taylor expansion of the accumulated rounding error". BIT Numerical Mathematics.

24. Rosenblatt, F. (1958). "The Perceptron: A Probabilistic Model For Information Storage And Organization in the Brain".

25. Werbos, Paul (1982). "Applications of advances in nonlinear sensitivity analysis" (PDF). System modeling and optimization. Springer.

26. Mead, Carver A.; Ismail, Mohammed (8 May 1989). Analog VLSI Implementation of Neural Systems (PDF). The Kluwer International Series in Engineering and Computer Science. 80. Norwell, MA: Kluwer Academic

27. David E. Rumelhart, Geoffrey E. Hinton & Ronald J. Williams , "Learning representations by back-propagating errors ," Nature', 323, pages 533–536 1986.

28. Smolensky, P. (1986). "Information processing in dynamical systems: Foundations of harmony theory.". In D. E. Rumelhart; J. L. McClelland; PDP Research Group (eds.). Parallel Distributed Processing: Explorations in the Microstructure of Cognition.

29. Ng, Andrew; Dean, Jeff (2012). "Building High-level Features Using Large Scale Unsupervised Learning".

30. Ian Goodfellow and Yoshua Bengio and Aaron Courville (2016). Deep Learning. MIT Press.

31. Werbos, P.J. (1975). Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences.

ДОДАТОК А

Приклади сегментації Bratz

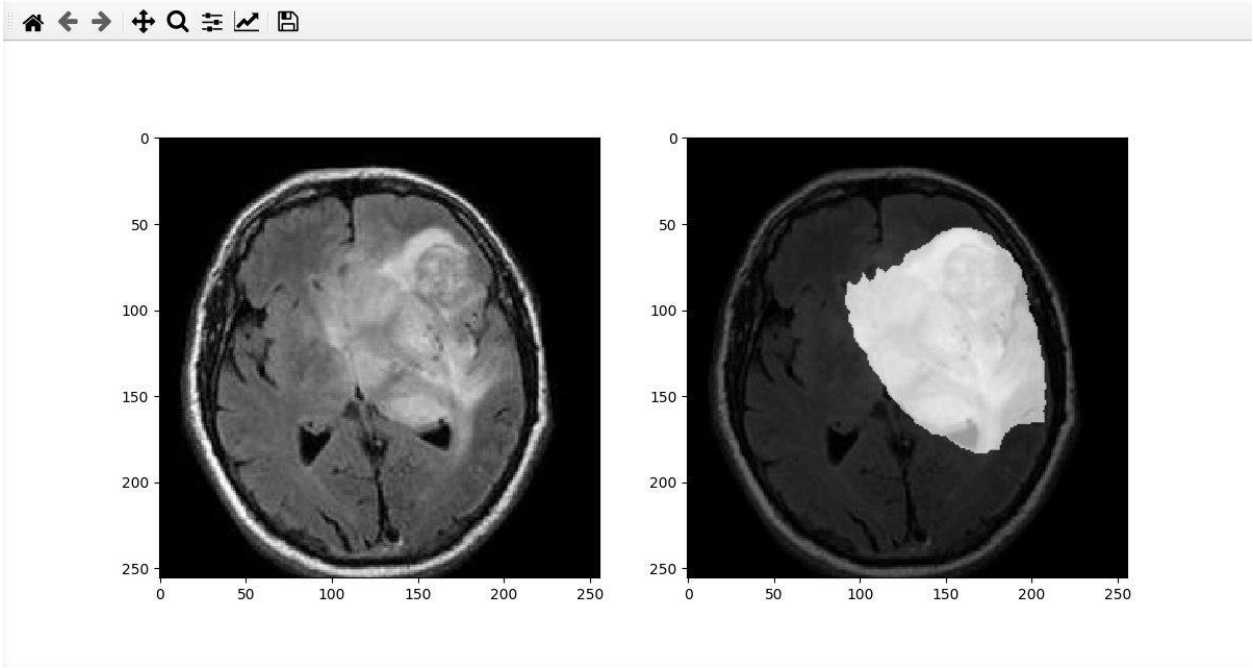

Рис А.1 дані отримані методом фотограмметрії.

