

ІНСТИТУТ ЕНЕРГЕТИЧНИХ МАШИН І СИСТЕМ ІМ. А. М. ПІДГОРНОГО
НАН УКРАЇНИ

Кваліфікаційна наукова
праця на правах рукопису

КОНЮХОВ ВЛАДИСЛАВ ДМИТРОВИЧ

УДК 004.8

ДИСЕРТАЦІЯ

**АНСАМБЛЕВІ АЛГОРИТМИ МАШИННОГО НАВЧАННЯ ДЛЯ
СЕГМЕНТАЦІЇ ЗОБРАЖЕНЬ НИЗЬКОЇ ЯКОСТІ**

Спеціальність: 113 Прикладна математика

Галузь знань: 11 Математика та статистика

Подається на здобуття наукового ступеня доктора філософії

Дисертація містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

 В.Д. Конюхов

Наукові Керівники:

Немченко Костянтин Едуардович, доктор фізико-математичних наук

Угрімов Сергій Вікторович, доктор технічних наук

Харків – 2025

АНОТАЦІЯ

Конюхов В. Д. Ансамблеві алгоритми машинного навчання для сегментації зображень низької якості. – Кваліфікована наукова праця на правах рукопису. Дисертація на здобуття наукового ступеня доктора філософії за спеціальністю 113 “Прикладна математика”. – Інститут енергетичних машин і систем ім. А.М. Підгорного НАН України, Харків, 2025.

Дисертаційна робота присвячена розробці нових підходів до сегментації медичних зображень з використанням ансамблевих методів.

Оскільки точність сегментації має велике значення в медичних дослідженнях, особливо в таких сферах, як рентгенологія, кардіологія, неврологія, офтальмологія, сегментація медичних зображень є важливим етапом у діагностиці захворювань, таких як остеоартрит, пухлини або інші патології, і потребує високої точності для визначення анатомічних структур, таких як хребці або органи. Дослідження спрямоване на покращення результатів сегментації за допомогою створення нових ансамблевих алгоритмів та поєднання кількох алгоритмів. Зокрема, було застосовано методи сегментації, які поєднують класичні підходи, такі, як пороговий аналіз, з новітніми техніками глибокого навчання.

Методи сегментації, зокрема з використанням глибокого навчання є одними з найбільш поширених і ефективних для сегментації зображень низької якості. Зокрема, методи сегментації на основі нейронних мереж, дозволяють автоматично знаходити об'єкти на зображеннях, навіть попри низьку якість зображення, яке обумовлене шумом, низьким контрастом чи артефактами. За допомогою навчання на різних наборах даних ці методи здатні покращити точність сегментації, тому що вони мають здібність до адаптації різних варіацій якості зображень.

Застосування ансамблевих методів при виконанні задач сегментації зображень стало важливим етапом розвитку цієї галузі. Оскільки медичні зображення часто містять шуми, артефакти та інші проблеми, використання передбачень отриманих від окремої моделі не завжди здатні ефективно виконувати

завдання при використанні різних типів даних. Для вирішення цих проблем було запропоновано використовувати ансамблеві методи, які дозволяють комбінувати кілька моделей для досягнення більш стабільних і точних результатів. Це в свою чергу дало можливість підвищити здатність працювати на зображеннях різної якості, зменшуючи вплив одиничних помилок окремих моделей, що є критично важливим у медичній сфері.

Завдяки можливості використання ансамблевих методів, можна значно підвищити якість сегментації навіть у складних випадках, коли навіть найкращі моделі можуть не давати необхідних результатів. Ключовою перевагою ансамблів є здатність досягати більш високих показників точності. Коли одна модель не може вирішити проблему, інші можуть допомогти, тим самим знижуючи ризик помилок.

Перш ніж починати розробку нових ансамблевих алгоритмів необхідно проаналізувати вже існуючі класичні методи сегментації, що саме і відбувається в першому розділі роботи. Для розуміння, як можна покращити вже існуючі ансамблеві методи, розглядаються та аналізуються особливості основних методів сегментації, включаючи порогові методи, метод К-середніх, метод зростання регіонів, метод вододіл та графові методи. Особливу увагу приділено проблемам, які виникають під час сегментації медичних зображень, таким як обробка зображень з низьким контрастом, високим рівнем шуму або складними анатомічними структурами. Отримані результати вказують на часткову компетентність традиційних методів, оскільки вони не завжди ефективні при роботі з такими складними даними, що ставить завдання для розвитку нових підходів.

Наступним кроком після аналізу традиційних методів є аналіз ансамблевих методів сегментації, який наведено у другому розділі. Завдяки підходу, який в них використовується, а саме – комбінації кількох передбачень, ці методи дозволяють знижувати помилки та забезпечувати більш стабільні та точні результати. В процесі сегментації медичних зображень ансамблеві методи мають значні переваги, оскільки вони дозволяють компенсувати недоліки окремих алгоритмів і досягати високої точності навіть у випадках коли медичне зображення є поганим. Для цього вивчаються різні техніки, які використовуються для об'єднання результатів кількох

моделей, що працюють з одними й тими ж даними, і оцінюються методи їх ефективності, зокрема через метрики точності, такі як DICE. Аналізуються найбільш популярні алгоритми комбінацій передбачень в наш час. Оцінка ефективності ансамблевих алгоритмів є ключовою складовою цього розділу, оскільки вона дозволяє порівняти отримані результати з результатами окремих моделей та визначити найкращий підхід для конкретних типів зображень.

Ключова частина роботи, а саме третій розділ, складається з розробки нових ансамблевих алгоритмів для сегментації зображень низької якості, зокрема рентгенівських знімків, що є найпопулярнішим видом знімків у медичній сфері. Розробка таких алгоритмів дозволяє значно підвищити точність сегментації, зокрема при роботі з анатомічними структурами, як хребці, що має велике значення для діагностики захворювань, пов'язаних із хребтом. Оскільки такі зображення часто містять шум, аномалії, неправильну експозицію, саме використання ансамблевих методів відіграє важливу роль у забезпеченні точних результатів сегментації. В даній роботі пропонується значна кількість нових ансамблевих алгоритмів, як окремих так і комбінованих. Запропоновано ансамблевий алгоритм комбінування передбачень – алгоритм усереднення форм об'єктів, також увага приділяється розробці алгоритмів на основі центрування. Для таких алгоритмів як усереднення з порогом, об'єднання, алгоритм усереднення форм об'єктів, запропоновано спочатку використовувати пошук середнього центра, а потім всі об'єкти зміщувати відносно цього центра. Застосування такого підходу дає можливість зменшити вплив негативних ефектів, наприклад шумів або артефактів. На базі всіх цих алгоритмів було створено більш ефективніші комбіновані алгоритми, які поєднали в собі всі основні переваги окремих ансамблевих алгоритмів і об'єднали їх. Також розглядаються двоетапні та багатоетапні алгоритми, що дозволяють досягти кращих результатів на складних зображеннях. Запропоновані алгоритми можна використовувати, як окремо так і разом, що робить їх більш універсальними. Для більш розгорнутого дослідження ефективності запропонованих алгоритмів було розглянуто їх застосування на різних нейронних мережах.

Для доведення застосування запропонованих алгоритмів на практиці, в четвертому розділі було розроблено програмне забезпечення для автоматизованого процесу отримання діагностичних параметрів хребців, що включає інтеграцію результатів сегментації з медичними інформаційними системами. Оскільки хребці є важливими анатомічними структурами, розроблена система дозволяє автоматично визначати їх розміри та форму на рентгенівських знімках, що є необхідним для правильної діагностики патологій хребта. Завдяки цьому лікарі можуть швидко та ефективно отримувати точні параметри хребців та ставити більш точні діагнози, що є важливим аспектом покращення процесу медичного обстеження.

Таким чином, отримані результати підтверджують доцільність та ефективність запропонованих підходів та дозволяють не тільки підвищити точність і ефективність сегментації медичних зображень, але й зменшити навантаження на медичних працівників, надаючи автоматизовані інструменти для швидкої та точної діагностики.

Ключові слова: машинне навчання; нейронні мережі; глибоке навчання; сегментація зображення; аналіз медичного зображення.

ABSTRACT

Koniukhov V. D. Ensemble machine learning algorithms for segmentation of low-quality images. – Qualifying scientific work on the rights of the manuscript. The dissertation on competition of a scientific degree of the doctor of philosophy on a specialty 113 “Applied Mathematics”. – A. Pidhornyi Institute of Power Machines and Systems of NAS of Ukraine, Kharkiv, 2025.

The dissertation is devoted to the development of new approaches to medical image segmentation using ensemble methods.

Since segmentation accuracy is of great importance in medical research, especially in fields such as radiology, cardiology, neurology, ophthalmology, medical image segmentation is an important step in the diagnosis of diseases such as osteoarthritis, tumors or other pathologies, and requires high accuracy to identify anatomical structures

such as vertebrae or organs. The research aims to improve segmentation results by creating new ensemble methods and combining multiple algorithms. In particular, segmentation methods that combine classical approaches such as threshold analysis with the latest deep learning techniques were applied.

Segmentation methods, particularly those using deep learning, are among the most common and effective for segmenting low-quality images. In particular, segmentation methods based on neural networks allow us to automatically find objects in images, even despite low image quality, which is caused by noise, low contrast, or artifacts. By training on different datasets, these methods can improve segmentation accuracy because they have the ability to adapt to different variations in image quality.

The use of ensemble methods in performing image segmentation tasks has become an important stage in the development of this field. Since medical images often contain noise, artifacts and other problems, the use of predictions obtained from a single model is not always able to effectively perform the task when using different types of data. To solve these problems, it was proposed to use ensemble methods that allow combining several models to achieve more stable and accurate results. This, in turn, made it possible to increase the ability to work on images of different quality, reducing the impact of single errors of individual models, which is critically important in the medical field.

Thanks to the possibility of using ensemble methods, it is possible to significantly improve the quality of segmentation even in complex cases, where even the best models may not give the required results. The key advantage of ensembles is the ability to achieve higher accuracy rates. When one model cannot solve the problem, others can help, thereby reducing the risk of errors.

Before starting the development of new ensemble methods, it is necessary to analyze the existing classical segmentation methods, which is exactly what happens in the first section of the work. To understand how to improve existing ensemble methods, the features of the main segmentation methods are analyzed, including threshold methods, the K-means method, region growing, watershed method and graph methods. Special attention is paid to the problems that arise during the segmentation of medical images, such as processing images with low contrast, high noise or complex anatomical structures.

The results obtained indicate the partial competence of traditional methods, since they are not always effective when working with such complex data, which poses a challenge for the development of new approaches.

The next step after the analysis of traditional methods is the analysis of ensemble segmentation methods. Due to the approach used in them, namely, the combination of several algorithms, these methods allow to reduce errors and provide more stable and accurate results. In the process of medical image segmentation, ensemble methods have significant advantages, as they allow to compensate for the shortcomings of individual algorithms and achieve high accuracy even in cases where the medical image is poor. To this end, various techniques are studied that are used to combine the results of several models working with the same data, and their effectiveness methods are evaluated, in particular through accuracy metrics such as the DICE coefficient. The most popular prediction combination methods are analyzed today. Evaluating the effectiveness of ensemble methods is a key component of this section, as it allows to compare the obtained results with the results of individual models and determine the best approach for specific types of images.

The key part of the work consists of developing new ensemble methods for segmentation of low-quality images, in particular X-ray images, which are the most popular type of images in the medical field. The development of such methods allows to significantly increase the accuracy of segmentation, in particular when working with anatomical structures such as vertebrae, which is of great importance for the diagnosis of diseases associated with the spine. Since such images often contain noise, anomalies, incorrect exposure, the use of ensemble methods plays an important role in ensuring accurate segmentation results. This work proposes a significant number of new ensemble methods, both separate and combined. An ensemble method of combining predictions is proposed - averaging over the shape of objects, attention is also paid to the development of methods based on centering. For such methods as averaging with threshold, union, averaging over the shape of objects, it is proposed to first use the search for the average center, and then shift all objects relative to this center. The use of this approach makes it possible to reduce the impact of negative effects, such as noise or artifacts. On the basis

of all these methods, more effective combined methods were created, which combined all the main advantages of individual ensemble methods and combined them. Two-stage and multi-stage algorithms are also considered, which allow achieving better results on complex images. The proposed methods can be used both separately and together, which makes them more versatile. For a more detailed study of the effectiveness of the proposed methods, their application on various neural networks was considered.

To prove the application of the proposed methods in practice, software was developed for the automated process of obtaining diagnostic parameters of the vertebrae, which includes the integration of segmentation results with medical information systems. Since the vertebrae are important anatomical structures, the developed system allows us to automatically determine their size and shape on X-ray images, which is necessary for the correct diagnosis of spinal pathologies. Thanks to this, doctors can quickly and efficiently obtain accurate parameters of the vertebrae and make more accurate diagnoses, which is an important aspect of improving the medical examination process.

Thus, the results obtained confirm the feasibility and effectiveness of the proposed approaches and allow not only to increase the accuracy and efficiency of medical image segmentation, but also to reduce the burden on medical professionals by providing automated tools for fast and accurate diagnosis.

Keywords: machine learning; neural networks; deep learning; image segmentation; medical image analysis.

СПИСОК ПУБЛІКАЦІЙ ЗДОБУВАЧА ЗА ТЕМОЮ ДИСЕРТАЦІЇ

Статті у наукових фахових виданнях, які включені до наукометричних баз даних (Scopus, Web of Science):

1. Koniukhov, V. D., Morgun, O. M., & Nemchenko, K. E. (2024). Impact of preprocessing and comparison of neural network ensemble methods for segmentation of the thoracic spine in X-ray images. *Radio Electronics, Computer Science, Control*, (4), 102–112. <https://doi.org/10.15588/1607-3274-2024-4-10>

2. Koniukhov, V. D. (2024). Ensemble method based on averaging shapes of objects using the pyramid method. *Radio Electronics, Computer Science, Control*, (4), 113–120. <https://doi.org/10.15588/1607-3274-2024-4-11>

3. Koniukhov, V. (2024). Improving the segmentation of the vertebrae using a multi-stage machine learning algorithm. *Radioelectronic and Computer Systems*, 2024(4), 83-90. <https://doi.org/10.32620/reks.2024.4.07>

Статті у наукових фахових виданнях України:

4. Koniukhov, V. D. (2024). Using an ensemble of neural networks for determining the diagnostic parameters of the vertebrae. *Journal of Mechanical Engineering – Problemy Mashynobuduvannia*, 27(1), 56–61. <https://doi.org/10.15407/pmach2024.01.056>

5. Koniukhov, V. D., & Ugrimov, S. V. (2024). Increasing the accuracy of determining the cardiothoracic ratio with the help of an ensemble of neural networks. *Journal of Mechanical Engineering – Problemy Mashynobuduvannia*, 27(2), 54–60. <https://doi.org/10.15407/pmach2024.02.054>

6. Koniukhov, V. D. (2024). Adaptive refinement of segmented object contour based on the brightness of neighboring pixels using the ensemble method. *Journal of Mechanical Engineering – Problemy Mashynobuduvannia*, 27(4), 73–78. <https://doi.org/10.15407/pmach2024.04.073>

7. Конюхов, В. Д. (2024). Ансамблеві методи на основі центрування для сегментації зображення. *Комп'ютерні системи та мережі*, 6(2), 122-130. <https://doi.org/10.23939/csn2024.02.122>

Список публікацій, які засвідчують апробацію матеріалів дисертації:

8. Конюхов, В. Д. (2024). Використання методів штучного інтелекту для діагностування серцево-судинних захворювань за допомогою кардіоторакального індексу. *Інформаційне суспільство: технологічні, економічні та технічні аспекти становлення: Матеріали Міжнародної наукової інтернет-конференції (Тернопіль, Україна; Опале, Польща, 9–10 липня 2024 р.)* (Вип. 90, с. 21–23).

9. Конюхов, В. Д. (2024). Ансамблеві алгоритми машинного навчання для сегментації зображень низької якості. *Тези доповідей конференції молодих вчених та спеціалістів ІПМаш НАН України «Сучасні проблеми машинобудування–2024»* (с. 6–7). Харків: Інститут енергетичних машин і систем ім. А. М. Підгорного.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ.....	13
ВСТУП.....	14
РОЗДІЛ 1. АНАЛІЗ АЛГОРИТМІВ СЕГМЕНТАЦІЇ МЕДИЧНИХ ЗОБРАЖЕНЬ	19
1.1 Особливості сегментації медичних зображень у порівнянні зі звичайними .	19
1.2 Аналіз існуючих алгоритмів	20
1.2.1 Порогові методи	20
1.2.2 Метод К-середніх	23
1.2.3 Алгоритм зростання регіонів.....	24
1.2.4 Алгоритм вододілу	26
1.2.5 Графові методи	29
1.2.6 Глибоке навчання для сегментації зображень	30
1.2.7 Порівняльний аналіз ефективності алгоритмів сегментації	33
1.2.8 Проблеми та виклики у сегментації медичних зображень	35
Висновки до першого розділу.....	36
РОЗДІЛ 2. АНСАМБЛЕВІ МЕТОДИ ДЛЯ СЕГМЕНТАЦІЇ ЗОБРАЖЕНЬ	38
2.1 Основи ансамблевих методів	38
2.2 Алгоритм усереднення з порогом	40
2.3 Ансамблевий алгоритм об'єднання	41
2.4 Методи оцінки ефективності ансамблевих алгоритмів	42
Висновки до другого розділу	43

РОЗДІЛ 3. РОЗРОБКА НОВИХ АНСАМБЛЕВИХ АЛГОРИТМІВ ДЛЯ СЕГМЕНТАЦІЇ ЗОБРАЖЕНЬ НИЗЬКОЇ ЯКОСТІ	45
3.1 Використані нейронні мережі та набори даних	45
3.2 Алгоритм усереднення форм об'єктів	52
3.3 Алгоритми на основі центрування	61
3.4 Комбіновані ансамблеві алгоритми	66
3.5 Оцінка точності та ефективності ансамблевих алгоритмів на різних наборах даних	70
3.6 Алгоритм адаптивного уточнення контуру на основі яскравості сусідніх пікселів	80
3.7 Двоетапний ансамблевий алгоритм сегментації зображень	90
3.8 Багатоетапний ансамблевий алгоритм сегментації зображень	98
Висновки до третього розділу	118
РОЗДІЛ 4. РОЗРОБКА КОМП'ЮТЕРНОЇ АВТОМАТИЗОВАНОЇ СИСТЕМИ ДЛЯ ОТРИМАННЯ ДІАГНОСТИЧНИХ ПАРАМЕТРІВ ХРЕБЦІВ	120
4.1 Постановка задачі розробки програмного забезпечення для визначення діагностичних параметрів хребців	120
4.2 Підготовка навчальних даних та тренування моделей	122
4.3 Реалізація програмного забезпечення автоматизованого процесу отримання діагностичних параметрів хребців грудного відділу	123
Висновки до четвертого розділу	127
ВИСНОВКИ	129
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	131
ДОДАТКИ	146

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

ААО — Ансамблевий алгоритм об'єднання

АУП— Алгоритм усереднення з порогом

АУФО — Алгоритм усереднення форм об'єктів

КТ — Комп'ютерна томографія

МРТ — Магнітно-резонансна томографія

ПЗ — Програмне забезпечення

УЗД — Ультразвукова діагностика

AVH — Anterior Vertebral Height

CBR — Case-Based Reasoning

DBSCAN — Density-based spatial clustering of applications with noise

DICE — Dice Similarity Coefficient

FAIR — Findable, Accessible, Interoperable, Reusable

FFNN — Feed-Forward Neural Network

GRNN — General Regression Neural Network

IoU — Intersection over Union

LSTM — Long short-term memory

MVH — Middle Vertebral Height

PVH — Posterior Vertebral Height

ROI — Region of Interest

SVM — Support Vector Machine

VDS — Virtual Distributed System

ВСТУП

Актуальність теми.

Сегментація зображень низької якості є вкрай актуальним питанням в наш час. Використання ансамблів нейронних мереж в задачах сегментації є важливим компонентом розвитку сучасних методів комп'ютерного зору. Такі найпопулярніші сфери використання методів сегментації, як медична та промислова мають значну кількість зображень низької якості через технічні проблеми пов'язані з апаратурою, яка використовується або зі спеціалістами, які цю апаратуру використовують. Саме через це й виникає необхідність у вдосконаленні вже існуючих та розробці нових алгоритмів, які зможуть більш ефективно виконувати свої завдання покращуючи при цьому точність сегментації.

Використання ансамблевого підходу в задачах сегментації зображень дало змогу продемонструвати покращення точності сегментації завдяки комбінуванню передбачень декількох моделей. Проведені донедавна дослідження довели, що методи які використовують ансамблевий підхід, мають змогу зменшити похибки окремих моделей і отримати більш надійні результати сегментації [1]. Застосування ансамблевих методів є необхідною частиною процесу сегментації зображень низької якості, оскільки ці методи здатні нівелювати втрату важливої інформації при використанні окремих моделей. В дослідженнях, які проводились для вивчення використання ансамблю нейронних мереж, було доказано, що застосування таких методів покращує точність сегментації зображень використовуючи різні набори даних. Отримання таких результатів стало можливим завдяки комбінуванню переваг різних моделей.

Використання методів глибокого навчання в медичній візуалізації сприяє розробці більш точних методів обробки зображень [2]. Але в свою чергу, для досягнення найкращих результатів необхідно враховувати якість наборів даних, які використовуються. В таких сферах, як медична, де насамперед точність відіграє ключову роль і якою не можна знехтувати, саме використання ансамблевих методів при їх сумісному використанні з зображеннями низької якості дозволить

забезпечити більш стабільні результати [3]. Також використання ансамблевих методів дає змогу мінімізувати вплив шумів та дефектів на вхідних зображеннях.

Мета і задачі дослідження.

Мета даного дослідження полягає у підвищенні ефективності сегментації зображень низької якості шляхом модифікації існуючих та розробки нових ансамблевих алгоритмів. Головна увага привертається до аналізу поведінки різних алгоритмів, як окремо так і в комбінованому вигляді, для того щоб вивчити їх вплив на покращення якості сегментації у порівнянні з використанням окремих моделей. Виходячи з цього, було поставлено наступні задачі для дослідження:

1. Проведення аналізу класичних методів сегментації зображень для визначення основних проблем, які можуть виникнути в процесі.
2. Розробити нові ансамблеві алгоритми для сегментації зображень низької якості.
3. Провести порівнянні ефективності сегментації окремих моделей та ансамблів з використанням запропонованих алгоритмів, визначивши ефективність кожного алгоритму.
4. Проаналізувати вплив якості зображень на точність сегментації з використанням запропонованих ансамблевих алгоритмів.

Об'єктом дослідження є процес сегментації зображень низької якості.

Предметом дослідження є методи та алгоритми побудови ансамблю нейронних мереж для сегментації зображень низької якості.

Методи дослідження.

При виконанні цієї роботи було використано наступні методи дослідження:

1. Аналіз літературних джерел для огляду існуючих методів сегментації зображень та ансамблевих методів.
2. Комп'ютерне моделювання для навчання моделей та отримання оцінки продуктивності різних видів ансамблів.
3. Експериментальні методи, які були використані для обробки зображень різної якості за допомогою ансамблів нейронних мереж і порівняння їх результатів з окремо використаними моделями.

4. Статистичний аналіз результатів, який виконував порівняння ефективності запропонованих алгоритмів сегментації за допомогою таких метрик як DICE та IoU.

Наукова новизна одержаних результатів полягає в наступному:

1. Вперше запропоновано алгоритм адаптивного уточнення контуру на основі яскравості сусідніх пікселів з використанням ансамблевих методів.
2. Вперше запропоновано ансамблеві алгоритми на основі центрування.
3. Вперше запропоновано ансамблевий алгоритм усереднення форм об'єктів.
4. Вперше запропоновано комбінації ансамблевих алгоритмів із використанням алгоритму усереднення форм об'єктів, а саме — комбінування алгоритму усереднення форм об'єктів та ансамблевого алгоритму об'єднання.
5. Вперше запропоновано використовувати різні ансамблеві алгоритми у двоетапному та багатоетапному алгоритмах.

Практичне значення одержаних результатів.

Запропоновані в дисертації алгоритми можуть знайти своє місце в практичній площині. Створені ансамблеві алгоритми для сегментації зображень низької якості можуть бути застосовані у медичній сфері або в будь-якій іншій сфері, яка потребує покращення якості сегментації. Ансамблеві алгоритми можуть бути використані в якості частини автоматизації процесів діагностики. Використання запропонованих алгоритмів для сегментації рентгенівських зображень дає змогу підвищити точність сегментації, особливо в тих випадках, коли зображення мають явні недоліки, такі, як шуми та артефакти. Таким чином це дозволить покращити якість існуючих автоматизованих рішень.

Розроблено додаток на мові Python для визначення діагностичних параметрів хребців грудної клітини за допомогою ансамблевих алгоритмів, а саме наступних хребців: Th8, Th9, Th10, Th11. Сегментація зображень була виконана за допомогою запропонованих ансамблевих алгоритмів.

Практична реалізація одержаних результатів полягає у передачі результатів досліджень ТОВ «РАДІОПРОМ» (м. Харків) для впровадження запропонованих методик, алгоритмів та бібліотек зі втіленими авторськими алгоритмами, що підтверджено відповідним актом. Результати роботи були впроваджені у

навчальний процес кафедри інформаційних технологій в фізико-енергетичних системах навчально-наукового інституту комп'ютерної фізики та енергетики Харківського національного університету імені В. Н. Каразіна при проведенні лекційних, лабораторних та практичних занять з курсів «Вступ до обробки даних» і «Обробка зображень та сигналів» для студентів 3 курсу бакалаврату та «Наближені методи розв'язання задач математичної фізики» і «Обробка даних фізичних експериментів» для студентів 1 курсу магістратури спеціальності 105 - «Прикладна фізика та наноматеріали», що підтверджено відповідним актом.

Особистий внесок здобувача.

Усі перелічені вище наукові результати роботи автор отримав самостійно. У співавторстві опубліковано праці [106] у якій дисертантові належить ідея комбінування та використання прогнозів ансамблів нейронних мереж для сегментації грудного відділу хребта, а також детальне вивчення застосування попередньої обробки рентгенівських зображень, [77] у якій дисертантові належить ідея застосування методів ансамблювання нейронних мереж для покращення визначення кардіо-торакального індексу.

Апробація результатів дослідження.

Основні результати дисертаційного дослідження були опубліковані на наступних наукових конференціях:

1. Дев'ятій науково-технічній міжнародній конференції “Комп’ютерне моделювання в наукоємних технологіях (КМНТ-2023)”, м. Харків, 2023.
2. Міжнародній науковій інтернет-конференції “Інформаційне суспільство: технологічні, економічні та технічні аспекти ставлення”, м. Тернопіль-Ополе, 2024.
3. Конференції молодих вчених та спеціалістів “Сучасні проблеми машинобудування – 2024”, м. Харків, 2024.
4. Десятій науково-технічній міжнародній конференції “Комп’ютерне моделювання в наукоємних технологіях (КМНТ-2024)”, м. Харків, 2024.

Публікації. Основні результати дисертації опубліковані у 9 наукових працях, зокрема 1 стаття у виданні, яке індексується у наукометричній базі Scopus; 2 статті у виданні, яке індексується у наукометричній базі Web of Science; 4 статті

опубліковані у виданнях, що належать до категорії Б; а також дві тези доповідей, представлені на науково-технічних конференціях.

Структура та обсяг роботи.

Дисертаційна робота складається із переліку умовних позначень та скорочень, вступу, чотирьох розділів, висновків, списку використаних джерел та додатків. Загальний обсяг роботи складає 166 сторінок, з яких 109 сторінок основного тексту, що містять 52 рисунки та 15 таблиць. Список використаних джерел налічує 128 найменувань. Обсяг додатків складає 20 сторінок.

РОЗДІЛ 1

АНАЛІЗ АЛГОРИТМІВ СЕГМЕНТАЦІЇ МЕДИЧНИХ ЗОБРАЖЕНЬ

1.1. Особливості сегментації медичних зображень у порівнянні зі звичайними

Аналізуючи основні відмінності між звичайними зображеннями та медичними, можна виділити наступні ключові пункти, які більш за все впливають на різність цих зображень:

1. Різноманітність типів: у медицині використовуються специфічні типи зображень, зокрема рентгенівські зображення, зображення зроблені за допомогою магнітно-резонансної томографії, зображення зроблені за допомогою комп'ютерної томографії, зображення зроблені за допомогою УЗД, тощо.

2. Велика роздільна здатність зображення: зазвичай медичні знімки мають вищу роздільну здатність в порівнянні зі звичайними зображеннями, що робить їх більш вимогливими для використання високоточних методів сегментації.

3. Висока точність та ефективність: один із головних факторів сегментації медичних зображень. Для лікування або діагностування захворювання в медицині приділяється велика увага до коректної ідентифікації патологій.

4. Шум та артефакти: одна з основних проблем в сегментації медичних зображень, яка значно ускладнює аналіз та діагностику.

5. Кількість даних: для досягнення високої точності сегментації зображень при використанні нейронних мереж ми потребуємо великої кількості даних. Найкращий результат досягається коли ми можемо отримати велику кількість різноманітних зображень для тренування моделей. Для виконання цієї задачі не в медичній практиці можна використовувати людей, які не дотичні до цієї сфери [4] але з медичними зображеннями інша справа, тільки кваліфіковані спеціалісти можуть виконати цю роль.

Роблячи висновок з вказаного вище, можна зрозуміти, що відмінність в сегментації між звичайними зображеннями і медичними грає велику роль. Зокрема

різноманітність типів, велика роздільна здатність, точність, кількість даних, шум та артефакти роблять сегментацію медичних зображень складною задачею. Саме з цієї причини й виникла необхідність створення спеціалізованих алгоритмів сегментації, які мають враховувати всі ключові аспекти та особливості медичних знімків.

1.2. Аналіз існуючих алгоритмів

1.2.1. Порогові методи

Порогова сегментація є одним із найпростіших методів сегментації. Вона використовується для розділу зображення на об'єкти, на основі значень інтенсивності пікселів [5]. В даному методі використовується вказане порогове значення і пікселі, які мають значення вище порогу, вважаються частиною однієї області, а решта – іншої.

Зазначений метод є тривіальним та має широкий спектр його використання. У пороговій сегментації є переваги та недоліки, основною перевагою є його простота та ефективність для деяких завдань, але в складних ситуаціях це може не мати того самого ж ефекту, коли наприклад зображення має різний рівень освітлення, шум, кількість відтінків тощо [5,6].

Методи порогової сегментації розділяються на декілька видів, котрі відрізняються способом визначення порогового значення та областями його застосування:

1. Глобальна порогова сегментація: в цьому методі використовується одне порогове значення для всіх елементів матриці пікселів зображення. Ті елементи, які мають значення вище за поріг є об'єктами, а інші фоном:

$$Output(x, y) = \begin{cases} Object, & \text{if } Input(x, y) > T \\ Background, & \text{otherwise} \end{cases}, \quad (1.1)$$

де T – поріг, $Input(x, y)$ – інтенсивність пікселя в заданих координатах, $Output(x, y)$ – результат сегментації.

2. Локальна порогова сегментація: в цьому методі існує декілька порогових значень котрі використовуються для деяких регіонів зображення одночасно. Такий спосіб допомагає врахувати деякі особливості зображення. Для отримання

локальної сегментації використовується відповідна формула:

$$Output(x, y) = \begin{cases} Object, & \text{if } Input(x, y) > T(x, y) \\ Background, & \text{otherwise} \end{cases},$$

де $T(x, y)$ – локально визначений поріг для пікселя.

3. Автоматичне визначення порогового значення: в даному випадку використовуються алгоритми для визначення більш оптимального порогового значення, наприклад метод Отсу. Цей метод визначає оптимальний поріг по дискримінантному критерію, а саме так, щоб максимізувати роздільність результуючих класів за рівнями сірого [7].

4. Ручний вибір порогового значення: цей метод дає змогу самостійно визначити порогове значення, потрібно врахувати особливості зображення та провести його аналіз. Використовується формула (1.1), де T – поріг вибраний вручну.

Для того щоб оцінити ефективність вказаних вище алгоритмів, проведемо порівняльний аналіз з завданням сегментації легень на рентгенівському зображенні. Використаємо всі 4 алгоритми, а результати занесемо в таблицю. Для визначення ефективності використаємо Коефіцієнт Жаккара, який використовується як метрика для визначення точності та має наступний вигляд:

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|}.$$

Для визначення ефективності було взято зображення із набору [8], розміром 256×256 пікселів в градації сірого. Для ручного методу було визначено поріг – 150. Після використання методів порогової сегментації було застосовано інверсію, тому що нас цікавлять саме легені.

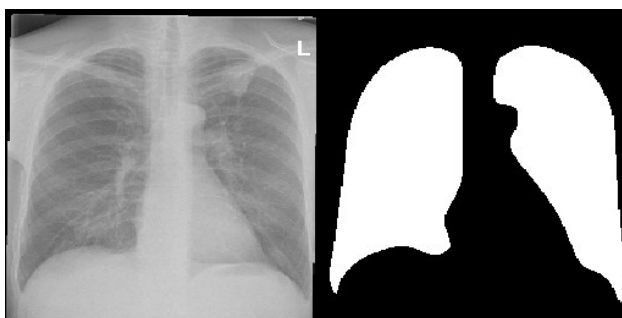


Рис. 1.1. Зображення легень та його маска

Базове зображення та його маска, згідно якої за допомогою коефіцієнта Жаккара було визначено точність, відображені на рисунку 1.1.

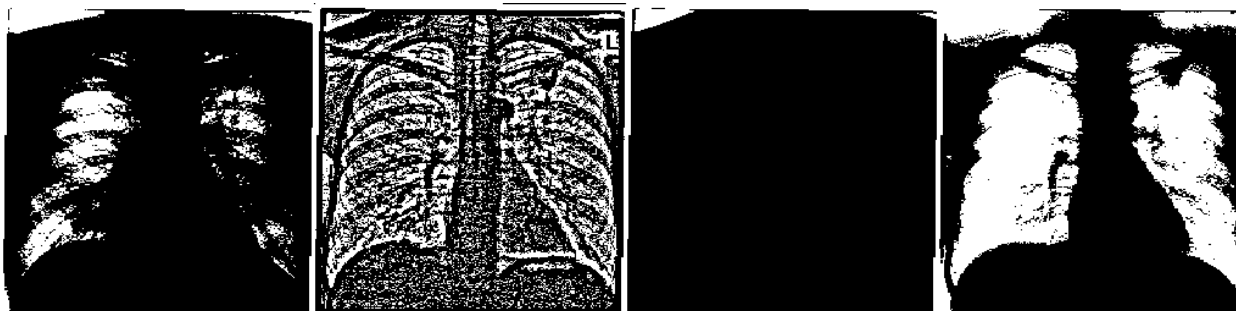


Рис. 1.2. Порогові методи сегментації для визначення легень. Зліва направо – глобальна, локальна, метод Отсу, ручна

Таблиця 1.1

Точність методів порогової сегментації, IoU %

Назва алгоритму	IoU %
Глобальна порогова сегментація	16.77
Локальна порогова сегментація	32.81
Автоматична сегментація (метод Отсу)	0.0
Ручна сегментація	61.06

Згідно з результатів отриманих з таблиці 1.1, при використанні порогових методів можна зробити висновок, що дані методи є простими та швидкими для обробки зображень, але вони мають свої обмеження, особливо коли мова йде про медичні знімки. Знімки грудної клітини можуть мати області з різним рівнем контрасту із-за різної щільності тканини. Це ускладнює використання глобального порогу, так як один і той же поріг може не підходити для різних областей. На рисунку 1.2 відображено складність застосування порогових методів для сегментації легень. Також однією із проблем є шуми. Медичні зображення дуже часто мають проблеми такого роду, вони можуть бути спричинені різними факторами, наприклад низькою розподільною здатністю або рухом пацієнта. В даному випадку порогова сегментація може посилити ці проблеми, і визначити

шуми, як частину інформації яка нас цікавить. І на останок грудна клітина містить серце, суди та легені, в свою чергу такий набір може спричинити перекриття інтенсивностей. Метод може невірно визначити ці тканини, як однорідні, що призведе до неточності в визначенні легень.

1.2.2. Метод К-середніх

Одним із популярних методів сегментації є метод К-середніх. Він використовується для сегментації зображень у багатьох сферах, в тому числі і в медичній. Даний метод поділяє зображення на K кластерів, де K – це параметр, що визначає кількість кластерів.

В деяких випадках, коли доречно використовувати попередню обробку або постобробку зображень, можливо застосувати кластеризацію. Кластеризація дозволяє об'єднувати подібні пікселі або сегменти, завдяки чому покращується якість деяких регіонів зображень та знижується рівень шуму. В якості методів кластеризації для попередньої обробки та постобробки можна використовувати методи K -середніх, Mean Shift або DBSCAN. Особливість застосування Mean Shift полягає в тому, що відсутня необхідність попереднього встановлення кількості кластерів.

У контексті медичного використання, цей метод може бути використаний для сегментації різних тканин або клітин. Він може сегментувати патологічні регіони, регіони з певними змінами або використовуватися для подальшого аналізу. Попри його простоту в використанні і реалізації, і відносній швидкості, він має свої недоліки, а саме неправильне визначення K може спровокувати погані результати сегментації зображення [9]. Від початкового розташування центроїдів може значно відрізнятись кінцевий результат, що робить його чутливим до початкових значень. Метод має наступний вигляд:

$$J(V) = \sum_{i=1}^c \sum_{j=1}^{c_i} (||x_i - v_j||)^2,$$

де $\|x_i - v_j\|$ - Евклідова відстань між x_i та v_j , c_i – кількість точок в даному i -тому кластері, c – кількість кластерних центрів.

При порівнянні результату отриманого за допомогою використання методу K -середніх для сегментації легень застосовуючи рисунок 1.1 та еталонної маски зображеної на рисунку 1.1, значення коефіцієнта Жаккара склало 41.34 %. На рисунку 1.3 можна побачити результат використання методу K -середніх для сегментації легень.

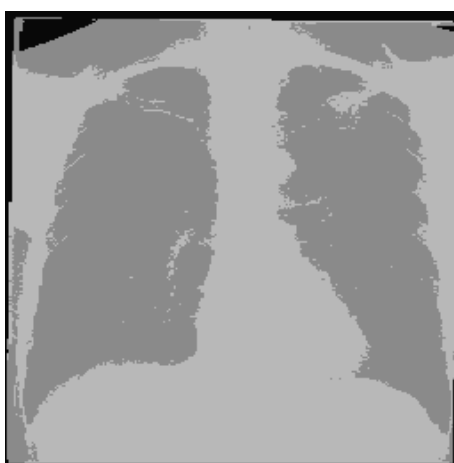


Рис. 1.3. Використання методу K -середніх для сегментації легень

1.2.3. Алгоритм зростання регіонів

Алгоритм зростання регіонів займає не останнє місце в списку методів сегментації зображень. Він є популярним і має широкий спектр використання. Метод базується на принципі групування сусідніх пікселів або груп пікселів в однакові регіони, які повинні мати схожі властивості [10]. Можливість автоматичного сегментування зображення є однією із основних його переваг. Також перевагою цього методу є його інтуїтивність і простота розуміння. Він ефективно проявляє себе у випадках, коли об'єкт має чіткі межі, а внутрішні риси однорідні. Він може добре сегментувати області зі змінною яскравістю та іншими властивостями. Але він також має свої недоліки, наприклад, як і методи порогової сегментації він може бути чутливим до шуму та інших артефактів, що в свою чергу призводить до зниження якості сегментації зображення. Також при складних

критеріях зростання регіонів або великому розмірі зображення обчислювальна складність може бути вимогливою.



Рис. 1.4. Блок-схема алгоритму зростання регіонів

Алгоритм зростання регіонів працює поступово розширюючи області у зображенні. Він починається з початкових точок, а згодом у регіон додаються нові пікселі, які відповідають певному критерію подібності. Відбувається цей процес у кілька кроків. На першому кроці, визначаються початкові точки або пікселі, з яких регіон виконуватиме свій ріст. Ці регіони можуть знаходитися вручну або визначатися автоматично за допомогою критерію, такого як яскравість, колір, або

текстура. Далі аналізуються сусідні пікселі навколо кожної точки і порівнюються із характеристиками вже включених у регіон пікселів. Приєднання пікселю до регіону відбувається в тому випадку, якщо елемент зображення відповідає критерію подібності, має схожу інтенсивність або знаходиться в межах допустимого порогу відхилення.

Можна застосовувати різні критерії подібності, які будуть впливати на кінцевий результат сегментації. Для отримання якісної сегментації необхідно правильно налаштувати параметри алгоритму.

Візуально алгоритм зростання регіонів можна відобразити за допомогою блок-схеми зображеної на рисунку 1.4.

Виконуючи порівняння точності з еталонною маскою наведеною на рисунку 1.1, значення коефіцієнту Жаккара склало 38.24 %. На рисунку 1.5 візуально продемонстровано результат алгоритму зростання.



Рис. 1.5. Результат алгоритму зростання регіонів

1.2.4. Алгоритм вододілу

Цей алгоритм використовується для розділення складної задачі на декілька менш складних, які є більш легкими частинами для розуміння та розв'язання. Він може бути доволі корисним в медичному контексті, якщо застосовувати його для інтерпретації багатьох кроків та систематичного і структурного підходу. За його

допомогою можна розбити медичне зображення на кілька регіонів, що мають схожі характеристики. Такий підхід дозволить виділити різні структури на зображеннях, наприклад, органи, тканини, патологічні утворення. В свою чергу алгоритм має обмеження, які часто породжують неправильні краї через нерівномірний розподіл сірого рівня в зображеннях [11].

Традиційний алгоритм має недоліки у вигляді пересегментації та впливу відбитого світла на зображенні. Використання морфологічного градієнту дозволяє отримати компонентний градієнт кольорового зображення в новому кольоровому просторі, який не схильний до вказаного раніше впливу [12].

Алгоритм базується на аналізі структури зображення та яскравості пікселів. Його ідея полягає в розумінні зображення як рельєфу, де кожен елемент зображення розглядається, як вершина, а границі між областями – як вододільні лінії. На рисунку 1.6 відображено результат алгоритму. В основі роботи цього алгоритму можна виділити наступні основні етапи:

1. Створення градієнту. В першу чергу потрібно розрахувати градієнт зображення, який повинен відображати зміну інтенсивності пікселів.

2. Використання міток. Зображення отримане на першому етапі подається для подальшого використання на вхід алгоритму, де кожен елемент зображення отримає початкову мітку.

3. Перетворення водорозділу. На цьому етапі використовуються маркери, які вказують області, що вже виділені. Процес вододілу починається з цих маркерів і вода (яка імітується за допомогою пікселів) відтікає поступово по зображенню, після чого вона об'єднує сусідні пікселі в басейни (регіони), які відповідають локальним максимумам градієнту.

4. Створення вододілу. Після того, як вода пройшла по всьому зображенню настає момент в якому формуються вододільні лінії, котрі будуть розділяти різні регіони на зображенні. Вони представляють границі між сегментами.

5. Оновлення міток. Після отримання границь сегментів, потрібно отримати однакові мітки. Це потрібно для ідентифікації окремих сегментів на зображенні.

6. Кінцева обробка. На останньому етапі можна провести додаткову обробку для корегування вже отриманих регіонів або згладити отримані границі між сегментами.

Алгоритм має наступний вигляд:

1. Призначення міток.

$$M(x, y) = \begin{cases} -1, & \text{якщо піксель належить об'єкту} \\ 1, & \text{якщо піксель належить фону} \end{cases}.$$

2. Оцінка відстані до границі.

$$F(x, y) = \begin{cases} 0, & \text{якщо } M(x, y) = -1 \\ \infty, & \text{якщо } M(x, y) = 1 \end{cases}.$$

3. Процес розповсюдження.

$$F(x, y) = \min \left(F(x, y), \min_{(x', y') \in N_4(x, y)} F(x', y') + 1 \right).$$

4. Розповсюдження міток.

$$M(x, y) = \begin{cases} M(x, y), & \text{якщо } M(x, y) = -1 \\ M(x, y) = \min M(x, y) \text{ та } M(x', y') + 1, & \forall (x', y') \in N_4(x, y) \end{cases}.$$

де, M – маска, що містить мітки; F – функція відстані до границі; $N_4(x, y)$ – множина пікселів сусідів в границях 4-пов'язаної області.



Рис. 1.6. Результат алгоритму вододілу

Завдяки використанню даного алгоритму для рентгенівського зображення приведенного на рисунку 1.1 було отримано значення коефіцієнта Жаккара – 31.95 %. На рисунку 1.6 наведено маску-результат застосування алгоритму вододілу.

1.2.5. Графові методи

Графові методи сегментації – це клас методів сегментації зображень, де вони представляють зображення як граф, в якому вершини представлені у вигляді елементів зображення, а ребра відповідають зв'язкам між ними. Графові методи побудовані за рахунок теорії графів для вирішення різного роду завдань сегментації зображень. Математичне моделювання, зокрема методи оптимізації та чисельні алгоритми, дозволяють покращити точність побудови графів для сегментації, зменшуючи вплив шуму та помилок в даних. Таким чином зображення у процесі розділяється на окремі сегменти, котрі відповідають існуючим об'єктам на вхідному зображенні. Графові методи сегментації будуються на поняттях “різниці” або “ваги” між пікселями. Для цього можна використовувати різні фактори, такі як наприклад, інтенсивність пікселів, тощо. Саме використовуючи ці різниці, методи графової сегментації визначають границі між сегментами.

