

Міністерство освіти і науки України  
Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук  
(повна назва)

Кафедра Штучного інтелекту  
(повна назва)

## КВАЛІФІКАЦІЙНА РОБОТА Пояснювальна записка

рівень вищої освіти другий (магістерський)

Розробка алгоритмів сегментації клітин на мікропрепаратах  
(тема)

Виконав:  
студент 2 курсу, групи СШМ-22-3  
Яковенко А. П.  
(прізвище, ініціали)

Спеціальність 122 Комп'ютерні науки  
(код і повна назва спеціальності)

Тип програми освітньо-наукова  
(освітньо-професійна або освітньо-наукова)

Освітня програма Системи штучного інтелекту  
(повна назва спеціалізації)

Керівник проф. Аврунін О. Г.  
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри \_\_\_\_\_  
(підпис)

В.О. Філатов  
(прізвище, ініціали)

2024 р.

Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук  
(повна назва)  
Кафедра Штучного інтелекту  
(повна назва)  
Рівень вищої освіти другий (магістерський)  
Спеціальність 122 Комп'ютерні науки  
(код і повна назва)  
Тип програми освітньо-наукова  
(освітньо-професійна або освітньо-наукова)  
Освітня програма Системи штучного інтелекту  
(повна назва)

ЗАТВЕРДЖУЮ:  
Зав. кафедри \_\_\_\_\_  
(підпис)  
« \_\_\_\_\_ » \_\_\_\_\_ 20 \_\_\_\_ р.

**ЗАВДАННЯ**  
НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові Яковенку Антону Павловичу  
(прізвище, ім'я, по батькові)

1. Тема роботи Розробка алгоритмів сегментації клітин на мікропрепаратах

затверджена наказом університету від 1 квітня 20 24 р. № 260Ст

2. Термін подання студентом роботи до екзаменаційної комісії 11 червня 20 24 р.

3. Вихідні дані до роботи Теоретичні матеріали про алгоритми сегментції зображень, результати досліджень ефективності конкретних методів сегментації

4. Перелік питань, що потрібно опрацювати в роботі \_\_\_\_\_

1) Огляд проблем сегментації зображень

2) Аналіз нейронних мереж та їх архітектури у контексті задач сегментації

3) Дослідження алгоритмів для виокремлення контурів об'єктів

4) Розробка алгоритму для поєднання нейромережових можливостей та методів додаткової обробки для якісної сегментації клітин на мікропрепаратах



## РЕФЕРАТ

Пояснювальна записка: 68 с., 28 рис., 1 дод., 30 джерел.

АЛГОРИТМИ СЕГМЕНТАЦІЇ, АРХІТЕКТУРА U-NET,  
БІОМЕДИЦИНА, КОМП'ЮТЕРНИЙ ЗІР, СЕГМЕНТАЦІЯ ЗОБРАЖЕНЬ.

Об'єкт дослідження – зображення клітин на мікропрепаратах.

Предмет дослідження – алгоритми сегментації біомедичних зображень.

Мета роботи – пошук оптимальних алгоритмів для сегментації зображень, та можливості оптимального їх поєднання з нейронною мережею.

Методи дослідження – аналіз наукових джерел, експериментальний підбір параметрів.

## **ABSTRACT**

Master's thesis contains: 68 pp., 28 fig., 1 ann., 30 references.

BIOMEDICINE, COMPUTER VISION, IMAGE SEGMENTATION,  
SEGMENTATION ALGORITHMS, U-NET ARCHITECTURE.

The object of the study is the image of cells on micro-preparations.

The subject of the research is algorithms for the segmentation of biomedical images.

The purpose of the work is to find optimal algorithms for image segmentation and the possibility of optimally connecting them with a neural network.

Research methods – analysis of scientific sources, experimental selection of parameters.

## ЗМІСТ

Вступ.....	8
1 Аналіз предметної галузі.....	10
1.1 Опис предметної галузі.....	10
1.2 Огляд сфери застосування алгоритмів сегментації біомедичних зображень.....	11
2 Постановка задачі.....	13
2.1 Класифікація обраної задачі.....	13
2.2 Визначення вимог до функціоналу.....	13
2.3 Вибір архітектури.....	14
3 Реалізація моделі.....	18
3.1 Види задач розпізнавання об'єктів.....	18
3.2 Сегментація.....	19
3.3 Архітектура U-Net.....	20
3.4 Підготовка даних.....	23
3.5 Розмітка об'єктів.....	27
3.6 Експорт зображень.....	32
3.7 Конвеєр даних.....	34
3.8 Оптимізація продуктивності конвеєра даних.....	36
3.9 Аугментація зображень.....	40
3.10 Завантаження та попередня обробка даних.....	41
3.11 Спосіб обробки зображень.....	44
3.12 Згортка.....	46
3.13 Реалізація моделі.....	49
3.14 Навчання нейронної мережі.....	50
3.15 Метрики та функції втрат.....	51
3.16 Проблеми детекції.....	54
3.17 Алгоритми обробки зображень.....	57
3.17.1 Морфологічна обробка.....	58