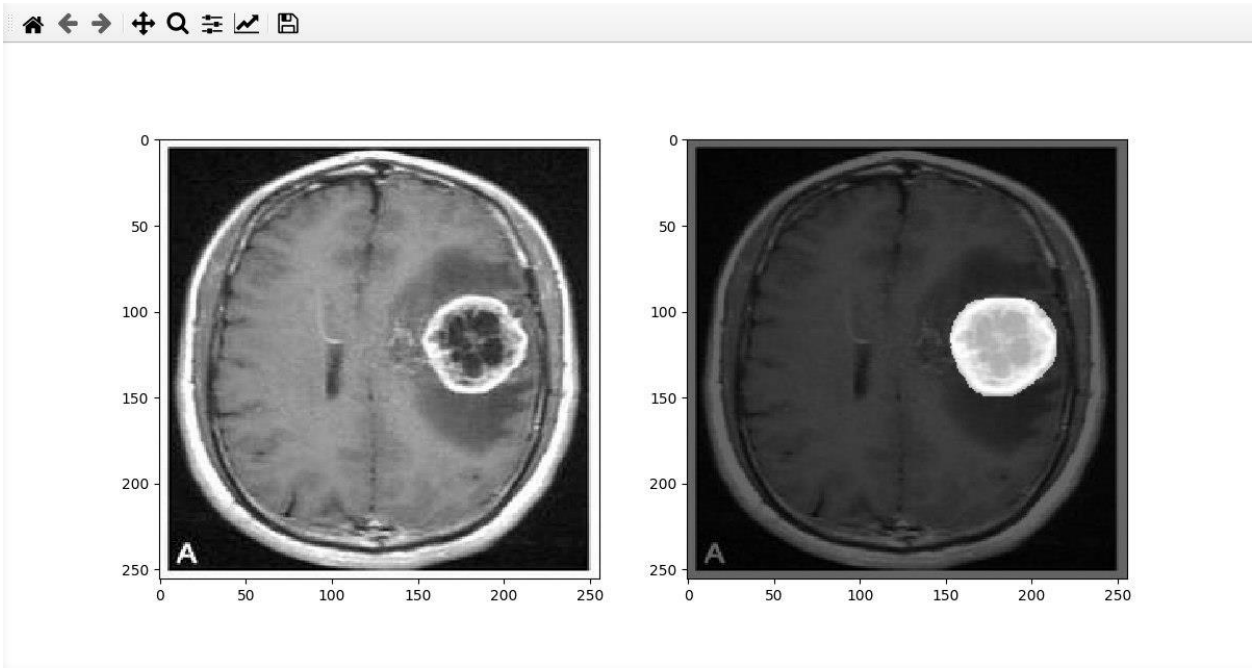

Рис А.2 дані отримані методом фотограмметрії.