Існує багато популярних графових методів, а саме: графове розрізання, графова морфологія, алгоритм мінімального перерізу у графі та GrabCut. Графове розрізання – це популярна техніка, яка поділяє зображення на передній план і фон, знаходячи оптимальний розріз у графічному представленні. Графова морфологія застосовує морфологічні операції до графів для отримання значних структур [13]. GrabCut розглядає проблему ефективного інтерактивного виділення об'єкта переднього плану у складному середовищі, фон якого неможливо тривіально виділити [14].

За допомогою мінімального перерізу у графі можна визначити такий набір ребер, який розділить граф на два компоненти, визначаючи мінімальну суму ваги. Такий підхід гарно підходить для розділення зображення на сегменти. Точність сегментації графового методу відображена на рисунку 1.7. Алгоритм мінімального перерізу у графі можна представити наступним чином: нехай $G = (V, E)$ – це граф

з множиною вершин V та множиною ребер E . Нехай s і t – джерело та стік, а $c(u, v)$ – це вага ребра (u, v) . Тоді отримаємо мінімальний переріз (S, T) , де $S \subseteq V$ і $T = V \setminus S$, задовольняє таким умовам:

1. $s \in S$ (джерело належить множині S)
2. $t \in T$ (стік не належить множині S)
3. Мінімальна сума ваг ребер, які перетинають межу між S і T :

$$\text{minCut} = \sum_{\substack{u \in S \\ v \in T}} c(v, u),$$

де S і T задовольняють умовам 1 і 2.

Значення коефіцієнта Жаккара для алгоритму мінімального перерізу, застосованого до рентгенівського зображення на рисунку 1.1 склало 16.13 %.



Рис. 1.7. Результат алгоритму мінімального перерізу

1.2.6. Глибоке навчання для сегментації зображень

Використання глибокого навчання для сегментації зображень в наш час стало дуже популярною складовою обробки медичних зображень. Використовуючи глибокі нейронні мережі, котрі розпізнають аномалії на медичних знімках, ми змогли досягти значного прогресу в автоматизації та точності сегментації зображень. Глибокі згорткові мережі призвели до прориву в обробці зображень [15]. Математичне моделювання в глибокому навчанні дозволяє підвищити точність та

ефективність сегментаційних алгоритмів, оптимізуючи їх для роботи з медичними зображеннями. Однією з найбільш поширених областей де глибоке навчання знайшло своє практичне використання є сегментація медичних зображень, таких як рентгенівські знімки, КТ знімки або МРТ знімки. Одним із найбільших напрямлень де використовуються ці методи є сегментація зображень для діагностування раку. Використовуючи великі обсяги анотованих даних, моделі глибокого навчання можуть вивчати складні закономірності та зв'язки в медичних зображеннях, полегшуючи точне виявлення, локалізацію та діагностику захворювань і аномалій [16], тим самим допомагаючи лікарю швидше визначити діагноз.

Сегментація зображень на основі глибокого навчання є процесом який включає в себе декілька етапів, а саме – підготовка даних, побудова моделі, навчання моделі та її оцінка. На етапі підготовки даних потрібно визначитись, чи будуть застосовуватись аугментовані дані, яка буде розмірність зображень, провести нормалізацію зображень, тощо. При побудові моделі глибокого навчання потрібно вибрати архітектуру нейронної мережі, яка буде використовуватись для сегментації, наприклад, можна використовувати згорткові нейронні мережі або згортково-рекурентні нейронні мережі. На наступному етапі проходить використання підготовлених даних для оптимізації параметрів та використання зворотного поширення помилки. В процесі відбувається адаптація параметрів моделі з використання оптимізатора для зменшення функції втрати. І на останньому етапі перевіряються передбачення моделі на тестових даних для оцінки її точності.

Для сегментації легень позначених на рис.1.1 була використана згорткова нейронна мережа VGG-Unet. VGG-Unet є гібридною нейронною мережею, яка об'єднує архітектуру VGG[17] та U-Net[18]. На рисунку 1.8 зображено архітектуру VGG-Unet, яка використовувалась в процесі тренування моделі для сегментації легень. Було проведено тренування моделі протягом 15 епох, при навчанні аугментація даних не використовувалась, для тренування моделі було використано 112 зображень розміром 256×256 пікселів. Завдяки рисунку 1.10 можна ознайомитись з графіками втрат та точності моделі в процесі оптимізації

параметрів. На рисунку 1.9 можна побачити передбачення для рентгенівського знімка зображеного на рисунку 1.1. Значення коефіцієнту Жаккара склало 93.89 %.

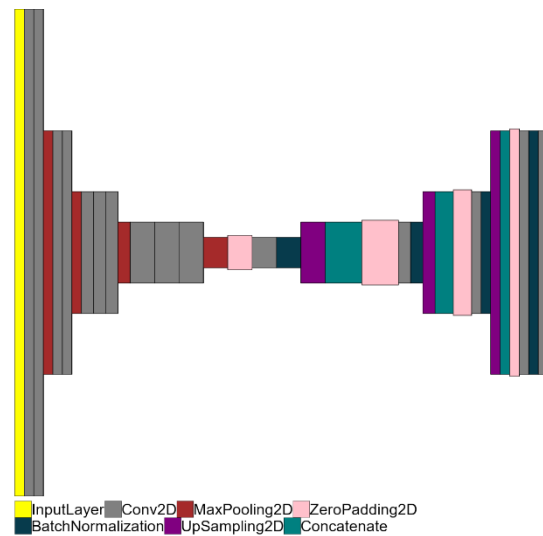


Рис. 1.8. Архітектура VGG-Unet



Рис. 1.9. Сегментація за допомогою VGG-Unet

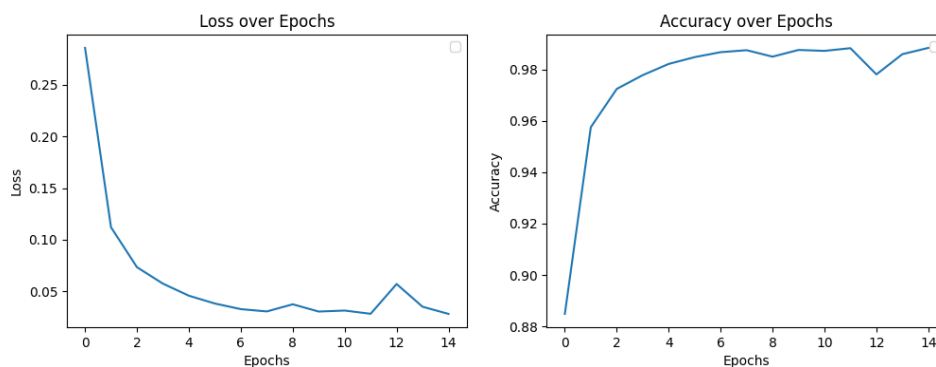


Рис. 1.10. Графік втрат та точності

1.2.7. Порівняльний аналіз ефективності алгоритмів сегментації

Розглянувши ключові алгоритми сегментації зображень, можна проаналізувати деякі аспекти їх використання та їхню ефективність при застосуванні в медичному контексті. Для того, щоб порівняти всі алгоритми, вони були застосовані для одного зображення, позначеного на рисунку 1.1. Точність усіх алгоритмів визначалась за допомогою коефіцієнту IoU у відсотковому вигляді. Було проведено порівняльний аналіз наступних методів:

- Порогова сегментація: було розглянуто 4 методи, а саме – глобальна порогова сегментація, локальна порогова сегментація, метод Отсу, ручна сегментація. Дивлячись таблицю 1 можна побачити, що всі методи дають різний результат, а самим найкращим результатом виявився метод ручної сегментації який отримав значення коефіцієнту Жаккара 61.06 %. Цей метод також займає 2 місце в діаграмі зображень на рисунку 1.11. Це може констатувати наступне – використання порогової сегментації в медичних зображеннях не є найкращим методом та не може гарантувати гарного результату, але в межах завдань комп'ютерного зору, його можна використовувати як допоміжний захід для основного інструменту.

- Метод К-середніх: автоматичний алгоритм який використовує кластеризацію для сегментації зображення, яка побудована на схожості кольорів та інтенсивності пікселів. Займає третє місце в рейтингу. В ролі самостійного алгоритму – не може гарантувати гарну точність. Значення коефіцієнту Жаккара склало 41.34 %.

- Алгоритм зростання регіонів: робить упор на збільшенні регіонів, котрі повинні обов'язково базуватись на критеріях схожості. Даний алгоритм може застосовуватись для завдань сегментації на зображеннях, які мають чітко визначені текстури. Виходячи з великої кількості різних видів медичних знімків, які можна отримати з рентгену, КТ або МРТ цей метод не може гарантувати задовільну точність для всіх цих випадків. Значення коефіцієнту Жаккара склало 38.24 %.

- Алгоритм вододілу: побудований на основі процесу імітації затоплення ландшафту. Основним з недоліків є те що при використанні з низькою

контрастністю або рівномірною яскравістю може видавати погані результати, тому що він повинен виявляти границі. Значення коефіцієнту Жаккара 31.95 %.

- **Графові методи:** ці методи використовують теорію графів для визначення розрізу, завдяки якому зображення буде розподілено на сегменти. В завданнях сегментації часто можуть виникати випадки коли проблеми з параметризацією та висока обчислювальна вартість можуть впливати на ефективність їх використання. В даному випадку значення коефіцієнту Жаккара склало 16.13 %.

- **Глибоке навчання:** алгоритм використовує нейронні мережі для навчання моделей, які будуть виконувати автоматичну сегментацію. Даний підхід є найрезультативнішим в даний час для усіх видів завдань. Він демонструє найкращу точність та можливість адаптуватися під різні вимоги. Використовуючи коефіцієнт Жаккара було отримано значення 93.89 %.

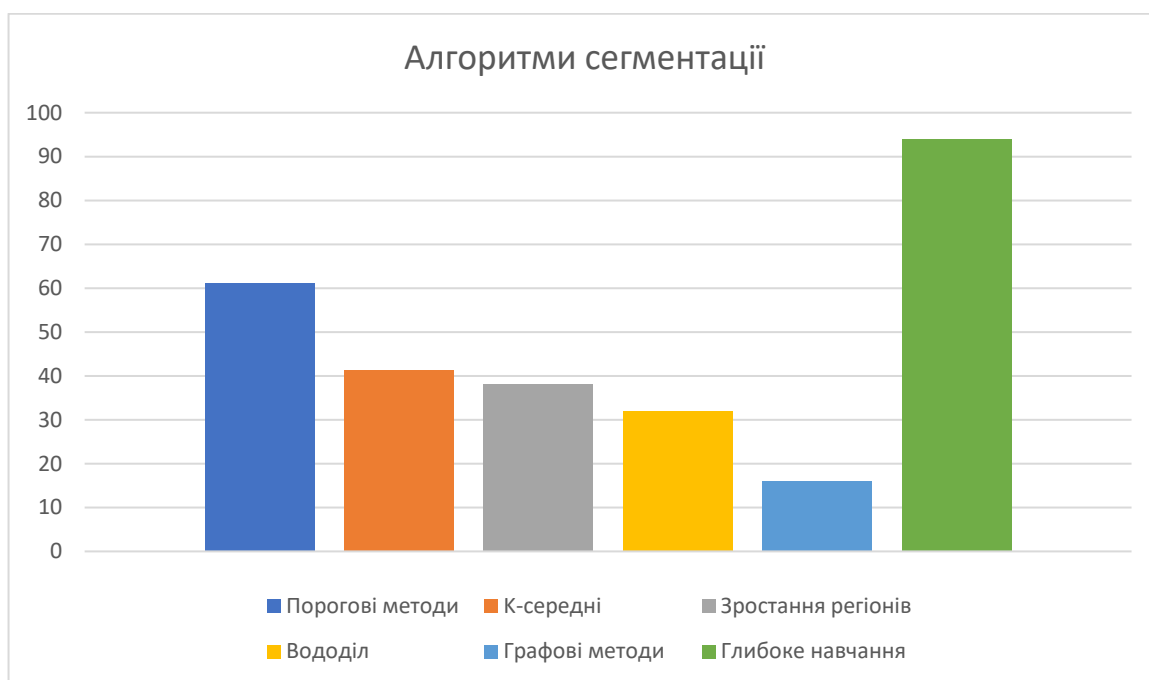


Рис. 1.11. Діаграма точності алгоритмів сегментації, IoU %

Кожен з вказаних вище методів має свої переваги та недоліки. Ці методи мають різні напрямки використання, але нас насамперед цікавлять завдання в контексті медичного характеру. Головним завданням сегментації зображень в медицині є в першу чергу точність, а потім інші фактори. Проаналізувавши всі

вказані вище алгоритми сегментації зображень можна з впевненістю заявити, що метод глибокого навчання є найкращим серед усіх інших і його подальше покращення та використання безсумнівно дадуть найкращий результат.

1.2.8. Проблеми та виклики у сегментації медичних зображень

Виконуючи завдання сегментації ми стикаємося з багатьма проблемами та викликами, тому що сегментація зображень – це дуже складний процес, а сегментація медичних зображень має ще більш складні виклики. Основними проблемами в медичних знімках є артефакти, шуми, погана експозиція. Вони можуть погіршити результат сегментації та призвести до неточностей. Також однією із проблем є низька контрастність деяких зображень, що в свою чергу також призводить до погіршення точності та ускладнення сегментації. Великі розміри зображень та великі обсяги даних, різне освітлення, погана контрастність та різниця в яскравості – це все першочергові часті проблеми у сегментації медичних зображень. Не варто забувати також про неоднорідність анатомічних структур, які в свою чергу теж впливають на точність визначення. Справді, побудова більш цілісних і глобальних анатомічних моделей залишається однією з найбільших завдань та можливостей у майбутньому через відсутність великих наборів даних, що дозволяють охарактеризувати складність анатомії людини [19]. З розвитком технологій, типи зображень стають все більш різноманітними, що створює додаткові проблеми при сегментації зображень з різними розмірами, масштабами, роздільною здатністю та режимами візуалізації [20].

Невід’ємною складовою кожної сегментації є зображення. Це питання становить дуже вагомую роль і воно відіграє не останню роль. Насамперед через доступ до медичних даних. Медичні дані – це дані які знаходяться у обмеженому доступі або взагалі є приватними. Тому нам потрібно враховувати конфіденційність даних при зборі та деякі етичні аспекти. Збір, зберігання та обробка персональних медичних даних є важливою складовою та може вимагати деяких додаткових заходів безпеки та згоди пацієнтів.

Для отримання гарних результатів потрібно мати якісну вибірку даних, ці дані мають бути належно анотовані, що вимагає втручання ручної роботи експертів. Це в свою чергу вимагає належного фаху у цих спеціалістів та досвіду і витрат часу, оскільки це потребує значної кількості часу.

Виходячи з усього зазначеного вище, можемо констатувати те, що перед сегментацією медичних зображень постають різні проблеми та виклики. Вирішення деяких з них є тривіальними, а інші можуть становити значну проблему. Але всі вони мають свій розв'язок і саме завдяки їх вирішенню можна покращити процес сегментації медичних зображень.

Висновки до першого розділу

В першому розділі було проведено аналітичний огляд ключових алгоритмів сегментації медичних зображень, було виконано дослідження їх точності та ефективності. Саме ці результати дозволили визначити основні переваги і недоліки кожного з методів в контексті їх застосування в медичній площині.

Розглянувши порогові методи стало зрозуміло, що вони є дуже простими та швидкими у застосуванні, але вони дуже чутливі до рівнів яскравості та контрасту, що в медичному використанні впливає на точність сегментації і їх ізольоване використання не є пріоритетним. І не останнім чинником є те, що найкращий результат показав ручний метод, коли ми саме зацікавлені в автоматизації сегментації.

Метод К-середніх теж є швидким, але коли мова йде про точність то він починає вимагати попереднього визначення кількості кластерів, що в свою чергу може вплинути на недостатню точність його роботи у разі якщо пікселі будуть розділені нерівномірно.

Алгоритм зростання регіонів може продемонструвати успішний результат в тих випадках, коли об'єкти мають схожі значення інтенсивності, що в нашому випадку не завжди можливо.

Алгоритм вододілу є успішним коли об'єкти на зображенні мають чіткі контури, що викликає сильні проблеми з його використанням коли зображення

мають наявність шуму.

Графові методи мають високі обчислювальні витрати, що потребує значних ресурсів, особливо при обробці великої кількості даних. Також вони можуть бути менш ефективними в плані сегментації зображень з нечіткими та поганими контурами.

Сегментація зображень на основі глибокого навчання є найуспішнішою із усіх вище вказаних методів. Вона виявилася потужною та гнучкою системою вирішення складних сегментаційних питань та показала свою здатність адаптуватися до різних умов та структур, що забезпечило їй високу точність.

Загальний висновок полягає в тому, що використання глибокого навчання є найефективнішим та найперспективнішим в питанні сегментації медичних зображень. Звісно кожен з розглянутих методів має свої переваги та недоліки і ми можемо підлаштовувати краще середовище для їх використання щоб отримати кращий результат але в плані роздільного використання всі вони значно програють глибокому навчанню і можуть бути використанні тільки, як допоміжні інструменти.

РОЗДІЛ 2

АНСАМБЛЕВІ МЕТОДИ ДЛЯ СЕГМЕНТАЦІЇ ЗОБРАЖЕНЬ

2.1. Основи ансамблевих методів

Одними із методів за допомогою яких можна підвищити точність сегментації, є ансамблеві методи у машинному навчанні. Основна ідея цих методів полягає в об'єднанні декількох моделей для збільшення точності результатів процесу сегментації, в порівнянні з використанням лише однієї моделі [21]. Також, за допомогою математичного моделювання таких методів, зокрема за допомогою статистичних та оптимізаційних технік, дозволяє більш точно оцінити ваги моделей в ансамблі та підвищити ефективність їх взаємодії. Завдяки цьому, стає можливим об'єднання переваг декількох моделей та зменшення їх недоліків. Комбінування декількох моделей для виконання однієї і тієї ж задачі може істотно зменшити варіативність та підвищити стійкість до надмірного перенавчання [22]. Це в свою чергу дозволяє уникнути недоліків, які можуть виникати через використання тільки однієї моделі.

Одним із найпопулярніших ансамблевих методів є беггінг [23]. Для того щоб скористатись цим методом необхідно навчити декілька моделей на різних наборах даних, після чого виконати комбінування передбачень. Одним із основних прикладів є випадковий ліс, який виконує тренування різних дерев рішень, в якому кожне дерево повинне вивчати різні підмножини даних та ознак, що в результаті зменшує варіативність використаної моделі [24, 25].

Наступний метод це – бустрепінг, який покращує моделі, навчаючи кожну наступну модель на основі помилок попередньої. При використанні алгоритму AdaBoost [26], відбувається збільшення ваги зразків, які були неправильно класифіковані попередніми моделями, такий підхід дає змогу моделям, які будуть наступними, сконцентруватися на більш складних для класифікації випадках [27]. Також немаловажним є той факт, що в деяких випадках дерева рішень використовуються, як базові моделі для AdaBoost. В багатьох задачах в теперішній

час зазвичай використовується Gradient Boosting, за допомогою якого стало можливим провести оптимізацію функції втрат, в основі якої лежить поступова побудова моделей, покращуючи результати попередніх [28]. Градієнтний бустинг займає не останнє місце серед ансамблевих методів, які використовуються під час обробки зображень. Завдяки градієнтному бустингу виконується покрокове навчання слабких моделей, що також є деревами рішень, такий підхід дає змогу покращити кінцеву якість прогнозів, а також виправляти помилки слабких моделей. Використання цього методу можна запропонувати, як для незалежної класифікації об'єктів, а також для постобробки.

Стекінг метод, який об'єднує декілька різних моделей, результати котрих є вхідними даними для мета-моделі, яка і робить кінцеве передбачення. Завдяки такому підходу, ми отримуємо можливість комбінувати нейронні мережі, дерева рішень, лінійну регресію, що значно покращує точність передбачення [29, 30]. Метод голосування, використовує декілька моделей, які роблять передбачення, після чого остаточне рішення приймається в процесі голосування цих моделей. Існує декілька видів голосування, таких як: мажоритарне або зважене голосування, де для кожної моделі необхідно визначити ваги залежно від її точності [31, 32].

Ансамблеві методи, мають як переваги так і недоліки. Ключовою особливістю використання ансамблевих методів є підвищення точності за рахунок об'єднання кількох моделей, кожна з яких має різний підхід до виконання своєї задачі. Саме комбінування різних моделей дає змогу підвищити стійкість результатів. Також однією із переваг є гнучкість ансамблів, що дає змогу застосовувати, як однорідні так і неоднорідні моделі. Це робить ансамблеві методи універсальними для широкого спектру задач. Основними ж недоліками є висока обчислювальна складність. Через необхідність тренування великої кількості моделей, зростає необхідність збільшення обчислювальних ресурсів. Також одним із основних недоліків є проблема з узагальненням на різних наборах даних, що в свою чергу є проблемою використаних моделей. В деяких випадках використання ансамблевих методів може продемонструвати гарні результати на навчальному наборі, але при експериментах на тестових даних або нових може отримувати

поганий результат. Це впливає з того, що деякі моделі, які використовуються в ансамблі можуть підлаштовуватись під специфіку окремого набору зображень, що веде до зниження потенціалу усього ансамблю.

Також при загальній думці – чим більше, тим краще, збільшення кількості моделей у ансамблі не завжди призводить до покращення точності сегментації. Завдяки деяким дослідженням, стало зрозуміло, що збільшення кількості моделей у ансамблі може впливати на зниження ефективності передбачення. Використання менш точних моделей, також призводить до погіршення результатів.

2.2. Алгоритм усереднення з порогом

Для того, щоб визначити ефективність нових запропонованих алгоритмів, необхідно виконати їх порівняння з уже існуючими алгоритмами. Одним із найпопулярніших та найпростіших алгоритмів у реалізації є – алгоритм усереднення з порогом, який зарекомендував себе, як гарний ансамблевий підхід для сегментації зображень. Цей алгоритм використовується для комбінування передбачень отриманих від різних моделей, для одного зображення. В кінці ми отримуємо маску, яка отримала усереднену інформацію з усіх передбачень [33].

Спочатку потрібно отримати передбачення від використаних моделей, для цього можна використовувати, як моделі одної архітектури так і моделі різних архітектур. Нехай дано N передбачень для використаних моделей, для одного i -го зображення. Значення інтенсивності пікселя i -ї матриці пікселів в координатах (x, y) буде $P_i(x, y)$, де $P_i(x, y) \in [0, 1]$, оскільки 0 відповідає класу фону, а 1 класу об'єкта. Визначивши це, можна отримати середнє значення для кожного елемента зображення (x, y) за формулою:

$$\bar{P}(x, y) = \frac{1}{N} \sum_{i=1}^N P_i(x, y).$$

Середнє значення пікселя $P_i(x, y)$ означає ймовірність його приналежності до класу об'єкта. Для отримання фінальної маски, необхідно використати поріг T . Завдяки використанню порогу, стає можливим визначення, чи відноситься елемент

зображення до об'єкта, чи він відноситься до класу фону. Найпопулярнішим порогом є $\frac{1}{2}$, для порівняння далі буде також використано поріг $\frac{2}{3}$, який дає змогу отримати впевнений результат для пікселя. Також можливо використовувати поріг який дорівнює 1. Такий підхід зараховує до фінальної маски тільки ті пікселі, які були отримані на передбаченнях усіх моделей та вказували на присутність об'єкта в цьому пікселі. Збільшення порогу, знижує ймовірність випадкових хибно-позитивних результатів, оскільки, для того щоб класифікувати елемент зображення, як частину об'єкта, потрібно більше позитивних результатів від усіх передбачень. Фінальна маска $M_{avg}(x, y)$ для елемента зображення в координатах (x, y) обчислюється за формулою:

$$M_{avg}(x, y) = \begin{cases} 255, & \bar{P}(x, y) > T \\ 0, & \bar{P}(x, y) \leq T \end{cases}$$

Використання даного алгоритму дає можливість покращити точність, оскільки результат фінальної маски залежить від декількох передбачень, що зменшує вплив випадкових результатів. В деякій мірі, такий підхід мінімізує вплив шуму. Також головною перевагою даного алгоритму є гнучкість у використанні порогу, який можна регулювати залежно від потреб.

2.3. Ансамблевий алгоритм об'єднання

Ансамблевий алгоритм об'єднання є дуже простим підходом для комбінування передбачень. Для побудови цього алгоритму використовуються булеві операції. Ансамблевий алгоритм об'єднання використовується для підсилення ефективності результатів окремих моделей, за рахунок створення однієї об'єднаної маски. Завдяки цьому алгоритму збільшується охоплення пікселів, котрі належать до класу об'єкта, в цьому випадку в фінальну маску вибираються всі пікселі, які були передбачені хоча б однією моделлю. Це можна виразити наступним чином:

$$M_{union} = \bigcup_{i=1}^N P_i, \quad (2.1)$$

де P_i – i -те передбачення, N – кількість моделей в ансамблі.

Використання даного алгоритму може збільшити вірогідність знаходження об'єкта, навіть попри те, що деякі моделі можуть створити хибні передбачення. Але попри таку перевагу, з'являється інший недолік – створення хибних результатів, де пікселі об'єкта можуть бути визначені, навіть якщо вони є тільки в деяких передбаченнях.

2.4. Методи оцінки ефективності ансамблевих алгоритмів

При використанні ансамблевих методів, постає питання оцінки їхньої ефективності. Якщо для двох моделей оцінка якості є очевидною — достатньо порівняти їхні значення DICE і вибрати модель з вищим показником, то для порівняння ансамблевого результату та результатів окремих моделей, які були використані в ансамблі ситуація є менш однозначною, і такий підхід може виявитися неефективним. Методи оцінки ефективності ансамблевих методів беззаперечно є важливою частиною в аналізі їх продуктивності. Завдяки їм ми можемо оцінити наскільки добре ансамбль може виконати своє завдання в порівнянні з моделями поодиночі. В основі всіх запропонованих методів лежить коефіцієнт DICE, який виступає в якості метрики для оцінки схожості двох множин:

$$DICE = \frac{2 \times |A \cap B|}{|A| + |B|}.$$

Для того, щоб оцінити ефективність ансамблевих методів ми можемо використовувати наступні методи:

Середній DICE. Використання цього метода дає змогу порівняти результат ансамблю з середнім значенням усіх моделей, які були використані в ансамблі. Такий підхід дає змогу отримати загальне уявлення про якість сегментації, враховуючи всі моделі, наскільки ансамблевий метод покращує, чи погіршує загальну якість передбачень в порівнянні з мережами поодиночі. В подальшому, для всіх тестів, які будуть виконані в цій роботі, буде використовуватись саме цей підхід. Формула має наступний вигляд:

$$\text{Mean Dice} = \frac{1}{N} \sum_{i=1}^N \text{Dice}_i,$$

де N – це кількість моделей в ансамблі, Dice_i – значення Дайс для i -ї моделі.

Медіанний DICE. Такий підхід дає змогу визначити медіанне значення показників DICE, завдяки чому це робить його менш чутливим к викидам в порівнянні з середнім. Насамперед це дуже ефективно коли деякі моделі отримують значно гірші результати. Цей підхід можна визначити за формулою:

$$\text{Median Dice} = \text{median}(\text{Dice}_1, \text{Dice}_2, \dots, \text{Dice}_N).$$

Максимальний DICE. Використання цього підходу, являє собою порівняння результату отриманого при використанні ансамблю та максимального значення DICE для однієї моделі серед моделей ансамбля. Цей метод дуже ефективний для оцінки найкращої продуктивності ансамблю, але він в свою чергу не дає повної картини, так як не бере до уваги точність інших моделей. Максимальне значення для найкращої моделі можна отримати за формулою:

$$\text{Max Dice} = \max(\text{Dice}_1, \text{Dice}_2, \dots, \text{Dice}_N).$$

Таким чином, можна зробити висновок, що для ефективної оцінки результатів роботи ансамблевих методів потрібно використовувати комплексний підхід, який зможе використати різні аспекти продуктивності моделей. Використовуючи декілька методів оцінки ефективності, ми маємо змогу більш точніше визначити якість роботи ансамблевого метода.

Висновки до другого розділу

У другому розділі були розглянуті ансамблеві алгоритми, які є важливою складовою машинного навчання. Використання яких дозволяє підвищити точність та стабільність використаних моделей. Об'єднання декількох моделей, для

отримання одного передбачення дає змогу компенсувати недоліки окремих моделей та зменшити їх негативний вплив.

Для оцінки ефективності ансамблевих алгоритмів є можливість застосовувати різні підходи, як окремо так і разом. Ці підходи допомагають зрозуміти та оцінити ефективність ансамблів в порівнянні з окремими моделями. Для більш точного вивчення ефективності слід використовувати не тільки один підхід, необхідно оцінювати ефективність ансамблю в порівнянні з найкращою моделлю та середнім значенням моделей, для того, щоб отримати розуміння, який підхід дає оптимальний результат.

РОЗДІЛ 3

РОЗРОБКА НОВИХ АНСАМБЛЕВИХ АЛГОРИТМІВ ДЛЯ СЕГМЕНТАЦІЇ ЗОБРАЖЕНЬ НИЗЬКОЇ ЯКОСТІ

3.1. Використані нейронні мережі та набори даних

Для проведення експериментів та більш детального визначення ефективності запропонованих алгоритмів було використано десять архітектур нейронних мереж, які були створені для сегментації зображень. Такий підхід дав змогу розгорнуто порівняти ефективність використаних підходів до сегментації зображень низької якості та виявити найбільш точні алгоритми.

Для виконання поставлених завдань були використані наступні архітектури: Dilated-UNet [34], FastUNet [35], Swin-Unet [36], UNet 3+ [37], RAUNet [38], Attention U-Net [39], Inception [40], Resnet101 U-Net [41], Resnet152v2 U-Net [42], Fcn8 MobileNet [43].

Переважає більшість нейронних мереж, які будуть використовуватись для досліджень, базується на архітектурі U-Net. Сама нейронна мережа U-Net може забезпечувати достатньо непогану ефективність в задачах сегментації, за рахунок її здатності відновлювати просторові особливості. Але для складних завдань U-Net не може похизуватися відмінними результатами, саме для цього й використовуються покращенні версії. Покращення можуть бути досягнуті за допомогою використання додаткових компонентів, таких, як наприклад рекурентні блоки. Також одним із покращень може бути використання LSTM-U-Net, це дає змогу використовувати сильні сторони U-Net об'єднуючи їх із можливостями LSTM, що в свою чергу відображається в моделюванні тимчасових залежностей. Завдяки використанню LSTM маємо змогу ефективно виконувати обробку частин зображень, за рахунок чого використовується більший контекст та отримуються покращенні результати.

Dilated-UNet – це модифікована архітектура U-Net, яка до базової архітектури додає дилатацію згорток для покращення отримання контекстної інформації при

сегментації. За рахунок використання дилатованих згорток, вдається захопити більш ширший контекст без критичного збільшення кількості параметрів. Принцип ефективного розширення поля сприйняття полягає в пропуску певних пікселів, замість використання сусідніх.

FastUNet – це покращена версія U-Net, яка була спеціально розроблена для досягнення більш високої швидкості та точності сегментації зображень. Це стало можливим завдяки зменшенню кількості параметрів та обчислювальних витрат, при всьому цьому вдалося покращити продуктивність архітектури в задачах сегментації. Такі нововведення дозволяють застосовувати запропоновану архітектуру в реальному часі, де швидкість відіграє ключову роль.

Swin-Unet побудована на базі Swin Transformer [44]. Ця архітектура уособлює в собі переваги трансформерів та згорткових нейронних мереж, завдяки чому вона здатна досягати високої продуктивності в задачах сегментації зображень. Використання Swin Transformer дало змогу краще отримувати інформацію, як локальну так і глобальну. Використання архітектур трансформерів, як підходу до сегментації зображень, в наш час набуло свого розвитку, оскільки це дозволяє гарантувати високоякісну обробку інформації. Swin Transformer адаптує класичну архітектуру Transformer для задач обробки зображень, завдяки чому стає можливим застосування ієрархічних структур та локальних вікон. Використання такого підходу у моделі Swin-Unet забезпечує нас ефективною глобальною інформацією, а також контекстними зв'язками під час виконання сегментації.

UNet 3+ стала розширенням базової архітектури U-Net. Її покращення базується на створенні масштабного бачення, за рахунок чого відбувається отримання інформації на різних рівнях. Також для збереження інформації та зменшення втрат деталей використовується декілька рівнів пропусків-з'єднань.

RAUNet є запропонованим варіантом для покращення виділення важливих ознак під час сегментації зображення, за рахунок використання механізму уваги та залишкових зв'язків. Для вирішення проблеми затухаючих градієнтів в цій архітектурі пропонується використання залишкових зв'язків, що допоможе передавати інформацію через мережу. Механізм уваги дає можливість підсилити

вагомі ознаки та зменшити вплив неважливих. Такий підхід дає можливість моделі зосередитись на ключових областях зображення.

Attention U-Net використовує механізм уваги для зосередження на найважливіших частинах зображення.

Inception має гарну здатність зберігати свою ефективність при одночасному використанні ядер згортки різних розмірів. Основні її переваги – це багатошаровість та паралельна обробка. Використання різних типів фільтрів та їх паралельна робота, дозволяє моделі отримувати кращі результати при сегментації.

Resnet101 U-Net – це ще один із варіантів архітектури U-Net, де Resnet101 використовується в якості кодера. Resnet101 має 101 шар та використовує залишкові зв'язки, що використовується для передачі інформації через велику кількість шарів. Пропускові з'єднання використовуються для передачі інформації між шарами кодера та декодера. Завдяки такому підходу вдається отримати покращення точності сегментації, оскільки модель здатна зберігати більше інформації про ознаки зображення.

Resnet152v2 U-Net має вже 152 шари та використовує залишкові зв'язки. Ця архітектура зменшує затухання градієнтів. Як і попередня архітектура вона складається з двох частин: кодера та декодера.

Остання використана архітектура Fcn8 MobileNet поєднує в собі переваги повнозгорткової мережі та ефективної мобільної мережі. Fcn8 MobileNet має легку архітектуру та все-таки дозволяє досягати хороших результатів при сегментації, завдяки використанню потужних механізмів виділення ознак.

Для всебічної оцінки ефективності запропонованих ансамблевих алгоритмів буде використано два набори даних. Обидва набори даних склалися з рентгенівських зображень грудної клітини в боковій проекції. Також всі зображення були приведені до однакового розміру, а саме 512×512 пікселів та градації сірого. Перша вибірка [45] складалась із 183 зображень. Для оптимізації параметрів моделей вибраних нейронних мереж було застосовано 146 зображень. Решта 37 зображень були використані для оцінки продуктивності моделей. В якості складної геометричної структури, яку необхідно було ідентифікувати, використовувався

регіон з чотирьох хребців: Th8, Th9, Th10, Th11. На рисунку 3.1 зображено декілька зображень, які знаходяться в першому наборі даних.

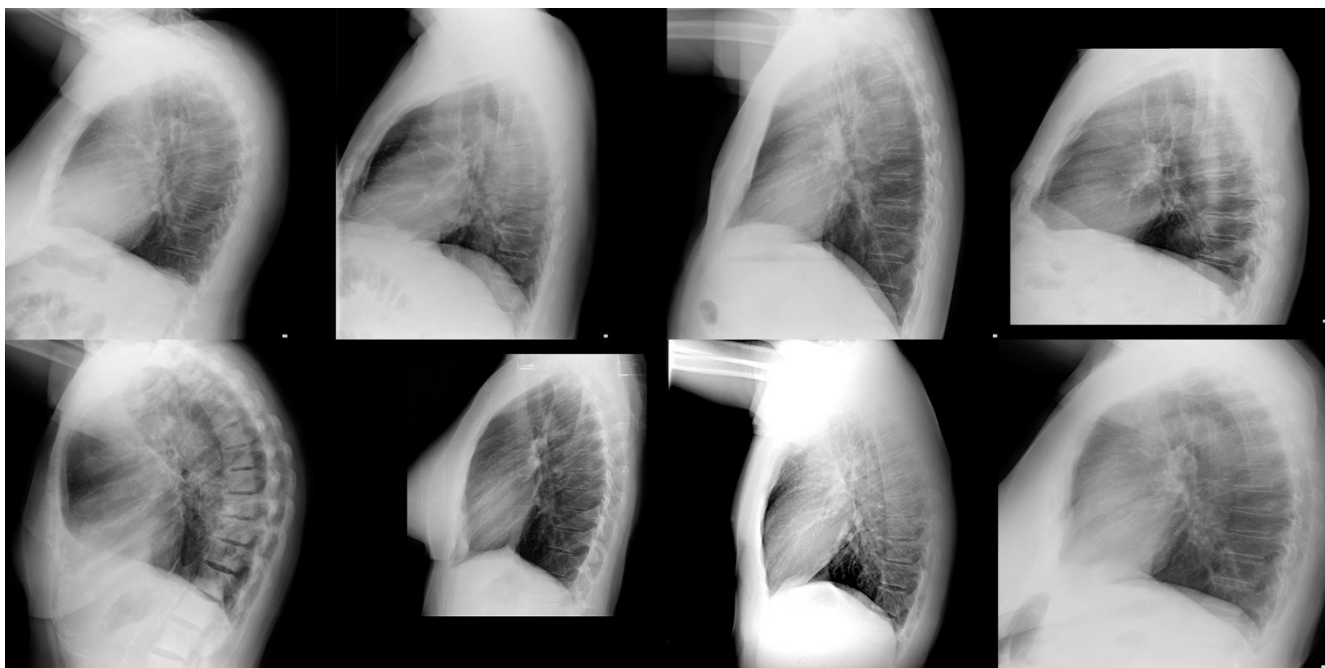


Рис. 3.1. Приклад зображень з першого набору даних

На рисунку 3.2 зображено приклад маски, яка використовувалась для тренування моделей. Де зліва зображена сама маска, яка відображає складну геометричну структуру, яку необхідно ідентифікувати, що складається з наступних хребців: Th8, Th9, Th10, Th11; а праворуч зображено накладання цієї маски на відповідне зображення, завдяки чому можна детально побачити в якому регіоні буде виконуватись ідентифікація складної геометричної структури.

Також було використано другу вибірку [46]. Яка складалась із 58 зображень, які в своїй переважній більшості були низької якості та в порівнянні з першою вибіркою були не такими якісними. Ця вибірка була спеціально використано для вивчення можливостей адаптивності запропонованих архітектур та ансамблевих алгоритмів. Оскільки в переважній більшості, моделі навчені на одному наборі даних демонструють погані результати на інших наборах, що унеможлиблює їх використання в процесі автоматизації.

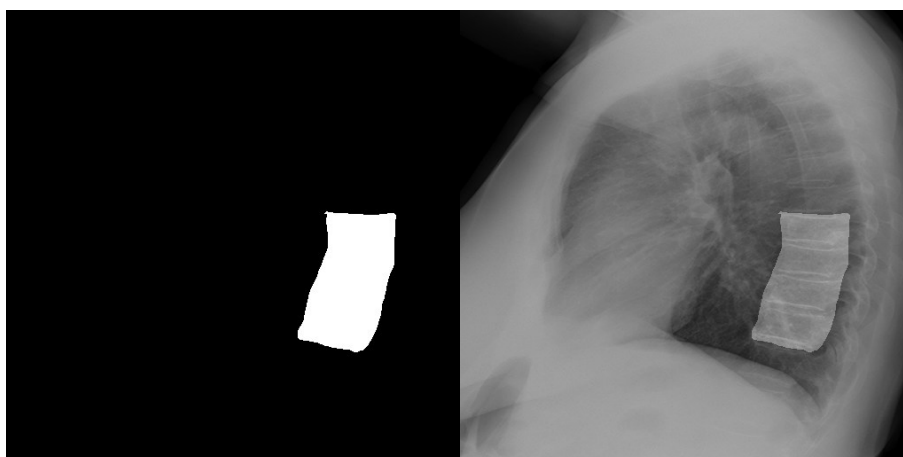


Рис. 3.2. Приклад маски для тренування моделей

На рисунку 3.3 зображені знімки, які входять до другого набору даних. Порівнявши зображення продемонстровані на рисунках 3.1 і 3.3, можна зробити висновок про їх якість. Як можна побачити, зображення демонструють різний рівень деталізації, що в свою чергу, може впливати на якість сегментації. На деяких зображеннях присутній низький контраст, що робить деталізацію менш чіткою. Також рівень освітлення відіграє значну роль, деякі знімки мають затемнені регіони, це теж ускладнює процес сегментації. Загалом на міжнародному рівні, думки про медичну корисність зображень низької, чи поганої якості, в порівнянні з критеріями зображення відрізняються [47]. Все-таки, необхідність опрацювання зображень такої якості в наш час залишається необхідним питанням.

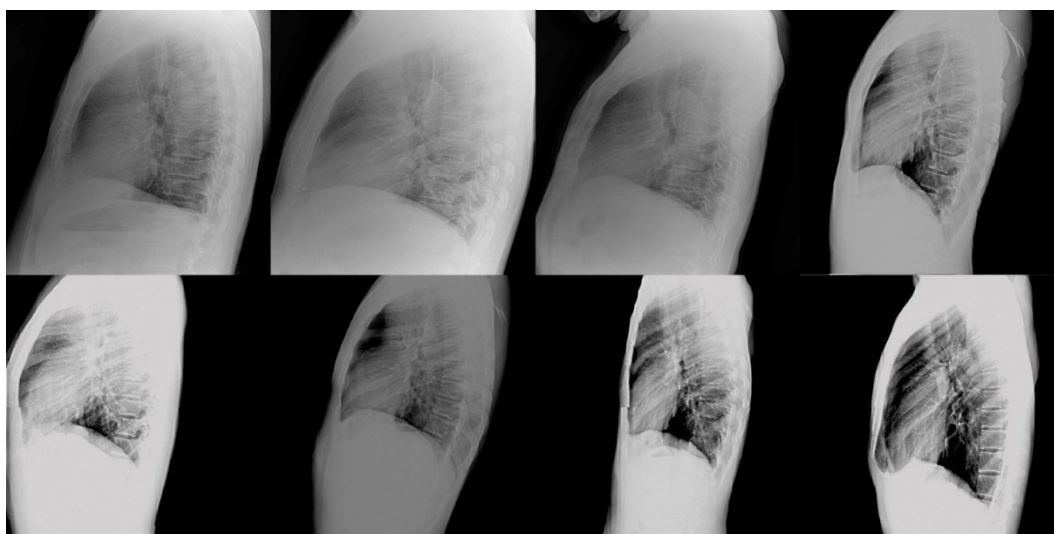


Рис. 3.3. Приклад зображень з другого набору даних

В таблиці 3.1 наведені результати навчених моделей, які будуть використанні в подальшому для створення ансамблів, що саме включає в себе використання декількох нейронних мереж [48]. Для кожної моделі було отримано результати із використанням тестових зображень, першого та другого наборів даних. З метою отримання більшої наочності, було отримано середнє, мінімальне та максимальне значення для всіх моделей. Для визначення подібності передбачення та правдивої маски використовувався коефіцієнт DICE.

Таблиця 3.1

Оцінка якості сегментації: значення DICE % для навчених моделей

Модель	Набір даних	Номер моделі										\bar{X}	min X	max X
		1	2	3	4	5	6	7	8	9	10			
Dilated-UNet	1	95.75	95.74	95.71	95.62	95.60	95.49	95.11	95.07	95.02	95.02	95.41	95.02	95.75
	2	79.40	78.59	81.59	78.88	80.17	80.18	80.34	80.09	80.22	80.21	79.97	78.59	81.59
FastUNet	1	91.35	90.66	90.08	89.54	89.41	88.95	88.86	88.68	87.68	87.30	89.25	87.30	91.35
	2	71.49	67.67	67.10	68.27	66.23	67.57	65.34	69.71	36.93	65.97	64.63	36.93	71.49
Swin-UNet	1	84.61	84.57	84.16	83.43	83.39	82.85	82.46	82.08	81.24	80.77	82.96	80.77	84.61
	2	67.58	66.14	60.75	69.00	63.05	62.10	63.07	60.30	62.48	65.56	64.00	60.30	69.00
UNet 3+	1	91.78	90.98	90.94	90.65	89.73	88.62	88.39	87.64	84.32	77.20	88.02	77.20	91.78
	2	68.34	60.41	63.96	62.04	63.91	58.39	66.93	61.33	34.01	36.56	57.59	34.01	68.34
RAUNet	1	91.61	91.45	90.21	89.88	89.79	89.73	88.93	88.85	88.20	82.53	89.12	82.53	91.61
	2	70.30	66.16	68.88	61.81	55.69	59.36	57.96	68.00	65.13	43.94	61.72	43.94	70.30
Attention U-Net	1	88.40	88.16	87.84	87.32	86.05	84.32	81.75	81.70	80.53	79.08	84.51	79.08	88.40
	2	50.34	62.19	71.82	61.80	30.85	44.88	36.12	45.08	49.32	37.65	49.00	30.85	71.82
Inception	1	83.87	83.24	82.23	81.74	81.65	80.06	78.96	78.80	76.38	73.89	80.08	73.89	83.87
	2	57.26	57.19	57.80	48.27	50.69	45.82	49.97	65.22	50.44	44.88	52.75	44.88	65.22
Resnet101 U-Net	1	90.79	87.31	85.93	85.77	83.59	83.35	83.22	82.50	78.04	69.98	83.05	69.98	90.79
	2	59.39	54.06	54.89	45.19	46.00	55.40	30.88	53.00	47.45	30.66	47.69	30.66	59.39
Resnet152v2 U-Net	1	94.95	93.67	93.50	92.09	88.04	80.98	79.01	78.48	72.91	48.30	82.19	48.30	94.95
	2	72.38	59.45	58.49	53.80	51.49	61.23	56.78	29.50	33.75	42.25	51.91	29.50	72.38
Fcn8 MobileNet	1	93.92	91.67	91.17	90.58	90.31	89.54	89.29	88.97	88.89	85.21	89.96	85.21	93.92
	2	76.96	76.26	69.05	72.67	59.08	68.82	60.36	71.57	68.90	62.48	68.62	59.08	76.96

Для того щоб детально вивчити поведінку ансамблів, була навчена велика кількість моделей. Це було зроблено для того, щоб зрозуміти, скільки моделей повинно бути використано, щоб отримати кращий результат. Для цього пропонується розглядати ансамблі, які містять моделі різної точності. Крім того,

використання моделей з різною точністю в ансамблі дозволяє оцінити доцільність поєднання високоточних та низькоточних моделей або ж визначити, чи варто використовувати лише моделі з високою точністю.