3.17.2 Алгоритм вододілу.....	59
3.17.3 Кластеризація методом k-середніх.....	60
3.17.4 Зменшення кількості відтінків.....	61
Висновки .....	63
Перелік джерел посилання .....	64
Додаток А Відомість кваліфікаційної роботи .....	68

## ВСТУП

У сучасному науковому світі розвиток методів комп'ютерного зору виявляється ключовим напрямком для удосконалення біологічних досліджень [8]. Поєднання високоточних зображень та алгоритмів аналізу відкриває нові можливості у розумінні клітинних процесів [12].

Сегментація клітин в мікропрепаратах грає важливу роль у визначенні структури та стану тканин, що є критичним для точної діагностики захворювань та вивчення біологічних механізмів. Мікропрепарати характеризуються великою різноманітністю клітинних структур та високим ступенем деталізації, що створює виклики для точної сегментації через ймовірну наявність шуму, нерівномірної контрастності зображення, обмеженої роздільної здатності фрагмента зображення з об'єктом, оклюзій, зміни положень об'єктів у хронологічному ряді зображень, тощо [20]. Висока варіабельність клітинних структур та можливість появи артефактів на зображеннях створюють проблеми для існуючих алгоритмів сегментації [22].

Використання глибокого навчання виявляється перспективним підходом для покращення точності сегментації та адаптації до різноманітних структур клітин на мікропрепаратах. Розроблені алгоритми повинні бути спеціально адаптовані до особливостей мікропрепаратів, враховуючи їхню велику варіабельність. Введення автоматичних механізмів виявлення та корекції помилок допомагає покращити надійність алгоритмів сегментації в умовах високої варіабельності [1].

Актуальність дослідження полягає у аналізі наявних методів сегментації зображень з визначенням їхніх переваг та недоліків, а також основних напрямків для подальшого вдосконалення для ефективного вирішення задачі сегментації клітин задля можливості використання цих алгоритмів, їх комбінацій або модифікацій в медичних дослідженнях.

В даній роботі розглядаються існуючі алгоритми сегментації зображень, оцінюється їх доцільність і ефективність використання у заданій предметній області. А також ймовірні шляхи покращення для отримання більш чітких результатів з оглядом на перспективу створення похідних алгоритмів, які гіпотетично даватимуть якісніші результати у межах задач сегментації клітин на біомедичних зображеннях.

## 1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ

### 1.1 Опис предметної галузі

Сегментація об'єктів на зображеннях – це процес автоматичного виділення та класифікації окремих об'єктів або регіонів зображення [10]. Вона відіграє важливу роль у комп'ютерному зорі, медичному зображенні, автоматизації виробництва, візуальному відслідковуванні об'єктів та в багатьох інших сферах. Сегментація зображення використовується для різних завдань, включаючи розпізнавання об'єктів, визначення їх меж, аналіз текстури та забезпечення кращого розуміння контексту зображення.

Існують різні методи сегментації зображень, які можна класифікувати за декількома критеріями, такими як тип даних (наприклад, зображення відео, медичні зображення, зображення з низьким чи високим роздільним здатністю), характеристики об'єктів, які потрібно сегментувати (форма, розмір, колір тощо), та призначення (віддалене спостереження, медична діагностика, рекреаційні цілі тощо) [10].

Одним з найпоширеніших методів сегментації є сегментація за допомогою порогового значення. Цей метод полягає в тому, що певне порогове значення встановлюється для пікселів зображення, і пікселі, які мають значення вище або нижче цього порогу, відносяться до окремих класів. Цей метод простий у реалізації, але може бути неефективним у випадках, коли зображення має складну текстуру або коли об'єкти мають різні рівні яскравості [2].