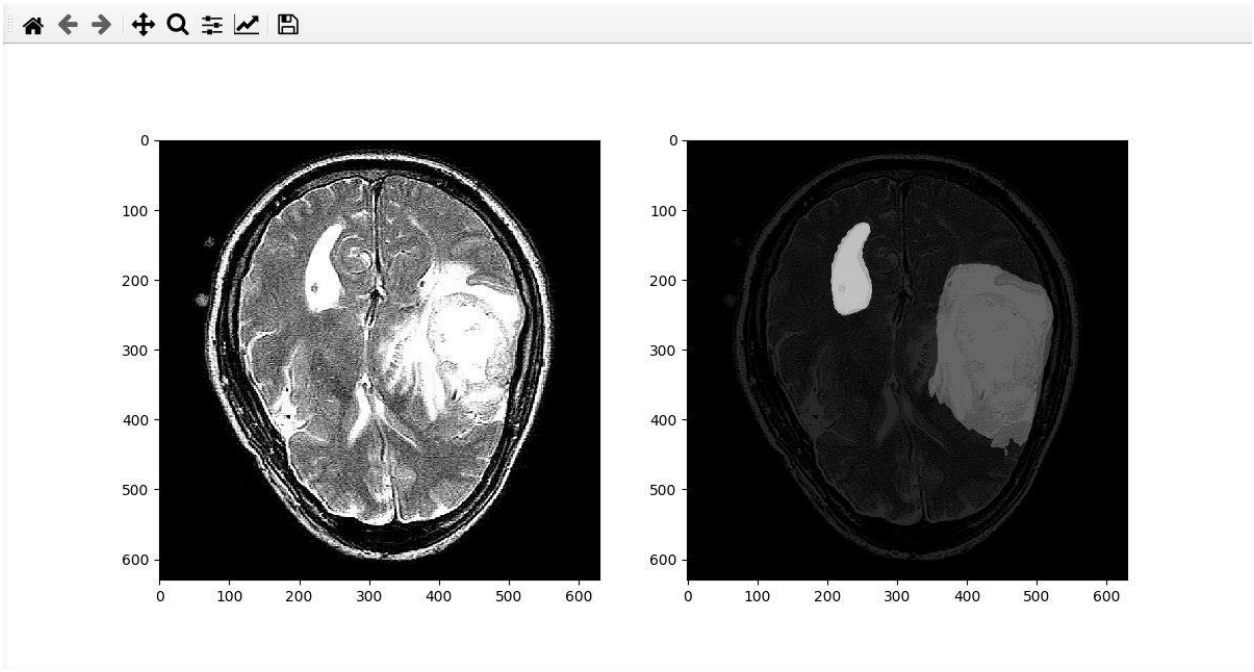

Рис А.3 дані отримані методом фотограмметрії.

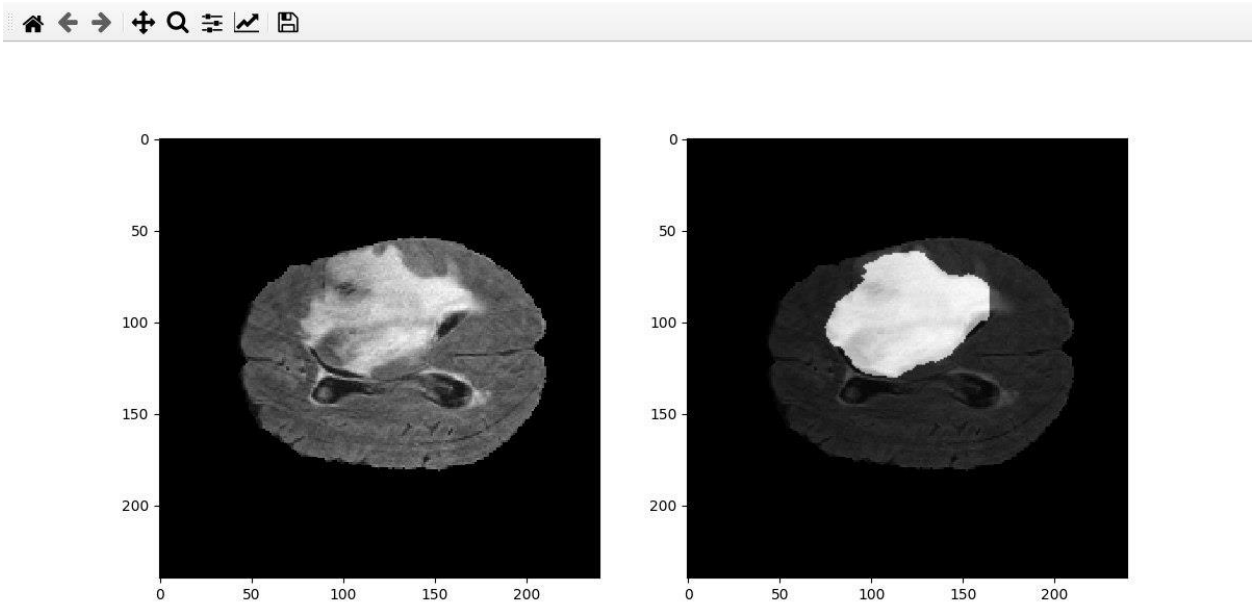

Рис А.4 дані отримані методом фотограмметрії.