Для моделей Dilated-UNet різниця між максимальним та мінімальним значенням склала 0.73 % та 3.00 % для першого та другого наборів відповідно; для моделей FastUNet — 4.04 % та 34.56 %; для моделей Swin-Unet — 3.84 % та 8.70 %; для моделей UNet 3+ — 14.58 % та 34.33 %; для моделей RAUNet — 9,07 % та 26,37 %; для моделей Attention U-Net — 9,32 % та 40,98 %; для моделей Inception — 9,98 % та 20,34 %; для моделей ResNet101 U-Net — 20,81 % та 28,73 %; для моделей ResNet152v2 U-Net — 46,64 % та 42,88 %; для моделей FCN8 MobileNet — 8,71 % та 17,88 %.

Аналіз розподілу точності всіх навчених моделей показує, що нейронна мережа Dilated-UNet має найменшу варіативність у порівнянні з іншими нейронними мережами. Це свідчить про те, що під час оптимізації параметрів моделі кожна модель з архітектурою Dilated-UNet є надзвичайно стабільною та добре адаптованою до даних. Натомість інші моделі, побудовані на основі різних нейронних мереж і характеризовані більшим розкидом між реалізаціями, можуть мати гіршу здатність до узагальнення та недостатньо ефективно обробляти зашумлені зображення.

Порівняння середніх значень точності першого та другого наборів для всіх навчених моделей показало, що різниця в значеннях DICE становить 26 %. Середнє значення DICE для першого набору склало 86 %, тоді як для другого – 60 %. Очевидно, така значна різниця зумовлена тим, що деякі нейронні мережі не здатні повною мірою адаптуватися до нових даних і демонструють значно гірші результати.

Отримані результати підкреслюють необхідність зваженого підходу до вибору нейронних мереж для виконання конкретних завдань. На жаль, не завжди є можливість використовувати глибокі складні моделі з великою кількістю параметрів, і часом доводиться віддавати перевагу більш компактним архітектурам. Проте, з огляду на отримані дані, вибір оптимальної нейронної мережі залишається

складним завданням для високовимогливих систем.

3.2 Алгоритм усереднення форм об'єктів

Першим запропонованим ансамблевим алгоритмом є – алгоритм усереднення форм об'єктів. Даний алгоритм використовується для комбінації результатів передбачень від моделей, які використовуються в ансамблі. Він деякою мірою схожий на алгоритм усереднення з порогом. Але є важливий нюанс, в той час коли в алгоритмі усереднення з порогом, необхідно отримати середнє значення для передбачень та порівняти його з заданим порогом, зазвичай це 0.5, в даному випадку відбувається усереднення форм об'єктів. Математичне моделювання цього процесу базується на врахуванні різних метрик відстані, що дозволяють коректно поєднувати форми об'єктів, одночасно зменшуючи вплив шуму та помилок прогнозів на результат. В основі цього алгоритму лежить функція трансформації відстані, яка використовується для обчислення відстані від пікселя об'єкта до найближчого пікселя фону на бінарному зображенні [49].

Для початку необхідно стисло визначити сам алгоритм, після чого вже детально розглянути кожний його етап. Усереднення передбачень за формою об'єктів відбувається за наступною формулою:

$$I_{avg} = (dm_{all} > 0) \times 255, \quad (3.1)$$

де I_{avg} – це агреговане передбачення, dm_{all} – це матриця відстаней усіх масок.

Матриця відстаней усіх масок визначається за формулою:

$$dm_{all} = \sum_{i=1}^n dm(mask_i), \quad (3.2)$$

де dm визначено за формулою:

$$dm(mask) = dt(mask) - dt(\neg mask), \quad (3.3)$$

де dt – це функція трансформації відстані.

Візуалізація роботи алгоритму, що відповідає формулами (3.1–3.3), представлена на рисунку 3.4.

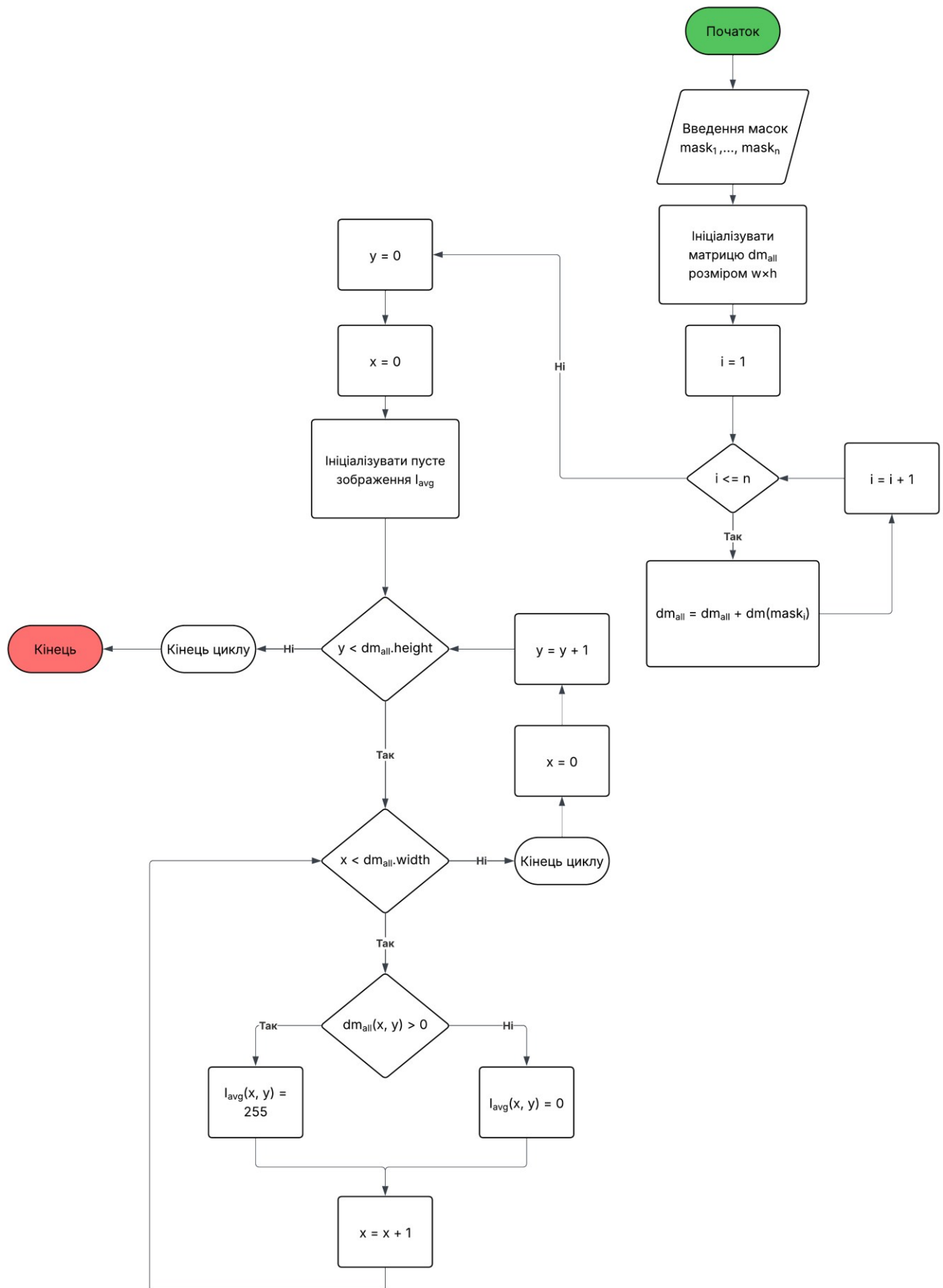


Рис. 3.4. Блок-схема алгоритму

Тепер необхідно детально розглянути кожний етап. Алгоритм починається з формули (3.3), де для маски відбувається обчислення трансформації відстані, яка потім записується в відповідну матрицю, котра відповідає розмірам маски. Візуально це можна побачити на рисунку 3.5.

0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	1	1	1	1	1	1	0	0	0
0	0	1	2	2	2	2	1	0	0	0
0	0	1	2	3	3	2	1	0	0	0
0	0	1	2	3	3	2	1	0	0	0
0	0	1	2	2	2	2	1	0	0	0
0	0	1	1	1	1	1	1	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0

Рис. 3.5. Приклад матриці відстаней, $dt(mask)$

На цьому рисунку для прикладу було використано зображення розміром 10×10 пікселів, де в ролі об'єкта виступає білий квадрат розміром 5×5 . Наведений приклад було отримано за допомогою використання $dt(mask)$. Для кожного пікселя об'єкта було обчислено відстань до найближчого пікселя фону. Саме через це у чорного фону, відстань для всіх пікселів дорівнює нулям. Для кожного пікселя об'єкта було записано відстань до найближчого пікселя фону у відповідний елемент матриці відстаней. Також в формулі (3.3) використовується інвертована маска, що зазначено, як $dt(\neg mask)$. На рисунку 3.6 зображено процес інверсії маски та отримання відстаней для неї.

2.8	2.2	2	2	2	2	2	2	2.2	2.8
2.2	1.4	1	1	1	1	1	1	1.4	2.2
2	1	0	0	0	0	0	0	1	2
2	1	0	0	0	0	0	0	1	2
2	1	0	0	0	0	0	0	1	2
2	1	0	0	0	0	0	0	1	2
2	1	0	0	0	0	0	0	1	2
2	1	0	0	0	0	0	0	1	2
2.2	1.4	1	1	1	1	1	1	1.4	2.2
2.8	2.2	2	2	2	2	2	2	2.2	2.8

Рис. 3.6. Приклад матриці відстаней для інвертованої маски, $dt(\neg mask)$

Після того, як було обчислено відстані для маски та інвертованої маски, в результаті було отримано дві матриці відстаней, приклади яких зазначені на рисунках 3.4 та 3.5. У процесі віднімання матриці відстаней інвертованої маски від матриці відстаней маски формується $dm(mask)$, яка представляє матрицю відстаней, де пікселі, розташовані поза об'єктом, мають від'ємні значення відстані, тоді як пікселі всередині об'єкта — додатні. Це зображено на рисунку 3.7.

-2.8	-2.2	-2	-2	-2	-2	-2	-2	-2.2	-2.8
-2.2	-1.4	-1	-1	-1	-1	-1	-1	-1.4	-2.2
-2	-1	1	1	1	1	1	1	-1	-2
-2	-1	1	2	2	2	2	1	-1	-2
-2	-1	1	2	3	3	2	1	-1	-2
-2	-1	1	2	3	3	2	1	-1	-2
-2	-1	1	2	2	2	2	1	-1	-2
-2	-1	1	1	1	1	1	1	-1	-2
-2.2	-1.4	-1	-1	-1	-1	-1	-1	-1.4	-2.2
-2.8	-2.2	-2	-2	-2	-2	-2	-2	-2.2	-2.8

Рис. 3.7. Приклад матриці відстаней, $dm(mask)$

На наступному етапі в формулі (3.2) відбувається процес додавання всіх матриць відстаней для використаних масок. Завдяки цьому отримуються нульові перетини на краях масок. Після того, як було виконано операції додавання для всіх масок, отримаємо dm_{all} , в якому, пікселі, які відносяться до фону, мають значення від'ємне, пікселі на краях масок дорівнюють нулю, а пікселі об'єкта мають позитивні значення. Таким чином, вираження $dm_{all} > 0$ перетворює всі пікселі, які мають значення більше нуля в 1, а фон перетворюється в 0 [50]. В результаті отримано бінарне зображення з агрегованим об'єктом. Для отримання зображення в градаціях сірого, всі пікселі необхідно нормалізувати, помноживши їх на 255.

При використанні даного алгоритму можуть виникати деякі проблеми пов'язані з великим розкидом розмірів складних геометричних структур, пустими передбаченнями та передбаченнями на яких складна геометрична структура знаходиться дуже далеко в порівнянні з іншими передбаченнями.

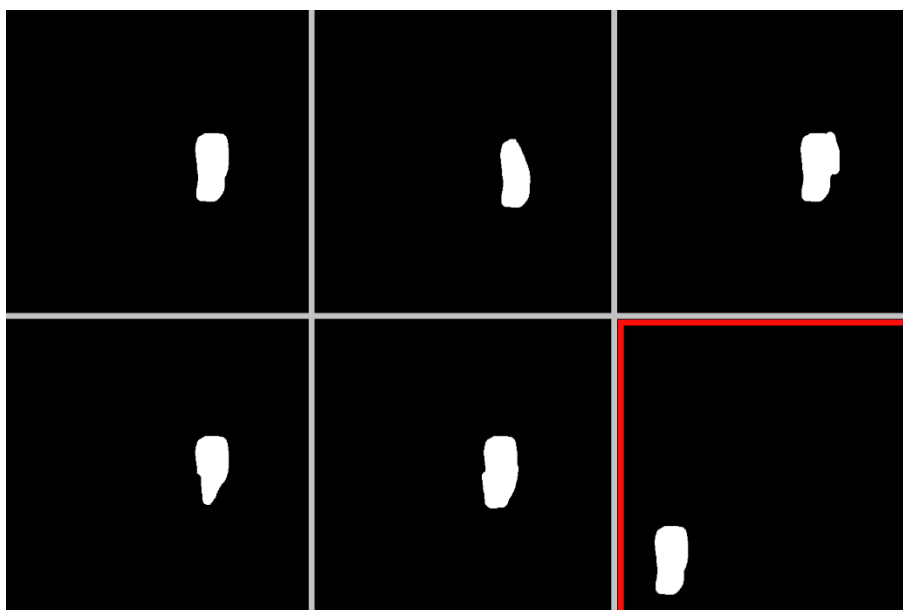


Рис. 3.8. Складна геометрична структура знаходиться дуже далеко від інших

На рисунку 3.8 відображено випадок, коли одна з складних геометричних структур розташована на передбачені, знаходиться далеко від розташування складних геометричних структур з інших передбачень. На рисунку зображено шість масок, в п'яти випадках складні геометричні структури знаходяться приблизно в одному місці, а шоста позначена червоною рамкою знаходиться взагалі в іншому місці в порівнянні з п'ятьма попередніми.

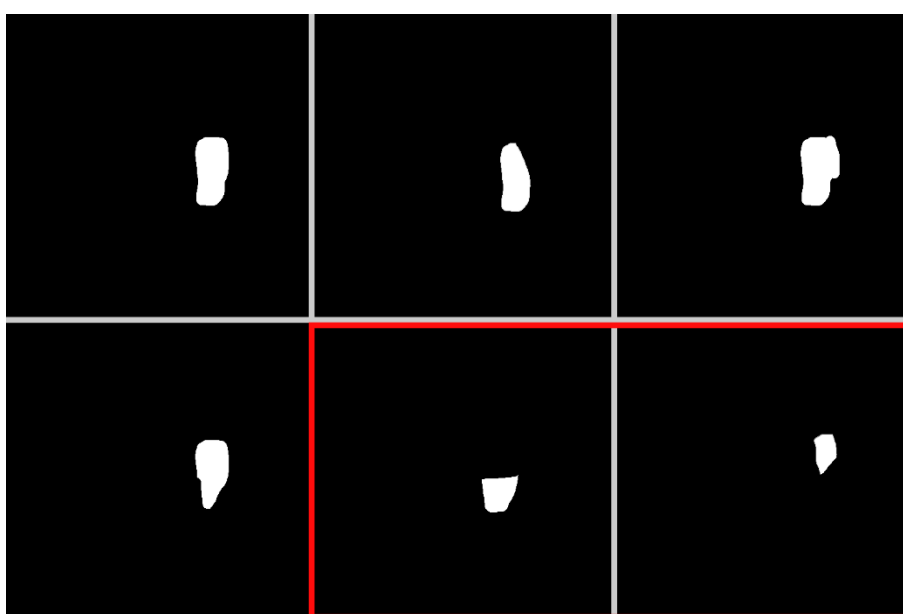


Рис. 3.9. Розміри складних геометричних структур значно відрізняються

На рисунку 3.9 зображено інший випадок, коли деякі складні геометричні структури на передбаченнях значно менші від інших. Чотири з шістьох передбачень, мають складні геометричні структури, розміри яких відрізняються не дуже значно. Два передбачення, які обведені червоною рамкою, мають складні геометричні структури, форми яких приблизно вдвічі менші в порівнянні з іншими чотирма передбаченнями. Такі випадки створюють деякі проблеми для використання даного алгоритму, що може призводити до його неефективного використання.

Тому для вирішення цих питань запропоновано наступні кроки: видаляти пусті передбачення, видаляти передбачення де складна геометрична структура знаходиться дуже далеко від інших та об'єднувати маленькі складні геометричні структури.

Для того, щоб видалити передбачення на яких складні геометричні структури знаходяться далеко від інших, потрібно на кожному передбаченні визначити центр мас для них, потім отримати середнє значення – центроїд, завдяки якому можна визначити, які центри знаходяться далеко від центроїда.

Нехай існує N передбачень, які зберігаються в наборі $P = \{P_1, P_2, \dots, P_N\}$. Для кожного передбачення P_i необхідно отримати центр мас $C_i = (x, y)$ за допомогою методу `get_center_of_mass`:

$$C_i = \text{get_center_of_mass}(P_i).$$

Наступним кроком є обчислення центроїда всіх центрів мас шляхом визначення їхнього середнього значення:

$$\bar{C} = \frac{1}{N} \sum_{i=1}^N C_i.$$

Для того, щоб визначити відстань між центроїдом та i -м центром мас, знайдемо Евклідову відстань до центроїда \bar{C} :

$$d_i = \sqrt{(x_i - \bar{x})^2 + (y_i - \bar{y})^2},$$

де d_i – це відстань від \bar{C} до C_i .

Необхідно встановити поріг T , за допомогою якого будуть визначатися центри, які знаходяться далеко від центроїда. В даній роботі було використано наступний поріг $T = \frac{512}{5}$, він був отриманий експериментальним методом. Так, як зображення, які використовувались були квадратними, а одна сторона дорівнювала 512, саме це значення й було використано. Знаючи поріг та відстань між центроїдом та центром мас i -го об'єкта, можна отримати індекси передбачень, об'єкти на яких знаходяться далеко від центроїду:

$$O = \{i | d_i > T\}.$$

Останній пункт – фільтрація передбачень, необхідно зберегти тільки ті передбачення, на яких об'єкт знаходиться в необхідній досяжності від центроїда:

$$P_{res} = \{P_i | i \notin O\}.$$

Для вирішення другої проблеми, коли розміри складних геометричних структур значно відрізняються, необхідно виконати їх фільтрацію по площі їх контурів, з наступним об'єднанням структур, якщо площа таких структур менше порога.

В першу чергу необхідно визначити середнє значення для площ контурів усіх передбачень:

$$\bar{A} = \frac{1}{N} \sum_{i=1}^N A_i,$$

де A_i – це площа контуру i -ї складної геометричної структури передбачення, N – кількість моделей в ансамблі.

Необхідно виконати фільтрацію масок за площею, буде дві множини масок: перша, де площа не менша за поріг; друга, де площа менша за поріг. Поріг визначимо за наступною формулою:

$$T = \frac{\bar{A}}{2}.$$

Тоді перша множина буде мати вигляд:

$$F = \{P_i | A_i \geq T\}.$$

та друга множина:

$$U = \{P_i | A_i < T\}.$$

Якщо множина U не пуста, тоді необхідно за допомогою формули (2.1) виконати об'єднання усіх масок, які знаходяться в U , в результаті чого отримаємо агреговану маску P_{union} .

Фінальний набір масок буде складатись з масок, які належать F та агрегованої маски P_{union} . Таким чином отримаємо набір масок, у якому маски з малими площами об'єднані в одну, а решта масок залишилась в початковому вигляді:

$$R = F \cup P_{union}.$$

В функції трансформації відстані можуть використовуватись різні типи відстаней. В залежності від типу відстані виконується або точне обчислення, або приблизне обчислення відстані. Під час дослідження було використано три типи відстані: мангеттенська відстань, Евклідова відстань, відстань Чебишева. Мангеттенську відстань можна визначити за формулою:

$$d_{man}(P, Q) = |x_1 - x_2| + |y_1 - y_2|,$$

де $P = (x_1, y_1)$ та $Q = (x_2, y_2)$

Евклідова відстань визначається за допомогою формули:

$$d_{eucl} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}, \quad (3.4)$$

Та відстань Чебишева була визначена за наступною формулою:

$$d_{cheb}(P, Q) = \max(|x_1 - x_2|, |y_1 - y_2|).$$

Для того, щоб визначити використання якого типу відстані буде гарантувати кращі результати, було проведено експеримент, де використовувалось три нейронні мережі: Dilated-UNet, Resnet152v2 U-Net, Fcn8 MobileNet. Для кожної нейронної мережі було навчено по дві моделі, які робили передбачення для другого набору даних, після чого використовувався запропонований алгоритм усереднення форм об'єктів, який усереднював передбачення двох моделей. Після отримання агрегованої маски, виконувалось порівняння агрегованої маски та правдивої маски за допомогою значення DICE. Для детального вивчення поведінки на різних моделях, було використано моделі, результати яких були майже однакові та мали велику та дуже велику різницю. Для прикладу наведено значення DICE для використаних моделей: Dilated-UNet 0.815864 та 0.801795; Resnet152v2 U-Net 0.603583 та 0.726723; Fcn8 MobileNet 0.337497 та 0.612248.

З результатів продемонстрованих у таблиці 3.2 можна побачити, що найкращою комбінацією типів відстані є використання d_{cheb} для $mask$ та d_{man} для $\neg mask$.

Таблиця 3.2

Визначення найкращого типу відстані, DICE

Модель	Тип відстані для $mask$	Тип відстані для $\neg mask$		
		d_{man}	d_{eucl}	d_{cheb}
Dilated-UNet	d_{man}	0.815348	0.815138	0.814433
	d_{eucl}	0.815864	0.815619	0.815225
	d_{cheb}	0.816295	0.816111	0.815664
Resnet152v2 U-Net	d_{man}	0.673017	0.671361	0.665645
	d_{eucl}	0.679662	0.677406	0.672185
	d_{cheb}	0.683878	0.682531	0.676914
Fcn8 MobileNet	d_{man}	0.389500	0.389279	0.386714
	d_{eucl}	0.392756	0.392113	0.390101
	d_{cheb}	0.395906	0.394678	0.391639

3.3 Алгоритми на основі центрування

В основі всіх алгоритмів, які будуть запропоновані нижче, лежить алгоритм центрування об'єктів. Основна його ідея полягає в знаходженні середнього центра для всіх складних геометричних структур передбачень та зміщення цих структур відносно середнього центра [51]. Якщо ми візьмемо N моделей (приблизно однакових по точності) та зробимо V передбачень для одного зображення, ми зможемо побачити, що в деяких випадках, форми та положення складних геометричних структур значно відрізняються. Навіть моделі майже з однаковими показниками точності в результаті створюють різні передбачення. Саме це й спонукало для використання ансамблю нейронних мереж та створення алгоритму центрування масок. Отримані передбачення від різних моделей зможуть доповнити друг друга та покращити надійність ансамблю. Використання центра у задачах обробки зображень має широкий спектр, наприклад, існує метод оцінки якості сегментації зображень, який використовує центри об'єктів для порівняння з еталонною сегментацією [52]. Існує ще один метод виявлення об'єктів, який використовує припущення про центральні точки об'єктів. Цей підхід фокусується на прогнозуванні центрів об'єктів замість традиційного методу з рамками, що обмежують об'єкт [53].

Перш за все, було запропоновано дві гіпотези: 1) чим більше моделей в ансамблі, тим менша відстань між середнім центром та центром еталонної маски; 2) чим менша відстань від середнього центра до центра передбачення тим більше значення DICE.

Для того щоб підтвердити чи спростувати ці гіпотези, потрібно дати поняття середнього центра. Для початку дамо визначення середньому центру. Центр мас твердого тіла – це унікальна точка, яка є середнім становищем усієї маси, з якої складається об'єкт [54]. Нехай дано n зображень $\{I_1, I_2, \dots, I_n\}$, де для кожного зображення існує центр мас $\{C_1, C_2, \dots, C_n\}$, де кожен центр має координати (x_i, y_i) . Середній центр можна отримати за формулою $C_{mean} = (x_{mean}, y_{mean})$, де

$$x_{mean} = \frac{1}{n} \sum_{i=1}^n x_i,$$

та

$$y_{mean} = \frac{1}{n} \sum_{i=1}^n y_i.$$

Послідовність усіх етапів роботи алгоритму представлена на рисунку 3.10.



Рис. 3.10. Блок-схема роботи алгоритму

При розробці цього алгоритму було отримано два завдання: довести, чи спростувати дві гіпотези, визначити ефективність запропонованих алгоритмів. Для дослідження двох гіпотез було обрано нейронну мережу Fcn8MobileNet. Використання даної мережі обумовлено її полегшеною архітектурою, яка швидко навчається при цьому зберігаючи високу продуктивність. Використовуючи перший набір даних було навчено 50 моделей. Набір даних було розділено наступним чином: 80 % для навчання; 10 % для валідації; 10 % для тестування. Під час тренування моделей кількість епох становила 100, розмір партії становив 32, для запобігання перенавчання використовувався метод EarlyStopping, який запобігав перенавчанню, якщо значення втрати перевірки не покращилося протягом десяти епох. Для виконання другого завдання було використано запропоновані моделі з підрозділу (3.1).

Для детального дослідження було обрано 10 ансамблів, які склалися з: 5, 10, 15, 20, 25, 30, 35, 40, 45, 50 моделей. В кожному тестовому зображенні відповідно до кількості моделей в ансамблі було отримано середній центр для складних геометричних структур, після чого було обчислено відстань від середнього центра до еталонного центра. Середнє значення відстані для всіх зображень у ансамблі було записано у таблицю. Відстань поміж точками вимірювалась по формулі (3.4).

Для дослідження другої гіпотези, для кожного зображення було використано 50 моделей навчених на першому наборі. Для кожного зображення було отримано відстань від центра складної геометричної структури передбачення до середнього центра по 50 маскам. Також для кожного зображення було отримано значення DICE. Для того щоб зрозуміти чи відстань від центра до середнього центра має вплив на DICE, було використано коефіцієнт кореляції Пірсона, для визначення залежності відстані від центра складної геометричної структури до середнього центра та коефіцієнту DICE. Коефіцієнт кореляції Пірсона зазвичай використовується для даних, які спільно нормально розподілені (тобто даних, наступних двовимірному нормальному розподілу) [55]. Коефіцієнт кореляції визначається за формулою:

$$r = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}},$$

де n – кількість пар значень, x_i та y_i значення X та Y , \bar{x} та \bar{y} середні значення вибірок X та Y .

Після того, як було отримано середнє значення центрів, потрібно кожну складну геометричну структуру на масці змістити відносно середнього центра. На рисунку 3.11. можна візуально побачити аналіз алгоритму. На цьому рисунку наведено еталоні маски створені людиною, складна геометрична структура виділена білим кольором – це регіон, який складається з 4 хребців. Червоні точки – це центри передбачень, синя точка – це середній центр, а зелена точка – це центр об'єкта еталонної маски.

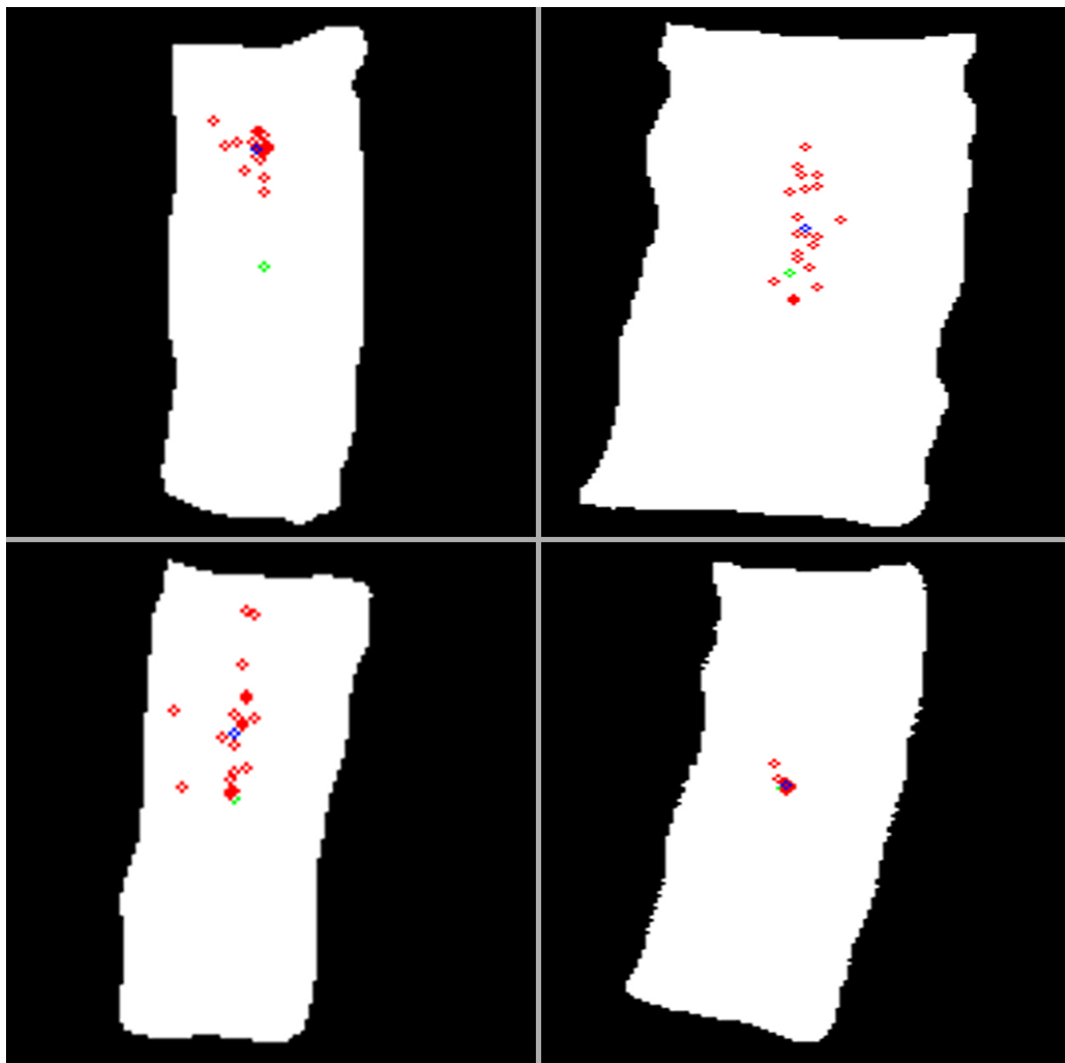


Рис. 3.11. Центри складних геометричних структур

Проаналізувавши зображення можна отримати багато інформації, наприклад, про форми складних геометричних структур, ідентифікацію яких необхідно виконати. В деяких випадках, центри складних геометричних структур розкидані далеко один від одного, що свідчить про велику варіативність форм передбачень кожної моделі. Так як, контури відрізняються по формі, також, по розкиду центрів складних геометричних структур на передбаченнях можна зрозуміти, що звичайний алгоритм має деякі виклики в цьому випадку. Наочно можна побачити користь використання середнього центра. Середній центр знаходиться дуже близько до еталонного центра.

Таблиця 3.3

Визначення відстані між середнім центром та еталонним центром

Тип відстані	Кількість моделей в ансамблі									
	5	10	15	20	25	30	35	40	45	50
Евклідова відстань	3.5889	3.4932	3.4556	3.4599	3.4898	3.5094	3.4918	3.4906	3.4724	3.4776

Як можна побачити з отриманих результатів, наведених в таблиці 3.3, постійне збільшення кількості моделей в ансамблі кардинально не зменшило відстань, що в свою чергу може значить, що запропонована гіпотеза вірна частково (до 15 моделей у ансамблі). Найкращий результат було отримано при використанні 15 мереж в ансамблі, таким чином це стало оптимальним значенням і подальше збільшення моделей не призводить до значних покращень.

Наступна гіпотеза, про те, що “чим менше відстань від центра маски до середнього центра, тим більше значення DICE”. Вона була спростована завдяки використанню коефіцієнта кореляції Пірсона. Отриманий результат склав 0.0269, це свідчить про те, що слабка позитивна залежність проглядається між змінними, але вона незначна та не може бути статистично важливою. Таким чином було зроблено висновок, що маски, центри котрих близько розташовані до еталонного

центра можуть мати значення DICE менше ніж маски центри яких розташовані далі. Для цих двох гіпотез було використано 50 моделей, що стало достатньо для отримання необхідних результатів.

3.4 Комбіновані ансамблеві алгоритми

В комп'ютерному зорі, для покращення якості сегментації та для отримання більш точних результатів можна використовувати комбінування різних алгоритмів сегментації. Такий підхід є потужним інструментом для вирішення актуальних завдань. Цей підхід дозволяє застосовувати переваги кожного алгоритму, нівелюючи їх недоліки та покращуючи точність. Можливість комбінування різних алгоритмів, створює для нас великі можливості в плані побудови різних варіацій комбінованих алгоритмів.

Комбінований метод, який спочатку використовує вирівнювання гістограми, після чого застосовує медіану фільтрацію та в кінці використовує алгоритм нечіткої кластеризації, дає можливість покращити сегментацію зображення та зменшити час обчислення [56]. Поєднання кластеризації на основі графів і семантичних функцій високого рівня, в результаті дало змогу отримати послідовну та значущу сегментацію [57]. Метод, який використовує ієрархічну багаторівневу схему сегментації з механізмом пошуку на основі дерева для перевірки узгодженості сегментованих регіонів та їх взаємозв'язку один з одним, забезпечуючи автоматичний адаптивний вибір параметрів продемонстрував високу ефективність для різних типів зображень [58]. Існує велика кількість робіт присвячених різним варіаціям використання комбінацій методів сегментації [59-66]. Наявність такої кількості робіт та результати отримані при дослідженнях підтверджують сенс використання комбінованого підходу для вирішення задач комп'ютерного зору.

В цьому підрозділі будуть наведені запропоновані комбінації ансамблевих алгоритмів. Під час сегментації зображень ансамблеві методи можуть включати різну кількість моделей, які можна використовувати паралельно або послідовно, а їх результати об'єднувати для покращення якості та точності сегментації. Кожний ансамблевий алгоритм має свої переваги та недоліки, кожен алгоритм по різному

виконує комбінування передбачень в ансамблі, тож комбінація із декількох алгоритмів може отримати більш точний результат. Доречний підбір комбінації допомагає справитися з шумом, врахувати складні форми складних геометричних структур та мінімізувати негативний вплив деяких ефектів. В таблиці 3.4 наведені запропоновані комбінації, вони будуть складатися з таких ансамблевих алгоритмів: алгоритм усереднення з порогом $T = 1/2$, алгоритм усереднення з порогом $T = 2/3$, алгоритм об'єднання для ансамблевих моделей сегментації, алгоритм усереднення форм об'єктів, а також для всіх запропонованих алгоритмів раніше, буде використано алгоритм центрування, завдяки чому матимемо змогу порівняти ефективність запропонованих ансамблевих алгоритмів та їх модифікованих версій, за допомогою центрування.

Таблиця 3.4

Комбінації ансамблевих алгоритмів

Назва комбінації	Алгоритми які застосовуються
Комбінація 1	алгоритм усереднення з порогом $T = 1/2$, алгоритм усереднення з порогом $T = 2/3$.
Комбінація 2	алгоритм усереднення з порогом $T = 1/2$, алгоритм усереднення з порогом $T = 2/3$, алгоритм усереднення з порогом $T = 1/2$ на основі центрування, алгоритм усереднення з порогом $T = 2/3$ на основі центрування.
Комбінація 3	алгоритм усереднення з порогом $T = 1/2$, алгоритм об'єднання для ансамблевих моделей сегментації.
Комбінація 4	алгоритм усереднення форм об'єктів, алгоритм об'єднання для ансамблевих моделей сегментації.
Комбінація 5	алгоритм об'єднання для ансамблевих моделей сегментації, алгоритм об'єднання для ансамблевих моделей сегментації на основі центрування.

Продовження таблиці 3.4

Комбінація 6	алгоритм усереднення форм об'єктів, алгоритм усереднення форм об'єктів на основі центрування.
Комбінація 7	алгоритм усереднення з порогом $T = 1/2$, алгоритм усереднення з порогом $T = 2/3$, алгоритм об'єднання для ансамблевих моделей сегментації, алгоритм усереднення форм об'єктів, алгоритм усереднення з порогом $T = 1/2$ на основі центрування, алгоритм усереднення з порогом $T = 2/3$ на основі центрування, алгоритм об'єднання для ансамблевих моделей сегментації на основі центрування, алгоритм усереднення форм об'єктів на основі центрування.

Сам процес використання будь-якої з комбінацій виглядає наступним чином. Нехай дано N моделей, які для i -ї матриці пікселів роблять V передбачень, в результаті чого утворюється множина передбачень $P = \{p_1, p_2, p_3 \dots p_N\}$. Множина i -ї комбінації позначена, як $Combination_i = \{m_1, m_2, m_3 \dots m_N\}$, де m_i – це i -тий ансамблевий алгоритм. Кожен ансамблевий алгоритм обробляє множину передбачень і генерує агреговану маску на основі результатів об'єднання цих передбачень:

$$R_i = m_i(P),$$

де R_i – агреговане зображення, що є результатом застосування i -го ансамблевого алгоритму.

Для того щоб отримати результат комбінування необхідно скористатись формулою:

$$I_{res} = f(R_1, R_2, R_3 \dots R_N),$$

де f – алгоритм усереднення, котрий приймає на вхід всі N зображень, отриманих в результаті виконання ансамблевих алгоритмів. В даному випадку в якості f використовувався алгоритм усереднення з порогом $T = 1/2$.

На рисунку 3.12 представлено блок-схему, що ілюструє послідовність виконання етапів алгоритму.

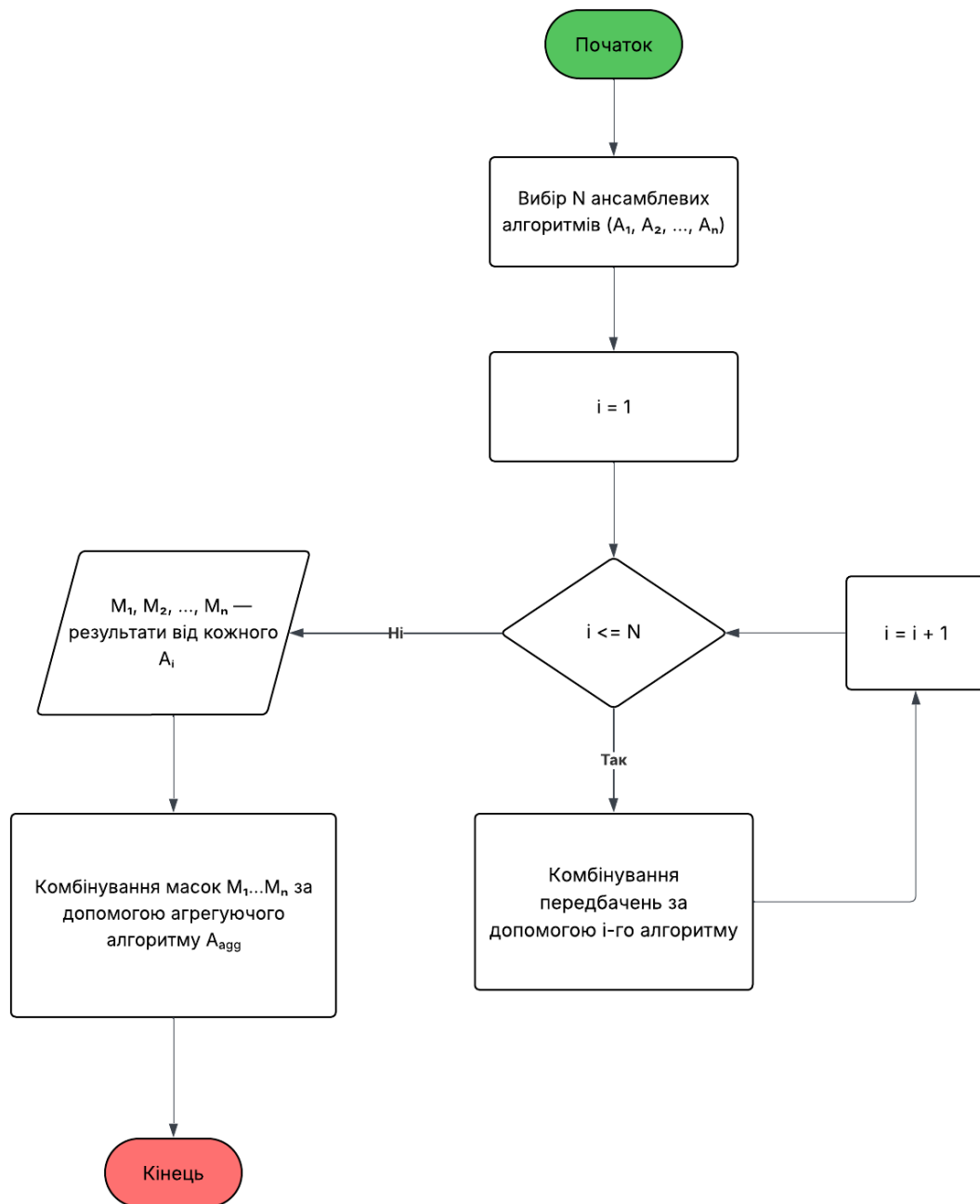


Рис. 3.12. Блок-схема алгоритму

3.5 Оцінка точності та ефективності ансамблевих алгоритмів на різних наборах даних

В цьому підрозділі представлено результати оцінки точності ансамблевих алгоритмів та комбінованих ансамблевих алгоритмів, перевірених на двох наборах даних. Порівняння проводяться за допомогою двох методів: порівняння на основі показника DICE у відсотковому вигляді; порівняння кількості випадків для кожного алгоритму, коли алгоритм показував результати краще для обраної нейронної мережі ніж його конкуренти. Для більш детального вивчення впливу ансамблевих алгоритмів на кінцевий результат, були використані різні варіації кількостей моделей у ансамблях. Мінімальна кількість моделей в ансамблі дорівнювала 2, а максимальна 10. Детальні результати запропонованих ансамблевих алгоритмів та комбінованих ансамблевих алгоритмів приведені в додатку В.

В першу чергу потрібно визначити необхідну кількість моделей в ансамблі для досягнення найкращого результату. Для цього в таблиці 3.5 наведено кількість моделей в ансамблі для кожного ансамблевого алгоритму, завдяки яким можливо отримати найкращий результат.

Застосування ансамблевого алгоритму усереднення з порогом $1/2$ на першому наборі даних продемонструвало такі результати: для досягнення найкращої точності було використано щонайменше 2 моделі та щонайбільше 5 моделей в ансамблі. Середня кількість моделей становила 4. Для другого набору даних оптимальним виявилось використання ансамблів, що містять від 2 до 7 моделей, при цьому середнє значення також дорівнювало 4.

Таблиця 3.5

Кількість моделей в ансамблі, коли було отримано найкращий результат

Мережа	Набір даних	Алгоритми без центрування					Алгоритми з центруванням				
		АУП $T=1/2$	АУП $T=2/3$	АУФО	ААО	АУП $T=1$	АУП $T=1/2$	АУП $T=2/3$	АУФО	ААО	АУП $T=1$
Dilated-UNet	1	5	6	5	2	2	3	3	3	2	2
	2	7	7	10	3	2	7	9	10	3	2
FastUNet	1	5	5	5	3	2	5	3	2	2	2

Продовження таблиці 3.5

FastUNet	2	4	4	2	3	2	3	3	2	4	2
Swin-UNet	1	2	5	2	2	2	5	5	5	2	2
	2	2	2	2	5	2	5	3	5	2	2
UNet 3+	1	5	5	5	2	2	5	3	3	2	2
	2	7	2	2	8	2	5	3	2	7	2
RAUNet	1	5	2	2	2	2	5	3	2	2	2
	2	5	3	2	2	2	3	3	2	2	2
Attention U-Net	1	3	3	3	2	3	5	3	3	2	2
	2	4	4	4	4	2	3	3	4	4	2
Inception	1	3	2	2	6	2	3	3	2	5	2
	2	3	3	3	10	2	3	3	3	9	2
Resnet101 U-Net	1	2	2	2	2	2	3	3	2	2	2
	2	3	2	3	9	2	3	3	3	10	2
Resnet152v2 U-Net	1	3	3	2	2	2	3	3	2	2	2
	2	2	3	2	2	2	3	3	2	2	2
Fcn8 MobileNet	1	4	5	3	2	3	2	2	2	2	2
	2	2	2	2	2	2	4	4	2	2	2

Аналіз отриманих результатів дозволив зробити висновок, що для забезпечення найкращої точності доцільно використовувати моделі, різниця між максимальною та мінімальною точністю яких не перевищує 5 %. Крім того, оптимальна кількість моделей в ансамблі становить 3–4. Поєднання моделей із високою та низькою точністю негативно впливає на кінцевий результат.