Інші методи сегментації використовуються для вирішення цих проблем, наприклад:

– сегментація за допомогою кластеризації – цей метод використовує алгоритми кластеризації, такі як k-середні, для групування пікселів на основі їхніх характеристик, таких як колір, яскравість або текстура [14];

– сегментація за допомогою різниці пікселів – цей метод використовує різницю у значеннях пікселів для визначення меж об'єктів [24]. Наприклад, алгоритми різниці пікселів можуть виділяти контур об'єктів за допомогою зміни яскравості або колірних відмінностей;

– сегментація за допомогою нейронних мереж – цей підхід використовує глибокі нейронні мережі для автоматичного вивчення ознак та класифікації областей зображення. Моделі, такі як конволюційні нейронні мережі (CNN), можуть бути навчені розпізнавати об'єкти на зображеннях та виконувати їх сегментацію з високою точністю [20].

Незалежно від методу, сегментація об'єктів на зображеннях є складним завданням, яке вимагає комбінації математичних методів, алгоритмів комп'ютерного зору та штучного інтелекту для досягнення точних результатів. Дослідження в цій області постійно розвивається, що призводить до вдосконалення та появи нових методів сегментації зображень з різними застосуваннями у наукових дослідженнях, медицині, промисловості та інших галузях.

## 1.2 Огляд сфери застосування алгоритмів сегментації біомедичних зображень

Алгоритми сегментації біомедичних зображень знаходять широке застосування у медичній діагностиці, дослідженні та лікуванні. Одним з основних напрямків їх використання є медичне зображення мозку. Сегментація мозку дозволяє виділити різні структури, такі як кора, біла речовина, жовч, та інші, що допомагає в аналізі морфологічних змін та відхилень. Крім того, алгоритми сегментації використовуються для автоматизованої аналізу медичних зображень з метою виявлення патологій, таких як ракові ураження, кістки, аномалії внутрішніх органів тощо [20].

Ще однією важливою областю застосування є сегментація зображень медичних знімків у рентгенівській томографії (СТ) та магнітно-резонансній

томографії (MRI). Це може включати сегментацію органів, судин, пухлин або уражень, що допомагає лікарям у точній діагностиці та плануванні лікування.

Застосування алгоритмів сегментації також відчутне в медичній інтервенції, наприклад, у хірургічному плануванні. Вони можуть допомагати визначити точне розташування пухлини або іншої патології перед операцією, що робить втручання більш точним і менш інвазивним.

Крім цього, алгоритми сегментації використовуються у дослідженні мозкової активності на базі функціональних магнітно-резонансних зображень (fMRI). Вони дозволяють ідентифікувати активні області мозку під час виконання певних завдань або стимулювання, що є ключовим для розуміння функцій мозку та розвитку нових методів лікування психічних та неврологічних розладів.

Нарешті, алгоритми сегментації використовуються у дослідженні медичних зображень в різних масштабах, включаючи мікроскопічні зображення для аналізу тканин та клітин. Вони можуть бути використані для автоматизованого виявлення або класифікації патологічних структур на клітинному рівні, що допомагає в дослідженні хвороб та розробці нових методів лікування.

Загалом, алгоритми сегментації біомедичних зображень є важливим інструментом у медичному дослідженні та практиці, допомагаючи в точній діагностиці, лікуванні та розумінні біологічних процесів.

## 2 ПОСТАНОВКА ЗАДАЧІ

### 2.1 Класифікація обраної задачі

Поставлена задача являє собою проведення дослідження щодо наявних алгоритмів поконтурної сегментації об'єктів, пошук найбільш оптимальних рішень їх модифікацій та комбінацій для роботи зі знімками клітин, зробленими через мікроскоп. Для подальшої обробки зображень методами нейронних мереж з метою розпізнавання конкретних екземплярів та відслідковування їх траєкторії.

Тобто це дослідницька підзадача, яку можна розглядати як фрагмент комплексної задачі для точного відслідковування переміщення клітин на мікропрепаратах.

### 2.2 Визначення вимог до функціоналу

Дана робота ставить на меті розробку системи для сегментації клітин на зображеннях, отриманих з мікроскопа. Сегментація клітин є важливим етапом у біомедичних дослідженнях, оскільки вона дозволяє автоматизувати процес аналізу та виявлення патологічних аномалій у клітинах.

Для досягнення поставленої мети необхідно розв'язати низку завдань:

- визначення методів сегментації, які дозволять ефективно виділяти клітини на зображеннях;
- відбір відповідної навчальної вибірки, яка містить розмічені зображення клітин для тренування системи;
- вибір мови програмування, середовища розробки та програмних інструментів, що найбільш відповідають поставленій задачі;
- аналіз і вибір архітектур нейронних мереж, які підходять для сегментації клітин на мікроскопічних зображеннях;

- визначення оптимального порядку та методів передобробки вхідних даних для підготовки їх до навчання та тестування системи;

- впровадження методів нормалізації форм клітин та інші дрібні параметри, які підвищують ефективність роботи системи.