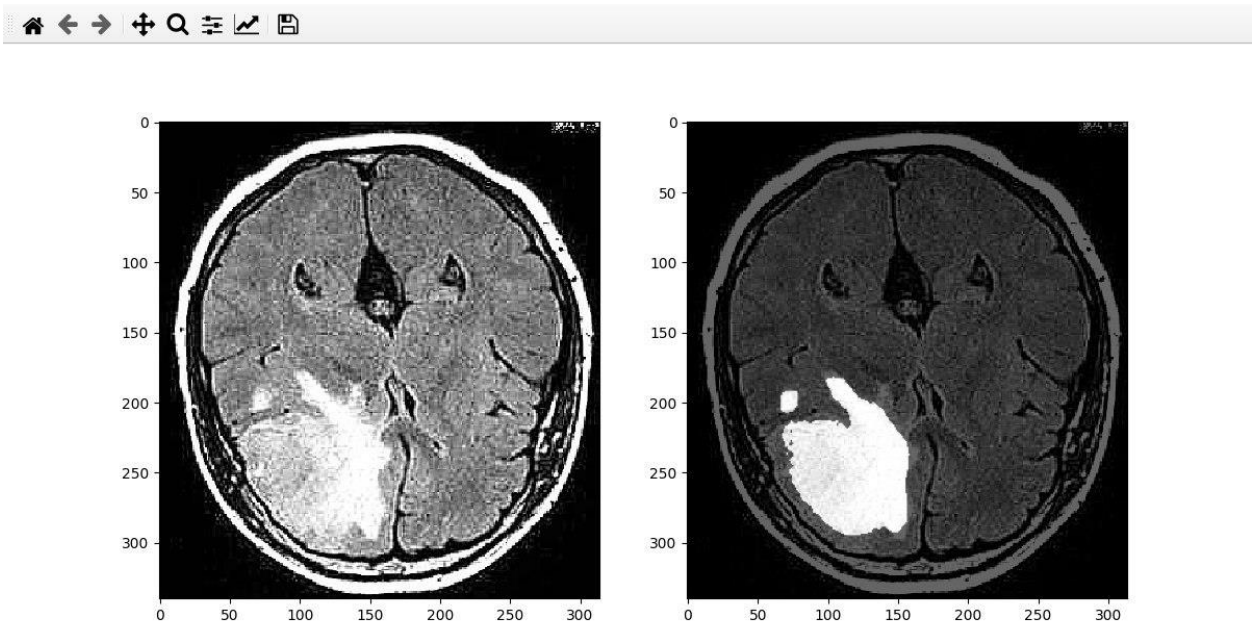

Рис А.5 дані отримані методом фотограмметрії.