Застосування ансамблевого алгоритму усереднення з порогом $2/3$ дало майже подібні результати: для першого набору було достатньо ансамблю з 2–6 моделей, а середня кількість становила 4. Для другого набору ансамблі містили від 2 до 7 моделей, а середнє значення дорівнювало 3.

Цей результат повністю узгоджується з висновками для алгоритму усереднення з порогом $1/2$: найкраща продуктивність досягається при використанні 3–4 моделей із мінімальною різницею точності, що не перевищує 5 %.

Застосування ансамблевого алгоритму усереднення форм об'єктів на першому наборі даних показало: для найкращої точності використовувалися

ансамблі з 2–5 моделей, а середня кількість становила 3. Для другого набору діапазон ансамблів розширився до 2–10 моделей, середнє значення дорівнювало 3.

Аналіз цього підходу вказує на доцільність використання ансамблів із 3 моделей, за умови, що різниця між точністю найкращої та найгіршої моделі не перевищує 3 %. Комбінування моделей із низькою точністю та високоточних моделей у ансамблях є невиправданим, оскільки це не забезпечує суттєвого покращення загальної продуктивності та може призводити до зниження ефективності ансамблю.

Ансамблевий алгоритм об'єднання продемонстрував свою ефективність для моделей із широким діапазоном точності: на першому наборі для досягнення оптимального результату використовувалися ансамблі з 2–6 моделей (середнє значення – 3), а на другому – від 2 до 10 моделей (середнє значення – 5).

Особливістю цього підходу є здатність покращувати результати навіть для моделей із низькою якістю передбачення, зокрема в діапазоні точності 30 %. Оптимальна кількість моделей в ансамблі для цього алгоритму – від 3 до 5.

Застосування алгоритму усереднення з порогом 1 продемонструвало значні обмеження: для першого набору було використано ансамблі з 2–3 моделей (середнє значення – 2), для другого – 2 моделі без варіативності.

Цей алгоритм значно звужує можливості ансамблювання, оскільки вимагає повного збігу всіх елементів складних геометричних структур. Враховуючи такі жорсткі обмеження, його використання не рекомендується.

Виходячи з наведених аргументів, можна зробити такі висновки: для першого набору даних оптимальним є використання ансамблю з 3 моделей, тоді як для другого набору доцільно застосовувати 4 моделі. Збільшення кількості моделей у другому випадку пояснюється нижчою якістю зображень у цьому наборі. Додавання ще однієї моделі до ансамблю сприяє підвищенню точності сегментації.

У таблиці В.11 з додатку В наведено середні значення для ансамблевих алгоритмів для 9 використаних ансамблів, що дає змогу отримати загальну картину. Проаналізувавши ці дані, було ухвалено рішення провести два види порівнянь: порахувати частоту використання кожного ансамблевого алгоритму для нейронних

мереж і порівняти середні значення для кожного з них. Загалом у таблиці представлено 20 випадків порівняння ансамблевих алгоритмів між собою. Кожен розглянутий випадок передбачає визначення найкращого ансамблевого алгоритму для конкретного набору нейронної мережі. Такий підхід обґрунтований тим, що детальний аналіз показав відсутність єдиного універсального алгоритму, який демонстрував би перевагу над іншими для всіх наборів даних і всіх архітектур нейронних мереж.

Розглянемо алгоритм усереднення з порогом $1/2$, який показав найкращий результат у 7 випадках із 20. Застосування цього алгоритму дало найкращі результати для нейронних мереж Dilated-UNet (обидва набори), UNet 3+ (перший набір), RAUNet (перший набір), Attention U-Net (перший набір), Resnet101 U-Net (перший набір) та Resnet152v2 U-Net (перший набір).

Алгоритм усереднення з порогом $2/3$ продемонстрував найкращий результат лише в одному випадку — для мережі Fcn8 MobileNet (перший набір).

Запропонований алгоритм усереднення форм об'єктів показав 3 позитивні результати для FastUNet (перший набір), Swin-Unet (перший набір) та Inception (перший набір).

Алгоритм об'єднання демонстрував найкращі результати виключно для другого набору даних, зокрема для мереж FastUNet, Swin-Unet, UNet 3+, RAUNet, Attention U-Net, Inception та Resnet101 U-Net.

Крім того, у 2 випадках позитивні результати були отримані завдяки запропонованому алгоритму усереднення форм об'єктів із центруванням, що покращило результати на другому наборі даних для мереж Resnet152v2 U-Net та Fcn8 MobileNet.

Отримані дані свідчать про те, що запропоновані автором алгоритми перевершили конкурентні підходи у 5 випадках із 20. Найкращі результати для другого набору даних показав алгоритм об'єднання.

Після порівняння кількості позитивних випадків для ансамблевих алгоритмів можна зіставити їхні середні значення. Ці результати наведені у таблиці 3.6. Для першого набору даних найкращим алгоритмом виявився алгоритм усереднення з

порогом $1/2$, а запропонований алгоритм усереднення форм об'єктів посів друге місце. Для другого набору найефективнішим став алгоритм об'єднання, тоді як авторський алгоритм знову зайняв другу позицію.

Таблиця 3.6

Порівняння середніх значень для ансамблевих алгоритмів, DICE %

Набір даних	Алгоритми без центрування					Алгоритми з центруванням				
	АУП $T=1/2$	АУП $T=2/3$	АУФО	ААО	АУП $T=1$	АУП $T=1/2$	АУП $T=2/3$	АУФО	ААО	АУП $T=1$
1	90.57	88.50	89.59	87.34	83.27	89.49	88.80	89.51	86.83	82.90
2	65.28	60.18	66.47	71.20	50.62	65.61	64.02	68.08	69.45	52.20

У таблиці В.12 з додатку В наведені максимальні результати для кожного ансамблевого алгоритму. З відповідних таблиць нейронних мереж В.1-В.10, де порівнювались ансамблеві алгоритми для 9 ансамблів, для кожного алгоритму було обрано найкращий результат з 9 ансамблів, після чого ці результати були занесені в таблицю.

Знову порівнюємо спочатку кількість найкращих результатів для кожного ансамблевого алгоритму, а потім їх середні значення. Алгоритм усереднення з порогом $1/2$ показав найкращий результат серед інших алгоритмів у 4 випадках із 20. Алгоритм усереднення форм об'єктів показав 2 найкращі результати. Алгоритм об'єднання був найкращим 12 разів. Алгоритми усереднення з порогом 1 і об'єднання на основі центрування продемонстрували по 1 позитивному випадку.

Порівняння середніх значень, отриманих у таблиці В.12 з додатку В, було занесено до таблиці 3.7. Переглянувши середні значення, можна зробити висновок, що для першого набору даних найкращий результат був отриманий алгоритмом усереднення з порогом $1/2$, авторський алгоритм зайняв третє місце з різницею 0.0075 від другого місця. Для другого набору даних найкращим алгоритмом став алгоритм об'єднання, а авторський алгоритм усереднення форм об'єктів посів друге місце.

Таблиця 3.7

Порівняння максимальних значень для ансамблевих алгоритмів, DICE %

Набір даних	Алгоритми без центрування					Алгоритми з центруванням				
	АУП $T=1/2$	АУП $T=2/3$	АУФО	ААО	АУП $T=1$	АУП $T=1/2$	АУП $T=2/3$	АУФО	ААО	АУП $T=1$
1	91.38	89.94	90.83	90.84	89.48	90.62	90.43	90.52	90.24	89.22
2	68.21	64.42	69.23	73.97	65.28	69.00	68.94	70.16	72.60	65.65

З отриманих результатів видно, що хоча авторські алгоритми і мають своє місце для застосування, а саме для досягнення найкращого результату в 8 випадках необхідно застосовувати ці алгоритми, проте їх відставання від алгоритму об'єднання або алгоритму усереднення з порогом $1/2$ є помітним. Також незаперечним фактом є те, що неможливо визначити універсальний ансамблевий алгоритм, який буде показувати найкращі результати завжди. Для вирішення такої проблеми було запропоновано використовувати комбінації ансамблевих алгоритмів, щоб підвищити точність окремих алгоритмів завдяки їх використанню в групі.

У таблиці В.23 з додатку В наведені результати порівняння середніх значень для комбінацій кожної нейронної мережі, отриманих з таблиць В.13-В.22 з додатку В, з найкращими конкурентними результатами, отриманими в таблиці В.11 з додатку В.

Після вивчення результатів цієї таблиці було зроблено наступні висновки: використання комбінованих алгоритмів показало найкращий результат у 17 випадках із 20. Застосування комбінованих алгоритмів дозволило отримати значення DICE на 0.79 % вище, ніж у конкурентних алгоритмів. Але якщо розглядати комбінації окремо та знайти комбінацію, яка буде найкращою для всіх нейронних мереж, ситуація виглядає дещо іншою. Для першого набору даних не вдалося отримати найкращу комбінацію, і найкращі результати продемонстрували конкурентні алгоритми. Однак для другого набору застосування авторської комбінації 4 дало найкращий результат порівняно з конкурентними алгоритмами.

Використання 4-ї комбінації підтверджує доцільність її використання для зображень низької якості, для яких власне й були запропоновані всі ці алгоритми. Саме загальне застосування запропонованих методик і дає змогу підвищити точність ідентифікації складних геометричних структур з другого набору даних.

Таблиця 3.8

Порівняння комбінацій та найкращих результатів ансамблевих алгоритмів,
DICE %

Набір даних	Конкурентний ансамблевий алгоритм	Комбінація						
		1	2	3	4	5	6	7
1	90.70	90.57	90.54	87.29	87.34	85.67	89.94	85.61
2	71.48	65.28	68.54	71.15	72.05	70.27	68.67	71.48

У таблиці 3.8 наведено порівняння середніх значень конкурентних алгоритмів і середніх значень комбінацій для пошуку найкращого алгоритму для всіх нейронних мереж.

Таблиця 3.9

Порівняння ансамблевих алгоритмів без центрування та з центруванням,
DICE %

Набір даних	Ансамблевий алгоритм				
	АУП $T=1/2$	АУП $T=2/3$	АУФО	ААО	АУП $T=1$
1	-1.09	+0.29	-0.08	-0.52	-0.36
2	+0.33	+3.84	+1.61	-1.75	+1.58

У таблиці 3.9 порівнюються результати ансамблевих алгоритмів без центрування та на основі центрування з таблиці 3.6. Ансамблеві алгоритми на основі центрування не показали найкращих результатів під час порівнянь в таблицях В.11 та В.12. Тому необхідно детальніше вивчити доцільність їх

використання для інтеграції в комбінації. Для цього запропоновано додати до таблиці наступні дані: $X_i = E_{center,i} - E_i$, де $E_{center,i}$ — це значення i -го ансамблевого алгоритму на основі центрування, а E_i — значення i -го алгоритму без центрування. Негативні значення свідчать про те, що застосування алгоритму центрування знижує точність сегментації, а позитивні — про покращення точності сегментації.

З результатів наведених в таблиці видно, що було розглянуто 10 випадків, з яких у 5 випадках застосування алгоритму центрування погіршувало точність, а в 5 випадках покращувало. Для другого набору даних у 4 з 5 випадків застосування алгоритму центрування дозволило підвищити точність. Один негативний випадок для другого набору даних був отриманий за допомогою ансамблевого алгоритму об'єднання, що вказує на те, що поєднання алгоритму об'єднання та алгоритмів центрування не є оптимальним, оскільки ансамблевий алгоритм об'єднання має властивість переносити всі пікселі передбачень на агреговану маску.

Якщо підсумувати всі результати, то для другого набору даних завдяки застосуванню ансамблевих алгоритмів на основі центрування точність можна підвищити на 1.84 % в середньому. Таким чином, порівнюючи ансамблеві алгоритми без центрування та на основі центрування, можна стверджувати, що їх використання може покращити результати при правильному підборі алгоритму для конкретної мережі. Для універсального використання рекомендується застосовувати ансамблеві алгоритми на основі центрування в комбінованих підходах.

При порівнянні результатів наведених у таблицях В.1–В.10, стало зрозумілим, що із 180 випадків у 158 випадках кращими були алгоритми без центрування, а 22 випадки отримали кращий результат завдяки використанню алгоритму центрування. На рис. 3.13 можна побачити кількість позитивних випадків для кожного алгоритму, коли результат алгоритму був кращим за середнє значення окремих моделей. Алгоритм усереднення з порогом $T=1/2$ отримав 54 позитивних випадків, тим самим він зміг продемонструвати узагальнюючу здатність. Половинний поріг дозволяє враховувати більше пікселів. Під час

використання алгоритму усереднення з порогом, при спробі збільшення порогу до $2/3$, кількість позитивних результатів суттєво зменшилась до 11. Це результат більш суворого порогового критерію, при якому необхідно враховувати лише найбільш схожі регіони. Алгоритм усереднення з порогом $T=1$ є найбільш суворим, цей алгоритм отримав лише 3 позитивних випадків, що свідчить про занадто обмежувальний сенс використання для більшості практичних задач. За допомогою алгоритму усереднення форм об'єктів вдалося отримати 15 позитивних випадків, що свідчить про його ефективність в задачах, де геометрична структура об'єктів більш-менш схожа. Ансамблевий алгоритм об'єднання забезпечив найбільшу кількість позитивних результатів – 75. Такий результат вдалося отримати в першу чергу за рахунок другого набору даних, тому що цей підхід максимально охоплює всі сегментовані регіони.

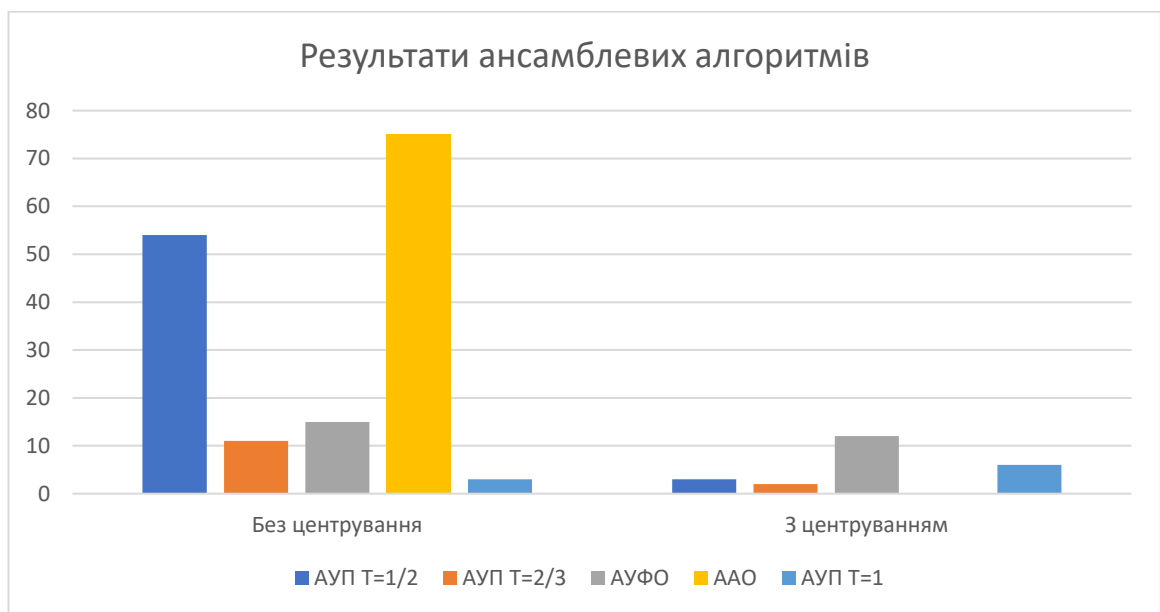


Рис. 3.13. Позитивні випадки ансамблевих алгоритмів для всіх моделей

Алгоритми, які використовують центрування показали менший результат в порівнянні з алгоритмами без центрування. Наприклад для алгоритму усереднення з порогом кількість результатів зменшилась з 54 до 3, а для алгоритму об'єднання взагалі стала 0. Водночас для алгоритму усереднення форм об'єктів центрування позитивно вплинуло на результат, де кількість позитивних випадків 15 і 12

відповідно. Це вказує на те, що використання алгоритму центрування найкраще працює в контексті алгоритмів, для яких важливо отримати чітку узгодженість між масками. З отриманих результатів, стає зрозумілим, що ефективність застосування алгоритму центрування залежить від типу ансамблевого алгоритму.

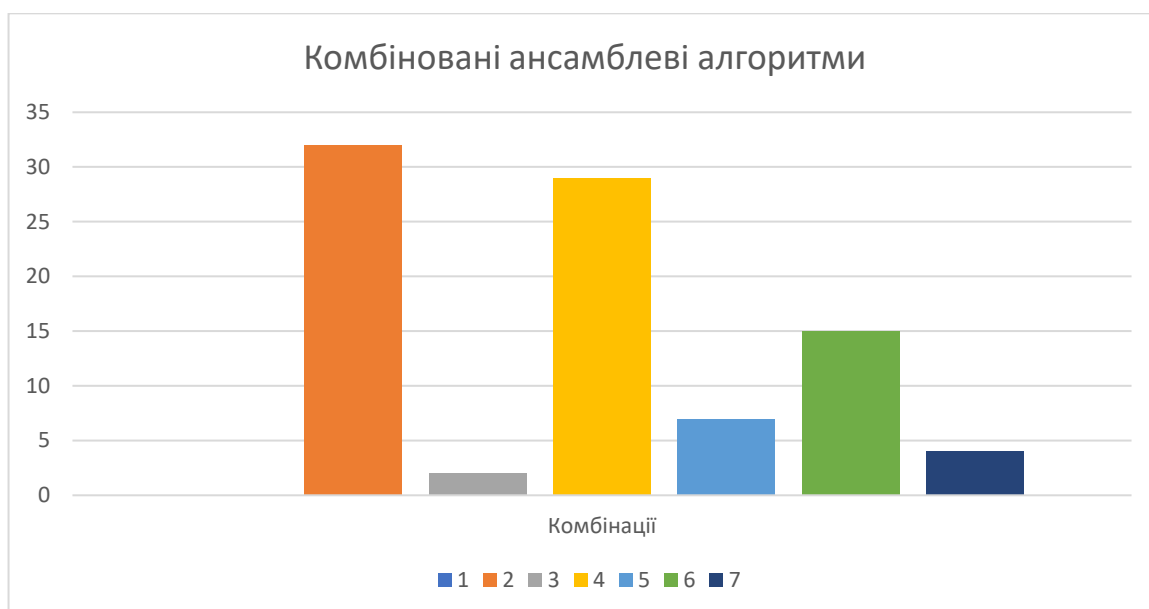


Рис. 3.14. Позитивні випадки комбінованих ансамблевих алгоритмів для всіх моделей, які краще звичайних ансамблевих алгоритмів

На рисунку 3.14 приведена кількість позитивних випадків для кожного комбінованого алгоритму, коли значення комбінованого алгоритму було кращим за максимальне значення конкурентного ансамблевого алгоритму. Як можна побачити, в 89 випадках із 180, комбіновані алгоритми показали кращий результат в порівнянні із звичайними ансамблевими алгоритмами.

Найкращий результат продемонструвала комбінація 2, вона отримала 32 позитивних випадки. Це свідчить про доцільність комбінації алгоритмів усереднення з порогом з різними порогами та алгоритмами на основі центрування. Результати комбінації 2, доводять здатність цього алгоритму вирішувати більшість задач сегментації, як для першого набору так, і для другого. Комбінація 4 також мала змогу продемонструвати конкурентний результат – 29, що свідчить про вдале використання алгоритму усереднення форм об'єктів та алгоритму об'єднання. Інші

алгоритми мають нижчі показники, тим не менш, вони показують результати кращі ніж ансамблеві алгоритми окремо.

Підсумувавши результати, можна констатувати наступне, після проведених експериментів на двох наборах даних, стало зрозуміло, що не існує універсального ансамблевого алгоритму, який би однаково покращував точність для всіх моделей. Результати варіювалися в залежності від конкретної нейронної мережі, що свідчить про відсутність єдиного алгоритму, здатного демонструвати постійну перевагу у всіх випадках.

Однак, застосування комбінацій ансамблевих алгоритмів виявилось ефективним для покращення точності на другому наборі даних. Це підтверджує доцільність комбінованих підходів, які дозволяють підвищити точність ідентифікації складних геометричних структур, особливо для моделей, що працюють з даними низької якості. Таким чином, результати вказують на те, що для досягнення кращих результатів необхідно використовувати комбіновані ансамблеві алгоритми. Це необов'язково повинні бути комбінації ансамблевих алгоритмів, це може бути комбінація ансамблевих алгоритмів використана для багатоетапного алгоритму сегментації.

Таким чином можна зробити загальний висновок, що сегментація низькоякісних зображень є дуже складним процесом і використовувати тільки один алгоритм є поганим підходом. Для вирішення проблеми ідентифікації складних геометричних структур на низькоякісних зображень обов'язково необхідно застосовувати багатоетапні послідовності різних алгоритмів. Тільки завдяки об'єднанню декількох алгоритмів маємо змогу значно покращити результат, що саме й буде доведено в наступних підрозділах, коли буде використано багатоетапний алгоритм сегментації з використанням ансамблевих алгоритмів.

3.6 Алгоритм адаптивного уточнення контуру на основі яскравості сусідніх пікселів

Методи сегментації зображень значною мірою залежать від точності визначення контурів складних геометричних структур. Це завдання особливо

актуальне для аналізу зображень у складних сферах, де точність є найголовнішим фактором. Наприклад, медична діагностика, робототехніка, системи автоматичного розпізнавання складних геометричних структур. Один із запропонованих підходів — алгоритм адаптивного уточнення контуру, який базується на аналізі яскравості сусідніх пікселів [67]. Цей алгоритм дозволяє коригувати попередньо визначені контури таким чином, щоб вони максимально точно описували краї об'єкта, використовуючи локальну інформацію про яскравість зображення.

Контур є однією з ключових характеристик об'єкта, що дозволяє визначити його форму, розміри, положення та інші параметри. Проте, часто початкові контури, отримані з використанням стандартних методів сегментації, таких як порогові алгоритми, метод Кенні, чи активні контури, виявляються неточними через низький контраст, наявність шумів або складну структуру зображення.

У багатьох реальних завданнях, наприклад, у медичній діагностиці, навіть незначна похибка у визначенні контуру може призвести до серйозних наслідків, таких як неправильний діагноз або некоректна оцінка розмірів патологічного утворення. Подібна ситуація має місце і в інших областях, зокрема в аналізі супутникових знімків або у виробничих системах, де важлива максимальна точність. Саме для таких випадків використовується алгоритм адаптивного уточнення контурів, що забезпечує можливість коригування меж об'єкта на основі локальних змін яскравості сусідніх пікселів.

Одним із наявних прикладів підходу до сегментації завдяки уточненню контурів об'єктів є SharpContour [68], який вирішує проблему неточної обробки кордонів об'єктів. Сегментація текстур, на основі матриці сірих рівнів співвідношень, використовує такі властивості, як контраст та однорідність для класифікації областей зображення [69]. Модель адаптивного контуру, яка базується на локальному налаштуванні, для медичних зображень дозволяє ідентифікувати межі об'єктів, враховуючи їх локальні особливості [70]. Алгоритм адаптивного коригування меж для багатомасштабної сегментації зображень дистанційного зондування, зосереджений на вдосконаленні точності визначення меж об'єктів шляхом адаптивного аналізу локальних характеристик [71]. Метод POLYCORE,

який використовує полігональний підхід для уточнення контурів у задачах сегментації інтраваскулярного ультразвуку забезпечує покращення точності сегментації судин шляхом адаптивного коригування контурів [72]. Підхід до сегментації зображень на основі активних контурів із корекцією зсуву [73] теж спрямований на подолання проблем, пов'язаних із неоднорідністю яскравості зображень, що часто ускладнює точну сегментацію. Результати отримані під час дослідження цих робіт довели, що використання адаптивних методів для уточнення контуру за рахунок локальних характеристик мають значний потенціал. Запропонований нижче алгоритм буде використовувати комбінацію алгоритму адаптивного уточнення контурів та ансамблевого алгоритму. Цей алгоритм базується на послідовному процесі у якому кожна точка контуру перевіряється на відповідність локальним умовам яскравості.

Цей алгоритм використовує ядро розміром $k \times k$, яке своїм центром накладається на елемент контуру, після чого відбувається рядкове порівняння значень яскравості елемента контуру та сусідніх пікселів, якщо яскравість сусіднього пікселя більша, то активний елемент контуру замінюється на нове значення, з координатами де був знайдений новий елемент, інакше елемент залишається частиною контуру. Було розглянуто два варіанти використання розмірів ядра: 3×3 і 5×5 . Візуалізацію використовуваних ядер можна побачити на рисунку 3.15. Зліва зображено ядро 3×3 , а праворуч 5×5 . Темний сірий квадрат – це точка контуру, до якої застосовується алгоритм, а сусідні світло-сірі квадрати – це сусіди, яких необхідно проаналізувати.

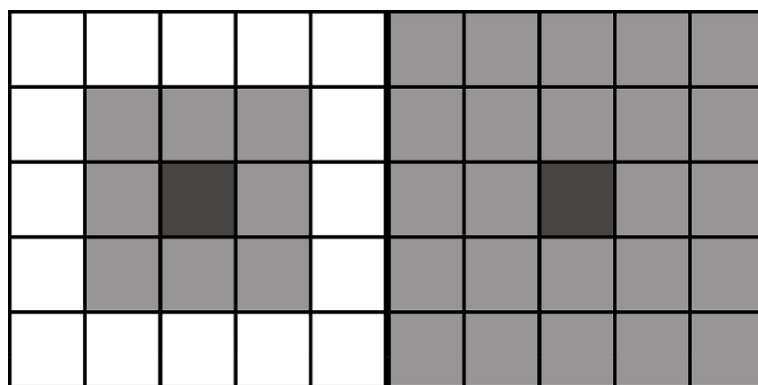


Рис. 3.15. Приклад використаних ядер

Нехай у нас є контур $C = \{(x_i, y_i)\}_{i=1}^N$ який був отриманий в результаті використання I_{res} , і є набором точок. Для кожної точки контуру (x_i, y_i) використовується область аналізу з розміром $k \times k$. Межі області аналізу для точки (x_i, y_i) задаються за формулою:

$$x_{min} = \max\left(0, x_i - \frac{k}{2}\right), x_{max} = \min\left(W - 1, x_i + \frac{k}{2}\right),$$

та

$$y_{min} = \max\left(0, y_i - \frac{k}{2}\right), y_{max} = \min\left(H - 1, y_i + \frac{k}{2}\right),$$

де W та H – кількість рядків і стовпців у матриці пікселів відповідно.

Пошук елемента зображення який яскравіший за поточний елемент контуру відбувається наступним чином. Для кожної точки (x_i, y_i) необхідно виконати перевірку інтенсивності елементів зображення у заданому вікні аналізу $(x, y) \in [x_{min}, x_{max}] \times [y_{min}, y_{max}]$. Якщо для точки (x_i, y_i) в результаті аналізу елементів зображення було знайдено елемент зображення (x_j, y_j) такий що:

$$I(x_j, y_j) > I(x_i, y_i) \text{ та } (x_j, y_j) \notin C,$$

тоді точка (x_j, y_j) додається в C_{new} , в іншому випадку в C_{new} додається (x_i, y_i) .

Після оновлення всіх точок, що належать контуру, формується контур C_{new} , який вже містить більш точні межі об'єкту. Наступним етапом цього алгоритму є використання ансамблевого алгоритму. Для досягнення кращих результатів цей алгоритм складається з двох частин: перша – адаптивне уточнення пікселів, друга – використання ансамблевого алгоритму. Для другого етапу можна використовувати будь-який ансамблевий алгоритм. В цій роботі для визначення ефективності цього алгоритму будуть використані два ансамблеві алгоритми.

Рисунок 3.16 демонструє поетапну структуру виконання алгоритму у вигляді

блок-схеми.



Рис. 3.16. Блок-схема алгоритму

На рисунку 3.17 наведено приклад використання алгоритму для передбачення поганої якості. В даному випадку модель видала погане передбачення, замість чотирьох хребців вона частково визначила тільки два. Червоним кольором було позначено контур об'єкта. В цьому прикладі до одного елемента контуру було застосовано ядро розміром 3×3 , яке можна побачити на зображенні та на яке вказує стрілка. В центрі ядра, зеленим колом позначено центр ядра, чорною рамкою обведено саме ядро.

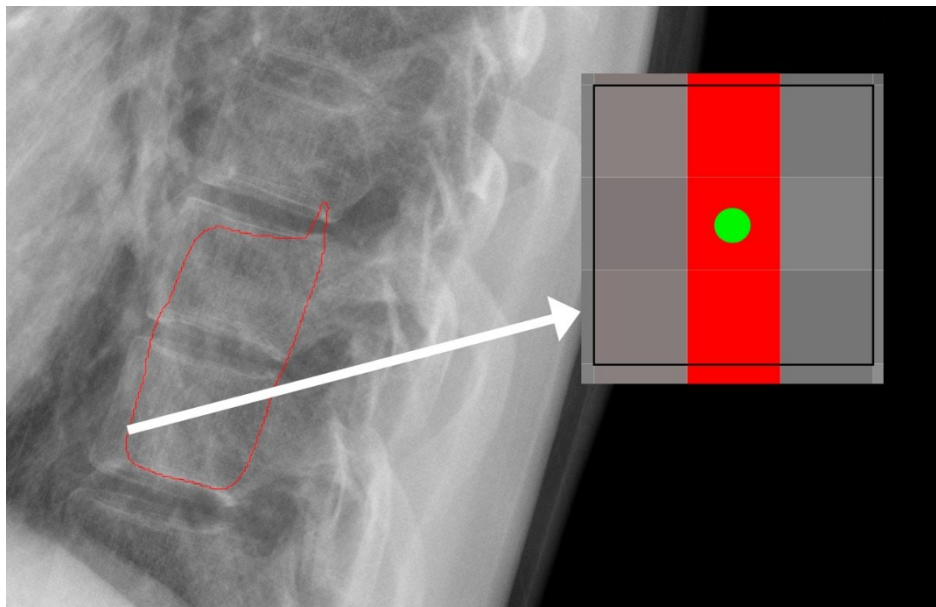


Рис. 3.17. Етап порівняння пікселів

На рисунку 3.18 можна побачити два контури, білий – це контур отриманий з передбачення, зелений – це контур, після застосування адаптивного алгоритму та ансамблевого алгоритму усереднення з порогом $T=1/2$.

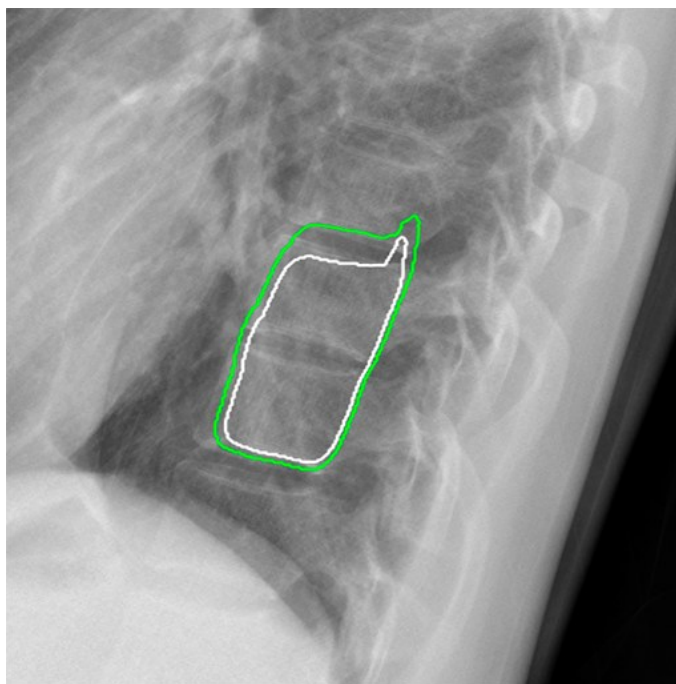


Рис. 3.18. Готовий результат адаптивного алгоритму

Для більш ретельного та детального вивчення алгоритму було запропоновано розглянути три варіанти цього алгоритму:

1. Сусідній елемент зображення зовні або всередині контуру та його інтенсивність більша за поточну.
2. Сусідній елемент зображення зовні контуру та його інтенсивність більша за поточну.
3. Сусідній елемент зображення усередині контуру та його інтенсивність більша за поточну.

Для оцінки продуктивності цього алгоритму було використано запропоновані моделі з підрозділу (3.1). В якості ансамблів використовувались ансамблі, які складались з 10 моделей. В таблиці 3.10 наведено результати використання адаптивного алгоритму без застосування ансамблевого алгоритму для десяти моделей.

Таблиця 3.10

Результати адаптивного алгоритму, DICE %

Модель	Набір	Середнє для окремих моделей	Метод та ядро						Покращення
			V1, $k=3$	V1, $k=5$	V2, $k=3$	V2, $k=5$	V3, $k=3$	V3, $k=5$	
Dilated-UNet	1	95.41	95.15	94.35	95.26	94.78	95.26	94.86	-
	2	79.97	79.85	79.40	79.72	79.34	80.07	80.08	0.11
FastUNet	1	89.25	88.98	88.25	89.15	88.77	89.07	88.68	-
	2	64.63	68.91	68.42	69.03	68.84	68.99	68.77	4.4
Swin-Unet	1	82.96	82.73	82.10	82.83	82.56	82.85	82.56	-
	2	64.00	64.91	64.67	65.11	65.19	64.76	64.51	1.19
UNet 3+	1	88.02	87.86	87.30	88.13	87.95	87.77	87.31	0.11
	2	57.59	64.43	64.03	64.70	64.71	64.32	63.98	7.12
RAUNet	1	89.12	88.97	88.34	89.17	88.90	88.91	88.48	0.05
	2	61.72	65.23	64.81	65.47	65.46	65.10	64.76	3.75
Attention U-Net	1	84.51	84.31	83.63	84.73	84.61	84.13	83.49	0.22
	2	49.00	58.29	57.88	58.65	58.73	58.11	57.65	9.73
Inception	1	80.08	79.87	79.59	80.64	80.86	79.52	78.93	0.78
	2	52.75	53.54	53.59	54.58	55.18	52.83	52.33	2.43

Продовження таблиці 3.10

Resnet101 U-Net	1	83.05	83.39	82.64	83.80	83.67	83.25	82.62	0.75
	2	47.69	61.97	61.50	62.23	62.16	61.92	61.56	14.54
Resnet152v2 U-Net	1	82.19	83.50	82.92	83.74	83.51	83.50	83.09	1.55
	2	51.91	64.07	63.72	64.22	64.08	64.07	63.89	12.31
Fcn8 MobileNet	1	89.96	89.61	88.83	89.58	89.00	89.93	89.71	-
	2	68.62	69.47	69.02	69.46	69.15	69.66	69.59	1.04

В таблиці 3.10 наведено 20 випадків для двох наборів даних, відповідно по 10 на кожний набір. У деяких випадках (чотирьох) із першим набором даних можна прослідити, що застосування даного алгоритму тільки погіршує точність. В інших же випадках покращення незначне. Але розглядаючи такі моделі, як Inception, Resnet101 U-Net, Resnet152v2 U-Net можна побачити, що в їх випадках саме використання запропонованого алгоритму дало покращення. Як можна побачити з загальної картини лідирує другий варіант, з ядром 3×3 , де вдалось покращити результати в 9 випадках, а з ядром 5×5 в 5 випадках. Для другого варіанту, разом було отримано 14 випадків покращення, що свідчить про його потенціал у використанні для зображень з поганими результатами. Для цих випадків середнє покращення склало 0.59 % в порівнянні із середнім значенням окремих моделей. У випадку із Resnet101 U-Net вдалося підвищити результат на 14.53 %. Отримані результати свідчать про доцільність використання цього алгоритму, оскільки за допомогою адаптивного алгоритму значення коефіцієнта DICE зросло на 3.75 %, але для покращення поганих результатів, точність яких знаходиться в діапазоні [40-80] %.

В таблицях 3.11, 3.12 порівнюються результати комбінування адаптивного алгоритму із ансамблевим алгоритмом та результати використання ансамблевого алгоритму.

Отримані результати в таблицях свідчать про покращення точності ідентифікації складних геометричних структур за допомогою адаптивного алгоритму уточнення контуру та ансамблевих алгоритмів особливо для другого набору даних. Саме завдяки комбінації адаптивного алгоритму та ансамблевого,

маємо можливість покращити якість сегментації передбачень поганої якості. Але використовуючи цей алгоритм, маємо можливість покращення не тільки для передбачень поганої якості, а також, і для добре передбачених масок. Для комбінованого алгоритму найбільше кращих результатів було отримано за допомогою третього алгоритму – 12 випадків покращень. На другому місці другий варіант та його 8 покращень. Особливо виділяються деякі випадки, котрі продемонстрували найкращий результат. Самий найкращий результат отримала Resnet101 U-Net для другого набору даних – 14.54 % для безансамблевого алгоритму. Використання даного алгоритму може покращити точність сегментації зображень для моделей, які видають поганий результат, а також цей алгоритм в комбінації з ансамблевим підвищує точність.

Завдяки використанню даного алгоритму, покращення для неансамблевого підходу склало 3.75 %, для ансамблевого алгоритму усереднення з порогом $T=1/2$ покращення склало 1.43 %, для ансамблевого алгоритму об'єднання 0.29 %.

Таблиця 3.11

Результати адаптивного алгоритму в комбінації з ансамблевим алгоритмом усереднення з порогом $T=1/2$, DICE %

Модель	Набір	Анс. алгоритм	Метод та ядро						Покращення
			V1, $k=3$	V1, $k=5$	V2, $k=3$	V2, $k=5$	V3, $k=3$	V3, $k=5$	
Dilated-UNet	1	95.98	95.85	95.01	95.86	95.31	95.94	95.60	-
	2	81.64	81.68	81.23	81.48	81.04	81.94	82.00	0.36
FastUNet	1	90.81	90.99	90.18	91.09	90.57	91.11	90.74	0.30
	2	69.96	71.59	71.12	71.64	71.41	71.71	71.55	1.75
Swin-UNet	1	84.34	83.98	83.36	83.99	83.62	84.17	83.97	-
	2	66.62	67.44	67.26	67.60	67.69	67.29	67.06	1.07
UNet 3+	1	90.66	90.87	90.29	91.07	90.87	90.76	90.30	0.41
	2	61.74	65.61	65.15	65.82	65.81	65.57	65.22	4.07
RAUNet	1	91.81	92.24	91.48	92.42	92.04	92.23	91.78	0.61
	2	64.67	66.15	65.72	66.37	66.36	65.98	65.62	1.70
Attention U-Net	1	88.39	88.77	88.02	89.15	89.03	88.57	87.87	0.76

Продовження таблиці 3.11

Attention U-Net	2	48.05	51.23	50.90	51.60	51.75	51.01	50.57	3.70
Inception	1	81.90	82.92	82.58	83.60	83.79	82.58	82.01	1.89
	2	55.82	54.54	54.68	55.60	56.23	53.79	53.26	0.41
Resnet101 U-Net	1	89.07	90.10	89.14	90.47	90.25	89.93	89.18	1.40
	2	50.00	52.28	51.91	52.47	52.41	52.23	51.95	2.47
Resnet152v2 U-Net	1	94.70	94.69	93.91	94.87	94.40	94.79	94.33	0.17
	2	57.23	59.45	59.12	59.57	59.48	59.38	59.20	2.34
Fcn8 MobileNet	1	92.28	91.79	90.95	91.62	90.90	92.19	92.07	-
	2	73.58	74.16	73.66	74.04	73.63	74.44	74.44	0.86

Таблиця 3.12

Результати адаптивного алгоритму в комбінації з ансамблевим алгоритмом
об'єднання, DICE %

Модель	Набір	Анс. алгоритм	Метод та ядро						Покращення
			V1 $k=3$	V1 $k=5$	V2 $k=3$	V2 $k=5$	V3 $k=3$	V3 $k=5$	
Dilated-UNet	1	94.80	94.31	93.54	94.14	93.31	94.93	94.99	0.19
	2	79.65	79.28	78.78	78.98	78.27	79.96	80.28	0.63
FastUNet	1	86.44	85.93	85.23	85.72	84.95	86.59	86.70	0.26
	2	72.41	72.09	71.61	71.92	71.36	72.58	72.72	0.31
Swin-UNet	1	79.70	79.35	78.80	79.22	78.64	79.81	79.85	0.15
	2	67.95	67.76	67.50	67.63	67.23	68.11	68.26	0.31
UNet 3+	1	89.01	88.62	87.88	88.49	87.74	89.10	89.18	0.17
	2	74.54	74.31	73.84	74.18	73.67	74.66	74.73	0.19
RAUNet	1	87.89	87.49	86.84	87.26	86.44	88.09	88.25	0.36
	2	73.48	73.21	72.72	73.13	72.67	73.57	73.59	0.11
Attention U-Net	1	86.66	86.24	85.57	86.07	85.28	86.80	86.90	0.24
	2	69.77	69.54	69.06	69.44	69.03	69.86	69.86	0.09
Inception	1	79.59	79.29	78.85	79.01	78.59	79.64	79.47	0.05
	2	66.43	66.55	66.60	66.68	66.84	66.27	66.16	0.41
Resnet101 U-Net	1	85.45	85.03	84.32	84.80	84.02	85.63	85.73	0.28
	2	65.74	65.41	64.90	65.28	64.77	65.85	65.91	0.17
Resnet152v2 U-Net	1	69.12	68.81	68.41	68.32	67.76	69.44	69.67	0.55
	2	59.11	58.91	58.61	58.60	58.18	59.33	59.52	0.41

Продовження таблиці 3.12

Fcn8 MobileNet	1	80.96	80.47	79.96	80.12	79.29	81.26	81.55	0.59
	2	66.62	66.38	66.06	66.06	65.48	66.91	67.22	0.29

3.7 Двоетапний ансамблевий алгоритм сегментації зображень

Процес сегментації зображень є доволі складним завданням в області комп'ютерного зору, особливо коли необхідно виконати сегментацію з складними формами, використовуючи зображення низької якості. Присутність шумів та артефактів додає значної складності до цього процесу. Для вирішення цих проблем було запропоновано двоетапний ансамблевий алгоритм сегментації, який дозволяє досягти більш високої точності за рахунок використання ансамблевих алгоритмів на двох різних етапах, що фактично дає змогу зменшити регіон інтересу, завдяки чому точність сегментації покращується. Алгоритм передбачає два етапи: на першому етапі виконується сегментація на зображенні оригінального розміру; на другому етапі для сегментації використовується уточнена локальна область, що концентрується навколо центра знайденої складної геометричної структури на попередньому кроці [74].

Основною ідеєю використання двоетапного підходу є вирішення проблем пов'язаних з сегментацією зображень низької якості або використання нових даних. Під час порівняння результатів ансамблевих алгоритмів для двох наборів даних, було доведено, що на нових або складних даних, результати гірші. Використання ансамблевого алгоритму на першому етапі дає змогу отримати попередню сегментацію з високим рівнем узагальнення, тоді як на другому етапі, зменшення регіону інтересу дозволяє зосередитись на деталях складної геометричної структури ідентифікація якої виконується. Завдяки використанню такого підходу вдається зменшити кількість помилок при сегментації.

Основною частиною всього навчання, як і завжди є дані, на яких буде проходити навчання. Від їх якості й кількості буде залежати кінцевий результат і успіх [75]. Як завжди спочатку потрібно провести тренування моделей. Для тренування моделей на першому етапі використовувався перший набір із підрозділу

3.1. Для навчання на другому етапі також використовувався перший набір, але зображення були модифіковані (виокремлені регіони інтересу). Всі зображення мали розміри 512×512 пікселів та були виконані в градації сірого. В якості нейронних мереж було запропоновано використовувати такі архітектури: Mobilenet Segnet, Mobilenet U-Net [76]. Завдяки використанню цих двох мереж мали змогу швидко навчити необхідні моделі та отримати хороший результат [77]. Для кожної архітектури було виконано навчання 10 моделей. Для оцінки продуктивності було використано другий набір даних. На рисунку 3.19 відображено приклад зображень для двох етапів.