Отримана система має наступні характеристики:

- здатність ефективно визначати межі клітин на зображеннях з мікроскопа;

- ефективне використання апаратних ресурсів під час тренування та видачі рішення;

- використання штучних нейронних мереж для покращення точності сегментації клітин.

Така система дозволить автоматизувати процес аналізу клітинних зображень, що має велике значення у біомедичних дослідженнях та медичній діагностиці.

### 2.3 Вибір архітектури

Задача, що розглядається є важливою віхою в області медичного діагностування та дослідження. Вона передбачає розпізнавання та виділення окремих клітин на зображенні з мікроскопа з метою аналізу їх структури та властивостей. Використання нейромереж для цієї задачі дозволяє автоматизувати процес та забезпечує точність, недосяжну для традиційних методів.

Однак вибір архітектури нейромережі для попиксельної сегментації клітин є складним завданням, яке вимагає уважного аналізу різних факторів. Зважаючи на специфіку завдання, було обрано ряд поширених архітектур та оцінена їхня відповідність для даної задачі.

Перш ніж вибрати конкретну архітектуру, необхідно чітко сформулювати завдання та вимоги до системи. Деякі ключові питання для

розуміння наскільки та чи інша архітектура доцільна для використання наведені нижче.

**Точність:** яка точність сегментації необхідна для конкретного застосування?

Кінцевий розмір зображення навчально-тренувального датасету складає 640\*512 пікселів. Середній розмір об'єкта складає 45 пікселів у висоту і ширину. Клітини мають округлу форму. Тобто приблизна площа об'єкта – 1560 пікселів. Важливо передбачити можливість зближення кількох об'єктів та їх накладання одне на одного, що викликатиме додаткові складнощі для виокремлення екземпляра. Отже, при роботі з таким набором даних необхідна досить висока точність детекції.

**Швидкодія:** чи потрібно відповідати в реальному часі, чи допускається обробка з затримкою?

Швидкість обробки зображення бажано мати високу. Але точність перевищує швидкість за пріоритетом. Швидкість може бути досягнута через зменшення кількості колірних каналів (що дозволяє зробити цільовий датасети без суттєвих втрат у інформативності). А також за допомогою алгоритмів обробки зображень, які будуть розглянуті в ході роботи.

**Ресурси:** які обчислювальні ресурси доступні для реалізації?

Кінцева система має бути ощадною до ресурсів, щоб забезпечити можливість її використання при невеликому запасі апаратних ресурсів.

**Розмір датасету:** який обсяг даних доступний для навчання моделі?

У поставленій задачі важливо мати змогу навчити модель на невеликому корпусі даних, щоб мати свободу застосовувати її нішево, під різні датасети без зайвих витрат часу, навчивши її на мінімально необхідній вибірці даних.

Відповівши на ці питання можна обрати найбільш підходящі кандидатури архітектури для даної задачі.

Опираючись на пошук і аналіз існуючих архітектурних рішень, які наразі мають широке застосування у подібних сферах діяльності, було

знайдено три вектори найбільш підходящих архітектурних моделей нейромереж.

Нижче наводиться короткий огляд кожної із них з описом основних властивостей:

– U-Net – це одна з найпоширеніших архітектур для задач сегментації зображень [22]. Вона має зворотний підключений шлях, що допомагає у відновленні точних контурів об'єктів на зображенні. U-Net добре показує себе при роботі з малими наборами даних та забезпечує високу точність, але може бути вимогливою до обчислювальних ресурсів [22];

– DeepLab – це інший популярний вибір для сегментації зображень. Він використовує конволюційні мережі глибокого навчання з різними варіантами моделі, такими як DeepLabv3 та DeepLabv3+ [23]. DeepLab часто відомий своєю здатністю ефективно працювати з великими зображеннями та забезпечувати високу швидкодію;

– Mask R-CNN – це ще одна потужна архітектура, яка поєднує в собі виявлення об'єктів та сегментацію. Вона використовує регіональні пропозиції (region proposals), що дозволяє виділяти та сегментувати об'єкти на зображенні з високою точністю [13].

Розглянувши вимоги до архітектури та гіпотетичні архітектурні моделі, можна зробити наступні висновки.

Розмір датасету грає визначну роль. Більшість архітектур, включаючи розглянуті, потребують великих датасетів для якісної роботи. Найменш вибагливою в розмірах датасету є U-Net. Вона може бути кращим вибором, оскільки вона має досить невелику кількість параметрів та добре показує себе при навчанні на невеликих об'ємах даних.