ДОДАТОК Б

Приклади датасету Bratz

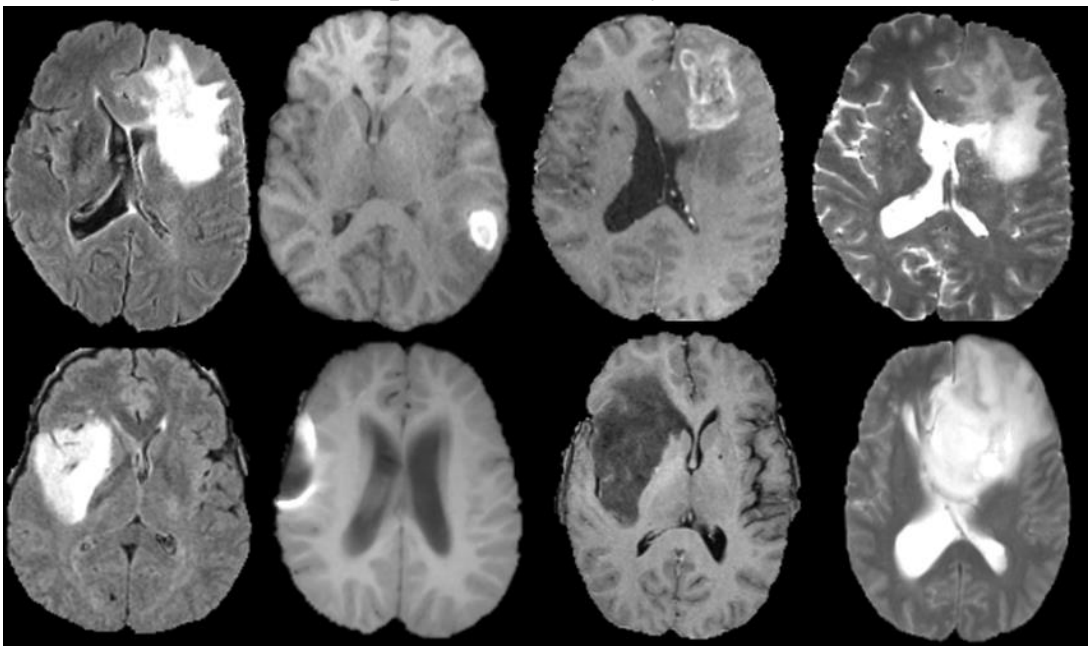

Рис Б.1 Приклад зображення з використаного набору даних

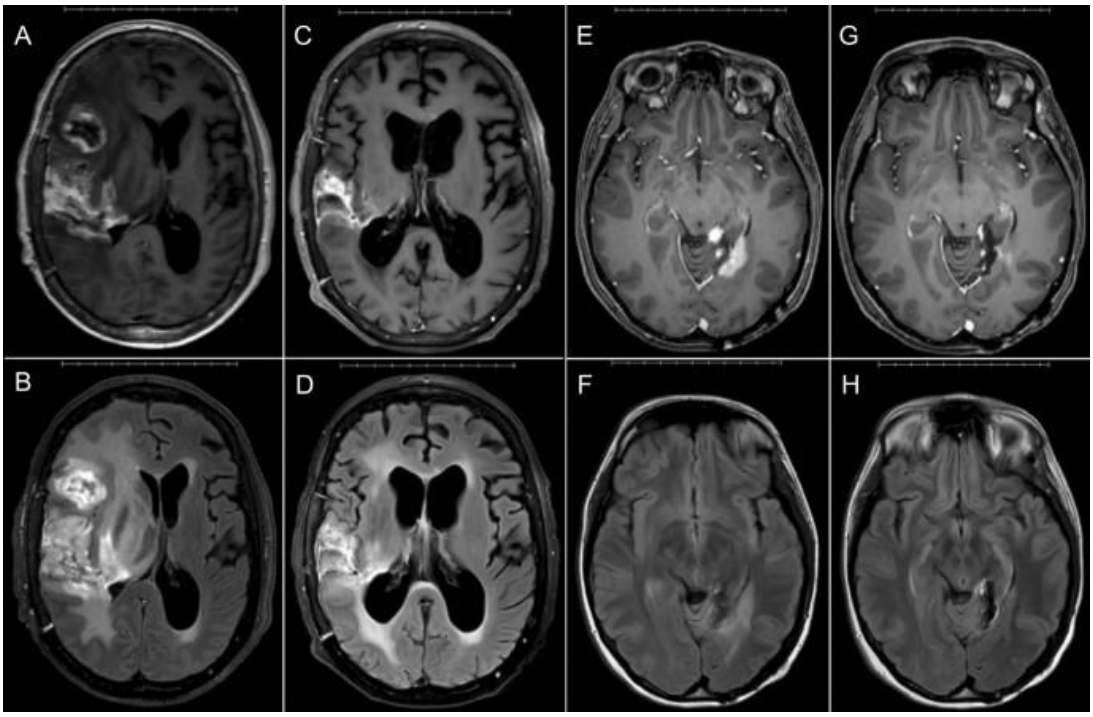

Рис Б.2 Приклад зображення з використаного набору даних