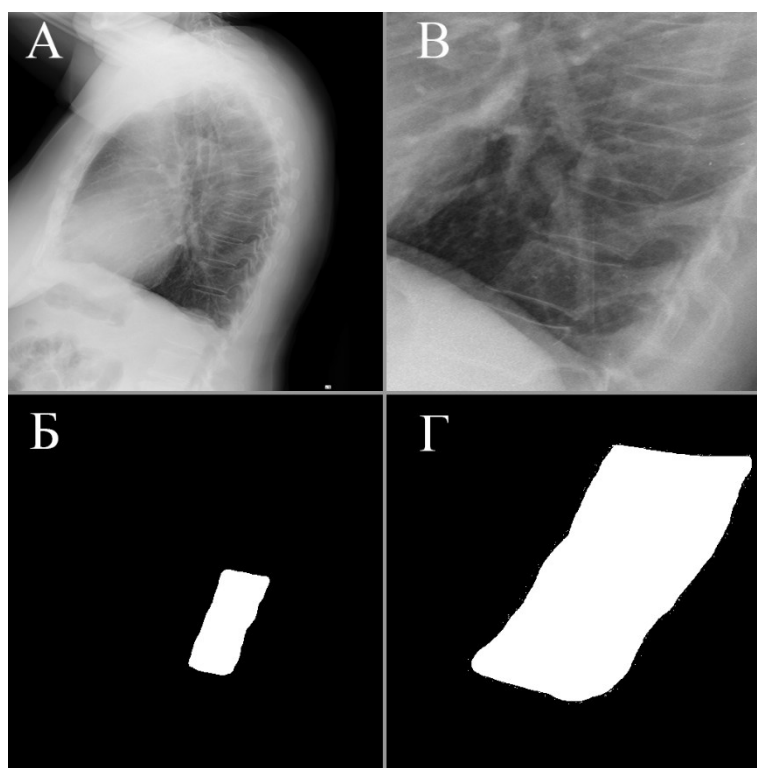


Рис. 3.19. Приклад зображень для тренування моделей. А – зображення для першого етапу, Б – маска для першого етапу, В – зображення для другого етапу, Г – маска для другого етапу

На першому етапі виконується сегментація всього вхідного зображення за допомогою будь-якого ансамблевого алгоритму (в даному випадку використовувався алгоритм усереднення з порогом $T=1/2$). Як результат використання ансамблевого алгоритму на першому етапі створює агреговану маску, яка в подальшому буде використана для другого етапу. На другому етапі необхідно

провести локальне уточнення сегментації. Де в якості локального уточнення виступає вирізаний регіон з першого етапу, який включає складну геометричну структуру, яку необхідно ідентифікувати та регіон навколо неї. В якості ансамблевого алгоритму для другого етапу було використано комбінацію 4.

Спершу потрібно знайти центр складної геометричної структури, яка була отримана на першому етапі. Координати центральної точки (x_c, y_c) можна обчислити за формулами:

$$x_c = \frac{x_{min} + x_{max}}{2}, \quad y_c = \frac{y_{min} + y_{max}}{2},$$

де (x_{min}, y_{min}) – це координати верхнього лівого кута об'єкта, а (x_{max}, y_{max}) – це координати нижнього правого кута.

Після того, як було отримано центральну точку, створюється регіон інтересу, який охоплює лише частину зображення, де безпосередньо розташована складна геометрична структура. Розміри цього регіону задаються наступними формулами:

$$W_{roi} = 0.33 \times W,$$

та

$$H_{roi} = 0.5 \times H,$$

де W – кількість стовпців матриці пікселів вхідного зображення, а H – це кількість рядків матриці пікселів вхідного зображення.

Такий вибір розмірів дає змогу сконцентруватися на найбільш цікавій частині зображення, при цьому зберігаючи достатню кількість інформації для сегментації. Після чого у межах виділеного регіону проводиться наступна сегментація, за допомогою ансамблевого алгоритму. Оскільки цей етап обмежений меншою кількістю непотрібної інформації, ансамбль має змогу провести більш точну сегментацію, ніж на попередньому кроці, враховуючи деталі, які можливо були втрачені на першому етапі сегментації. Робота алгоритму наочно представлена у

вигляді блок-схеми на рисунку 3.20.

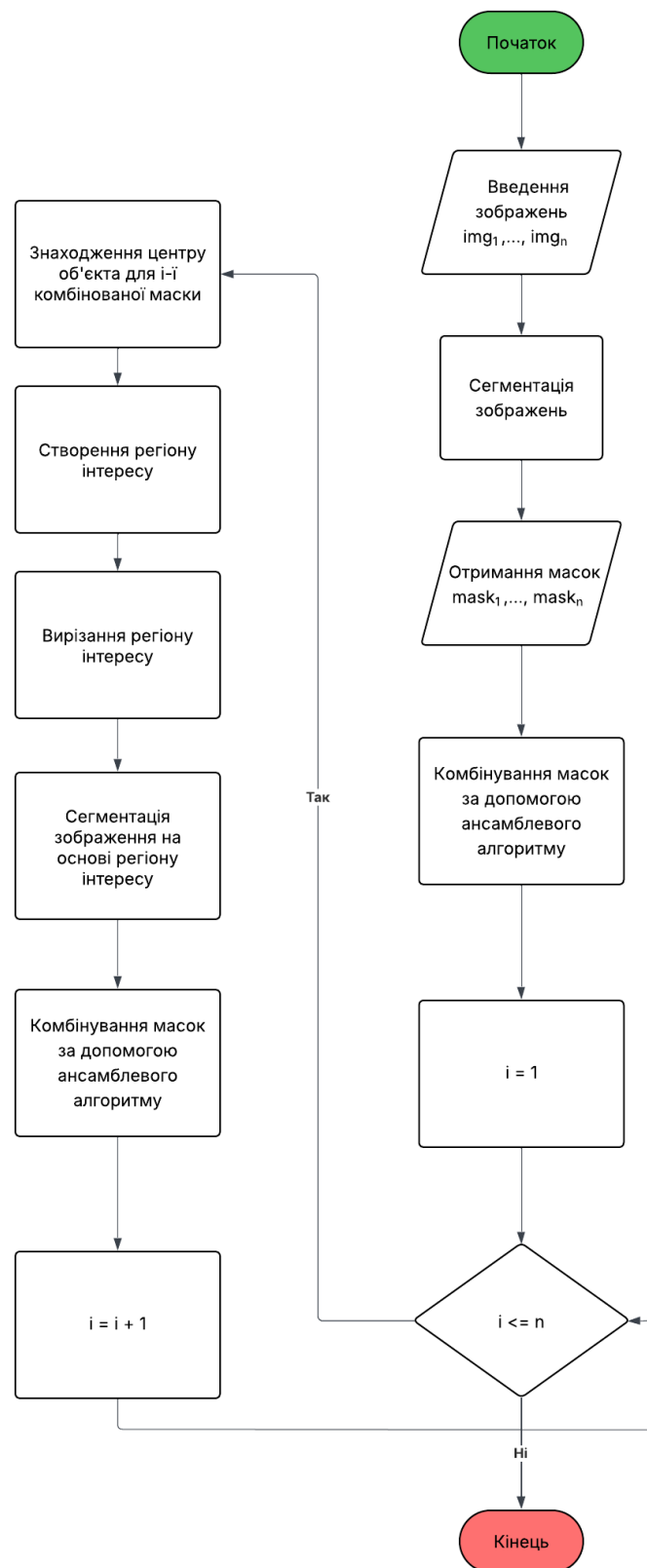


Рис. 3.20. Блок-схема алгоритму

Візуально виконання цього алгоритму відображено на зображенні 3.21. Де на зображенні А – це перший етап, на якому білим кольором відображено контур ідентифікованої складної геометричної структури, червоне коло – це центр цієї структури, а червоний прямокутник – це регіон інтересу, обраний для другого етапу. Зображення Б і є відображенням регіону інтересу, зеленим контуром позначено складну геометричну структуру, яку було отримано після сегментації другого етапу.

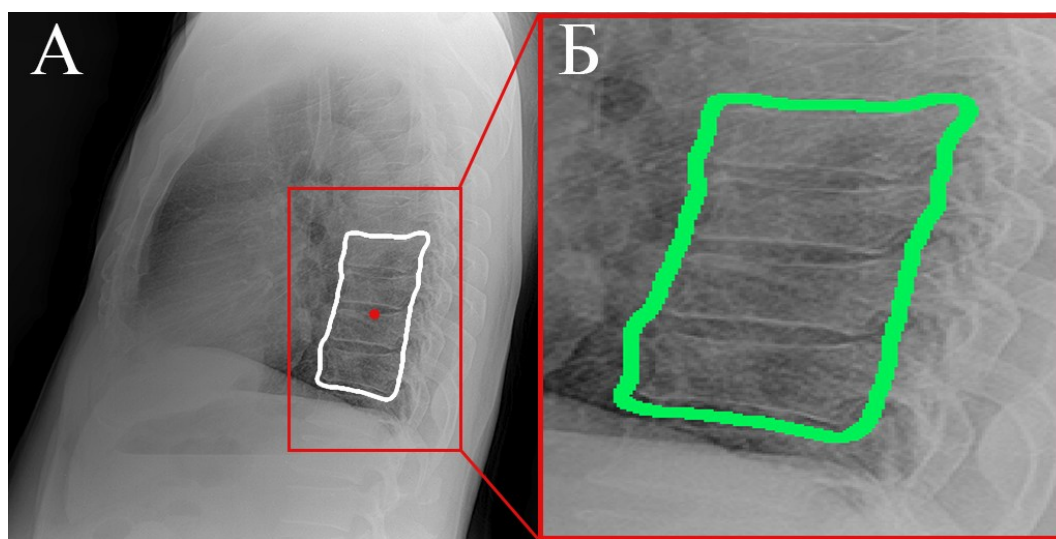


Рис. 3.21. Двоетапний алгоритм. А – перший етап, Б – другий етап

Результати передбачень для першого етапу представлені на рисунку 3.22. Контури виділені білим кольором – це контури еталонних масок, чорні контури – це контури передбачених масок. Для аналізу наведено два зображення: А – знімок №14, Б – знімок №23. В даному випадку було використано дві окремі моделі Mobilenet Segnet. Для зображень, які знаходяться ліворуч, було використано першу модель, а для тих зображень, що праворуч – другу. В даному випадку можна ознайомитись з якістю сегментації, для цього використовується порівняння двох контурів: контуру еталонної маски та контуру передбачення. Проаналізувавши таким чином зображення, можна зрозуміти, що форма складних геометричних структур на передбаченнях деколи знаходиться поза еталонною зоною пошуку або форма передбачуваної складної геометричної структури в рази менша від еталонної. Також, аналізуючи дане зображення, можна помітити, що контури, які були отримані в результаті передбачення мають значні нерівності. Причиною

спричинення таких похибок може бути низький контраст відносно фону або недостатній обсяг навчальних даних. Саме для вирішення таких проблем і було запропоновано використовувати різні види ансамблевих алгоритмів у двоетапному алгоритмі.

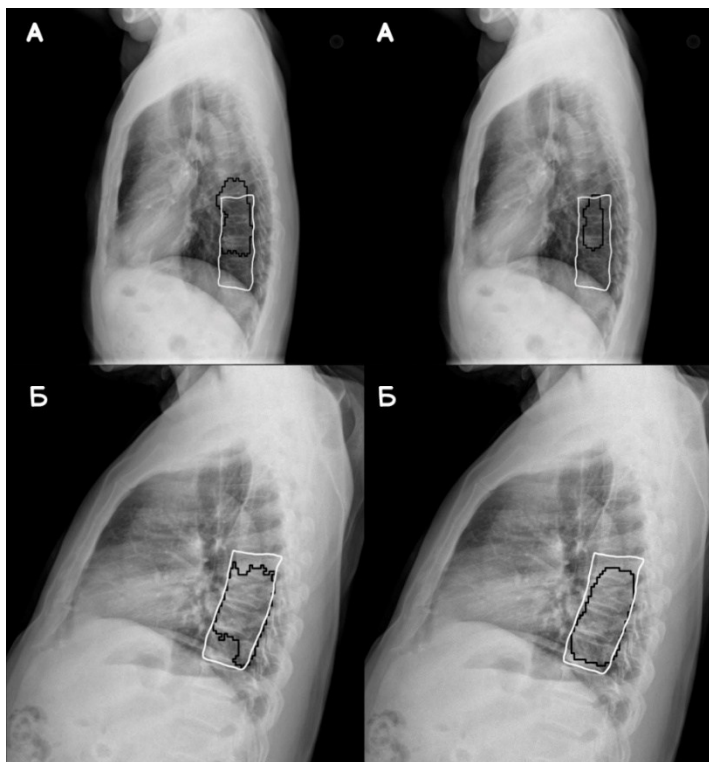


Рис. 3.22. Приклади передбачень на першому етапі

На рисунку 3.23 наведено результати окремих передбачень, а також використання ансамблевого алгоритму для двох нейронних мереж на першому етапі. Червоною лінією позначено результат усереднення для певної мережі. Зелена лінія – це результат ансамблевого алгоритму.

Як можна зрозуміти з діаграм наведених на рисунку 3.23, результат окремих моделей може бути кращим за результати ансамблевого алгоритму для одного зображення, а для іншого може бути навпаки. Такі сумнівні результати для першого етапу зумовлені, як вмістом шуму так і малим радіаційним контрастом, слабкими межами хребта. Передбачення для другого етапу показано на рисунку 3.24, для тих же вхідних зображень, що використовувались на рисунку 3.19. За допомогою використання другого етапу вдалось підвищити значення DICE для зображення

№14 з 54.30 % до 82.00 %, а для зображення №23 із 72.50 % до 92.30 %. В таблиці 3.13 приведено загальні результати.

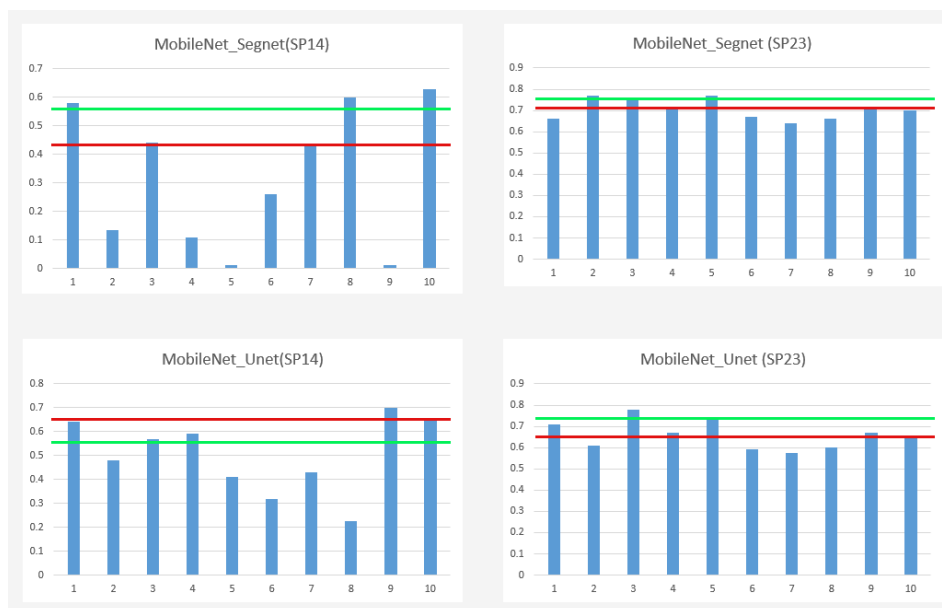


Рис. 3.23. Результати передбачень для окремих моделей та ансамблевого алгоритму

Таким чином, можна зробити висновок, що використання двоетапного алгоритму дозволяє покращити якість передбачень. Завдяки зменшенню регіону інтересу під час виконання другого етапу, ми маємо можливість сконцентруватися на найнеобхіднішій частині зображення ігноруючи зайве або артефакти.

Під час експериментів цей алгоритм продемонстрував ефективність на нових даних. Такий підхід є хорошим інструментом для вирішення задач сегментації, завдяки поєднанню переваги глобального і локального підходів. Його використання дозволяє досягати кращих результатів сегментації навіть у складних умовах.

В таблиці 3.13 наведено ефективність використання двоетапного алгоритму. Було розглянуто використання двох нейронних мереж Mobilenet Segnet та Mobilenet U-Net, оцінку продуктивності яких було виконано на двох наборах даних. Порівняння результатів свідчить про значну користь цього алгоритму. Точність на першому етапі краща ніж середнє значення для всіх моделей, а точність другого етапу краща ніж точність першого. Запропонований підхід – знаходження спочатку

глобальних ознак, а потім локальних, доводить його користь використання для будь-яких нейронних мереж або наборів даних.

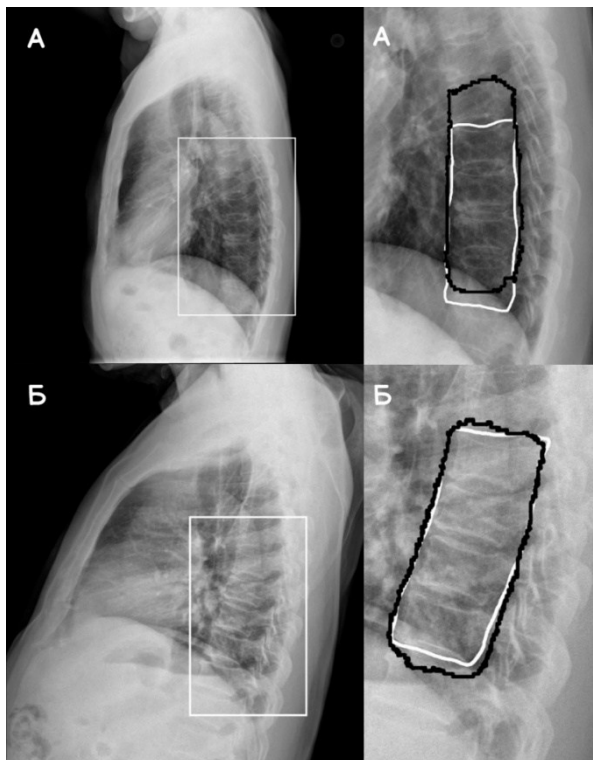


Рис. 3.24. Приклади передбачень на другому етапі, білі контури – це контури еталонних масок, чорні контури – це контури передбачених масок. А – знімок №14, Б – знімок №23

Таблиця 3.13

Результати використання двоетапного алгоритму з використанням ансамблевого алгоритму, DICE %

Модель	Набір даних	Безансамблевий алгоритм	Ансамблевий алгоритм	Покращення
Mobilenet Segnet	1	83.02	87.56	4.54
	2	68.67	76.02	7.35
Mobilenet U-Net	1	87.48	92.12	4.64
	2	76.29	82.68	6.39

3.8 Багатоетапний ансамблевий алгоритм сегментації зображень

Сегментація рентгенівських зображень грудної клітини є складною задачею в комп'ютерному зорі. В даному випадку запропонований алгоритм буде використовуватись саме для сегментації рентгенівських зображень але його можна оптимізувати та застосовувати, як універсальний алгоритм для сегментації любых зображень. Для детальної сегментації хребців було запропоновано багатоетапний алгоритм, який розбиває весь процес на послідовні етапи, що дає змогу виконувати задачу використовуючи результат попереднього кроку, чим саме ми забезпечуємо більш точну деталізацію.

Сегментація хребта та хребців на рентгенівських знімках є безперечно однією з найскладніших задач в сегментації медичних зображень. Основні складнощі полягають в особливостях анатомічної структури хребта, різній якості рентгенівських зображень, наявності патологій та інших проблем. З усього цього можна виділити більш проблематичний аспект – це якість рентгенівських зображень. Знімки відрізняються за якістю, роздільною здатністю, рівнем контрастності, артефактами. Це все додає вагомості складності до процесу сегментації. Саме для вирішення таких проблем, наукова спільнота почало розробляти спеціальні методи сегментації хребта та хребців. Існуючі наукові праці здебільшого можна поділити на три групи: розробка нових нейронних мереж, розробка нових або використання існуючих ансамблевих методів, розробка або використання інших методів.

Використання деяких алгоритмів видалення шуму дає змогу покращити якість зображення та сегментацію хребців. Наприклад, запропонований метод видалення шуму та подвійної рентгенівської абсорбціометрії [78] дав змогу більш точно виділяти контури хребців. Отримання такого результату стало можливим тільки завдяки розробці спеціального алгоритму видалення шуму. Використання звичайних алгоритмів, які використовують для переважної більшості звичайних зображень, наприклад Гауссове згладжування, тільки погіршують картину при сегментації медичних зображень. Перший метод за допомогою якого можна покращити точність сегментації хребта або хребців є створення власної нейронної

мережі. Для таких завдань розроблюються спеціальні мережі, які використовують головні ознаки хребта або хребців і завдяки отриманим даним виконують сегментацію. Такими мережами є MDR2-UNet [79], TransResUNet [80], CHASPPRAU-Net [81], VerseDiff-UNet [82], Multi-task Contexture Learning Network [83], MBNet [84], RFPNet [85], Sequential Conditional Reinforcement Learning network [86]. Наступним за популярністю методом є використання ансамблевих методів або неансамблевих методів, таких як VertXNet [87]; 2-етапна модель глибокого навчання для локалізації орієнтирів на рентгенограмах хребта [88]; локалізація ключових точок для суглобів хребця [89]; бімодальний метод сегментації [90]; SpineCLUE [91]; статистична модель форми та позиції кількох хребців для сегментації КТ зображень [92, 93]; напіваавтоматична сегментація хребців на об'ємних МР зображеннях за допомогою статистичної моделі форми та позиції [94]; використання YOLOv4 та ResUNet для виявлення переломів хребців на рентгенівських знімках [95]; автоматична сегментація шийних хребців для рентгенівських знімків [96]; сегментація хребців за допомогою графів [97]; використання YOLACT для сегментації хребців [98]; метод послідовного навчання з умовним підкріпленням для сегментації хребців [99]; використання R2U-Net для сегментації хребців [100]; метод який використовує ResNet-UNet для сегментації хребців [101]; SPINEPS [102]; комбінований підхід для сегментації хребців за допомогою U-Net та U-GCN [103]; локалізація хребта за допомогою виявлення точок інтересу [104]; метод який використовує машину опорних векторів та гістограму з орієнтованих градієнтів [105]. Всі ці методи використовують різні підходи та мають варіативність у покращенні точності.

Запропонований багатоетапний алгоритм сегментації хребців складається з наступних етапів:

1. Сегментації регіону хребта, яка складається з чотирьох хребців на вхідному зображенні.
2. Знаходження центра складної геометричної структури з маски отриманої на першому етапі та вирізання прямокутного регіону, який включає сегментовану структуру.

3. Сегментація регіону із чотирьох хребців на зображенні, отриманому на другому етапі.

4. Вирізання прямокутного регіону одержаної складної геометричної структури отриманої на третьому етапі.

5. Сегментація окремих хребців на зображенні отриманому на четвертому етапі.

6. Видалення артефактів на окремих хребцях.

7. Побудова середньої лінії хребта.

8. Горизонтальне вирівнювання хребців відносно середньої лінії з сьомого кроку.

9. Вертикальне вирівнювання за допомогою алгоритму обчислення відстаней між хребцями.

10. Вирізання прямокутного регіону, який включає один хребець.

11. Сегментація окремого хребця, з зображення отриманого на десятому етапі.

Візуально весь алгоритм можна побачити на рисунку 3.25. На рисунку детально відображено всі етапи алгоритму. Перші три етапи в запропонованому алгоритмі вже були наведені в попередньому підрозділі 3.7, де було надано визначення двоетапного алгоритму, на їх основі будується новий багатоетапний алгоритм. В цьому випадку цей алгоритм наведений, як трьохетапний, тільки тому, що другий етап є проміжний і не вважається повним етапом. Надалі ці три кроки будуть пропущені, оскільки вони вже були описані раніше. Подальший опис алгоритму почнеться з четвертого етапу.

Варто відзначити, хоч три етапи зазначені в підрозділі 3.7 не будуть детально розглянуті, необхідно визначити їх налаштування. Для всіх етапів буде використовуватись нейронна мережа Dilated-UNet. Для першого та третього етапів буде використано два ансамблі по 10 моделей, відповідно до кожного етапу.

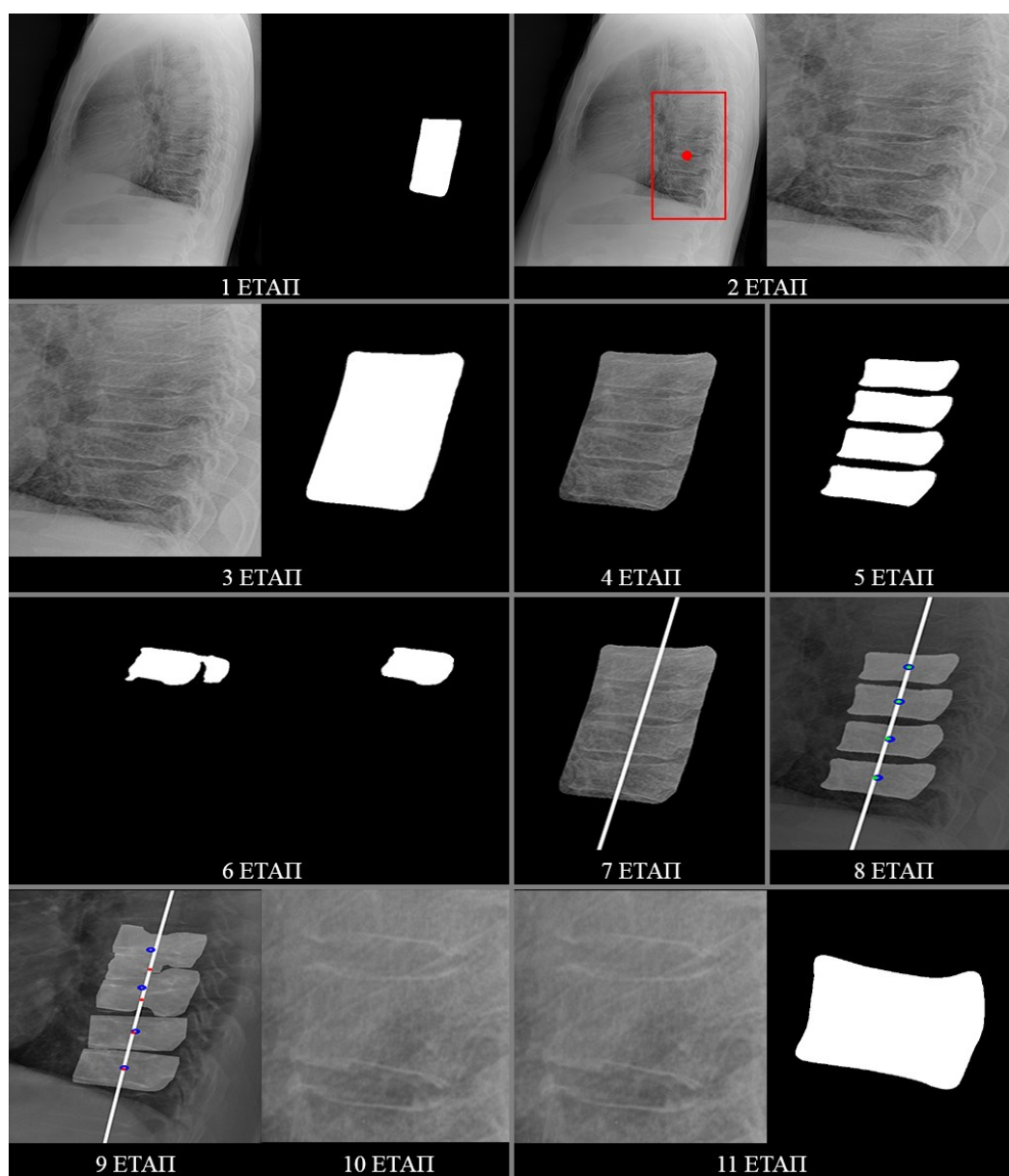


Рис. 3.25. Багатоетапний алгоритм сегментації хребців

Четвертий етап відображений зображенням, яке складається з регіону, який включає чотири складні геометричні фігури, які необхідно буде ідентифікувати: Th8, Th9, Th10, Th11. На п'ятому етапі відображено передбачення, яке було отримане після сегментації зображення наведеного на четвертому етапі. На рисунку шостого етапу можна побачити результат видалення артефактів з передбачення. Для кожної складної геометричної структури з кожного передбачення застосовується алгоритм видалення артефактів. На сьомому етапі проведено білу лінію, яка проходить через середину області об'єкта. На рисунку восьмого етапу приведено регіон зображення де вже знайдені центри для хребців, де синім

кольором позначено центри ідентифікованих складних геометричних структур, а саме – хребців, а зеленим кольором позначено ці центри зміщенні відносно середньої лінії хребта. На рисунку дев'ятого етапу відбувається вирівнювання хребців вертикально. На десятому етапі за допомогою значень отриманих на попередньому етапі виконується вирізання регіону інтересу для подальшої сегментація структур. І нарешті, на 11 етапі відбувається сегментація окремих хребців.

Четвертий етап.

На даному етапі необхідно отримати зображення, яке буде використовуватись для сегментації чотирьох складних геометричних структур, а саме – хребців. Для цього використовується маска з третього етапу та зображення з другого етапу. Нехай M – це матриця пікселів маски третього етапу, а I – матриця пікселів зображення другого етапу, де H – це кількість рядків матриці пікселів, а W – кількість стовпців матриці пікселів. В першу чергу необхідно створити пусту матрицю пікселів заповнену чорними пікселями:

$$S(i, j) = 0, \forall i \in [1, H], j \in [1, W].$$

Наступним кроком необхідно перетворити маску M у бінарну маску:

$$T(i, j) = \begin{cases} 255, & M(i, j) \geq 127 \\ 0, & \text{в іншому випадку} \end{cases}$$

Маючи бінарну маску ми можемо легко отримати контур C . Після чого необхідно всі елементи зображення всередині контуру заповнити білим кольором:

$$S(i, j) = \begin{cases} 255, & (i, j) \in C_{inside} \\ 0, & \text{в іншому випадку} \end{cases}$$

де C_{inside} – це множина пікселів, визначених всередині контуру C .

Останнім кроком є вирізання необхідного регіону:

$$R(i, j) = \begin{cases} I(i, j), & S(i, j) = 255 \\ 0, & \text{в іншому випадку} \end{cases}$$

де $R(i, j)$ – матриця пікселів, що містить сегментовані регіони з вирізаними хребцями.

На рисунку 3.26 зображено четвертий етап. Зліва – це вхідне зображення, по центру – маска, праворуч – результат вирізання. Зображена маска є результатом використання ансамблевого алгоритму. Для кожного вхідного зображення використовується N моделей, які в свою чергу роблять передбачення. Після того, як були отримані необхідні передбачення можна використовувати будь-який ансамблевий алгоритм, який виконає їх комбінування.

Як результат виконання даного етапу, маємо зображення яке складається з чотирьох хребців та невеликої ділянки довкола них. Такий підхід в подальшому дасть змогу виконувати сегментацію чотирьох хребців в указаному регіоні інтересу, що значно підвищує точність подальшої сегментації в порівнянні, якщо б ми виконували сегментацію хребців використовуючи вхідне зображення.

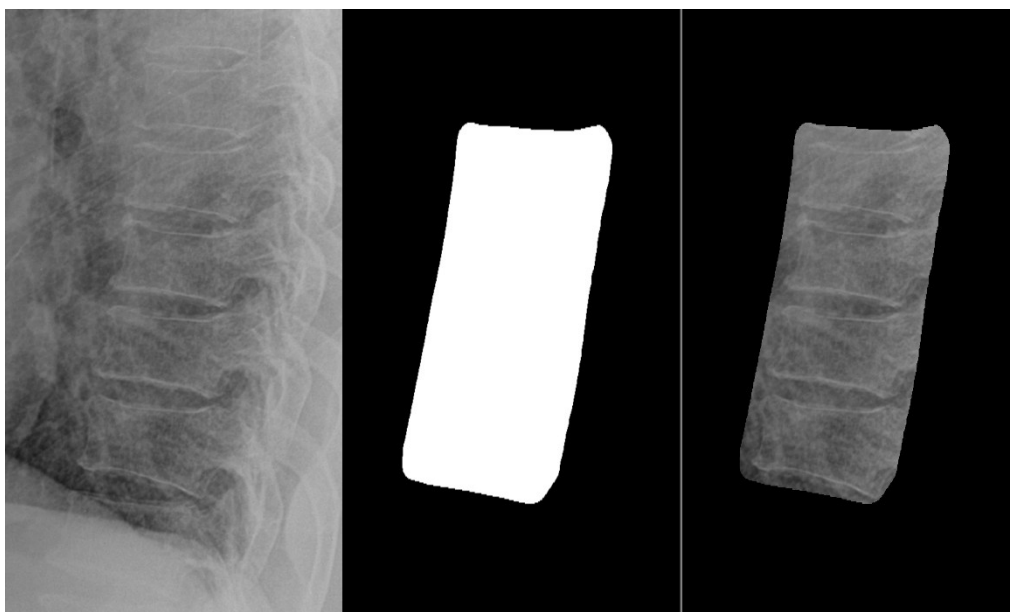


Рис. 3.26. Вирізання області з хребцями на четвертому етапі

П'ятий етап.

Для сегментації окремих хребців було використано чотири набори даних. Кожен набір складався із 182 зображень, розміром 512×512 пікселів в градації

сірого. Для тренування моделей було використано 160 зображень, а для оцінки продуктивності 22. Оскільки при використанні різних методів попередньої обробки рентгенівських зображень, ці методи можуть розмивати, або робити межі менш чіткими, тому попередня обробка даних не виконувалась [106]. Перший набір використовувався для сегментація хребця Th8, другий для Th9, третій для Th10 та четвертий для сегментації Th11. Для кожного набору даних було проведено тренування 10 моделей, які утворили ансамбль. В результаті було отримано 40 моделей, де кожні 10 моделей виконували сегментацію окремого хребця. Приклад використаних зображень можна побачити на рисунку 3.27.

В якості ансамблевого алгоритму можна використовувати любий доречний алгоритм, який має найкращі результати (в даному випадку було використано комбінацію 4). Після того, як оптимізація параметрів моделей була успішно проведена, необхідно використати ансамблевий алгоритм для комбінування окремих передбачень. В результаті цього етапу ми отримаємо комбіновані передбачення для всіх окремих хребців.



Рис. 3.27. Приклад зображень та масок для п'ятого етапу. Зліва направо – зображення для тренування моделей, маска Th8, маска Th9, маска Th10, маска Th11

Шостий етап.

Під час сегментації хребців іноді можуть траплятися артефакти на хребцях, здебільшого це виступаючі фрагменти або регіони, що не відповідають сегментованій складній геометричній структурі. Дуже важливо прибрати ці дефекти, оскільки їх наявність в подальших етапах дуже негативно вплине на

алгоритми вирівнювання хребців, тому що буде необхідно визначати центр хребця, а при використанні хребця, який має дефекти центр буде визначено некоректно. Звісно, форма хребця зміниться після цієї операції і алгоритм навіть може зачепити правильні частини хребця. Це не вплине значним чином на погіршення знаходження центру.

Суть даного алгоритму полягає в визначенні контуру складної геометричної структури, знаходженні мінімального прямокутника, що охоплює її (чотири крайні точки), аналіз розподілення білих елементів зображення поміж сторонами та видалення ліній де кількість білих елементів зображення менша за середню. Таким чином існує два варіанти видалення: горизонтальне та вертикальне. Нехай I – матриця пікселів вхідного зображення, C – контур об'єкта, а $B = \{b_1, b_2, b_3, b_4\}$ – вершини мінімального прямокутника, що охоплює складну геометричну структуру, відлік вершин починається з лівої верхньої. На рисунку 3.28 показано знаходження мінімального прямокутника, що охоплює хребець. На даному рисунку дуже гарно видно артефакт, який знаходиться праворуч від хребця, і з'єднаний з ним за допомогою тонкої перемички. Використання такого зображення для пошуку його центра призведе до різкого погіршення виконання алгоритму далі. Оскільки знайшовши центр хребця невірно, а в випадку з наявністю артефакту так і буде, знайдений центр не буде відповідати дійсному центра хребця.

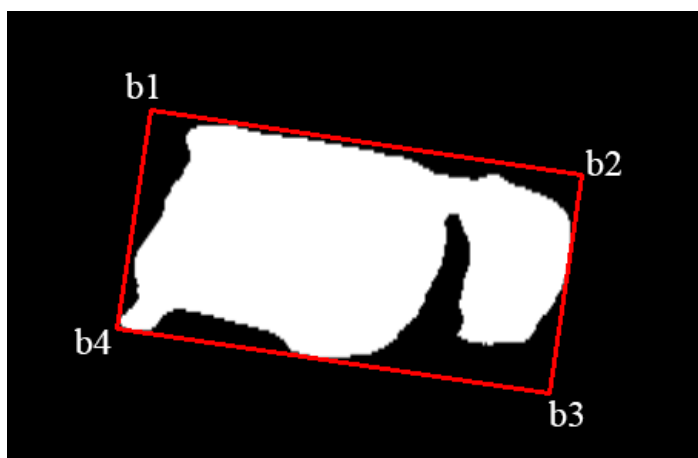


Рис. 3.28. Знаходження мінімального прямокутника, що охоплює хребець

В першу чергу необхідно визначити сторони для видалення вертикальних артефактів, отримати ліву та праву сторони можна за формулами:

$$S_{v,left} = [b_1, b_4], \quad (3.5)$$

$$S_{v,right} = [b_2, b_3], \quad (3.6)$$

та сторони для видалення горизонтальних артефактів:

$$S_{h,bottom} = [b_4, b_3], \quad S_{h,top} = [b_1, b_2].$$

Тепер необхідно визначити кількість точок для усіх сторін. Для лівої та правої:

$$N_{left} = \max(|x_1 - x_4|, |y_1 - y_4|), \quad (3.7)$$

$$N_{right} = \max(|x_2 - x_3|, |y_2 - y_3|), \quad (3.8)$$

для нижньої та верхньої:

$$N_{bottom} = \max(|x_4 - x_3|, |y_4 - y_3|), \quad N_{top} = \max(|x_1 - x_2|, |y_1 - y_2|).$$

Так, як для протилежних сторін кількість точок може бути різною, необхідно вибрати однакову кількість:

$$N_v = \max(N_{left}, N_{right}), \quad (3.9)$$

$$N_h = \max(N_{bottom}, N_{top}).$$

Наступним кроком необхідно побудувати задану кількість точок для кожної сторони. Для лівої та правої:

$$x_{left} = \text{linspace}(x_1, x_4, N_v), \quad (3.10)$$

$$y_{left} = \text{linspace}(y_1, y_4, N_v), \quad (3.11)$$

$$x_{right} = \text{linspace}(x_2, x_3, N_v), \quad (3.12)$$

$$y_{right} = \text{linspace}(y_2, y_3, N_v). \quad (3.13)$$

Об'єднуємо координати в масив точок:

$$S_{left} = \{(x_i, y_i) | x_i \in x_{left}, y_i \in y_{left}, i = 1, \dots, N_v\}, \quad (3.14)$$

$$S_{right} = \{(x_i, y_i) | x_i \in x_{right}, y_i \in y_{right}, i = 1, \dots, N_v\}. \quad (3.15)$$

Аналогічним чином для нижньої та верхньої сторін:

$$\begin{aligned}x_{bottom} &= linspace(x_4, x_3, N_h), & y_{bottom} &= linspace(y_4, y_3, N_h), \\x_{top} &= linspace(x_1, x_2, N_h), & y_{top} &= linspace(y_1, y_2, N_h).\end{aligned}$$

де $linspace()$ – це функція створення рівномірно розташованих чисел за вказаний інтервал.

Об'єднуємо координати в масив точок:

$$\begin{aligned}S_{bottom} &= \{(x_i, y_i) | x_i \in x_{bottom}, y_i \in y_{bottom}, i = 1, \dots, N_h\}, \\S_{top} &= \{(x_i, y_i) | x_i \in x_{top}, y_i \in y_{top}, i = 1, \dots, N_h\}.\end{aligned}$$

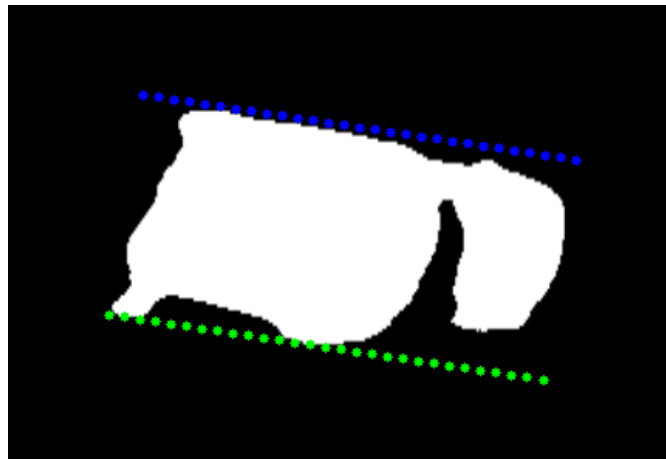


Рис. 3.29. Знаходження точок для нижньої та верхньої сторін, з інтервалом 10

На рисунку 3.29 відображено знаходження точок для нижньої сторони (зелені точки) та верхньої сторони (сині точки). На рисунку відображена кожна десята точка, для того щоб було легше візуально зрозуміти.

Тепер, коли поміж протилежними сторонами є однакова кількість точок, за допомогою двох протилежних точок можна знайти кількість білих елементів зображення між ними:

$$Np_{h,i} = count_white_pixels(p_{1,i}, p_{2,i}, I),$$

де $p_1 \in S_{bottom}, p_2 \in S_{top}$.

$$Np_{v,i} = count_white_pixels(p_{1,i}, p_{2,i}, I),$$

де $p_1 \in S_{left}, p_2 \in S_{right}$.

Спочатку знаходимо середнє значення білих елементів зображення для протилежних сторін:

$$avg_h = \frac{1}{N_h} \sum_{i=1}^{N_h} Np_{h,i},$$

$$avg_v = \frac{1}{N_v} \sum_{i=1}^{N_v} Np_{v,i}.$$

Визначаємо порогове значення:

$$T_h = 0.5 \times avg_h,$$

$$T_v = 0.5 \times avg_v.$$

На рисунку 3.30 можна побачити жовті лінії між синіми та зеленими точками, саме в кожній такій лінії і відбувається визначення кількості білих елементів зображення, а потім, якщо кількість менше за поріг, жовта лінія перетворюється в чорну.

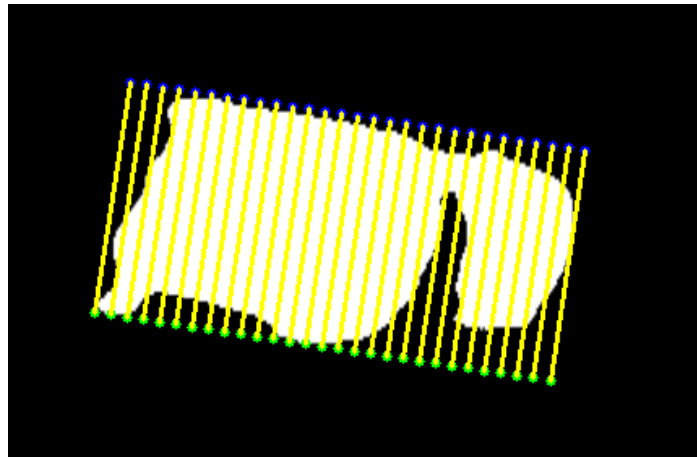


Рис. 3.30. Знаходження середнього значення кількості білих елементів зображення поміж сторонами

Якщо $Np_{h,i} < T_h$, то необхідно нарисувати чорну лінію між p_1 та p_2 . Після того, як чорні лінії були нарисовані, необхідно знайти найбільший контур на зображенні S_{max} , який, і є результатом видалення артефактів. Для того, щоб нівелювати обрізання необхідних елементів зображення, будемо використовувати

розширення зображення за допомогою певного структуруючого елемента, використовуємо функцію `dilate()`. Для якої вкажемо ядро розміром 10×10 та кількість ітерацій 1.

На рисунку 3.31 продемонстровано результат роботи алгоритму. Білий хребець є первинним зображенням. Сірий хребець поверх білого – це результат алгоритму. Червоне коло – це центр первинного хребця, а зелене коло – це центр хребця без артефактів. Можна побачити, що місце розташування центрів відрізняється.

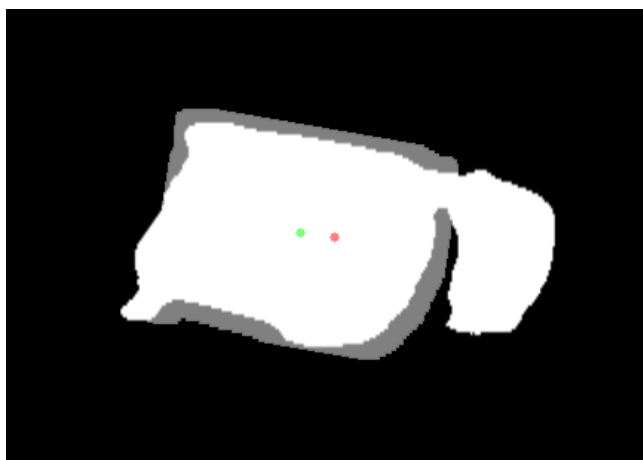


Рис. 3.31. Результат алгоритму видалення артефактів

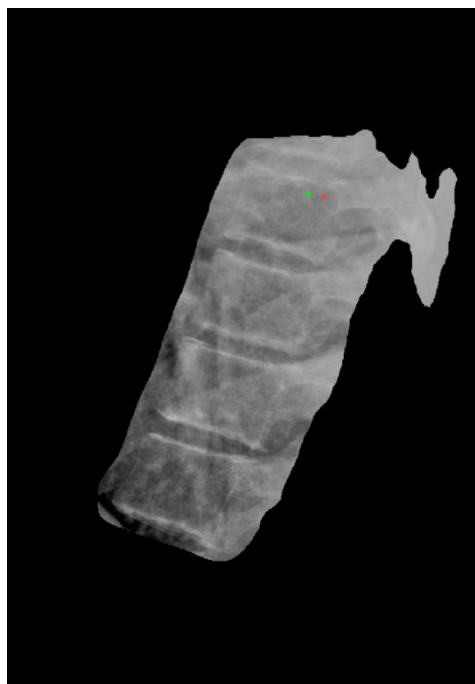


Рис. 3.32. Розташування центрів первинного хребця та хребця без артефактів на зображенні четвертого етапу

На рисунку 3.32 добре видно доречність використання даного алгоритму. На рисунку з четвертого етапу відображено два центри: первинний центр та центр без артефакту. Завдяки використанню даного алгоритму маємо змогу побачити, що зелене коло знаходиться до істинного центра хребця ближче ніж червоне. Це в подальшому дуже допоможе, коли необхідно буде ізолювати виключно тільки один хребець.