Тож, U-Net це – найбільш релевантна архітектура для нинішньої задачі, яка буде взята для виконання роботи. Однак інші дві наведені архітектури (DeepLab та Mask R-CNN) є альтернативою U-Net при інакших характеристиках вхідного датасету.

Швидкодія: DeepLab має високу швидкодію без втрати точності, DeepLab може бути кращим варіантом, якщо в подібній задачі будуть наявні дані з великою кількістю різноманітних класів клітин на досить великих зображеннях. В такому випадку DeepLab було б використовувати доцільніше.

Складність сегментації: У нинішньому датасеті форма клітин доволі проста. Але якщо клітини на зображенні мають складні форми або структури, Mask R-CNN може забезпечити більш точну сегментацію, враховуючи контекст об'єкта.

В межах даного дослідження використовується лише найбільш релевантна архітектура. Інші рішення представлені теоретично, для порівняння можливостей.

## 3 РЕАЛІЗАЦІЯ МОДЕЛІ

### 3.1 Види задач розпізнавання об'єктів

Розпізнавання об'єктів комп'ютерним зором – це процес автоматичного визначення та класифікації об'єктів на зображеннях або відео за допомогою алгоритмів та моделей машинного навчання. Цей процес зазвичай включає в себе три основні етапи: класифікацію, локалізацію та сегментацію. Кожен із цих етапів може бути кінцевим результатом. Однак у даній роботі основний акцент робиться саме на сегментації.

Задача класифікації є першою і найгрубішою в контексті точності результатів. Вона полягає в тому, щоб визначити, який об'єкт або об'єкти присутні на зображенні або відео. Модель аналізує вхідне зображення і призначає йому один або кілька класів, що відповідають об'єктам, які вона розпізнала на зображенні. Наприклад, у випадку розпізнавання обличч, модель може визначити, що на зображенні зображено людину [7].

Більш інформативна задача – локалізація. Вона визначає місцезнаходження об'єктів на зображенні або відео. В результаті локалізації модель вказує на місце розташування об'єкта або об'єктів, а також надає рамки (або прямокутники) для виділення цих об'єктів на вхідному зображенні [6]. Наприклад, модель може вказати на те, де саме на зображенні знаходиться обличчя людини та обвести його прямокутником.

І третій етап або ж тип розпізнавання об'єктів – це сегментація. Цей етап визначає пікселі, які відповідають кожному об'єкту на зображенні, розділяючи зображення на різні сегменти або області [6]. Це набагато більш точний підхід до визначення меж об'єктів, оскільки він дозволяє розділити зображення на окремі частини, які відповідають кожному об'єкту. Наприклад, у випадку сегментації людського обличчя, модель виділить кожне обличчя на зображенні шляхом призначення кожному пікселю значення, яке вказує, чи належить він до обличчя або ні.

### 3.2 Сегментація

Сегментація в медичній сфері, зокрема на зображеннях клітин, є дуже важливим завданням, яке допомагає лікарям аналізувати та діагностувати різні захворювання. Зображення клітин часто отримуються за допомогою мікроскопів, і точна сегментація допомагає ідентифікувати та аналізувати різні структури клітин та їх органели.

Основні виклики в цій області полягають у наступному:

- різні типи клітин і структур. Зображення клітин може містити різні типи клітин і структур, такі як ядра, клітинні мембрани, цитоплазму тощо. Точна сегментація дозволяє ідентифікувати та розділити ці структури для подальшого аналізу;

- різна густина і контрастність. Зображення клітин може мати різну густоту і контрастність, що може ускладнювати сегментацію, особливо при наявності шуму або артефактів на зображенні;

- великий обсяг даних. Медичні зображення можуть мати великий обсяг даних, що ускладнює ручну сегментацію. Тому автоматизовані методи сегментації часто використовуються для обробки великого обсягу зображень.

У медичній сфері використовуються різні методи сегментації клітинних зображень, включаючи класичні методи та сучасні техніки машинного навчання. Ось кілька прикладів методів сегментації, які часто використовуються:

- методи порогової сегментації. Ці методи використовуються для виділення областей клітин на зображенні, встановлюючи певні порогові значення для яскравості пікселів;

- методи активних контурів (згорткових нейронних мереж). Ці методи використовують активні контури або сегментують клітини на основі нейронних мереж, що дає можливість більш точної сегментації, особливо в умовах великої варіабельності;

– семантичні сегментуючі нейронні мережі (U-Net, SegNet). Ці сучасні архітектури нейронних мереж спеціально призначені для сегментації зображень. Вони використовуються в медичній сфері для точної сегментації клітин та їх структур.