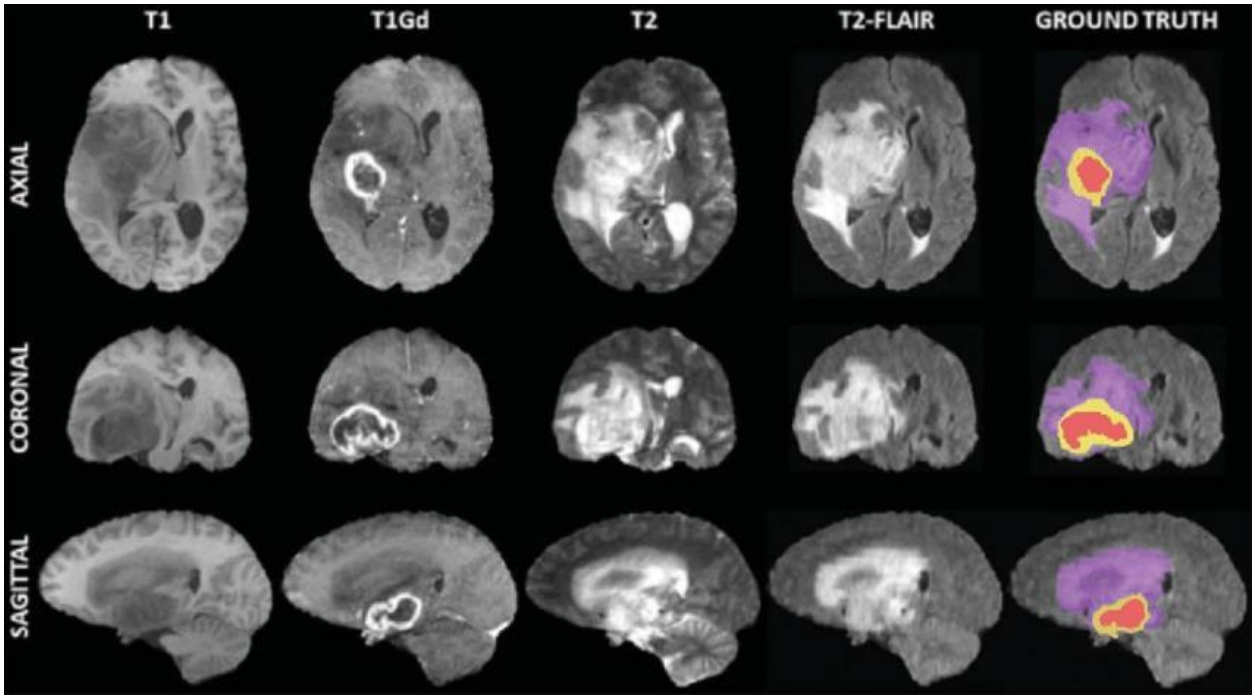

Рис Б.3 Приклад зображення з використаного набору даних

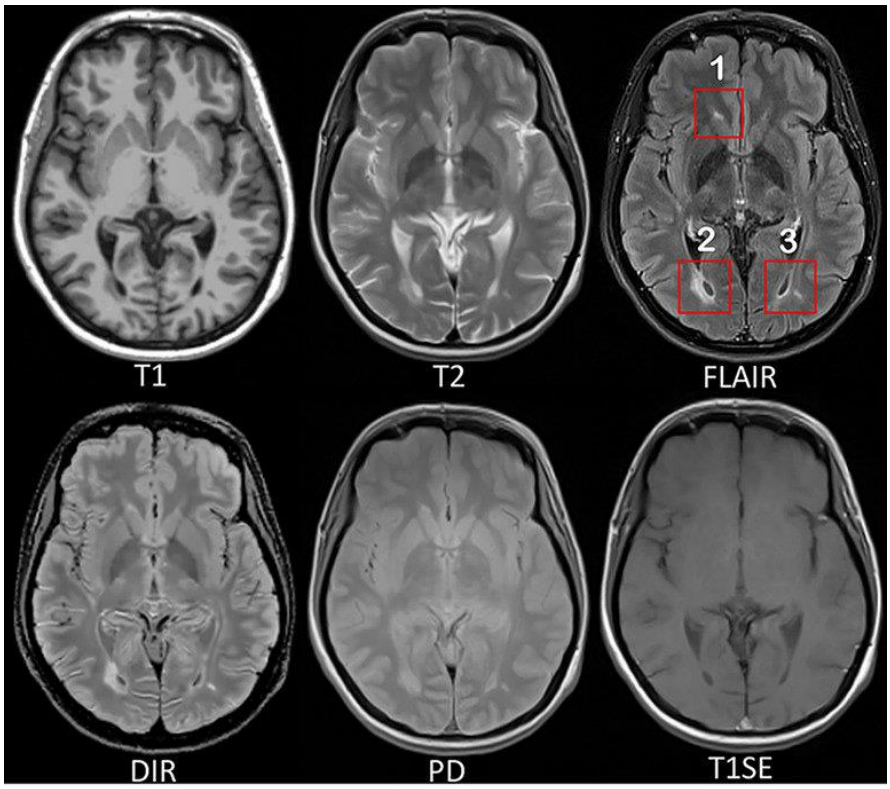

Рис. Б.4 Приклад зображення з використаного набору даних