Сьомий етап.

Створення середньої лінії на масці хребта дає змогу потім вирівняти окремі хребці відносно середньої лінії, завдяки чому маємо змогу покращити розташування хребця на первинному зображенні. Процес виконання даного алгоритму можна описати як: 1) побудова мінімального обмежувального прямокутника для заданої складної геометричної структури; 2) знаходження точок лівої та правої сторони прямокутника; 3) проведення лінії між протилежними точками; 4) знаходження крайніх протилежних точок на контурі хребта; 5) обчислення центральної точки між точками з пункту 4; 6) видалення 15 % початкових та 15 % кінцевих середніх точок; 7) апроксимація та екстраполяція середніх точок.

Нехай I – це маска хребта, C – контур об'єкта, а $B = \{b_1, b_2, b_3, b_4\}$ – вершини мінімального прямокутника, що охоплює складну геометричну структуру. Використовуючи формулу (3.5) та (3.6) визначимо ліву сторону та праву відповідно. За допомогою формули (3.7) визначимо необхідну кількість точок для лівої сторони, за допомогою формули (3.8) визначимо кількість точок для правої сторони. Використовуючи формулу (3.9) отримаємо рівномірну кількість точок для обох сторін. Після чого необхідно визначити координати x та y для точок лівої сторони використовуючи формули (3.10) та (3.11) відповідно. Аналогічно використовуючи формули (3.12) та (3.13) визначаємо для правої сторони. Тепер маємо точки лівої сторони S_{left} отримані за формулою (3.14) та точки S_{right} отримані за допомогою формули (3.15). Маючи точки лівої сторони та правої, маємо

змогу провести між ними лінію та знайти точки перетину, які належать контуру і є протилежними:

$$P_{left,i}, P_{right,i} = \text{get_intersection_points}(S_{left,i}, S_{right,i}, I).$$

Знайдемо середню відстань між точками перетину:

$$avg_{length} = \frac{1}{N_v} \sum_{i=1}^{N_v} \text{dist}(P_{left,i}, P_{right,i}),$$

де $\text{dist}()$ – функція знаходження відстані між двома точками.

Тепер видалимо всі точки відстань яких менша за $\frac{avg_{length}}{1.5}$ та отримаємо послідовність точок *Intersections*, яка складається з лівих, та правих точок, що перетинають контур. Створимо послідовність *Midpoints*, яка буде складатися з середніх точок, середню точку можна отримати, якщо скласти координати лівої та правої точок, і розділити навпіл:

$$M_p = \frac{P_{left,x} + P_{right,x}}{2}, \frac{P_{left,y} + P_{right,y}}{2}.$$

Видалимо перші 15 % середніх точок та 15 % останніх точок:

$$Midpoints = Midpoints[start_index:end_index],$$

де $start_index = 0.15 \times N_v, end_index = 0.85 \times N_v$.

Тепер необхідно виконати екстраполяцію для середніх точок. Отримаємо екстрапольовану послідовність точок за допомогою використання апроксимації поліному другого ступеня вздовж вертикальної осі:

$$Extrapolated = \text{polynomial_approximation}(Midpoints, 2, N_v \times 2).$$

де $\text{polynomial_approximation}()$ – функція, яка виконує апроксимацію точок, використовуючи поліном 2 ступеня та повертає вдвічі більше точок ніж було передано.

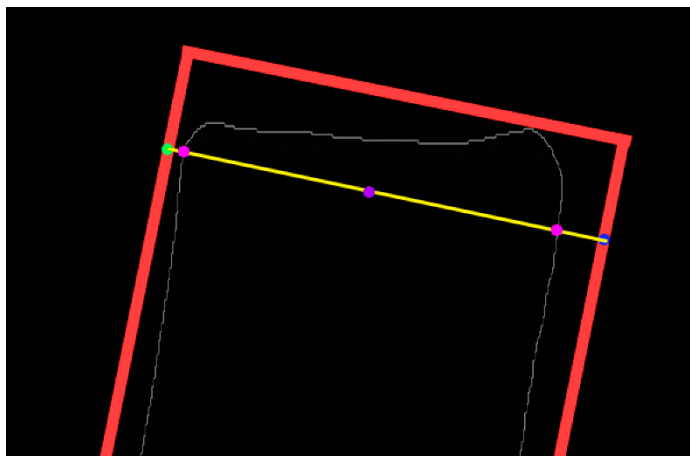


Рис. 3.33. Знаходження середньої точки

На рисунку 3.33 візуально відображено процес отримання однієї з середніх точок. Червоний прямокутник навколо контуру хребта – це мінімальний прямокутник, який охоплює контур. Зелена точка – це точка $S_{left,i}$, синя точка – $S_{right,i}$. Жовтим кольором проведена лінія між цими точками, а рожевим кольором продемонстровані точки перетину контуру та лінії, яка з'єднує $S_{left,i}$ та $S_{right,i}$. Наостанок фіолетова точка – це середня точка. Таким чином на вказаному рисунку наведено знаходження тільки однієї середньої точки, використовуючи *Midpoints* на зображення наносяться усі середні точки.

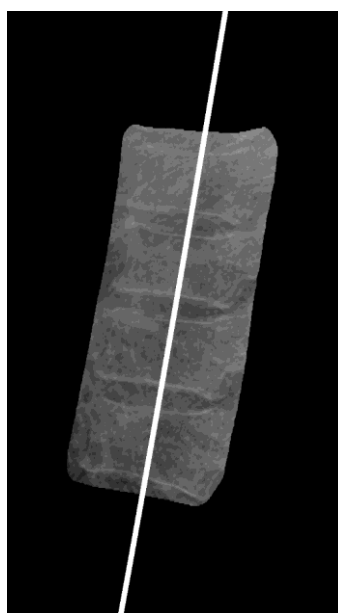


Рис. 3.34. Середня лінія через хребет

На рисунку 3.34 відображено фінальну версію середньої лінії, яка складається з точок послідовності *Extrapolated* та проходить через весь хребет. Як можна побачити із зображення, середня лінія точно проходить через істинні центри хребців.

Восьмий етап.

Для того, щоб покращити остаточну точність сегментації хребця необхідно правильно визначити розмір та координати прямокутника, який потрібно вирізати, а потім в якому буде виконано сегментацію хребця. Маючи чотири центри хребців, $Centers = \{C_1, C_2, C_3, C_4\}$, необхідно знайти найближчу точку E_j яка відноситься до середньої лінії *Extrapolated*. Знаходження E_j дасть змогу замінити C_i на E_j , що в свою чергу значно покращить координати центра прямокутника, який потрібно вирізати. Для цього потрібно, для кожної точки C_i із *Centers* знайти найближчу точку E_j із *Extrapolated*, використовуючи Евклідову відстань вказану формулою (3.4) для точок (C_i, E_j) отримаємо необхідну відстань, де найближча точка E_j' визначається як:

$$E_j' = \arg \min_{E_j \in Extrapolated} d(C_i, E_j).$$

Після чого змінюємо координати центрів хребців:

$$C_i' = E_j'.$$

Як результат даної операції отримаємо нову послідовність центрів хребців $AlignedCenters = \{C_1', C_2', C_3', C_4'\}$.

На рисунку 3.35 відображено результат використання даного етапу. Синім кольором позначені початкові центри хребців, зеленим кольором позначено вирівняні центри за допомогою алгоритму. Як можна побачити, зелені центри знаходяться на середній лінії хребта, що дає змогу краще визначити подальші розміри прямокутника.

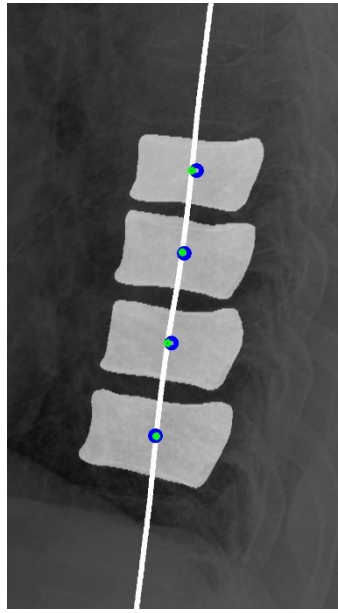


Рис. 3.35. Вирівнювання центральних точок хребців

Дев'ятий етап.

Існує деяка кількість випадків коли на попередніх етапах хребці були сегментовані погано. Хребці можуть знаходитись між собою на різних відстанях, вони можуть дуже відрізнятись по формі, що робить процес вирізання дуже складним. Для того, щоб вирішити ці проблеми, можна використовувати алгоритм вертикального вирівнювання хребців. Завдяки його використанню можна знайти відстані між хребцями $D_1 = d(C'_1, C'_2)$; $D_2 = d(C'_2, C'_3)$; $D_3 = d(C'_3, C'_4)$, де $d()$ – це Евклідова відстань. Знаючи відстані між хребцями, можемо відкоригувати розташування центрів хребців, якщо відстані дуже відрізняються.

В першу чергу необхідно визначити чи вірна відстань між усіма хребцями. Для цього використаємо чотири умови:

$$A_1 = D_3 > D_2 \wedge D_3 < 1.19 \times D_2,$$

$$A_2 = D_2 > D_1 \wedge D_2 < 1.19 \times D_1,$$

$$A_3 = D_3 < D_2 \wedge D_2 < 1.08 \times D_3,$$

$$A_4 = D_2 < D_1 \wedge D_1 < 1.08 \times D_2.$$

Необхідно порахувати кількість вірних умов:

$$counter = \sum_{i=1}^4 A_i.$$

Якщо $counter \geq 2$, тоді відстані між хребцями правильні, і використовувати алгоритм для їх виправлення непотрібно, в іншому ж випадку потрібно застосувати вказаний нижче алгоритм. Розглянемо декілька випадків.

Випадок 1. Якщо $D_1 > D_2 \wedge 1.19 \times D_2 < D_1$ та $D_2 > D_3 \wedge 1.19 \times D_2 < D_3$ то отримаємо нові відстані $D'_1 = 1.094 \times D_2$ та $D'_3 = 0.925 \times D_2$. Після чого змінимо координати для $C'_1 = (C'_2 - D'_3)$ та $C'_4 = (C'_3 + D'_1)$. Інші центри C_2 та C_3 залишаємо.

Випадок 2, якщо $D_1 > D_2 \wedge 1.19 \times D_2 < D_1$ та $D_2 < D_3 \wedge 1.08 \times D_2 < D_3$ отримаємо нові відстані $D'_2 = 0.914 \times D_1$. Після чого змінимо координати для $C'_2 = (C'_3 - D'_2)$ та $C'_3 = (C'_2 + D'_2)$. Інші центри C_1 та C_4 залишаємо.

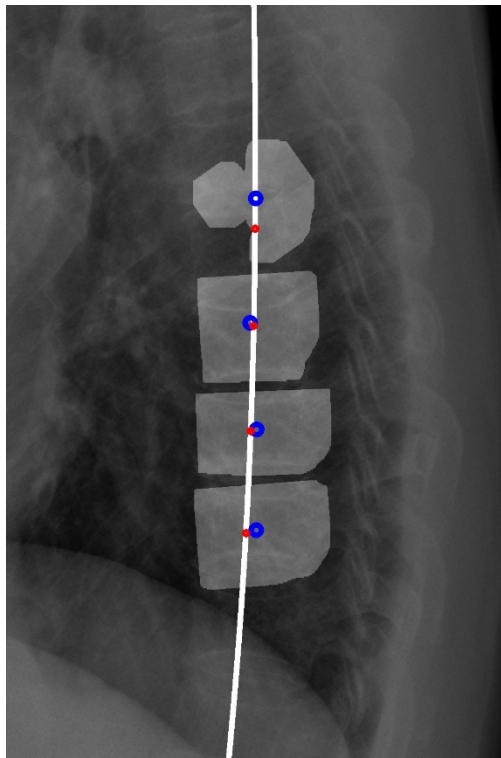


Рис. 3.36. Вирівнювання центральних точок хребців, якщо відстані між ними некоректні

В результаті отримаємо нові центри для хребців, які мали відхилення. На рисунку 3.36 візуально можна побачити результат алгоритму. Синім кольором позначено початкові центри хребців, а червоним центри після використання алгоритму вирівнювання. Завдяки застосуванню даного алгоритму можна вірно налаштувати відстані між хребцями та змінити центри.

Десятий етап.

Після того, як хребці було вирівняно, маємо необхідність вирізати регіон зображення в який входить сам хребець. Після того, як було отримано центральну точку, створюється регіон інтересу, який охоплює лише частину зображення де безпосередньо розташовано складну геометричну структуру. Визначимо коефіцієнти для висоти та ширини: $H_c = 1.7, W_c = 1.9$. Розміри цього регіону задаються наступними формулами:

$$W_{roi} = D_i \times W_c,$$

та

$$H_{roi} = D_i \times H_c,$$

де D_i – це i – та відстань між двома хребцями.

Одинадцятий етап.

Останній етап в алгоритмі де відбувається сегментація хребця на зображенні яке складається виключно із зображення хребця та деяких країв інших хребців. Для сегментації окремого хребця було використано один набір даних. Набір складався із 182 зображень розміром 512×512 пікселів в градації сірого. Для тренування моделей було використано 160 зображень, а для оцінки продуктивності 22. Попередня обробка даних не виконувалась. Для сегментації було створено ансамбль з 10 моделей. В результаті було отримано 40 моделей, де кожні 10 моделей виконували сегментацію окремого хребця. На рисунку 3.37 можна побачити зображення та маску для тренування моделей.

В якості ансамблевого алгоритму було використано комбінацію 4. Результатом даного етапу стала сегментація окремого хребця. Якщо порівнювати результати сегментації окремих хребців з 5 етапу та 11 етапу, то можна зрозуміти, що використання багатоетапного ансамблевого алгоритму сегментації має значне покращення.

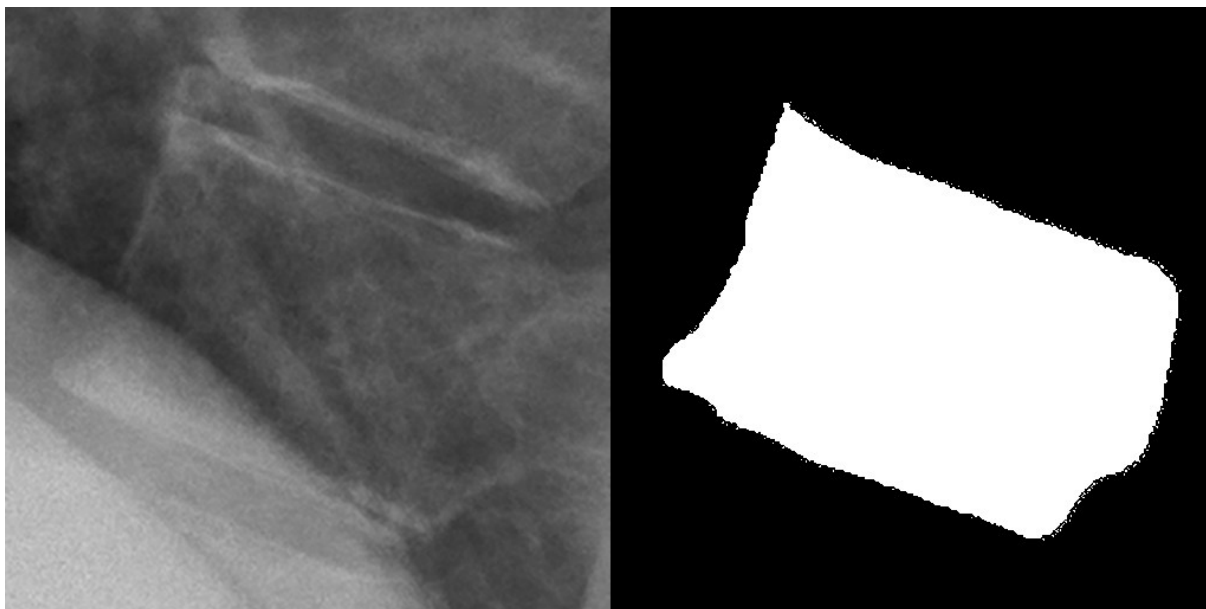


Рис. 3.37. Зображення та маска для сегментації окремого хребця

В таблиці 3.14 наведено ефективність використання багатоетапного алгоритму. Для цього в таблиці показано порівняння використання даного алгоритму для безансамблевого підходу та ансамблевого підходу. Наведені результати підтверджують доцільність використання ансамблевих алгоритмів у запропонованого алгоритму для сегментації зображень низької якості. Як можна побачити з даних хребця Th8, покращення склало 12.76 %, для хребця Th9 склало 11.89 %, у Th10 покращення 11.40 %, покращення для останнього хребця склало 15.25 %. Застосування багатоетапного ансамблевого алгоритму в середньому підвищило точність на 12.85 %. З усього цього можна зробити наступний висновок, приріст точності алгоритму залежить від початкової базової точності через ефект згладжування та обмеження покращення. При низькій точності, алгоритм має більше простору для виправлення помилок, тому покращення є значним. Однак при високій точності, простір для поліпшень обмежений, оскільки більшість прикладів

вже класифіковані правильно, і алгоритм додає лише незначне вдосконалення. Використання цього алгоритму дає змогу вирішити проблеми, які були розглянуті в підрозділах де порівнювалась ефективність використання ансамблевих алгоритмів. Цей алгоритм виступає в якості універсального методу вирішення існуючих проблем сегментації. Завдяки його використанню можна отримувати гарантоване покращення точності сегментації незалежно від використаної нейронної мережі.

Таблиця 3.14

Результати використання багатоетапного алгоритму, DICE %

Хребець	Безансамблевий алгоритм	Ансамблевий алгоритм	Покращення
Th8	72.73	85.49	12.76
Th9	79.16	91.05	11.89
Th10	80.21	91.61	11.40
Th11	76.89	92.14	15.25

Висновки до третього розділу

У третьому розділі було описано вибір архітектур нейронних мереж, що використовувались для сегментації рентгенівських зображень, а також вибір наборів даних на яких виконувалась сегментація. Було висвітлено ансамблевий алгоритм усереднення форм об'єктів, а також алгоритми на основі центрування, які використовують центроване вирівнювання об'єктів перед застосуванням ансамблевих алгоритмів. На базі запропонованих алгоритмів було реалізовано комбіновані ансамблеві алгоритми сегментації, які поєднали в собі різні підходи для досягнення більш точної сегментації. Для того, щоб оцінити точність та ефективність запропонованих ансамблевих алгоритмів було виконано порівняння оцінок точності. Для незалежних результатів було розглянуто вплив розроблених алгоритмів на нові та вже навчені дані. Також було запропоновано ансамблевий алгоритм адаптивного уточнення контуру на основі яскравості сусідніх пікселів.

Одним з розглянутих найперспективніших алгоритмів був багатоетапний ансамблевий алгоритм сегментації, який можна використовувати напрямую або можна застосовувати його двоетапну версію. Така варіативність одного алгоритму дає змогу використовувати різну версію для різних задач.

Детальний аналіз ансамблевих алгоритмів продемонстрував деякі складнощі в плані визначення універсального алгоритму сегментації для всіх нейронних мереж. В результаті чого було запропоновано реалізацію комбінації ансамблевих алгоритмів, а також комбінації різних підходів, як наприклад багатоетапний ансамблевий алгоритм. Використання комбінованих ансамблевих алгоритмів дає можливість покращити результати в порівнянні з використанням окремих алгоритмів.

З отриманих даних для другого набору даних, який використовувався, як низькоякісні зображення для оцінки продуктивності, було отримано наступні результати для запропонованих алгоритмів:

1. Комбінація 4 мала перевагу перед окремими ансамблевими алгоритмами та покращила результати на 0.79 %.

2. Використання адаптивного алгоритму уточнення контуру на основі яскравості сусідніх пікселів для неансамблевого підходу покращило результати на 5.66 %, для ансамблевого алгоритму усереднення з порогом 1/2 на 1.87 %, для ансамблевого алгоритму об'єднання покращення складало 0.29 %.

3. Завдяки використанню двоетапного алгоритму мали змогу отримати середнє покращення для чотирьох хребців, яке складало 5.73 %.

4. Використання багатоетапного алгоритму дало змогу отримати середнє покращення для чотирьох хребців 12.82 %.

Це підтверджує гіпотезу про необхідність використання комбінації різних алгоритмів для сегментації зображень низької якості, оскільки найкращий результат продемонстрував ансамблевий багатоетапний алгоритм сегментації.

Основні результати опубліковані в роботах [48, 49, 50, 51, 67, 74, 75, 77, 106].

РОЗДІЛ 4

РОЗРОБКА КОМП'ЮТЕРНОЇ АВТОМАТИЗОВАНОЇ СИСТЕМИ ДЛЯ ОТРИМАННЯ ДІАГНОСТИЧНИХ ПАРАМЕТРІВ ХРЕБЦІВ

4.1. Постановка задачі розробки програмного забезпечення для визначення діагностичних параметрів хребців

Сучасна медицина стикається з необхідністю створення автоматизованих систем аналізу здоров'я пацієнтів. Існують різні напрямки ПЗ для медичної сфери: машинний аналіз особистих справ, визначення патологій за допомогою медичних зображень та інші. Основна увага при розробці медичного ПЗ приділяється технічному реалізуванню та етичним аспектам [107]. Чого можна досягти значною мірою використовуючи принципи FAIR [108]. Одним із головних напрямків є діагностика захворювань опорно-рухового апарату, включно з патологічними змінами в хребцях. Рентгенологічні дослідження є найпопулярнішим та найдешевшим методом візуалізації хребта, однак для їх інтерпретації потрібен кваліфікований лікар та деякий час. В зв'язку з цим виникає необхідність розробки ПЗ, яке здатне автоматично визначати діагностичні параметри хребців, допомагаючи лікарям, як інструмент для підтримки прийняття рішень, а не повністю замінити людське судження [109].

Також, при розробці ПЗ, дуже популярним є використання віртуальних розподілених систем (VDS), завдяки їх використанню можливо покращити продуктивність продукту, що буде розроблено. Використання VDS дозволяє розподіляти навантаження, за рахунок чого можна обробляти більш великі обсяги даних.

Основною ціллю розробки є створення ПЗ для автоматичного аналізу рентгенівських зображень, направлено на визначення геометричних характеристик хребців, висоти передньої, середньої та задньої частини хребця. За допомогою отриманих даних можливо отримати необхідну інформацію про такі захворювання, як остеопороз, протрузії, грижі або викривлення хребта.

Проблема діагностики захворювань хребта є актуальною в наш час, через деякі виклики:

- Збільшення числа захворювань опорно-рухового апарату.
- Складність ручної інтерпретації даних.
- Потрібність в стандартизації діагностики.

Існуюче програмне забезпечення для діагностування захворювань хребта використовує різні підходи: алгоритми штучних нейронних мереж (FFNN та GRNN), і метод опорних векторів (SVM) [110]; підхід на основі аналізу попередніх випадків CBR [111]; 2D методи машинного навчання [112-117]; метод який об'єднує інформацію про хвороби хребта у вигляді онтологій для покращення медичної діагностики та візуалізації за допомогою 3D [118]; метод використання інерціальних вимірювальних пристроїв, як менш інвазивну та економічно вигідну альтернативу для скринінгу захворювань хребта [119]; та цілий апаратно-програмний комплекс, призначений для реабілітації пацієнтів із захворюваннями хребта [120].

Основна відмінність запропонованого ПЗ полягає в методі сегментації рентгенівських зображень, для цього буде використано багатоетапний ансамблевий метод сегментації, завдяки якому буде отримано покращення точності сегментації. Ключові етапи постановки задачі можна поділити на декілька пунктів:

1. Аналіз предметної області

- Аналіз існуючих методів діагностування захворювань хребта.
- Визначення набору діагностичних параметрів.

2. Вибір методів обробки даних

- Використання методу регіону інтересу (ROI).
- Використання машинного навчання для сегментації хребта та хребців.

3. Підготовка даних

- Створення бази для тренування моделей.

4. Розробка алгоритмів

- Розробка алгоритмів для видалення артефактів.
- Створення алгоритмів для сегментації хребта та хребців.
- Створення алгоритмів для отримання діагностичних параметрів хребців.

5. Розробка графічного інтерфейсу користувача

- Створення зручного інтерфейсу для користування з легким отриманням результатів.

Розробка ПЗ для визначення діагностичних параметрів хребців є доволі складною задачею, яка включає в себе інтеграцію технологій машинного навчання та медичної візуалізації. Вирішення цієї задачі може значно підвищити точність та оперативність діагностування захворювань, що важливо в зв'язку зі збільшенням попиту на такі послуги.

4.2 Підготовка навчальних даних та тренування моделей

Як було зазначено в підрозділі 4.1, в запропонованому ПЗ буде використовуватись багатоетапний ансамблевий алгоритм сегментації. Основною ідеєю покращення точності сегментації зображення є використання ансамблю нейронних мереж. Такий підхід потребує великої кількості навчених моделей. Для задовільного використання цього алгоритму необхідно навчити 140 моделей. Всього буде сім ансамблів, кожен з яких буде складатись із 20 моделей. Всі зображення, які будуть використовуватись матимуть однакові розміри 512×512 пікселів та відтворені в градації сірого. В даному випадку не використовувались методи попередньої обробки зображень. Заради тренування моделей сегментації для ідентифікації складної геометричної структури, а саме хребта, на початковому етапі було використано 170 зображень. Аналогічну кількість було використано, і

для тренування моделей, які будуть виконувати ідентифікацію хребта у вирізаному прямокутнику. Заради тренування моделей, які будуть виконувати ідентифікацію хребців Th8, Th9, Th10, Th11 на вирізаному хребті, було використано по 160 зображень для навчання кожного хребця відповідно. Наостанок для фінального етапу де на вирізаному хребті необхідно вирізати область з хребцем і в ній виконати сегментацію хребця було використано 224 зображення.

В основі всіх навчених моделей лежить нейронна мережа Dilated-UNet. Реалізація тренування моделей була виконана за допомогою мови програмування Python 3.8 [121]. Вибір версії Python обумовлений сумісністю з іншими API. Для використання обчислювальних можливостей графічних процесорів було використано CUDA Toolkit 11.2 [122], також, для побудови та тренування моделей були використано бібліотеки Tensorflow 2.9.1 [123] та Keras 2.9.0 [124]. При навчанні всіх моделей, фіксована кількість епох не використовувалась, для того, щоб запобігти перенавчанню моделі, використовувався метод EarlyStopping, завдяки якому модель переставала навчатись коли її подальші результати не могли продемонструвати кращий результат. В якості моніторингу було обрано коефіцієнт DICE, в якості витримки використовувались 10 епох, що означало зупинку тренування моделі, якщо протягом 10 епох значення точності не покращувались. Всі моделі було збережено в форматі h5, кінцева вага всіх 140 навчених моделей склала 16.2 ГБ.

4.3 Реалізація програмного забезпечення автоматизованого процесу отримання діагностичних параметрів хребців грудного відділу

Для побудови графічного інтерфейсу користувача було обрано Tkinter [125]. Розробка ПЗ для автоматизації процесу отримання діагностичних параметрів хребців включає кілька головних етапів, а саме: завантаження, обробка, сегментація, аналіз зображення, а також створення відповідної звітності.

На початковому етапі користувачу необхідно завантажити в програму одне, або декілька рентгенівських зображень грудної клітини в боковій проекції. Інтерфейс реалізовано так, що користувач має змогу переключатись між

завантаженими зображеннями, так як вони відображаються у вигляді списку. Центральна частина інтерфейсу складається з елементу в якому виводиться вибране зображення. Для полегшення навігації між етапами використовуються вкладки, які розташовані над зображенням. Кожна вкладка відповідає окремому етапу: вибране зображення, перегляд сегментованих масок, а також фінальні результати аналізу.

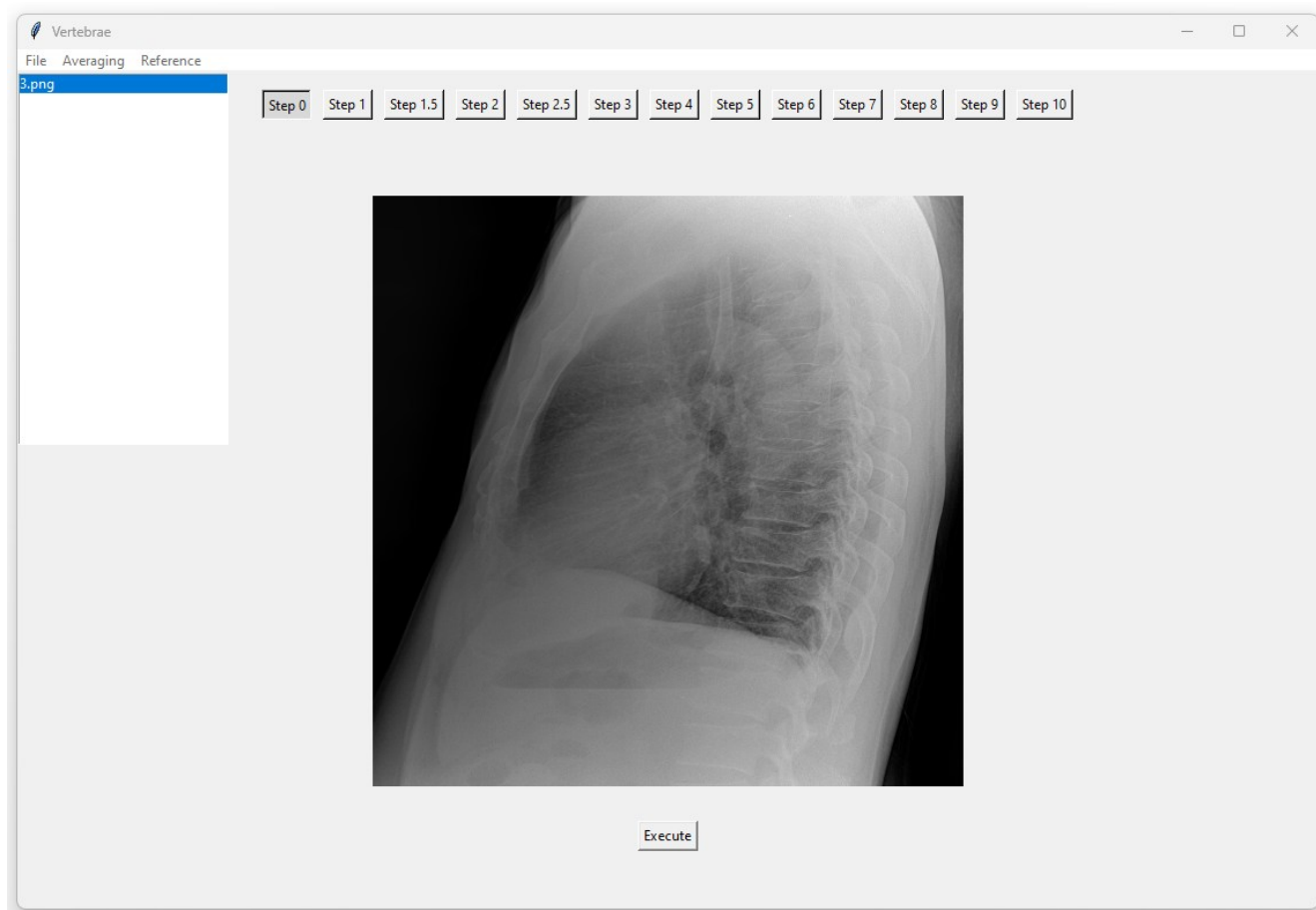


Рис. 4.1. Інтерфейс програми

На рисунку 4.1 відображено графічний інтерфейс користувача запропонованої програми. Активація вкладки Step 0, відображає в центрі екрану користувачу оригінальне зображення, яке він вибрав для аналізу.

Головною особливістю ПЗ є використання ансамблевого алгоритму сегментації, а саме комбінації 4 для всіх етапів сегментації, за допомогою адаптивного алгоритму уточнення контуру.

За допомогою кнопки Execute, користувач запускає виконання

багатоетапного алгоритму. В основі якого лежить використання багатоетапного алгоритму сегментації, завдяки якому запропоноване ПЗ значно відрізняється від інших аналогів, які використовують сегментацію в один етап. Таким чином, використання цього методу в ПЗ дозволяє домогтися стабільності і точності передбачень шуканих регіонів хребта, навіть для знімків з високим рівнем шуму, низьким контрастом. Після того, як кнопка Execute була натиснута відбувається процес сегментації, алгоритм виділяє маски хребців на зображенні. Коли сегментація всіх хребців була виконана, необхідно провести аналіз отриманих хребців. Для цього потрібно визначити ключові параметри: передню висоту, середню висоту та задню висоту. З метою більш детального аналізу, програма обчислює співвідношення AVH/PVN та MVN/PVN , які дозволяють діагностувати можливі деформації чи патології. Існує декілька способів для отримання значень висот [126-128].

В даному випадку буде запропоновано метод визначення передньої висоти, середньої висоти та задньої висоти за допомогою проведення паралельних відрізків.

На рисунку 4.2 відображено весь процес знаходження висот. На першому етапі необхідно побудувати мінімальний прямокутник який оточує хребець. Другий етап починається із знаходження нижньої сторони цього прямокутника (зелений відрізок) та проведення синього відрізка.

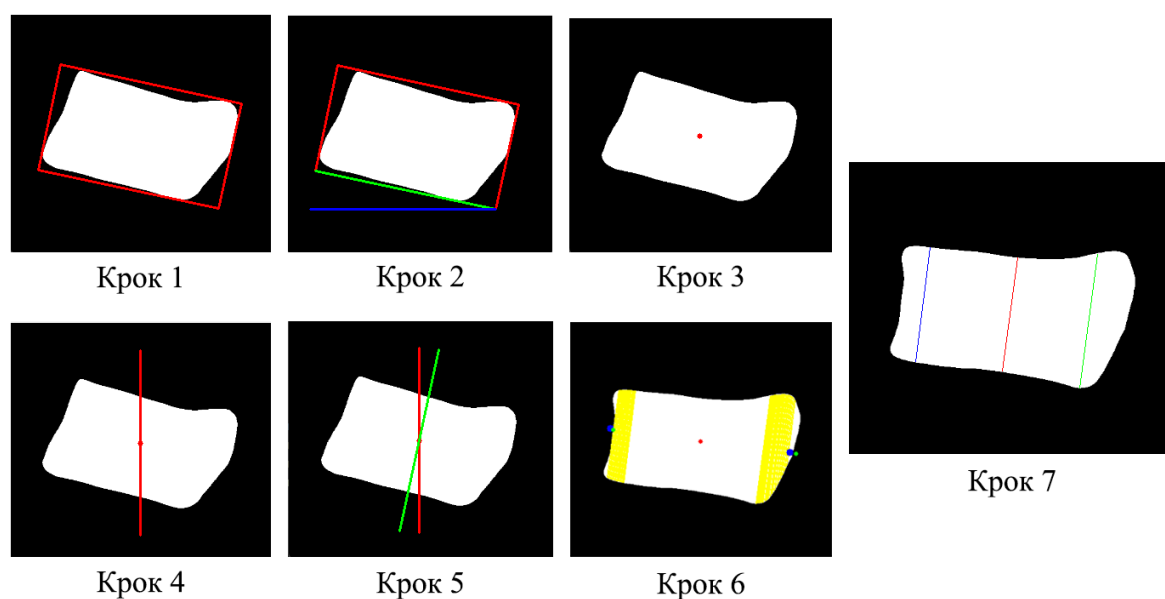


Рис. 4.2. Процес знаходження діагностичних параметрів хребця

На цьому етапі за допомогою цих двох відрізків можна отримати кут α між ними. На третьому кроці необхідно визначити центр хребця. Після чого на четвертому етапі проводиться відрізок через центр хребця. Так, як кут проведеного відрізка через центр хребця не відповідає куту положення хребця, необхідно повернути червоний відрізок на кут α отриманий раніше, в результаті чого на п'ятому кроці отримаємо зелений відрізок. На шостому кроці необхідно побудувати таку кількість паралельних відрізків до середнього відрізка, які перекриватимуть весь хребець. Після чого необхідно залишити 20 % відрізків ліворуч та 30 % відрізків праворуч. Ці відрізки позначені жовтим кольором на шостому кроці. Для того щоб отримати ліву висоту, візьмемо крайній відрізок з лівої внутрішньої сторони, аналогічно зробимо для правої сторони. На сьомому кроці отримаємо хребець з трьома відрізками, де синій відрізок – AVH, червоний відрізок – це MVH та зелений відрізок – це PVH. На рисунку 4.3 показано перемикання між різними етапами алгоритму. Дане зображення демонструє можливість перегляду стану кожного етапу. Користувач має змогу переконатися в доцільності використання ансамблевого алгоритму на вказаному етапі. Такий підхід дає змогу зрозуміти якість виконаної сегментації.

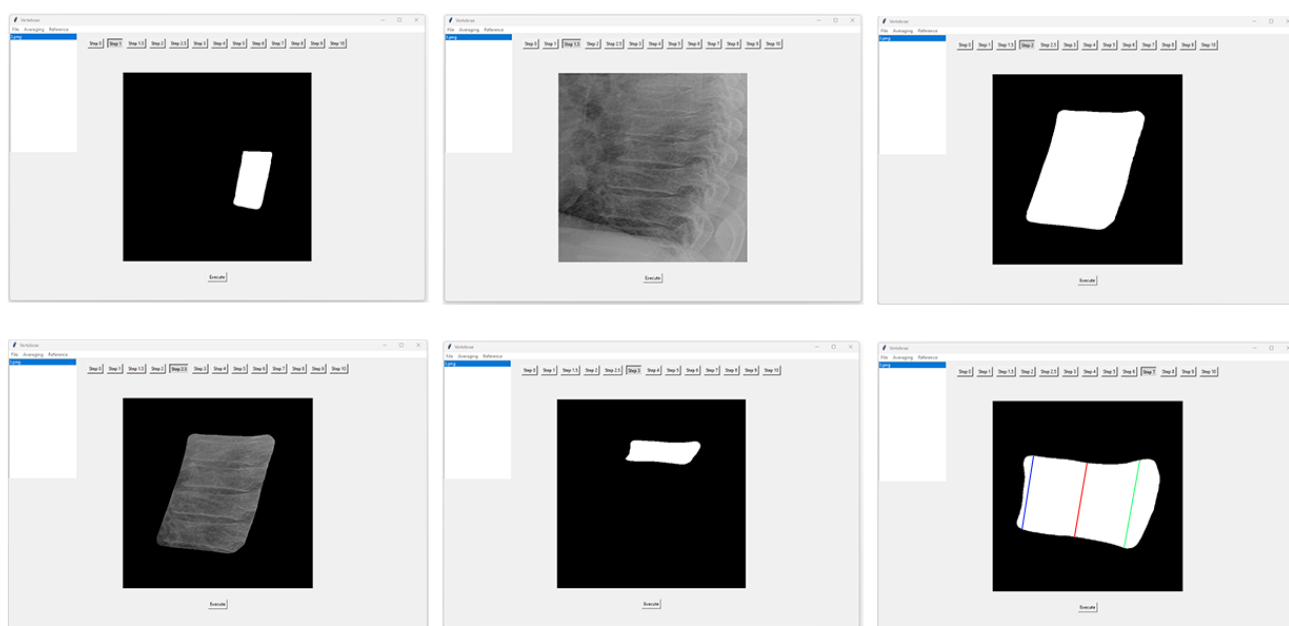


Рис. 4.3. Відображення зображень на різних етапах

Отримані дані виводяться на екран у вигляді графічної інформації: відповідні лінії висот відображаються поверх масок. Також усі обчислені числові дані автоматично зберігаються у файл формату Excel. У звіті кожен рядок відповідає окремому зображенню, а колонки включають такі дані: назва зображення, AVH, MVH, PVH, співвідношення AVH/PVH та MVH/PVH. На рисунку 4.4 зображено приклад звіту формату Excel для трьох зображень.

	A	B	C	D	E	F	G	H
1	Number	Vertebra	Height			Ratio		
2			AVH	MVH	PVH	AVH/PVH	MVH/PVH	
3	00000139.png	Th8	116.499	114.337	122.577	0.95042	0.93278	
4		Th9	115.521	112.588	116.709	0.98982	0.96469	
5		Th10	125	118.68	125.96	0.99238	0.9422	
6		Th11	128.32	130.269	128.32	1	1.01519	
7	00000141.png	Th8	115.248	121.565	117.444	0.9813	1.03509	
8		Th9	115.948	112.792	117.881	0.9836	0.95683	
9		Th10	120.814	115.767	133.462	0.90523	0.86742	
10		Th11	121.078	123.309	140.769	0.86012	0.87596	
11	00000144.png	Th8	107.201	116.4	123.471	0.86823	0.94274	
12		Th9	126.827	122.711	130.943	0.96857	0.93714	
13		Th10	133.15	128.736	131.244	1.01452	0.98089	
14		Th11	131.552	130.602	135.665	0.96968	0.96268	

Рис. 4.4. Відображення діагностичних параметрів хребців у Excel файлі

Таким чином розроблене програмне забезпечення не лише автоматизує рутинні обчислення, але й значно спрощує аналіз, дозволяючи зосередитися на діагностичних висновках та ухваленні рішень.

Висновки до четвертого розділу

В цьому розділі було запропоновано комп'ютерну автоматизовану систему для отримання діагностичних параметрів хребців грудної клітки, яка може покращити медичну діагностику. Ця система дозволяє користувачу загрузити рентгенівські зображення грудної клітки в боковій проекції та отримувати сегментовані хребці, а також обчислювати критично важливі параметри, такі як AVH, MVH і PVH. Окрім цього, система забезпечує обчислення співвідношень

AVH/PVH та MVH/PVH, які зберігаються у вигляді таблиць у форматі Excel, що є зручним для аналізу та подальшого використання.

Реалізація функціоналу включає інтерактивний графічний інтерфейс, що дозволяє користувачу завантажувати, як одне так і кілька зображень для аналізу. Користувач має змогу переглядати результати обробки на різних етапах, перемикаючись між вкладками. Це значно спрощує процес візуалізації отриманих результатів.

Система є важливим інструментом для об'єктивізації діагностики та підвищення її точності, що сприяє прийняттю обґрунтованих клінічних рішень. Вона також забезпечує зниження ручної праці, мінімізуючи людський фактор, і пов'язані з ним помилки. Застосування таких рішень відкриває нові можливості для розвитку медичних досліджень та практики, формуючи підґрунтя для інтеграції штучного інтелекту в охорону здоров'я.

ВИСНОВКИ

У дисертаційній роботі розв'язано актуальну науково-прикладну задачу модифікації існуючих та створенні нових ансамблевих алгоритмів з метою підвищення ефективності сегментації зображень низької якості.

Отримано наступні результати дослідження:

- Проведено аналіз існуючих класичних алгоритмів сегментації зображень, виявлено їхні переваги та обмеження при роботі з низькоякісними зображеннями;
- Досліджено ансамблеві алгоритми сегментації, зокрема, їх вплив на підвищення точності результатів. Отримані дані свідчать про недоцільність використання окремих ансамблевих алгоритмів для вирішення задач сегментації зображень низької якості;
- Запропоновано використання адаптивного алгоритму уточнення контуру на основі яскравості сусідніх пікселів та ансамблевого алгоритму, що дало змогу покращити результати сегментації: на 1.87 % для ансамблевого методу усереднення з порогом $1/2$, на 0.29 % для методу об'єднання масок;
- Запропоновано новий підхід в комбінуванні ансамблевих алгоритмів, що дало змогу об'єднувати різні кількості ансамблевих алгоритмів для підвищення точності сегментації в порівнянні з використанням окремих ансамблевих алгоритмів. Завдяки чому стало можливим врахувати специфіку кожного ансамблевого алгоритму та поліпшити точність сегментації низькоякісних зображень. Запропонована комбінація алгоритму усереднення форм об'єктів та ансамблевого алгоритму об'єднання забезпечила покращення точності сегментації на 0.79 %;
- Запропоновані алгоритми на основі центрування, під час дослідження продемонстрували найкращі результати саме для другого набору даних, що дозволило підвищити середню точність сегментації на 1.84 %;
- Запропоновано новий підхід в використанні ансамблевих алгоритмів та багатоетапного алгоритму, що дозволяє використовувати його, як універсальний метод для вирішення проблем сегментації низькоякісних зображень, не дивлячись

на використанні нейронні мережі, проведені експериментальні дослідження цього підходу підтвердили ефективність його використання для будь-яких нейронних мереж, що дало змогу підвищувати точність сегментації низькоякісних зображень в будь-яких випадках. Реалізація двоетапного підходу дозволила досягти середнього покращення на 5.73 %, реалізація багатоетапного алгоритму сприяла отриманню середнього покращення на 12.82 % для тих самих хребців;

- Проведено експериментальні дослідження на великій кількості нейронних мереж, які підтвердили ефективність використання запропонованих алгоритмів для конкретних нейронних мереж у порівнянні з існуючими аналогами;

- Створено комп'ютерну автоматизовану систему, яка реалізує запропоновані алгоритми та надає можливість отримувати діагностичні параметри хребців, зокрема AVH, MVH, PVH та розраховувати їх співвідношення.