### 3.3 Архітектура U-Net

U-Net – це нейромережева архітектура, яка була спеціально розроблена для семантичної сегментації зображень. Її назва виникла від її схожості з літерою «U» через її форму. Розроблена в 2015 році Ольфом Ронненбергом (Olaf Ronneberger), Філіппом Фішером (Philipp Fischer) та Томасом Брохаусом (Thomas Brox) з Університету Фрайбурга в Німеччині, вона швидко стала дуже популярною в області медичного зображення, особливо в аналізі зображень медичного обладнання [17].

Основна ідея за U-Net полягає у використанні архітектури енкодер-декодер, де енкодер відповідає за поступове зменшення роздільної здатності (зображення зменшується у розмірах), а декодер – за поступове збільшення зображення до оригінального розміру. Це дозволяє моделі отримати широке поле зору та відновити точні деталі.

Архітектура є згортковою [17]. Згорткові нейронні мережі зазвичай містять кілька шарів згортки, а нейрони в різних шарах відповідають областям зображення різних масштабів [16]. Таким чином, для багат шарової згорткової нейронної мережі зручно отримувати характеристики зображення різних масштабів. Ця здатність згорткових нейронних мереж може бути посилена за допомогою розширення структури мережі. Завдяки підтримці достатньої кількості вибірових даних більш глибокі згорткові нейронні мережі здатні вирішувати багато дуже складних проблем, пов'язаних із візуальними функціями, і іноді ці мережі можуть демонструвати здатність виходити за межі людського зору [4].

Архітектурна модель має структуру, що умовно поділяється на 5 компонентів: енкодер, з'єднання, декодер, канали зв'язку між енкодером та декодером, та завершальний шар активації [22].

Втім, саме по собі архітектурне рішення є умовним каркасом для нейромережі. Вона задає ефективну базу, яка після додаткового опрацювання може виконувати поставлені задачі. Перед цим важливо зрозуміти можливі ризики, спричинені архітектурними недоліками та оцінити можливість їх перекриття алгоритмами чи запозиченням елементів інших архітектур для створення гібридної моделі. Тож в наступних розділах даної роботи будуть розглянуті пропоновані методи попередньої, проміжної та пост-обробки даних для досягнення оптимальних результатів у контексті задачі.

Архітектура U-net (приклад для 32x32 пікселів у найнижчій роздільній здатності). Кожне синє поле відповідає карті багатоканальних функцій. Кількість каналів вказана у верхній частині вікна. Розмір x-у надається в нижньому лівому краї поля. Білі поля представляють скопійовані карти об'єктів. Стрілки позначають різні операції.

Архітектура мережі проілюстрована на рисунку 3.1. Вона складається з енкодера (ліворуч) і декодера (права сторона). Згортковий шар відповідає типовій архітектурі згорткової мережі. Він складається з повторюваного застосування двох згорток 3x3 (згортка без доповнення), за кожною з яких слідує випрямлена лінійна одиниця (ReLU) і операція об'єднання 2x2 max з кроком 2 для зменшення дискретизації. На кожному кроці зменшення дискретизації подвоюється кількість функційних каналів. Кожен крок у шляху декодування складається з підвищення дискретизації карти функцій, за якою слідує згортка 2x2 («згортка вгору»), яка вдвічі зменшує кількість каналів функцій, об'єднання з відповідною обрізаною картою функцій зі звужувального шляху та двох згорток 3x3, кожна з яких супроводжується ReLU. Кадрування необхідне через втрату пікселів межі під час кожної згортки. На останньому шарі згортка 1x1 використовується для

відображення кожного 64-компонентного вектора ознак у потрібну кількість класів. Всього мережа має 23 згорткових шарів. Щоб дозволити безперебійне розміщення вихідної карти сегментації, важливо вибрати розмір вхідної плитки таким чином, щоб усі операції максимального об'єднання  $2 \times 2$  застосовувалися до шару з рівними розмірами  $x$  та  $y$  [13].