Результати роботи є вагомим внеском у розвиток технологій обробки низькоякісних медичних зображень, забезпечуючи підвищення якості діагностики та спрощення процесу отримання важливих параметрів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Yang, Y., Lv, H. & Chen, N. (2023). A Survey on ensemble learning under the era of deep learning. *Artificial Intelligence Review*, 56, 5545–5589. <https://doi.org/10.1007/s10462-022-10283-5>
2. Zhang, H., Qie, Y. (2023). Applying Deep Learning to Medical Imaging: A Review. *Applied Sciences*, 13(18), 10521. <https://doi.org/10.3390/app131810521>
3. St. Göb, S., Sawant, F.X., Erick, C., Schmidkonz, A., Ramming, M., Lang, E.W., Wittenberg, T., & Götz, Th.I. (2023). Comparing ensemble methods combined with different aggregating models using micrograph cell segmentation as an initial application example. *Journal of Pathology Informatics*, 14, 100304. <https://doi.org/10.1016/j.jpi.2023.100304>
4. Krizhevsky, A., Sutskever, I., & Hinton, G.E. (2012). ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, 60, 84 - 90. <https://doi.org/10.1145/3065386>
5. Sonka, M., Hlavac, V., & Boyle, R. (2014). Image Processing, Analysis, and Machine Vision (4th ed.). *Cengage Learning*.
6. Castleman, K. R. (1996). Digital Image Processing. *Prentice-Hall*.
7. Otsu, N. (1979). A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1), 62–66. <https://doi.org/10.1109/TSMC.1979.4310076>
8. Jaeger, S., Candemir, S., Antani, S., Wáng, Y. X. J., Lu, P. X., & Thoma, G. (2014). Two public chest X-ray datasets for computer-aided screening of pulmonary diseases. *Quantitative Imaging in Medicine and Surgery*, 4(6), 475–477. <https://doi.org/10.3978/j.issn.2223-4292.2014.11.20>
9. Zheng, X., Lei, Q., Yao, R., & others. (2018). Image segmentation based on adaptive K-means algorithm. *Journal of Image and Video Processing*, 2018(68). <https://doi.org/10.1186/s13640-018-0309-3>

10. Adams, R., & Bischof, L. (1994). Seeded region growing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(6), 641–647. <https://doi.org/10.1109/34.295913>
11. Liu, H., Zhang, W., Wang, F., Sun, X., Wang, J., Wang, C., & Wang, X. (2023). Application of an improved watershed algorithm based on distance map reconstruction in bean image segmentation. *Heliyon*, 9(4), e15097. <https://doi.org/10.1016/j.heliyon.2023.e15097>
12. Wu, Y., & Li, Q. (2022). The Algorithm of Watershed Color Image Segmentation Based on Morphological Gradient. *Sensors*, 22(21), 8202. <https://doi.org/10.3390/s22218202>
13. Zaki, N., Qin, W., & Krishnan, A. (2023). Graph-based methods for cervical cancer segmentation: Advancements, limitations, and future directions. *AI Open*, 4, 42–55. <https://doi.org/10.1016/j.aiopen.2023.08.006>
14. Rother, C., Kolmogorov, V., & Blake, A. (2004). "GrabCut": Interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics (TOG)*, 23(3), 309–314. <https://doi.org/10.1145/1015706.1015720>
15. LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521, 436–444. <https://doi.org/10.1038/nature14539>
16. Li, M., Jiang, Y., Zhang, Y., & Zhu, H. (2023). Medical image analysis using deep learning algorithms. *Frontiers in Public Health*, 11. <https://doi.org/10.3389/fpubh.2023.1273253>
17. Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*. <https://arxiv.org/abs/1409.1556>
18. Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional networks for biomedical image segmentation. *Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 234–241. https://doi.org/10.1007/978-3-319-24574-4_28

19. Wang, R., Lei, T., Cui, R., Zhang, B., Meng, H., & Nandi, A. K. (2022). Medical image segmentation using deep learning: A survey. *IET Image Processing*, 16(5), 1243-1267. <https://doi.org/10.1049/ipr2.12419>
20. Yu, Y., Wang, C., Fu, Q., Kou, R., Huang, F., Yang, B., Yang, T., & Gao, M. (2023). Techniques and Challenges of Image Segmentation: A Review. *Electronics*, 12(5), 1199. <https://doi.org/10.3390/electronics12051199>
21. Opitz, D., & Maclin, R. (1999). Popular Ensemble Methods: An Empirical Study. *Journal of Artificial Intelligence Research*, 11, 169–198.
22. Dietterich, T. G. (2000). Ensemble Methods in Machine Learning. *International Workshop on Multiple Classifier Systems*, 1–15.
23. Breiman, L. (1996). Bagging predictors. *Machine Learning*, 24(2), 123–140. <https://doi.org/10.1007/BF00058655>
24. Breiman, L. (2001). Random Forests. *Machine Learning*, 45(1), 5–32. <https://doi.org/10.1023/A:1010933404324>
25. Zhou, Z.-H. (2012). Ensemble Methods: Foundations and Algorithms. *CRC Press*.
26. Freund, Y., & Schapire, R.E. (1999). A Short Introduction to Boosting. *Journal of Japanese Society for Artificial Intelligence*, 14(5), 771-780.
27. Freund, Y., & Schapire, R. E. (1997). A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. *Journal of Computer and System Sciences*, 55(1), 119–139. <https://doi.org/10.1006/jcss.1997.1504>
28. Witten, I. H., Frank, E., & Hall, M. A. (2011). Data Mining: Practical Machine Learning Tools and Techniques. *Elsevier*.
29. Schapire, R. E. (1999). Theoretical views of boosting and applications. In *Lecture Notes in Computer Science*, 46–59.
30. Caruana, R., & Niculescu-Mizil, A. (2006). An Empirical Comparison of Supervised Learning Algorithms. *Proceedings of the 23rd International Conference on Machine Learning*, 161–168.
31. Kuncheva, L. I. (2004). Combining Pattern Classifiers: Methods and Algorithms. *Wiley-Interscience*.

32. Rokach, L. (2010). Ensemble-based classifiers. *Artificial Intelligence Review*, 33(1–2), 1–39.
33. de Zarzà, I., de Curtò, J., Hernández-Orallo, E., & Calafate, C. T. (2023). Cascading and Ensemble Techniques in Deep Learning. *Electronics*, 12(15), 3354. <https://doi.org/10.3390/electronics12153354>
34. Saadati, D., Manzari, O.N., & Mirzakuchaki, S. (2023). Dilated-UNet: A Fast and Accurate Medical Image Segmentation Approach using a Dilated Transformer and U-Net Architecture. *ArXiv*, *abs/2304.11450*.
35. An, S., Xu, L., & Deng, Z. (2025). FastUNet: Fast hierarchical multi-patch underwater enhancement network for industrial recirculating aquaculture. *Pattern Recognition*, 157, 110928. <https://doi.org/10.1016/j.patcog.2024.110928>
36. Cao, H., Wang, Y., Chen, J., Jiang, D., Zhang, X., Tian, Q., & Wang, M. (2021). Swin-Unet: Unet-like Pure Transformer for Medical Image Segmentation. *ECCV Workshops*. https://doi.org/10.1007/978-3-031-25066-8_9
37. Huang, H., Lin, L., Tong, R., Hu, H., Zhang, Q., Iwamoto, Y., Han, X., Chen, Y., & Wu, J. (2020). UNet 3+: A Full-Scale Connected UNet for Medical Image Segmentation. *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1055-1059.
38. Ni, Z.-L., Bian, G.-B., Zhou, X.-H., Hou, Z.-G., Xie, X.-L., Wang, C., Zhou, Y.-J., Li, R.-Q., & Li, Z. (2019). RAUNet: Residual Attention U-Net for Semantic Segmentation of Cataract Surgical Instruments. In T. Gedeon, K. Wong, & M. Lee (Eds.), *Neural Information Processing* (pp. 117-126). ICONIP 2019. *Lecture Notes in Computer Science*, vol 11954. Springer, Cham. https://doi.org/10.1007/978-3-030-36711-4_13
39. Oktay, O., Schlemper, J., Folgoc, L.L., Lee, M.J., Heinrich, M.P., Misawa, K., Mori, K., McDonagh, S.G., Hammerla, N.Y., Kainz, B., Glocker, B., & Rueckert, D. (2018). Attention U-Net: Learning Where to Look for the Pancreas. *ArXiv*, *abs/1804.03999*.
40. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S.E., Anguelov, D., Erhan, D., Vanhoucke, V., & Rabinovich, A. (2014). Going deeper with convolutions. *2015 IEEE*

Conference on Computer Vision and Pattern Recognition (CVPR), 1-9.
<https://doi.org/10.1109/CVPR.2015.7298594>

41. He, K., Zhang, X., Ren, S., & Sun, J. (2015). Deep Residual Learning for Image Recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770-778. <https://doi.org/10.1109/cvpr.2016.90>

42. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Identity Mappings in Deep Residual Networks. *European Conference on Computer Vision*.
https://doi.org/10.1007/978-3-319-46493-0_38

43. Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., & Adam, H. (2017). MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *ArXiv, abs/1704.04861*.

44. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., & Guo, B. (2021). Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 9992-10002.
<https://doi.org/10.1109/ICCV48922.2021.00986>

45. Stanford AIMI. (n.d.). *CheXpert Chest X-Rays*. AIMI Center.
<https://aimi.stanford.edu/datasets/chexpert-chest-x-rays>

46. Vindr.ai. (n.d.). *Vindr.ai datasets: SpineXR*. Vindr.ai.
<https://vindr.ai/datasets/spinexr>

47. Kjelle, E., & Chilanga, C. (2022). The assessment of image quality and diagnostic value in X-ray images: a survey on radiographers' reasons for rejecting images. *Insights into imaging*, 13(1), 36. <https://doi.org/10.1186/s13244-022-01169-9>

48. Конюхов, В. Д. (2024). Використання методів штучного інтелекту для діагностування серцево-судинних захворювань за допомогою кардіоторакального індексу. *Інформаційне суспільство: технологічні, економічні та технічні аспекти становлення*, 9-10 липня 2024 р., Тернопіль. Тернопіль, Випуск 90, 21-23.

49. Koniukhov, V. D. (2024). Ensemble method based on averaging shapes of objects using the pyramid method. *Radio Electronics, Computer Science, Control*, (4), 113–120. <https://doi.org/10.15588/1607-3274-2024-4-11>

50. Конюхов, В. Д. (2024). Ансамблеві алгоритми машинного навчання для сегментації зображень низької якості. *Тези доповідей конференції молодих вчених та спеціалістів ІПМаш НАН України «Сучасні проблеми машинобудування–2024»* (с. 6–7). Харків: Інститут енергетичних машин і систем ім. А. М. Підгорного.

51. Конюхов, В. Д. (2024). Ансамблеві методи на основі центрування для сегментації зображення. *Комп'ютерні системи та мережі*, 6(2), 122-130. <https://doi.org/10.23939/csn2024.02.122>

52. Charles, J. J., Kuncheva, L. I., Wells, B., & Lim, I. S. (2006). An evaluation measure of image segmentation based on object centres. In A. Campilho & M. S. Kamel (Eds.), *Image analysis and recognition* (Vol. 4141, pp. 232-240). Lecture Notes in Computer Science. Springer. https://doi.org/10.1007/11867586_27

53. Chen, H., & Zheng, H. (2020). Object Detection Based on Center Point Proposals. *Electronics*, 9(12), 2075. <https://doi.org/10.3390/electronics9122075>

54. Gahramanova, A. (2019). Locating centers of mass with image processing. *Undergraduate Journal of Mathematical Modeling: One + Two*, 10(1), Article 1. <https://doi.org/10.5038/2326-3652.10.1.4906>

55. Schober, P., Boer, C., & Schwarte, L. A. (2018). Correlation coefficients: Appropriate use and interpretation. *Anesthesia & Analgesia*, 126(5), 1763-1768. <https://doi.org/10.1213/ANE.0000000000002864>

56. Dakhare, S., Chowhan, H., & Chandak, M.B. (2014). Combined Approach for Image Segmentation. *ArXiv*, *abs/1406.0074*. <https://doi.org/10.14445/22312803/IJCTT-V11P125>

57. Jiao, X., Chen, Y., & Dong, R. (2020). An unsupervised image segmentation method combining graph clustering and high-level feature representation. *Neurocomputing*, 409, 83–92. <https://doi.org/10.1016/j.neucom.2020.05.073>

58. Cote, M., & Saeedi, P. (2014). Hierarchical image segmentation using a combined geometrical and feature-based approach. *Journal of Data Analysis and Information Processing*, 2(4), 117–136. <https://doi.org/10.4236/jdaip.2014.24014>

59. Murashov, D. M. (2022). Method for combining image segmentation maps on the basis of information redundancy and variation of information minimization.

Optoelectronics and Instrumentation Processing, 58, 457–464.
<https://doi.org/10.3103/S8756699022050119>

60. Ghane, N., Vard, A., Talebi, A., & Nematollahy, P. (2017). Segmentation of White Blood Cells From Microscopic Images Using a Novel Combination of K-Means Clustering and Modified Watershed Algorithm. *Journal of medical signals and sensors*, 7(2), 92–101.

61. Ma, Y., & Zhang, G. (2022). Noisy image segmentation model combining PDE and improved variational level set method. *Academic Journal of Mathematical Sciences*, 3(1). <https://doi.org/10.25236/AJMS.2022.030106>

62. Băluțoiu, M. A., Sturzu, D., Boiangiu, C.-A., Voncilă, M.-L., Tarbă, N., & Vlăsceanu, G. V. (2021). On how to combine image segmentation algorithms using entropy. In *2021 20th RoEduNet Conference: Networking in Education and Research (RoEduNet)*, Iasi, Romania, 1-5. <https://doi.org/10.1109/RoEduNet54112.2021.9638272>

63. Monteiro, F. C., & Campilho, A. C. (2005). Spectral methods in image segmentation: A combined approach. In J. S. Marques, N. Pérez de la Blanca, & P. Pina (Eds.), *Pattern recognition and image analysis. IbPRIA 2005* (Vol. 3523, pp. 271-278). Springer. https://doi.org/10.1007/11492542_24

64. Dagher, I., & Abboud, E. (2024). Combining contour-based and region-based in image segmentation. *F1000Research*, 12, 1312. <https://doi.org/10.12688/f1000research.140872.3>

65. Aljahdali, S., & Zanaty, E. A. (2008). Combining multiple segmentation methods for improving the segmentation accuracy. In *2008 IEEE Symposium on Computers and Communications* (pp. 649-653). IEEE. <https://doi.org/10.1109/ISCC.2008.4625766>

66. Yang, T., Zhou, S., Xu, A., & Yin, J. (2020). A Method for Tree Image Segmentation Combined Adaptive Mean Shifting with Image Abstraction. *Journal of Information Processing Systems*, 16(6), 1424–1436. <https://doi.org/10.3745/JIPS.02.0151>

67. Koniukhov, V. D. (2024). Adaptive refinement of segmented object contour based on the brightness of neighboring pixels using the ensemble method. *Journal of*

Mechanical Engineering – Problemy Mashynobuduvannia, 27(4), 73–78.
<https://doi.org/10.15407/pmach2024.04.073>

68. Zhu, C., Zhang, X., Li, Y., Qiu, L., Han, K., & Han, X. (2022). SharpContour: A contour-based boundary refinement approach for efficient and accurate instance segmentation. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 4382–4391). <https://doi.org/10.1109/CVPR52688.2022.00435>

69. Aouat, S., Ait-hammi, I., & Hamouchene, I. (2021). A new approach for texture segmentation based on the Gray Level Co-occurrence Matrix. *Multimedia Tools and Applications*, 80(24027–24052). <https://doi.org/10.1007/s11042-021-10634-4>

70. Ma, D., Liao, Q., Chen, Z., Liao, R., & Ma, H. (2019). Adaptive local-fitting-based active contour model for medical image segmentation. *Signal Processing: Image Communication*, 76, 201–213. <https://doi.org/10.1016/j.image.2019.05.006>

71. Judah, A., Hu, B., & Wang, J. (2014). An algorithm for boundary adjustment toward multi-scale adaptive segmentation of remotely sensed imagery. *Remote Sensing*, 6(5), 3583–3610. <https://doi.org/10.3390/rs6053583>

72. Bransby, K. M., Bajaj, R., Ramasamy, A., Çap, M., Yap, N., Slabaugh, G., Bourantas, C., & Zhang, Q. (2024). POLYCORE: Polygon-based contour refinement for improved intravascular ultrasound segmentation. *Computers in Biology and Medicine*, 182, 109162. <https://doi.org/10.1016/j.compbiomed.2024.109162>

73. Zia, H., Soomro, S., & Choi, K. N. (2024). Image segmentation using bias correction active contours. *IEEE Access*, 12, 60641–60655. <https://doi.org/10.1109/ACCESS.2024.3391052>

74. Koniukhov, V. (2024). Improving the segmentation of the vertebrae using a multi-stage machine learning algorithm. *Radioelectronic and Computer Systems*, 2024(4), 83–90. <https://doi.org/10.32620/reks.2024.4.07>

75. Koniukhov, V. D. (2024). Using an ensemble of neural networks for determining the diagnostic parameters of the vertebrae. *Journal of Mechanical Engineering – Problemy Mashynobuduvannia*, 27(1), 56–61. <https://doi.org/10.15407/pmach2024.01.056>

76. Alsenan, A., Youssef, B. B., & Alhichri, H. (2021). A deep learning model based on MobileNetV3 and UNet for spinal cord gray matter segmentation. In *2021 44th International Conference on Telecommunications and Signal Processing (TSP)* (pp. 244-248). <https://doi.org/10.1109/TSP52935.2021.9522652>
77. Koniukhov, V. D., & Ugrimov, S. V. (2024). Increasing the accuracy of determining the cardiothoracic ratio with the help of an ensemble of neural networks. *Journal of Mechanical Engineering – Problemy Mashynobuduvannia*, 27(2), 54–60. <https://doi.org/10.15407/pmach2024.02.054>
78. Roussel, N. (2018). Denoising of Dual Energy X-ray Absorptiometry Images and Vertebra Segmentation.
79. Kim, D. H., Jeong, J. G., Kim, Y. J., et al. (2021). Automated vertebral segmentation and measurement of vertebral compression ratio based on deep learning in X-ray images. *Journal of Digital Imaging*, 34(5), 853–861. <https://doi.org/10.1007/s10278-021-00471-0>
80. Chen, L., Yu, Z., Huang, J., Shu, L., Kuosmanen, P., Shen, C., Ma, X., Li, J., Sun, C., Li, Z., Shu, T., & Yu, G. (2023). Development of lung segmentation method in X-ray images of children based on TransResUNet. *Frontiers in Radiology*, 3, 1190745. <https://doi.org/10.3389/fradi.2023.1190745>
81. Saeed, M. U., Dikaio, N., Dastgir, A., Ali, G., Hamid, M., & Hajjej, F. (2023). An Automated Deep Learning Approach for Spine Segmentation and Vertebrae Recognition Using Computed Tomography Images. *Diagnostics*, 13(16), 2658. <https://doi.org/10.3390/diagnostics13162658>
82. Zhang, Z., et al. (2023). Introducing shape prior module in diffusion model for medical image segmentation. In *2023 6th International Conference on Mechatronics, Robotics and Automation (ICMRA)* (pp. 185-190). IEEE. <https://doi.org/10.1109/ICMRA59796.2023.10708363>
83. Chang, H., Gao, G., Chen, Y., & Zheng, H. (2024). Multi-task contexture learning network for automated vertebrae segmentation and tumor diagnosis from MRI. *Computers and Electrical Engineering*, 113, 109032. <https://doi.org/10.1016/j.compeleceng.2023.109032>

84. Tran, V.L., Lin, H.Y., & Liu, H.W. (2021). MBNet: A multi-task deep neural network for semantic segmentation and lumbar vertebra inspection on X-ray images. In H. Ishikawa, C.L. Liu, T. Pajdla, & J. Shi (Eds.), *Computer Vision – ACCV 2020* (Vol. 12626, pp. 560–577). Springer. https://doi.org/10.1007/978-3-030-69541-5_38
85. Wang, Z., Zhu, J., Fu, S., Mao, S., & Ye, Y. (2023). RFPNet: Reorganizing feature pyramid networks for medical image segmentation. *Computers in Biology and Medicine*, 163, 107108. <https://doi.org/10.1016/j.compbiomed.2023.107108>
86. Zhang, D., Chen, B., & Li, S. (2021). Sequential conditional reinforcement learning for simultaneous vertebral body detection and segmentation with modeling the spine anatomy. *Medical image analysis*, 67, 101861. <https://doi.org/10.1016/j.media.2020.101861>
87. Chen, Y., Mo, Y., Readie, A., et al. (2024). VertXNet: An ensemble method for vertebral body segmentation and identification from cervical and lumbar spinal X-rays. *Scientific Reports*, 14, 3341. <https://doi.org/10.1038/s41598-023-49923-3>
88. Cina, A., Bassani, T., Panico, M., et al. (2021). 2-step deep learning model for landmarks localization in spine radiographs. *Scientific Reports*, 11, 9482. <https://doi.org/10.1038/s41598-021-89102-w>
89. Pisov, M., Kondratenko, V., Zakharov, A., Petraikin, A., Gomboleviskiy, V.A., Morozov, S.P., & Belyaev, M. (2020). Keypoints Localization for Joint Vertebra Detection and Fracture Severity Quantification. *International Conference on Medical Image Computing and Computer-Assisted Intervention*. https://doi.org/10.1007/978-3-030-59725-2_70
90. Liaskos, M., Savelonas, M. A., Asvestas, P. A., Lykissas, M. G., & Matsopoulos, G. K. (2020). Bimodal CT/MRI-Based Segmentation Method for Intervertebral Disc Boundary Extraction. *Information*, 11(9), 448. <https://doi.org/10.3390/info11090448>
91. Zhang, S., Chen, M., Wu, J., Zhang, Z., Li, T., Xue, C., & Kong, Y. (2024). SpineCLUE: Automatic Vertebrae Identification Using Contrastive Learning and Uncertainty Estimation. <https://doi.org/10.48550/arXiv.2401.07271>

92. Rasoulia, A., Rohling, R. N., & Abolmaesumi, P. (2013). A statistical multi-vertebrae shape+pose model for segmentation of CT images. *Proceedings of SPIE 8671, Medical Imaging 2013: Image-Guided Procedures, Robotic Interventions, and Modeling*, 86710P. <https://doi.org/10.1117/12.2007448>
93. Rasoulia, A., Rohling, R., & Abolmaesumi, P. (2013). Lumbar spine segmentation using a statistical multi-vertebrae anatomical shape+pose model. *IEEE transactions on medical imaging*, 32(10), 1890–1900. <https://doi.org/10.1109/TMI.2013.2268424>
94. Suzani, A., Rasoulia, A., Fels, S., Rohling, R. N., & Abolmaesumi, P. (2014). Semi-automatic segmentation of vertebral bodies in volumetric MR images using a statistical shape+pose model. *Proceedings of SPIE 9036, Medical Imaging 2014: Image-Guided Procedures, Robotic Interventions, and Modeling*, 90360P. <https://doi.org/10.1117/12.2043847>
95. Cheng, L.-W., Chou, H.-H., Cai, Y.-X., Huang, K.-Y., Hsieh, C.-C., Chu, P.-L., Cheng, I.-S., & Hsieh, S.-Y. (2024). Automated detection of vertebral fractures from X-ray images: A novel machine learning model and survey of the field. *Neurocomputing*, 566, 126946. <https://doi.org/10.1016/j.neucom.2023.126946>
96. Al Arif, S. M. M. R., Knapp, K., & Slabaugh, G. (2018). Fully automatic cervical vertebrae segmentation framework for X-ray images. *Computer methods and programs in biomedicine*, 157, 95–111. <https://doi.org/10.1016/j.cmpb.2018.01.006>
97. Meng, D., Mohammed, E., Boyer, E., & Pujades, S. (2021). Vertebrae Localization, Segmentation and Identification Using a Graph Optimization and an Anatomic Consistency Cycle. *Computerized medical imaging and graphics: the official journal of the Computerized Medical Imaging Society*, 107, 102235. https://doi.org/10.1007/978-3-031-21014-3_32
98. Chen, C., Namdar, K., Wu, Y., Hosseinpour, S., Shroff, M.M., Doria, A.S., & Khalvati, F. (2022). Automating Cobb Angle Measurement for Adolescent Idiopathic Scoliosis using Instance Segmentation. <https://doi.org/10.48550/arXiv.2211.14122>
99. Zhang, D., Chen, B., & Li, S. (2021). Sequential conditional reinforcement learning for simultaneous vertebral body detection and segmentation with modeling the

spine anatomy. *Medical image analysis*, 67, 101861.
<https://doi.org/10.1016/j.media.2020.101861>

100. Seo, J. W., Lim, S. H., Jeong, J. G., et al. (2021). A deep learning algorithm for automated measurement of vertebral body compression from X-ray images. *Scientific Reports*, 11, 13732. <https://doi.org/10.1038/s41598-021-93017-x>

101. Masood, R. F., Taj, I. A., Khan, M. B., Qureshi, M. A., & Hassan, T. (2022). Deep learning based vertebral body segmentation with extraction of spinal measurements and disorder disease classification. *Biomedical Signal Processing and Control*, 71(Part B), 103230. <https://doi.org/10.1016/j.bspc.2021.103230>

102. Moller, H.K., Graf, R., Schmitt, J., Keinert, B., Atad, M., Sekuboyina, A.K., Streckenbach, F., Schon, H., Kofler, F., Kroencke, T., Bette, S., Willich, S.N., Keil, T., Niendorf, T., Pischon, T., Endemann, B., Menze, B.H., Rueckert, D., & Kirschke, J.S. (2024). SPINEPS - Automatic Whole Spine Segmentation of T2-weighted MR images using a Two-Phase Approach to Multi-class Semantic and Instance Segmentation. *European radiology*. <https://doi.org/10.1007/s00330-024-11155-y>

103. Zhang, J., Liu, F., Xu, J., Zhao, Q., Huang, C., Yu, Y., & Yuan, H. (2023). Automated detection and classification of acute vertebral body fractures using a convolutional neural network on computed tomography. *Frontiers in Endocrinology*, 14, 1132725. <https://doi.org/10.3389/fendo.2023.1132725>

104. Benjelloun, M., & Mahmoudi, S. (2009). Spine localization in X-ray images using interest point detection. *Journal of digital imaging*, 22(3), 309–318. <https://doi.org/10.1007/s10278-007-9099-3>

105. Sa, R., Owens, W., Wiegand, R., & Chaudhary, V. (2016). Fast scale-invariant lateral lumbar vertebrae detection and segmentation in X-ray images. In *Proceedings of the 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 1054–1057. <https://doi.org/10.1109/EMBC.2016.7590884>

106. Koniukhov, V. D., Morgun, O. M., & Nemchenko, K. E. (2024). Impact of preprocessing and comparison of neural network ensemble methods for segmentation of the thoracic spine in X-ray images. *Radio Electronics, Computer Science, Control*, (4), 102–112. <https://doi.org/10.15588/1607-3274-2024-4-10>

107. Fruehwirt, W., & Duckworth, P. (2021). Towards better healthcare: What could and should be automated? *Technological Forecasting and Social Change*, 172, 120967. <https://doi.org/10.1016/j.techfore.2021.120967>
108. Parciak, M., Suhr, M., Schmidt, C., et al. (2023). FAIRness through automation: Development of an automated medical data integration infrastructure for FAIR health data in a maximum care university hospital. *BMC Medical Informatics and Decision Making*, 23, 94. <https://doi.org/10.1186/s12911-023-02195-3>
109. Göndöcs, D., & Dörfler, V. (2024). AI in medical diagnosis: AI prediction & human judgment. *Artificial Intelligence in Medicine*, 149, 102769. <https://doi.org/10.1016/j.artmed.2024.102769>
110. Ansari, S., Sajjad, F., Ul-Qayyum, Z., Naveed, N., & Shafi, I. (2013). Diagnosis of Vertebral Column Disorders Using Machine Learning Classifiers. *2013 International Conference on Information Science and Applications (ICISA)*, 1-6. <https://doi.org/10.1109/ICISA.2013.6579446>
111. Palchunov, D., Yakhyayeva, G., & Yasinskaya, O. (2015). Software system for the diagnosis of the spine diseases using case-based reasoning. In *2015 International Conference on Biomedical Engineering and Computational Technologies (SIBIRCON)* (pp. 205-210). Novosibirsk, Russia. <https://doi.org/10.1109/SIBIRCON.2015.7361884>
112. Oktay, A. B., Albayrak, N. B., & Akgul, Y. S. (2014). Computer aided diagnosis of degenerative intervertebral disc diseases from lumbar MR images. *Computerized Medical Imaging and Graphics*, 38(7), 613-619. <https://doi.org/10.1016/j.compmedimag.2014.04.006>
113. Azimi, P., Yazdanian, T., Benzel, E. C., Aghaei, H. N., Azhari, S., Sadeghi, S., & Montazeri, A. (2020). A Review on the Use of Artificial Intelligence in Spinal Diseases. *Asian spine journal*, 14(4), 543–571. <https://doi.org/10.31616/asj.2020.0147>
114. Chen, K., Zheng, L., Zhao, H., & Wang, Z. (2024). Deep Learning-Based Intelligent Diagnosis of Lumbar Diseases with Multi-Angle View of Intervertebral Disc. *Mathematics*, 12(13), 2062. <https://doi.org/10.3390/math12132062>

115. Xuan, J., Ke, B., Ma, W., Liang, Y., & Hu, W. (2023). Spinal disease diagnosis assistant based on MRI images using deep transfer learning methods. *Frontiers in public health*, 11, 1044525. <https://doi.org/10.3389/fpubh.2023.1044525>
116. Maki, S., Furuya, T., Inoue, M., Shiga, Y., Inage, K., Eguchi, Y., Orita, S., & Ohtori, S. (2024). Machine Learning and Deep Learning in Spinal Injury: A Narrative Review of Algorithms in Diagnosis and Prognosis. *Journal of Clinical Medicine*, 13(3), 705. <https://doi.org/10.3390/jcm13030705>
117. Xiao, B., Zhu, M., Du, E., Liu, W., Ma, J., Huang, H., Gong, J., Diacinti, D., Zhang, K., Gao, B., Liu, H., Jiang, R., Ji, Z., Xiong, X., He, L., Wu, L., Xu, C., Du, M., Wang, X., Chen, L., Wu, K., Yang, L., Xu, M., Diacinti, D., Dou, Q., Kwok, T., & Wáng, Y. (2022). A software program for automated compressive vertebral fracture detection on elderly women's lateral chest radiograph: Ofeye 1.0. *Quantitative Imaging In Medicine And Surgery*, 12(8), 4259-4271. doi:10.21037/qims-22-433
118. Lee, S., Lee, S., Seo, D., Yoo, K.-H., & Kim, S. (2015). Development of ontology and 3D software for the diseases of spine. *Advances in Multimedia*, 2015, 420848. <https://doi.org/10.1155/2015/420848>
119. Pelc, M., Vilimkova Kahankova, R., Blaszcyszyn, M., et al. (2023). Initial study on an expert system for spine diseases screening using inertial measurement unit. *Scientific Reports*, 13, 10440. <https://doi.org/10.1038/s41598-023-36798-7>
120. Panchenko, O. A., Śliwiński, Z., Radchenko, S. M., & Kabantseva, A. V. (2015). Hardware-software medical rehabilitation complex for vertebral diseases patients. *Fizjoterapia Polska*, 15, 6-11.
121. Python Software Foundation. (2019, October 14). *Python 3.8.0 release notes*. Python.org. <https://www.python.org/downloads/release/python-380/>
122. NVIDIA. (2021). *CUDA 11.2.0 download archive*. <https://developer.nvidia.com/cuda-11.2.0-download-archive>
123. TensorFlow. (n.d.). *TensorFlow version documentation*. <https://www.tensorflow.org/versions>
124. Chollet, F. (n.d.). *Keras: Deep learning for humans*. <https://keras.io/>

125. Python Software Foundation. (n.d.). *Tkinter — Python interface to Tcl/Tk*. <https://docs.python.org/3/library/tkinter.html>
126. Kanezaki, S., Miyazaki, M., Ishihara, T., Notani, N., & Tsumura, H. (2018). Magnetic resonance imaging evaluation of intervertebral disc injuries can predict kyphotic deformity after posterior fixation of unstable thoracolumbar spine injuries. *Medicine*, 97(e11442). <https://doi.org/10.1097/MD.00000000000011442>
127. Marie-Hardy, L., Mohsinaly, Y., Pietton, R., Stencel-Allemand, M., Khalifé, M., Bonaccorsi, R., Barut, N., & Pascal-Moussellard, H. (2022). Efficiency of a novel vertebral body augmentation system (Tektona™) in non-osteoporotic spinal fractures. *BMC Musculoskeletal Disorders*, 23, 10.1186/s12891-022-05272-2. <https://doi.org/10.1186/s12891-022-05272-2>
128. Hsu, W. E., Su, K. C., Chen, K. H., Pan, C. C., Lu, W. H., & Lee, C. H. (2019). The Evaluation of Different Radiological Measurement Parameters of the Degree of Collapse of the Vertebral Body in Vertebral Compression Fractures. *Applied bionics and biomechanics*, 2019, 4021640. <https://doi.org/10.1155/2019/4021640>

ДОДАТКИ

Додаток А

Список опублікованих праць за темою дисертації

Статті у наукових фахових виданнях, які включені до наукометричних баз даних (Scopus, Web of Science):

1. Koniukhov, V. D., Morgun, O. M., & Nemchenko, K. E. (2024). Impact of preprocessing and comparison of neural network ensemble methods for segmentation of the thoracic spine in X-ray images. *Radio Electronics, Computer Science, Control*, (4), 102–112. <https://doi.org/10.15588/1607-3274-2024-4-10>
2. Koniukhov, V. D. (2024). Ensemble method based on averaging shapes of objects using the pyramid method. *Radio Electronics, Computer Science, Control*, (4), 113–120. <https://doi.org/10.15588/1607-3274-2024-4-11>
3. Koniukhov, V. (2024). Improving the segmentation of the vertebrae using a multi-stage machine learning algorithm. *Radioelectronic and Computer Systems*, 2024(4), 83-90. <https://doi.org/10.32620/reks.2024.4.07>

Статті у наукових фахових виданнях України:

4. Koniukhov, V. D. (2024). Using an ensemble of neural networks for determining the diagnostic parameters of the vertebrae. *Journal of Mechanical Engineering – Problemy Mashynobuduvannia*, 27(1), 56–61. <https://doi.org/10.15407/pmach2024.01.056>
5. Koniukhov, V. D., & Ugrimov, S. V. (2024). Increasing the accuracy of determining the cardiothoracic ratio with the help of an ensemble of neural networks. *Journal of Mechanical Engineering – Problemy Mashynobuduvannia*, 27(2), 54–60. <https://doi.org/10.15407/pmach2024.02.054>
6. Koniukhov, V. D. (2024). Adaptive refinement of segmented object contour based on the brightness of neighboring pixels using the ensemble method. *Journal of*

Mechanical Engineering – Problemy Mashynobuduvannia, 27(4), 73–78.
<https://doi.org/10.15407/pmach2024.04.073>

7. Конюхов, В. Д. (2024). Ансамблеві методи на основі центрування для сегментації зображення. *Комп'ютерні системи та мережі*, 6(2), 122-130.
<https://doi.org/10.23939/csn2024.02.122>

Список публікацій, які засвідчують апробацію матеріалів дисертації:

8. Конюхов, В. Д. (2024). Використання методів штучного інтелекту для діагностування серцево-судинних захворювань за допомогою кардіоторакального індексу. *Інформаційне суспільство: технологічні, економічні та технічні аспекти становлення: Матеріали Міжнародної наукової інтернет-конференції (Тернопіль, Україна; Ополе, Польща, 9–10 липня 2024 р.)* (Вип. 90, с. 21–23).

9. Конюхов, В. Д. (2024). Ансамблеві алгоритми машинного навчання для сегментації зображень низької якості. *Тези доповідей конференції молодих вчених та спеціалістів ІПМаш НАН України «Сучасні проблеми машинобудування–2024»* (с. 6–7). Харків: Інститут енергетичних машин і систем ім. А. М. Підгорного.

Апробація результатів дослідження


1. Дев'ята науково-технічна міжнародна конференція “Комп'ютерне моделювання в наукоємних технологіях (КМНТ-2023)”, м. Харків, 2023.

2. Міжнародна наукова інтернет-конференція “Інформаційне суспільство: технологічні, економічні та технічні аспекти ставлення”, м. Тернопіль-Ополе, 2024.

3. Конференції молодих вчених та спеціалістів “Сучасні проблеми машинобудування - 2024”, м. Харків, 2024.

4. Десята науково-технічна міжнародна конференція “Комп'ютерне моделювання в наукоємних технологіях (КМНТ-2024)”, м. Харків, 2024.

Додаток Б



ЗАТВЕРДЖУЮ
 Директор навчально-наукового інституту
 комп'ютерної фізики та енергетики
 Харківського національного університету
 імені В.Н.Каразіна
 Ірина ГАРЯЧЕВСЬКА
 20.01.2025

АКТ
 впровадження результатів дисертаційної роботи
 Конюхова Владислава Дмитровича

Комісія у складі: заступник директора з наукової роботи ННІ КФЕ - к.ф.-м.н., доц. Марушенко Ілля Миколайович, заступник директора з навчальної роботи ННІ КФЕ – доктор філософії Протектор Денис Олегович, завідувач кафедри Інформаційних технологій в фізико-енергетичних системах - к.ф.-м.н., доц. Сухов Руслан Володимирович, склали акт про впровадження результатів дисертаційної роботи Конюхова В.Д. Дмитровича на тему «Ансамблеві алгоритми машинного навчання для сегментації зображень низької якості», подану на здобуття наукового ступеню доктора філософії за спеціальністю 113 - «Прикладна математика» у навчальний процес навчально-наукового інституту комп'ютерної фізики та енергетики Харківського національного університету імені В. Н. Каразіна.

Склад впровадження:

- Алгоритм адаптивного уточнення контурів об'єктів на зображеннях із високим рівнем шуму, що враховує яскравість сусідніх пікселів і використовує ансамблеві методи для підвищення точності сегментації.
- Методи комбінування ансамблевих підходів у паралельній та послідовній схемах для підвищення точності діагностичних оцінок.
- Програмний комплекс для автоматизованого аналізу рентгенівських знімків, який забезпечує точне визначення діагностичних параметрів грудних хребців.

Комісія встановила, що результати дисертаційної роботи Конюхова В.Д. були впроваджені у навчальний процес кафедри інформаційних технологій в фізико-енергетичних системах навчально-наукового інституту комп'ютерної фізики та енергетики Харківського національного університету імені В. Н. Каразіна при проведенні лекційних, лабораторних та практичних занять з курсів «Вступ до обробки даних» і «Обробка зображень та сигналів» для студентів 3 курсу бакалаврату та «Наближені методи розв'язання задач математичної фізики» і «Обробка даних фізичних експериментів» для студентів 1 курсу магістратури спеціальності 105 - «Прикладна фізика та наноматеріали».




Голова комісії		Ілля МАРУЩЕНКО
Члени комісії		Денис ПРОТЕКТОР
		Руслан СУХОВ

Рис. Б.1. Акт впровадження у навчальний процес кафедри інформаційних технологій в фізико-енергетичних системах навчально-наукового інституту комп'ютерної фізики та енергетики Харківського національного університету імені В. Н. Каразіна

А К Т
про впровадження результатів
дисертаційної роботи доктора філософії
у виробництво

Даним актом стверджується, що результати дисертаційної на тему «Ансамблеві алгоритми машинного навчання для сегментації зображень низької якості», що представлена на здобуття наукового ступеня доктора філософії виконаної Конюховим Владиславом Дмитровичем впроваджені у ТОВ «Радіопром».

1. Вид впроваджуваних результатів: методика, алгоритми, розроблені бібліотеки зі втіленими авторськими алгоритмами;

2. Новизна отриманих результатів: у дисертації запропоновано новий підхід до сегментації зображень, який включає алгоритм адаптивного уточнення контуру на основі аналізу яскравості сусідніх пікселів із використанням ансамблевих методів. Введено комбіновані ансамблеві методи, що поєднують усереднення за формою об'єктів та ансамблевий метод об'єднання. На основі запропонованих методів створено Python-додаток та окрему бібліотеку для визначення діагностичних параметрів грудних хребців, у якому сегментація зображень виконується із застосуванням розроблених алгоритмів.

3. Практичне впровадження результатів: розроблені в дисертації алгоритми були реалізовані у вигляді окремої програмної бібліотеки, яка була застосована в медичних дослідженнях для діагностики захворювань за реальними рентгенівськими зображеннями

4. Значущість отриманих результатів: отримані результати відкривають нові можливості для медичної діагностики, значно підвищуючи точність аналізу зображень та ефективність автоматизованих методів. Вони сприяють удосконаленню медичної візуалізації, розширюючи її застосування та підвищуючи рівень діагностичних технологій, що має критичне значення для сучасної медицини.

Заступник директора
з наукової роботи

ІЄМС НАЦІОНАЛЬНОЇ АКАДЕМІЇ

ІНСТИТУТ
ЕНЕРГЕТИЧНИХ
СИСТЕМ
ІМ.А.М.ПІДГОРНОГО
НАЦІОНАЛЬНОЇ
АКАДЕМІЇ
№03534570
УКРАЇНА

12 лютого 2025 р.
м.п.

Здобувач

Владислав КОНЮХОВ

11 лютого 2025 р.

Директор ТОВ «РАДІОПРОМ»

Олег МОРГУН
10 лютого 2025 р.
м.п.



Рис. Б.2. Акт впровадження ТОВ «РАДІОПРОМ»

ВІДГУК
на дисертацію Конюхова Владислава Дмитровича
«Ансамблеві алгоритми машинного навчання для сегментації зображень
низької якості», представлену на здобуття наукового ступеня доктора
філософії за спеціальністю 113 – Прикладна математика

Завдання, поставлене у цій дисертації, полягає у розробці та практичному використанні ансамблевих алгоритмів для сегментації зображень низької якості, що є актуальним напрямком у сучасному машинному навчанні. Сегментація зображень відіграє ключову роль у різних сферах комп'ютерного зору, включаючи обробку супутникових знімків, робототехніку, а також медичну діагностику, де особливо важливо точно аналізувати зображення, отримані з рентгенографії чи МРТ. Проте низька якість зображень, обумовлена наявністю шуму, низьким контрастом та відсутністю чітких контурів, створює значні труднощі у цьому процесі. У цій роботі розглядаються ансамблеві методи, які поєднують результати декількох моделей, що дозволяє значно покращити точність сегментації в умовах низької якості зображень. Використання ансамблю моделей, на відміну від застосування однієї, дає змогу компенсувати слабкі сторони окремих алгоритмів та підвищити їхню ефективність. Основними перевагами ансамблевих підходів є підвищена стабільність і точність, зменшення ймовірності перенавчання та зниження варіативності результатів. Це дозволяє уникнути ситуацій, коли одна модель значною мірою впливає на кінцевий результат, що особливо важливо при роботі з неоднорідними даними.

Разом із тим, сегментація зображень низької якості залишається складним завданням, що супроводжується рядом викликів. Наприклад, нечіткі або відсутні контури об'єктів ускладнюють класифікацію пікселів, а наявність шуму та артефактів може спотворювати результати сегментації. У рамках цієї дисертації для подолання таких проблем застосовано ансамблеві глибокі нейронні мережі, зокрема модифіковані архітектури U-Net, адаптовані для комбінування переваг різних моделей.

Ансамблеві методи мають широке практичне застосування, зокрема у медичній діагностиці, де вони дозволяють точно сегментувати органи та патологічні утворення на зображеннях низької якості. Крім того, вони можуть бути використані для аналізу знімків, отриманих з дронів чи супутників, що є корисним у різних наукових і прикладних дослідженнях.

Основна мета цієї дисертації – розробка та практичне використання ансамблевих підходів для обробки рентгеновських зображень легенів та хребців з метою покращення діагностики певних захворювань. Отримані результати підтверджують ефективність ансамблевих методів у вирішенні задач сегментації та вже використані в певних системах медичної діагностики.

Таким чином, вивчення дисертації, актуальність проведених досліджень, відповідність отриманих результатів поставленим цілям, суттєвий добуток та новизна результатів, а також наявність практичного застосування дозволяє зробити висновок, що автор дисертації Владислав Конюхов заслуговує на присвоєння йому ступня доктора філософії за спеціальністю «Прикладна математика».

Директор ТОВ «РАДІОПРОМ»
кандидат фіз.-мат. наук




Рис. Б.3. Відгук на дисертацію

Додаток В

Таблиця В.1

Результати ансамблевих алгоритмів для Dilated-UNet, DICE %

Кількість моделей у ансамблі	Набір даних	Алгоритми без центрування					Алгоритми з центруванням				
		АУП $T=1/2$	АУП $T=2/3$	АУФО	ААО	АУП $T=1$	АУП $T=1/2$	АУП $T=2/3$	АУФО	ААО	АУП $T=1$
2	1	95.94	95.53	95.79	96.06	95.41	95.26	95.26	95.63	95.87	95.26
	2	79.50	77.28	79.12	81.35	76.52	76.96	76.96	79.48	80.57	76.96
3	1	95.92	95.79	95.90	95.93	95.25	95.96	95.96	95.86	95.81	95.17
	2	81.28	80.36	80.00	81.71	76.48	81.17	81.17	80.70	81.17	76.53
4	1	95.96	95.74	95.88	95.90	95.05	95.63	95.63	95.77	95.74	94.90
	2	81.08	80.27	79.39	81.10	75.53	80.32	80.32	80.53	80.77	75.45
5	1	96.00	95.75	95.91	95.71	94.90	95.88	95.52	95.80	95.54	94.79
	2	81.42	79.88	80.34	80.72	74.80	81.23	79.59	81.03	80.75	74.60
6	1	95.96	95.79	95.89	95.62	94.76	95.79	95.79	95.78	95.46	94.61
	2	81.50	80.47	80.35	80.43	74.37	80.99	80.99	81.11	80.60	74.05
7	1	95.96	95.79	95.82	95.59	94.01	95.88	95.68	95.77	95.38	93.90
	2	81.67	80.64	80.55	80.03	74.00	81.60	80.83	81.34	80.50	73.52
8	1	95.93	95.76	95.86	95.14	93.84	95.80	95.58	95.77	94.83	93.64
	2	81.57	80.51	80.51	80.02	73.75	81.37	80.52	81.30	80.38	73.02
9	1	95.95	95.76	95.82	95.05	93.77	95.85	95.68	95.72	94.63	93.55
	2	81.60	80.47	80.58	79.76	73.38	81.48	81.17	81.37	80.23	72.74
10	1	95.98	95.74	95.85	94.80	93.39	95.74	95.59	95.71	94.28	92.97
	2	81.64	80.49	80.65	79.66	73.27	81.39	80.92	81.40	80.09	72.43

Таблиця В.2

Результати ансамблевих алгоритмів для FastUNet, DICE %

Кількість моделей у ансамблі	Набір даних	Алгоритми без центрування					Алгоритми з центруванням				
		АУП $T=1/2$	АУП $T=2/3$	АУФО	ААО	АУП $T=1$	АУП $T=1/2$	АУП $T=2/3$	АУФО	ААО	АУП $T=1$
2	1	90.68	90.09	90.99	91.73	90.25	89.03	89.03	90.63	90.93	89.03
	2	70.36	67.71	71.70	75.27	68.77	69.64	69.64	72.21	73.96	69.64
3	1	90.32	90.98	91.09	91.74	89.88	90.15	90.15	90.60	90.48	87.21
	2	70.70	67.41	70.64	75.63	64.47	72.79	72.79	71.54	74.69	65.21
4	1	90.76	90.22	91.20	89.66	89.69	89.91	89.91	90.56	88.39	86.79
	2	70.73	68.38	71.06	75.40	62.09	70.85	70.85	71.84	74.69	63.26

Продовження таблиці В.2

5	1	91.13	91.43	91.22	88.84	88.83	90.45	89.12	90.62	87.92	86.96
	2	70.03	66.61	70.76	75.28	60.33	72.34	68.60	71.80	74.54	61.38
6	1	90.39	91.33	90.93	88.53	88.60	89.91	89.91	90.24	87.40	86.07
	2	70.08	66.64	70.97	75.27	59.80	70.76	70.76	71.75	74.35	60.51
7	1	90.54	91.16	91.00	86.84	88.57	89.98	89.57	90.34	86.25	85.78
	2	70.35	66.84	71.20	74.40	57.83	72.22	69.68	71.95	73.36	58.72
8	1	90.60	90.67	90.79	86.71	87.96	89.89	89.00	90.18	85.98	84.87
	2	70.31	67.40	70.25	73.42	56.43	70.44	68.13	70.88	72.52	56.84
9	1	90.58	90.65	90.97	86.72	84.66	90.25	89.76	90.50	85.83	82.04
	2	69.60	65.10	68.58	73.39	40.78	70.45	68.37	69.77	72.45	41.62
10	1	90.81	90.61	90.73	86.45	83.35	89.87	88.92	90.14	85.35	81.47
	2	69.97	65.37	68.92	72.42	40.13	70.86	69.11	69.97	71.99	41.06

Таблиця В.3

Результати ансамблевих алгоритмів для Swin-Unet, DICE %

Кількість моделей у ансамблі	Набір даних	Алгоритми без центрування					Алгоритми з центруванням				
		АУП $T=1/2$	АУП $T=2/3$	АУФО	ААО	АУП $T=1$	АУП $T=1/2$	АУП $T=2/3$	АУФО	ААО	АУП $T=1$
2	1	85.75	83.75	85.66	84.45	84.70	85.29	85.29	85.61	84.16	85.29
	2	68.00	63.65	67.45	69.93	63.23	64.36	64.36	68.61	69.10	64.36
3	1	85.21	84.54	85.58	83.37	84.65	85.14	85.14	85.48	83.02	85.15
	2	67.11	60.78	65.26	69.31	56.08	65.46	65.46	68.26	66.66	58.82
4	1	85.28	84.77	85.55	82.08	84.56	85.23	85.23	85.37	82.04	85.01
	2	67.63	62.18	65.82	70.37	55.27	64.02	64.02	69.24	67.07	58.07
5	1	85.55	84.78	85.63	82.14	82.25	85.37	85.44	85.61	81.87	82.35
	2	67.51	61.40	66.70	70.51	51.42	65.62	62.85	69.25	67.41	54.72
6	1	85.18	84.53	85.34	81.76	81.68	85.26	85.26	85.29	81.67	82.12
	2	67.05	60.25	65.84	70.24	49.20	64.26	64.00	68.26	67.52	53.04
7	1	84.83	84.47	85.11	81.41	81.19	84.62	84.92	84.86	81.29	81.59
	2	66.62	60.19	66.08	69.94	48.36	65.07	63.17	68.39	67.84	52.49
8	1	84.78	84.35	85.03	81.31	79.87	84.68	84.81	84.79	81.14	80.69
	2	66.54	59.65	64.73	69.69	45.61	64.15	62.06	67.29	68.24	49.82
9	1	84.57	84.19	84.81	80.34	79.82	84.68	84.55	84.67	80.45	80.68
	2	66.54	59.87	64.71	68.44	44.74	65.23	63.21	67.61	67.69	49.13
10	1	84.35	83.95	84.57	79.70	79.65	84.16	84.15	84.22	79.86	80.19
	2	66.63	60.90	66.11	67.96	44.38	64.61	62.83	68.30	67.38	48.72

Таблиця В.4

Результати ансамблевих алгоритмів для UNet 3+, DICE %

Кількість моделей у ансамблі	Набір даних	Алгоритми без центрування					Алгоритми з центруванням				
		АУП $T=1/2$	АУП $T=2/3$	АУФО	ААО	АУП $T=1$	АУП $T=1/2$	АУП $T=2/3$	АУФО	ААО	АУП $T=1$
2	1	91.09	88.90	90.76	93.65	88.70	89.44	89.44	91.17	91.80	89.44
	2	63.74	61.55	68.80	71.76	66.01	65.69	65.69	69.09	71.42	65.69
3	1	91.47	89.78	90.96	93.64	88.36	90.97	90.97	91.19	91.71	88.79
	2	64.35	61.00	67.10	73.18	61.84	68.34	68.34	68.16	72.04	62.56
4	1	92.36	88.67	90.86	92.98	87.02	89.74	89.74	91.00	91.09	87.07
	2	63.41	60.83	66.32	73.74	58.80	66.76	66.76	67.54	72.30	59.67
5	1	93.44	90.23	91.20	90.42	86.90	91.35	88.88	90.99	89.27	86.43
	2	64.22	61.42	65.90	73.75	56.70	68.38	65.66	67.30	72.98	57.50
6	1	92.49	90.19	90.83	90.43	84.46	89.88	89.88	90.69	89.14	84.12
	2	64.07	60.34	65.31	73.81	52.86	66.82	66.61	66.93	72.89	53.41
7	1	91.92	90.06	90.65	89.52	84.36	90.33	89.07	90.47	88.11	84.03
	2	64.61	61.18	66.07	74.62	51.95	68.25	66.31	67.54	73.83	52.53
8	1	91.81	89.82	90.33	89.25	83.44	89.55	88.33	90.16	87.71	83.06
	2	64.39	60.52	65.64	74.67	50.39	67.02	64.71	67.43	73.82	51.37
9	1	91.16	89.12	89.53	89.10	81.30	89.70	88.76	89.60	87.21	80.77
	2	63.27	58.11	63.95	74.65	38.71	67.08	64.90	66.27	73.65	41.14
10	1	90.66	88.37	88.24	89.02	74.46	88.68	87.69	88.73	86.72	73.50
	2	61.74	55.06	62.18	74.55	31.00	66.11	63.44	64.93	73.34	33.68

Таблиця В.5

Результати ансамблевих алгоритмів для RAUNet, DICE %

Кількість моделей у ансамблі	Набір даних	Алгоритми без центрування					Алгоритми з центруванням				
		АУП $T=1/2$	АУП $T=2/3$	АУФО	ААО	АУП $T=1$	АУП $T=1/2$	АУП $T=2/3$	АУФО	ААО	АУП $T=1$
2	1	91.70	92.00	92.65	90.53	92.51	90.91	90.91	91.91	90.36	90.91
	2	67.81	64.09	70.37	75.27	66.90	67.87	67.87	71.14	73.89	67.87
3	1	91.53	91.56	92.09	90.02	91.32	91.46	91.46	91.69	89.48	89.67
	2	70.34	67.36	69.57	74.54	63.88	71.44	71.44	70.61	73.18	65.50
4	1	91.69	91.52	91.65	89.83	89.17	91.29	91.29	91.57	89.03	87.60
	2	68.19	64.78	68.17	74.55	58.06	68.15	68.15	69.12	73.42	59.54
5	1	92.28	91.07	91.47	89.60	88.00	91.57	89.70	91.16	88.59	86.23
	2	66.38	61.16	66.28	74.56	52.88	68.84	64.02	67.51	73.02	53.87

Продовження таблиці В.5

6	1	92.16	90.66	91.30	89.58	86.83	91.10	91.10	90.98	88.59	85.03
	2	65.83	60.73	65.62	74.60	50.85	65.60	65.35	67.07	72.84	51.93
7	1	92.16	90.71	91.38	89.08	85.26	91.53	90.34	91.07	88.21	83.56
	2	65.24	59.51	63.81	73.16	46.85	65.56	62.06	65.17	71.48	47.75
8	1	91.94	90.52	91.34	88.13	85.16	90.81	89.34	90.84	87.16	82.95
	2	65.70	60.04	65.00	73.44	46.78	65.03	61.29	65.98	72.08	47.43
9	1	91.61	90.32	91.00	87.99	84.94	90.96	90.05	90.61	86.94	82.72
	2	65.86	60.53	65.11	73.44	46.08	66.07	63.68	66.22	72.26	46.86
10	1	91.81	89.98	90.22	87.90	78.03	90.27	88.82	89.77	86.78	77.81
	2	64.68	58.02	64.04	73.48	37.65	64.71	60.66	65.34	72.28	39.77

Таблиця В.6

Результати ансамблевих алгоритмів для Attention U-Net, DICE %

Кількість моделей у ансамблі	Набір даних	Алгоритми без центрування					Алгоритми з центруванням				
		АУП $T=1/2$	АУП $T=2/3$	АУФО	ААО	АУП $T=1$	АУП $T=1/2$	АУП $T=2/3$	АУФО	ААО	АУП $T=1$
2	1	89.79	86.42	86.72	91.62	83.82	85.71	85.71	87.87	90.09	85.71
	2	56.05	50.11	61.01	67.02	56.20	57.04	57.04	62.29	66.00	57.04
3	1	91.54	89.65	88.68	87.94	83.94	89.05	89.05	88.90	87.22	84.36
	2	63.63	58.20	66.55	74.38	56.03	66.20	66.20	67.76	74.54	56.45
4	1	91.53	88.00	88.10	88.10	81.15	88.28	88.28	88.71	87.25	81.77
	2	63.77	58.39	67.15	74.93	51.92	64.65	64.65	68.08	75.40	53.73
5	1	91.00	87.76	87.91	87.35	78.24	90.21	86.94	88.77	86.23	79.06
	2	60.57	49.35	64.77	69.91	32.68	50.48	46.33	65.94	57.03	34.95
6	1	90.09	86.81	87.25	87.38	74.68	88.41	88.41	88.01	86.18	75.48
	2	56.47	47.32	61.96	69.87	27.95	48.93	48.05	63.64	59.51	31.47
7	1	88.96	85.85	86.38	87.39	71.82	88.79	85.83	87.13	85.97	73.21
	2	53.23	44.36	59.18	69.88	24.45	49.86	45.94	62.12	60.70	28.80
8	1	89.10	84.09	85.08	87.33	70.21	86.76	83.02	86.12	85.72	71.89
	2	50.14	44.03	58.32	69.95	23.38	49.19	44.96	61.18	61.48	28.09
9	1	88.99	83.03	85.04	86.72	68.30	87.51	84.70	86.15	85.36	69.86
	2	49.72	44.50	58.04	69.79	22.61	51.41	47.20	60.66	62.55	28.04
10	1	88.39	82.19	83.06	86.66	66.82	85.44	82.42	84.90	85.05	68.28
	2	48.05	43.11	57.49	69.77	20.91	50.35	46.23	59.84	63.09	25.92

Таблиця В.7

Результати ансамблевих алгоритмів для Inception, DICE %

Кількість моделей у ансамблі	Набір даних	Алгоритми без центрування					Алгоритми з центруванням				
		АУП $T=1/2$	АУП $T=2/3$	АУФО	ААО	АУП $T=1$	АУП $T=1/2$	АУП $T=2/3$	АУФО	ААО	АУП $T=1$
2	1	82.92	77.89	84.76	84.28	82.80	82.41	82.41	84.22	84.13	82.41
	2	59.08	50.93	55.83	61.05	53.02	52.75	52.75	56.61	61.02	52.75
3	1	82.99	77.40	84.19	84.27	80.96	84.02	84.02	84.09	84.38	80.33
	2	59.15	51.12	56.04	63.13	50.68	57.86	57.86	57.57	62.91	49.70
4	1	82.93	76.04	83.89	84.37	79.20	82.51	82.51	83.68	84.35	78.25
	2	57.23	47.97	53.13	63.38	43.44	53.59	53.59	55.04	63.21	41.94
5	1	82.82	75.25	83.65	84.39	77.97	83.62	81.36	83.62	84.48	76.92
	2	56.55	47.10	52.23	63.45	41.80	55.55	49.38	54.43	63.64	39.87
6	1	82.49	74.31	83.27	84.40	76.29	82.41	82.41	83.29	84.41	74.82
	2	55.12	45.25	51.28	63.47	39.46	51.35	51.31	53.52	63.02	37.07
7	1	82.14	73.52	82.86	84.34	74.61	82.94	81.16	82.79	84.28	73.10
	2	54.95	44.78	51.27	63.70	37.77	53.15	48.91	53.31	63.20	35.34
8	1	82.49	75.06	83.48	79.60	74.61	82.40	80.75	83.36	80.57	72.83
	2	56.83	46.85	54.14	66.38	37.67	52.95	48.70	56.18	64.98	35.17
9	1	82.22	74.04	82.87	79.59	71.70	82.44	81.09	82.69	80.25	70.35
	2	56.59	46.43	53.67	66.42	35.91	54.34	50.82	55.73	65.13	33.57
10	1	81.91	72.93	81.97	79.59	67.13	81.79	80.20	82.20	80.50	65.14
	2	55.83	45.22	53.51	66.44	33.90	51.89	48.79	55.38	65.08	31.86

Таблиця В.8

Результати ансамблевих алгоритмів для Resnet101 U-Net, DICE %

Кількість моделей у ансамблі	Набір даних	Алгоритми без центрування					Алгоритми з центруванням				
		АУП $T=1/2$	АУП $T=2/3$	АУФО	ААО	АУП $T=1$	АУП $T=1/2$	АУП $T=2/3$	АУФО	ААО	АУП $T=1$
2	1	91.83	88.78	89.59	90.05	88.08	86.93	86.93	88.92	89.48	86.93
	2	60.36	55.91	60.10	60.72	55.94	55.36	55.36	60.23	56.61	55.36
3	1	91.24	87.83	88.28	88.78	83.07	89.18	89.18	88.19	88.16	84.36
	2	60.48	55.29	60.84	61.78	49.32	56.13	56.13	62.20	57.15	50.18
4	1	89.80	86.85	87.43	88.74	80.20	87.56	87.56	87.66	87.51	80.95
	2	56.81	50.64	59.58	61.66	42.91	54.21	54.21	61.42	56.94	44.65
5	1	88.32	85.39	86.84	88.75	76.77	88.08	84.89	87.37	87.25	77.34
	2	55.12	47.09	59.19	61.72	38.46	55.31	51.41	61.01	57.44	41.41

Продовження таблиці В.8

6	1	89.38	84.92	86.65	87.68	74.34	86.44	86.44	87.17	86.68	75.55
	2	55.44	48.33	58.82	62.96	35.95	54.65	54.32	61.13	58.35	40.04
7	1	90.12	84.43	85.70	87.28	70.00	88.37	85.89	86.87	86.49	73.31
	2	52.81	43.58	58.11	62.89	31.27	54.18	52.05	60.46	58.27	35.28
8	1	89.41	84.88	84.85	87.22	66.31	87.73	83.76	86.30	86.60	69.95
	2	51.58	44.31	57.36	63.66	29.52	54.78	51.16	60.46	58.77	34.55
9	1	89.84	84.48	83.05	85.46	60.36	88.22	85.94	85.44	85.80	64.81
	2	51.87	44.06	56.03	65.75	22.47	55.13	52.07	61.03	59.67	30.42
10	1	89.07	82.24	82.38	85.45	54.82	86.79	83.64	84.86	85.28	62.37
	2	50.01	41.07	56.89	65.75	20.01	54.39	50.65	60.92	59.86	26.95

Таблиця В.9

Результати ансамблевих алгоритмів для Resnet152v2 U-Net, DICE %

Кількість моделей у ансамблі	Набір даних	Алгоритми без центрування					Алгоритми з центруванням				
		АУП $T=1/2$	АУП $T=2/3$	АУФО	ААО	АУП $T=1$	АУП $T=1/2$	АУП $T=2/3$	АУФО	ААО	АУП $T=1$
2	1	95.26	94.38	94.83	94.67	93.94	93.75	93.75	94.65	94.10	93.75
	2	66.19	60.33	72.35	78.25	69.58	70.42	70.42	73.56	76.45	70.42
3	1	95.44	94.43	94.71	94.39	92.28	94.63	94.63	94.40	93.86	91.93
	2	63.32	60.50	70.34	78.01	64.14	73.35	73.35	71.68	76.07	65.27
4	1	95.27	93.95	94.31	94.33	89.71	93.91	93.91	94.15	93.74	89.77
	2	62.01	57.56	68.41	77.63	60.01	70.05	70.05	70.30	75.81	61.59
5	1	95.07	93.48	93.16	94.36	83.51	94.32	93.00	92.98	93.79	83.62
	2	60.35	53.45	67.05	77.35	56.00	70.74	68.86	70.01	75.92	57.55
6	1	95.20	93.75	93.12	82.78	82.20	92.33	92.33	91.80	83.83	81.77
	2	61.50	54.42	68.47	70.83	53.07	71.27	70.03	70.82	70.87	55.90
7	1	95.37	94.20	93.26	73.97	82.16	92.33	91.73	91.92	74.95	81.34
	2	63.34	56.91	66.24	60.22	45.35	64.68	62.53	68.16	58.78	49.13
8	1	95.16	93.03	92.64	73.81	78.00	92.20	91.32	91.70	74.84	77.36
	2	61.17	52.53	66.32	60.20	37.45	64.56	62.26	67.93	58.99	41.45
9	1	94.88	91.99	91.44	69.28	74.26	90.97	90.09	90.25	71.36	73.79
	2	58.53	49.40	67.35	58.59	30.11	62.70	60.88	68.62	57.09	32.26
10	1	94.57	90.08	88.91	69.12	54.09	89.92	88.76	89.24	70.92	54.21
	2	57.23	49.14	67.50	59.12	22.61	62.06	58.57	69.08	57.83	26.07

Таблиця В.10

Результати ансамблевих алгоритмів для Fcn8 MobileNet, DICE %

Кількість моделей у ансамблі	Набір даних	Алгоритми без центрування					Алгоритми з центруванням				
		АУП $T=1/2$	АУП $T=2/3$	АУФО	ААО	АУП $T=1$	АУП $T=1/2$	АУП $T=2/3$	АУФО	ААО	АУП $T=1$
2	1	92.92	94.04	93.73	91.21	94.44	93.44	93.44	93.31	91.09	93.44
	2	77.21	76.74	76.95	76.52	76.67	76.43	76.43	77.11	76.49	76.43
3	1	93.19	94.16	93.79	88.54	94.48	92.60	92.60	92.99	88.06	92.63
	2	76.29	76.57	75.76	74.79	71.78	76.26	76.26	76.51	75.59	71.65
4	1	93.38	94.18	93.42	87.59	93.82	92.62	92.62	92.59	87.21	92.22
	2	75.66	74.97	74.69	74.11	69.10	76.62	76.62	75.99	74.69	69.50
5	1	93.23	94.47	93.14	87.45	92.44	91.96	92.43	92.37	87.10	91.01
	2	74.58	72.10	73.22	73.59	58.62	76.01	73.36	74.88	74.24	61.32
6	1	92.87	94.17	93.00	86.32	91.22	92.25	92.25	92.23	86.02	89.30
	2	74.60	73.43	73.00	73.18	54.99	74.19	73.84	74.93	74.06	59.03
7	1	92.79	93.47	92.81	85.29	90.56	92.05	92.18	92.07	85.15	88.54
	2	73.63	70.72	72.93	72.85	49.98	74.11	71.34	74.97	73.82	52.88
8	1	92.63	93.50	92.69	84.37	90.38	91.95	91.85	91.78	84.32	88.14
	2	74.05	70.43	73.45	71.73	48.90	73.18	70.43	75.31	72.38	51.85
9	1	92.57	93.22	92.62	83.57	88.85	91.60	92.05	91.74	83.57	86.96
	2	73.49	72.27	73.98	71.16	48.28	73.12	71.14	75.37	71.03	51.05
10	1	92.29	93.12	92.42	80.97	88.72	91.64	91.73	91.46	81.22	86.37
	2	73.59	72.39	74.28	66.63	44.45	71.83	69.94	75.57	67.28	49.72

Таблиця В.11

Порівняння середніх значень точності ансамблів, DICE %

Мережа	Набір даних	Алгоритми без центрування					Алгоритми з центруванням				
		АУП $T=1/2$	АУП $T=2/3$	АУФО	ААО	АУП $T=1$	АУП $T=1/2$	АУП $T=2/3$	АУФО	ААО	АУП $T=1$
Dilated-UNet	1	95.96	95.74	95.86	95.53	94.49	95.75	95.63	95.76	95.28	94.31
	2	81.25	80.04	80.16	80.53	74.68	80.73	80.28	80.92	80.56	74.37
FastUNet	1	90.64	90.79	90.99	88.58	87.97	89.94	89.48	90.42	87.61	85.58
	2	70.24	66.83	70.45	74.50	56.74	71.15	69.77	71.30	73.62	57.58
Swin-UNet	1	85.05	84.37	85.25	81.84	82.04	84.94	84.98	85.10	81.72	82.56
	2	67.07	60.98	65.86	69.60	50.92	64.75	63.55	68.36	67.66	54.35
UNet 3+	1	91.82	89.46	90.37	90.89	84.34	89.96	89.20	90.45	89.20	84.13
	2	63.76	60.00	65.69	73.86	52.03	67.16	65.82	67.24	72.92	53.06

Продовження таблиці В.11

RAUNet	1	91.87	90.93	91.46	89.19	86.80	91.10	90.33	91.07	88.35	85.16
	2	66.67	61.80	66.44	74.12	52.21	67.03	64.95	67.57	72.72	53.39
Attention U-Net	1	89.93	85.98	86.47	87.83	75.44	87.80	86.04	87.40	86.56	76.62
	2	55.74	48.82	61.61	70.61	35.13	54.23	51.84	63.50	64.48	38.28
Inception	1	82.54	75.16	83.44	82.76	76.14	82.73	81.76	83.33	83.04	74.91
	2	56.81	47.30	53.46	64.16	41.52	53.72	51.35	55.31	63.57	39.70
Resnet101 U-Net	1	89.89	85.53	86.09	87.71	72.66	87.70	86.03	86.98	87.03	75.06
	2	54.94	47.81	58.55	62.99	36.21	54.90	53.04	60.99	58.12	39.87
Resnet152v2 U-Net	1	95.14	93.26	92.93	82.97	81.13	92.71	92.17	92.34	83.49	80.84
	2	61.52	54.92	68.22	68.91	48.70	67.76	66.33	70.02	67.53	51.07
Fcn8 MobileNet	1	92.87	93.81	93.07	86.14	91.66	92.23	92.35	92.28	85.97	89.85
	2	74.79	73.29	74.25	72.73	58.09	74.64	73.26	75.63	73.28	60.38

Таблиця В.12

Порівняння максимальних значень точності ансамблів, DICE %

Мережа	Набір даних	Алгоритми без центрування					Алгоритми з центруванням				
		АУП $T=1/2$	АУП $T=2/3$	АУФО	ААО	АУП $T=1$	АУП $T=1/2$	АУП $T=2/3$	АУФО	ААО	АУП $T=1$
Dilated-UNet	1	96.00	95.79	95.91	96.06	95.41	95.96	95.96	95.86	95.87	95.26
	2	81.67	80.64	80.65	81.71	76.52	81.60	81.17	81.40	81.17	76.96
FastUNet	1	91.13	91.43	91.22	91.74	90.25	90.45	90.15	90.63	90.93	89.03
	2	70.73	68.38	71.70	75.63	68.77	72.79	72.79	72.21	74.69	69.64
Swin-Unet	1	85.75	84.78	85.66	84.45	84.70	85.37	85.44	85.61	84.16	85.29
	2	68.00	63.65	67.45	70.51	63.23	65.62	65.46	69.25	69.10	64.36
UNet 3+	1	93.44	90.23	91.20	93.65	88.70	91.35	90.97	91.19	91.80	89.44
	2	64.61	61.55	68.80	74.67	66.01	68.38	68.34	69.09	73.83	65.69
RAUNet	1	92.28	92.00	92.65	90.53	92.51	91.57	91.46	91.91	90.36	90.91
	2	70.34	67.36	70.37	75.27	66.90	71.44	71.44	71.14	73.89	67.87
Attention U-Net	1	91.54	89.65	88.68	91.62	83.94	90.21	89.05	88.90	90.09	85.71
	2	63.77	58.39	67.15	74.93	56.20	66.20	66.20	68.08	75.40	57.04
Inception	1	82.99	77.89	84.76	84.40	82.80	84.02	84.02	84.22	84.48	82.41
	2	59.15	51.12	56.04	66.44	53.02	57.86	57.86	57.57	65.13	52.75

Продовження таблиці В.12

Resnet101	1	91.83	88.78	89.59	90.05	88.08	89.18	89.18	88.92	89.48	86.93
U-Net	2	60.48	55.91	60.84	65.75	55.94	56.13	56.13	62.20	59.86	55.36
Resnet152v2	1	95.44	94.43	94.83	94.67	93.94	94.63	94.63	94.65	94.10	93.75
U-Net	2	66.19	60.50	72.35	78.25	69.58	73.35	73.35	73.56	76.45	70.42
Fcn8	1	93.38	94.47	93.79	91.21	94.48	93.44	93.44	93.31	91.09	93.44
MobileNet	2	77.21	76.74	76.95	76.52	76.67	76.62	76.62	77.11	76.49	76.43

Таблиця В.13

Результати комбінованих ансамблевих алгоритмів для Dilated-UNet, DICE %

Кількість моделей у ансамблі	Набір даних	Комбінація						
		1	2	3	4	5	6	7
2	1	95.94	95.90	96.04	96.06	95.87	95.78	95.85
	2	79.50	80.21	81.38	81.35	81.04	79.70	81.05
3	1	95.92	95.87	95.84	95.93	95.72	95.90	95.63
	2	81.28	81.74	81.69	81.71	81.00	80.76	80.99
4	1	95.96	95.93	95.86	95.90	95.68	95.86	95.63
	2	81.08	81.49	81.09	81.10	80.36	80.53	80.34
5	1	96.00	96.03	95.68	95.71	95.43	95.94	95.40
	2	81.42	81.83	80.71	80.72	79.97	81.16	79.95
6	1	95.96	95.97	95.56	95.62	95.30	95.91	95.25
	2	81.50	81.75	80.43	80.43	79.70	81.12	79.69
7	1	95.96	95.96	95.53	95.59	95.20	95.89	95.14
	2	81.67	82.02	80.03	80.03	79.43	81.33	79.43
8	1	95.93	95.95	95.14	95.14	94.50	95.91	94.50
	2	81.57	81.94	80.02	80.02	79.41	81.25	79.41
9	1	95.95	95.94	95.05	95.05	94.27	95.87	94.27
	2	81.60	82.00	79.76	79.76	79.06	81.38	79.06
10	1	95.98	95.93	94.80	94.80	93.77	95.87	93.77
	2	81.64	81.99	79.66	79.66	78.96	81.41	78.96

Таблиця В.14

Результати комбінованих ансамблевих алгоритмів для FastUNet, DICE %

Кількість моделей у ансамблі	Набір даних	Комбінація						
		1	2	3	4	5	6	7
2	1	90.68	90.71	91.73	91.73	91.04	90.99	91.04
	2	70.36	73.10	75.16	75.27	74.78	72.38	74.68
3	1	90.32	90.32	91.74	91.74	90.61	91.06	90.61
	2	70.70	73.87	75.53	75.63	75.32	71.79	75.24
4	1	90.76	90.57	89.66	89.66	88.08	91.12	88.08
	2	70.73	73.60	75.38	75.40	74.97	72.13	74.96
5	1	91.13	90.58	88.84	88.84	87.05	91.04	87.05
	2	70.03	73.50	75.26	75.28	74.83	71.91	74.88
6	1	90.39	90.17	88.53	88.53	86.59	90.70	86.59
	2	70.08	73.24	75.24	75.27	74.82	71.93	74.90
7	1	90.54	90.16	86.84	86.84	85.00	90.73	85.00
	2	70.35	73.38	74.39	74.40	73.45	72.08	73.62
8	1	90.60	90.23	86.71	86.71	84.81	90.56	84.81
	2	70.31	72.02	73.42	73.42	72.61	71.03	72.77
9	1	90.58	90.33	86.72	86.72	84.65	90.91	84.65
	2	69.60	71.63	73.39	73.39	72.43	70.09	72.61
10	1	90.81	90.43	86.45	86.45	84.17	90.60	84.17
	2	69.97	72.18	72.42	72.42	71.55	70.27	71.70

Таблиця В.15

Результати комбінованих ансамблевих алгоритмів для Swin-Unet, DICE %

Кількість моделей у ансамблі	Набір даних	Комбінація						
		1	2	3	4	5	6	7
2	1	85.75	85.76	84.59	84.45	84.12	85.78	84.25
	2	68.00	68.72	69.75	70.64	69.95	69.02	70.55
3	1	85.21	84.99	83.37	83.37	82.78	85.50	82.78
	2	67.11	68.31	69.14	71.32	68.61	68.56	71.10
4	1	85.28	85.14	82.08	82.08	81.46	85.39	81.46
	2	67.63	68.25	70.19	71.43	69.14	69.82	70.49
5	1	85.55	85.42	82.14	82.14	81.06	85.71	81.06
	2	67.51	68.49	70.45	71.98	69.18	69.84	71.24

Продовження таблиці В.15

6	1	85.18	85.16	81.76	81.76	80.88	85.31	80.88
	2	67.05	67.91	70.23	71.41	69.14	68.56	70.75
7	1	84.83	84.64	81.41	81.41	80.56	85.05	80.56
	2	66.62	67.87	69.91	71.12	69.04	68.63	70.41
8	1	84.78	84.69	81.31	81.31	80.50	84.97	80.50
	2	66.54	67.87	69.65	70.67	69.04	67.61	70.22
9	1	84.57	84.58	80.34	80.34	79.49	84.77	79.49
	2	66.54	68.15	68.39	69.36	67.55	67.89	68.43
10	1	84.35	84.23	79.70	79.70	78.80	84.37	78.80
	2	66.63	67.75	67.89	69.25	66.89	68.72	68.17

Таблиця В.16

Результати комбінованих ансамблевих алгоритмів для UNet 3+, DICE %

Кількість моделей у ансамблі	Набір даних	Комбінація						
		1	2	3	4	5	6	7
2	1	91.09	91.98	93.65	93.65	93.14	91.52	93.14
	2	63.74	69.64	71.75	71.76	71.75	69.39	71.74
3	1	91.47	92.11	93.64	93.64	92.80	91.58	92.80
	2	64.35	69.09	73.18	73.18	73.34	68.39	73.34
4	1	92.36	92.51	92.98	92.98	91.63	91.42	91.63
	2	63.41	68.54	73.74	73.74	73.63	67.85	73.63
5	1	93.44	93.03	90.42	90.42	88.52	91.49	88.52
	2	64.22	69.21	73.75	73.75	73.84	67.62	73.84
6	1	92.49	92.36	90.43	90.43	88.42	91.24	88.42
	2	64.07	68.50	73.81	73.81	73.72	67.20	73.71
7	1	91.92	91.84	89.52	89.52	87.22	91.04	87.22
	2	64.61	69.42	74.62	74.62	74.33	67.86	74.32
8	1	91.81	91.72	89.25	89.25	86.75	90.72	86.75
	2	64.39	68.95	74.67	74.67	74.23	67.74	74.26
9	1	91.16	91.22	89.10	89.10	86.33	90.12	86.33
	2	63.27	68.28	74.65	74.65	73.98	66.78	74.02
10	1	90.66	90.79	89.02	89.02	86.00	89.52	86.00
	2	61.74	67.56	74.55	74.55	73.63	65.53	73.79

Таблиця В.17

Результати комбінованих ансамблевих алгоритмів для RAUNet, DICE %

Кількість моделей у ансамблі	Набір даних	Комбінація						
		1	2	3	4	5	6	7
2	1	91.70	91.88	90.53	90.53	89.63	92.32	89.63
	2	67.82	71.88	75.12	75.27	75.19	71.50	75.04
3	1	91.53	91.59	90.02	90.02	88.79	91.96	88.79
	2	70.34	72.50	74.44	74.54	74.06	70.77	73.95
4	1	91.69	92.00	89.83	89.83	88.22	91.87	88.22
	2	68.19	71.04	74.44	74.55	74.24	69.60	74.20
5	1	92.28	92.46	89.60	89.60	87.66	91.63	87.66
	2	66.38	70.53	74.47	74.56	73.96	68.04	73.91
6	1	92.16	92.45	89.58	89.58	87.58	91.49	87.58
	2	65.83	68.91	74.57	74.60	73.99	67.58	74.02
7	1	92.16	92.36	89.08	89.08	86.93	91.57	86.93
	2	65.24	67.56	73.09	73.12	72.52	65.78	72.48
8	1	91.94	92.10	88.13	88.13	85.82	91.39	85.82
	2	65.70	67.69	73.35	73.40	72.98	66.76	72.89
9	1	91.61	91.75	87.99	87.99	85.56	91.01	85.56
	2	65.86	67.92	73.34	73.40	72.99	66.80	72.85
10	1	91.81	91.87	87.90	87.90	85.39	90.30	85.39
	2	64.68	67.22	73.39	73.44	72.92	66.12	72.84

Таблиця В.18

Результати комбінованих ансамблевих алгоритмів для Attention U-Net, DICE %

Кількість моделей у ансамблі	Набір даних	Комбінація						
		1	2	3	4	5	6	7
2	1	89.79	90.62	91.64	91.62	91.60	88.52	91.62
	2	56.05	64.34	66.99	67.02	67.28	62.98	67.24
3	1	91.54	91.46	87.94	87.94	86.33	89.62	86.33
	2	63.63	68.31	74.37	74.38	74.83	69.02	74.70
4	1	91.53	91.70	88.10	88.10	86.17	89.35	86.17
	2	63.77	69.34	74.92	74.93	75.45	69.52	75.32
5	1	91.00	91.70	87.35	87.35	84.50	89.47	84.50
	2	60.57	61.64	69.91	73.47	65.47	67.54	73.49

Продовження таблиці В.18

6	1	90.09	91.04	87.38	87.38	84.30	88.84	84.30
	2	56.47	59.57	69.87	73.43	65.94	65.21	73.34
7	1	88.96	90.46	87.39	87.39	83.93	87.99	83.93
	2	53.23	57.55	69.88	73.44	66.14	63.88	73.17
8	1	89.10	89.69	87.33	87.33	83.60	86.95	83.60
	2	50.14	55.99	69.95	73.51	66.34	62.98	72.75
9	1	88.99	89.56	86.72	86.72	83.00	87.02	83.08
	2	49.72	56.34	69.79	73.29	66.45	62.49	72.46
10	1	88.39	89.24	86.66	86.66	82.71	85.74	82.71
	2	48.05	55.48	69.77	73.27	66.60	61.49	72.43

Таблиця В.19

Результати комбінованих ансамблевих алгоритмів для Inception, DICE %

Кількість моделей у ансамблі	Набір даних	Комбінація						
		1	2	3	4	5	6	7
2	1	82.92	83.16	83.52	84.28	84.09	84.80	83.35
	2	59.08	59.97	61.75	61.05	62.89	57.52	63.41
3	1	82.99	83.76	83.74	84.27	83.98	84.51	83.46
	2	59.15	61.38	63.18	63.23	65.06	58.54	65.12
4	1	82.93	83.37	83.86	84.37	83.94	84.27	83.44
	2	57.23	59.09	63.24	63.49	65.50	56.19	65.38
5	1	82.82	83.63	83.94	84.39	84.06	84.06	83.62
	2	56.55	59.38	63.35	63.56	65.63	55.36	65.44
6	1	82.49	83.11	84.00	84.40	83.99	83.83	83.59
	2	55.12	57.23	63.29	63.58	65.45	54.51	65.33
7	1	82.14	83.06	83.90	84.34	83.86	83.32	83.44
	2	54.95	57.57	63.42	63.81	65.56	54.42	65.31
8	1	82.49	83.02	79.12	79.60	78.12	83.86	77.66
	2	56.83	58.79	65.79	66.59	66.49	57.22	66.06
9	1	82.22	82.95	79.12	79.59	77.94	83.30	77.50
	2	56.59	59.09	65.82	66.64	66.55	56.83	66.10
10	1	81.91	82.74	79.20	79.59	78.06	82.82	77.69
	2	55.83	57.90	65.91	66.65	66.51	56.51	66.13

Таблиця В.20

Результати комбінованих ансамблевих алгоритмів для Resnet101 U-Net, DICE %

Кількість моделей у ансамблі	Набір даних	Комбінація						
		1	2	3	4	5	6	7
2	1	91.83	90.86	90.05	90.05	89.24	89.57	89.24
	2	60.36	61.26	60.64	62.20	59.14	60.51	60.81
3	1	91.24	90.50	88.78	88.78	87.50	88.84	87.50
	2	60.49	60.85	61.74	65.35	59.90	62.83	64.18
4	1	89.80	89.81	88.74	88.74	87.09	88.34	87.09
	2	56.81	58.72	61.61	65.43	59.42	62.19	64.10
5	1	88.32	89.05	88.75	88.75	86.95	88.07	86.95
	2	55.12	57.93	61.68	65.47	59.55	62.02	64.15
6	1	89.38	89.32	87.68	87.68	85.65	87.81	85.65
	2	55.44	58.58	62.94	65.80	60.51	62.12	64.50
7	1	90.12	89.67	87.28	87.28	85.37	87.56	85.37
	2	52.81	56.71	62.89	65.74	60.34	61.67	64.27
8	1	89.41	89.38	87.22	87.22	85.40	86.91	85.40
	2	51.58	57.07	63.66	65.94	61.13	61.54	64.44
9	1	89.84	89.75	85.45	85.45	83.19	86.12	83.19
	2	51.87	57.36	65.75	68.24	62.94	61.91	66.92
10	1	89.07	89.17	85.45	85.45	83.00	85.57	83.10
	2	50.01	56.85	65.75	68.23	63.01	62.04	66.98

Таблиця В.21

Результати комбінованих ансамблевих алгоритмів для Resnet152v2 U-Net, DICE %

Кількість моделей у ансамблі	Набір даних	Комбінація						
		1	2	3	4	5	6	7
2	1	95.26	95.24	94.59	94.67	94.07	94.92	93.99
	2	66.19	76.44	78.31	78.25	77.19	73.78	77.25
3	1	95.44	95.22	94.39	94.39	93.65	94.77	93.65
	2	63.32	73.88	78.01	78.01	76.88	72.06	76.87
4	1	95.27	95.14	94.33	94.33	93.34	94.50	93.34
	2	62.01	72.40	77.63	77.63	76.47	70.93	76.46
5	1	95.07	95.03	94.36	94.36	93.27	93.43	93.27
	2	60.35	72.01	77.35	77.35	76.42	70.55	76.46

Продовження таблиці В.21

6	1	95.20	94.37	82.78	82.78	79.05	92.56	79.05
	2	61.50	72.67	70.83	70.83	68.74	71.37	68.72
7	1	95.37	94.23	73.97	73.97	69.87	92.54	69.87
	2	63.34	67.66	60.22	62.70	57.36	68.53	60.28
8	1	95.16	94.24	73.81	73.81	69.62	92.18	69.62
	2	61.16	67.05	60.20	62.68	57.30	68.40	60.13
9	1	94.88	93.48	69.28	69.28	65.10	90.93	65.10
	2	58.53	65.64	58.59	62.41	55.39	69.14	59.87
10	1	94.57	93.40	69.12	69.14	64.59	89.86	64.77
	2	57.23	65.43	59.12	63.23	55.83	69.86	60.87

Таблиця В.22

Результати комбінованих ансамблевих алгоритмів для Fcn8 MobileNet, DICE %

Кількість моделей у ансамблі	Набір даних	Комбінація						
		1	2	3	4	5	6	7
2	1	92.92	92.73	91.02	91.21	90.49	93.38	90.31
	2	77.21	77.14	76.23	76.52	76.18	76.93	75.90
3	1	93.19	92.16	88.08	88.54	86.53	93.11	86.10
	2	76.29	76.18	74.67	74.79	74.28	76.75	74.17
4	1	93.38	92.71	87.57	87.59	85.64	92.71	85.63
	2	75.66	76.96	74.04	74.11	73.42	76.29	73.43
5	1	93.23	92.15	87.44	87.45	85.59	92.44	85.58
	2	74.58	76.40	73.56	73.73	72.84	75.61	73.21
6	1	92.87	92.25	86.32	86.32	84.09	92.35	84.09
	2	74.60	75.83	73.16	73.32	72.31	75.50	72.63
7	1	92.79	91.92	85.29	85.29	82.96	92.20	82.96
	2	73.63	75.34	72.84	72.99	71.86	75.44	72.22
8	1	92.63	91.98	84.37	84.37	81.87	92.04	81.87
	2	74.05	75.27	71.73	72.62	70.50	75.69	71.74
9	1	92.57	91.39	83.57	83.57	80.95	91.93	80.95
	2	73.49	74.39	71.16	72.10	69.63	75.86	71.21
10	1	92.29	91.56	80.97	80.97	77.90	91.72	77.90
	2	73.59	73.87	66.63	69.34	64.24	75.95	68.01

Таблиця В.23

Порівняння середніх значень точності комбінацій, DICE %

Кількість моделей у ансамблі	Набір даних	Конкурентний ансамблевий алгоритм	Комбінація						
			1	2	3	4	5	6	7
2	1	95.96	95.96	95.94	95.50	95.53	95.08	95.88	95.05
	2	81.25	81.25	81.66	80.53	80.53	79.88	80.96	79.88
3	1	90.79	90.64	90.39	88.58	88.58	86.89	90.86	86.89
	2	74.50	70.24	72.95	74.46	74.50	73.86	71.51	73.93
4	1	85.05	85.05	84.96	81.86	81.84	81.07	85.20	81.09
	2	69.60	67.07	68.15	69.51	70.80	68.73	68.74	70.15
5	1	91.82	91.82	91.95	90.89	90.89	88.98	90.96	88.98
	2	73.86	63.76	68.80	73.86	73.86	73.60	67.60	73.63
6	1	91.87	91.87	92.05	89.18	89.19	87.29	91.50	87.29
	2	74.12	66.67	69.47	74.02	74.10	73.65	68.11	73.57
7	1	89.93	89.93	90.61	87.83	87.83	85.13	88.17	85.14
	2	70.61	55.74	60.95	70.61	72.97	68.28	65.01	72.77
8	1	82.76	82.54	83.20	82.27	82.76	82.00	83.86	81.53
	2	64.16	56.81	58.93	63.97	64.29	65.52	56.34	65.36
9	1	89.89	89.89	89.72	87.71	87.71	85.93	87.64	85.94
	2	62.99	54.94	58.37	62.96	65.82	60.66	61.87	64.49
10	1	95.14	95.14	94.49	82.96	82.97	80.28	92.86	80.29
	2	68.91	61.51	70.36	68.92	70.34	66.84	70.51	68.55