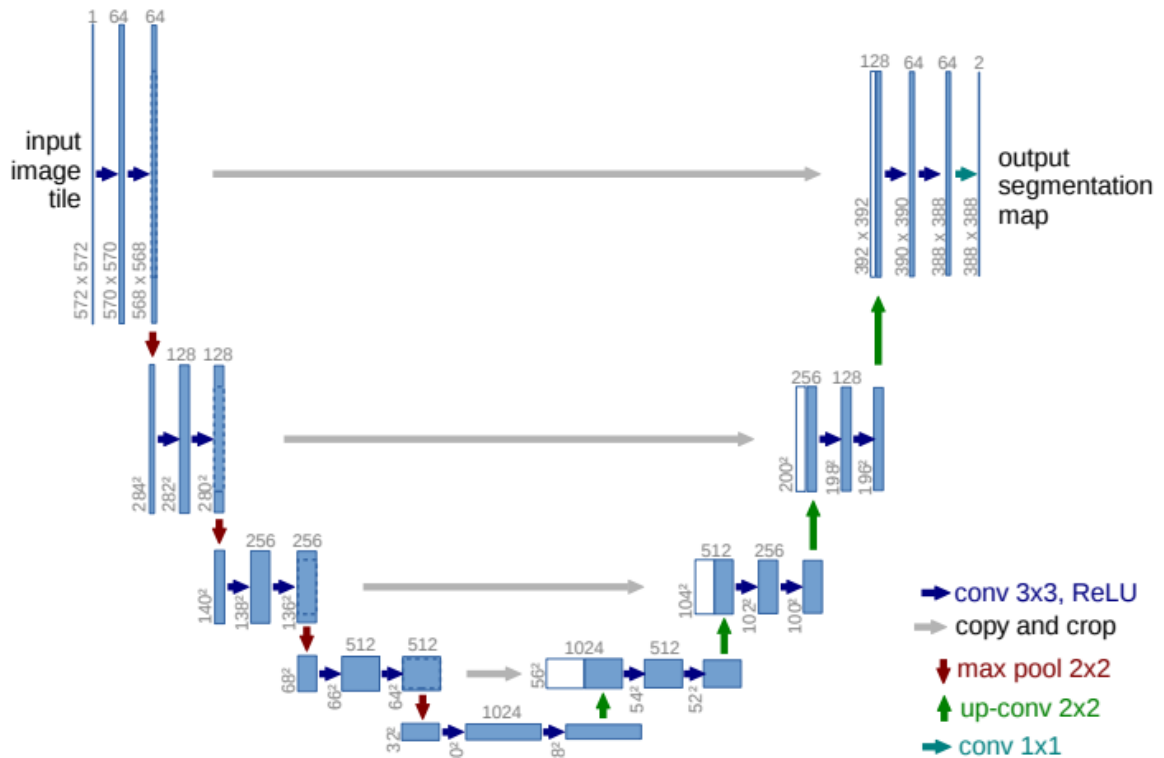


Рисунок 3.1 – Архітектурна модель U-net

Однією з основних проблем U-Net є обмежена обробка контексту. Хоча архітектура забезпечує механізми збереження контексту в енкодері через з'єднання та зворотні зв'язки, вона все ж зазвичай працює на основі локальних змін. Це може призводити до пропуску важливих глобальних ознак та контекстуальної інформації, що часто важливо для точної сегментації.

Ще однією проблемою U-Net є її схильність до навчання та загальної стабільності. Зокрема, в медичних дослідженнях, де набір даних може бути

обмеженим, і класи незбалансовані, модель може стикатися з проблемами перенавчання або недонавчання, що призводить до низької загальної продуктивності.

Ще проблемою U-Net є її чутливість до розміру вхідних зображень та наявності артефактів. Зображення, які відрізняються від розмірів, на яких модель навчалася, можуть призводити до втрати деталей або погіршення якості сегментації. Крім того, артефакти або шум на вхідних зображеннях можуть призводити до неправильних передбачень та погіршення результатів.

Хоча U-Net відмінно справляється з сегментацією зображень, вона може бути обмеженою у використанні для інших завдань машинного навчання, таких як виявлення об'єктів або відновлення зображень. Це обмеження полягає в її архітектурному дизайні, який спеціалізується на сегментації та не завжди підходить для інших завдань.

### 3.4 Підготовка даних

Задача полягає у створенні системи, яка б добре працювала в умовах обмеженості інформації. В тому числі з даними від одного-двох джерел і можливістю її швидко натренувати «з нуля». Задля цього буде використовуватись відносно невеликий фрагмент відео, нарізаний на кадри, які використовуватимуться як набори вхідних та навчальних даних.

Наступним етапом, після нарізки, стає розподіл датасету на навчальну та тренувальну частини. Найпростіший спосіб це зробити – лінійно розділити датасет навпіл. Такий спосіб матиме ефективність на відносно великих пропорціях. Наприклад, хронологічно перші 80% датасету обрати у якості навчальних даних. На інших 20%, відповідно, перевірити ефективність результатів роботи системи. Даний підхід має перевагу в можливості відслідковувати результат на динамічному фрагменті інформації (цілісній частині відео). Недолік системи – часткове і

нерівномірне покриття даних розміткою. Інакше кажучи, певний набір клітин чи навіть їх типів, які присутні на перших кадрах, максимально будуть проаналізовані нейромережею. А кадри для тренування можуть містити інші типи, форми об'єктів чи сторонній шум, до якого система не буде адаптована, що призведе до гірших результатів. Тож наведений метод доцільно використовувати для імітації динамічності середі, реакції системи на нову інформацію. Його раціонально застосовувати для «стрес тесту» алгоритмів та системи в цілому. Однак для навчання її для практичного використання таким методом недоцільно через відносно високу кількість навчальних даних та відносно низьку результативність.

Наведений рисунок 3.2 демонструє даний тип вибірки. На даному умовному прикладі фрейми 0-6 обрані для навчання, а 7-9 – для тренування. Система отримає максимум інформації про клітину, позначену синім і лише близько половини інформації про жовту та зелену клітини. При тому синьої клітини в тренувальній вибірці немає. Тобто деякі типи об'єктів зафіксовані максимально, а жовта і зелена – не дуже. При тому система може зафіксувати певну закономірність що дані два класи можуть знаходитись лише у нижній половині кадру. На вибірці для оцінки навчання вони ж знаходяться у верхній частині. Отже їх детекція і якість сегментації буде гіршою, особливо у випадку надкладання кількох об'єктів між собою. З цього впливає більш низька оцінка за істину в контексті конкретного датасету, однак більш наближена до реальної після роботи з іншими датасетами.

Більш реперезентативним варіантом вибірки даних стане методика N-ного кадру (рисунок 3.3). Суттю методики є вибір пропорції розділення сукупності кадрів на два кластери – навчальний набір та тренувальний. І сортування усіх кадрів до цих виборок відповідно пропорції. Для прикладу,  $\frac{1}{2}$ . Таким чином кожен другий кадр буде потрапляти до навчальної вибірки протягом всього відео.

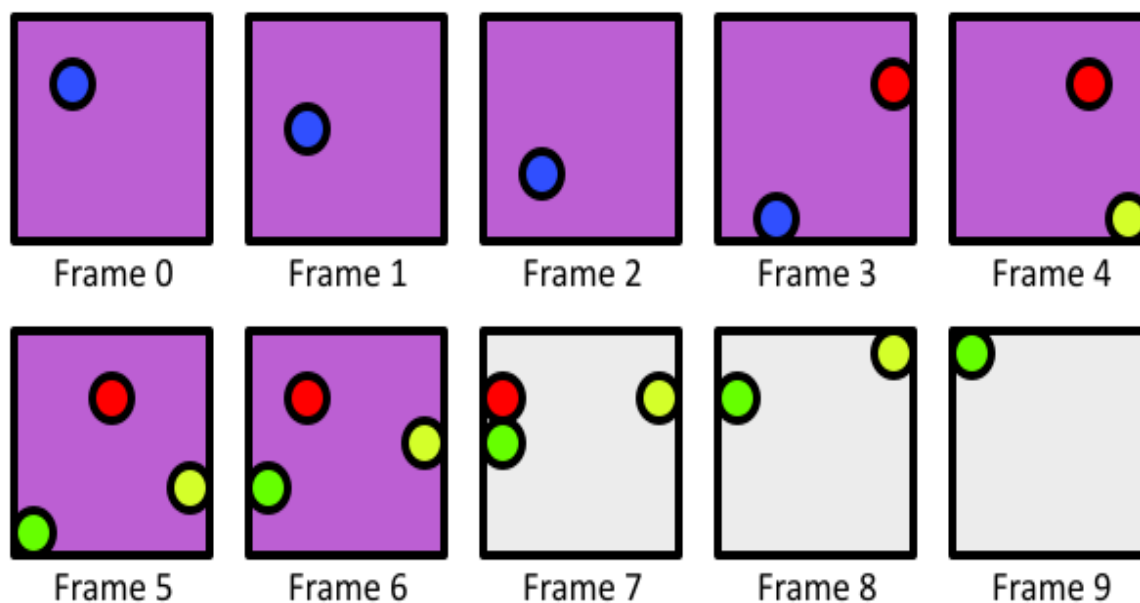


Рисунок 3.2 – Лінійний поділ датасету

Однак на невеликих вибірках, як показують практичні результати, оптимальніше обирати більший за половину відсоток вхідних матеріалів для тренування. Так, пропорції  $2/3$ ,  $3/4$ , або  $4/5$  збережуть невелику частку даних, достатню для перевірки роботи системи і нададуть максимум із доступних даних для навчання. В порівнянні з попередньою методикою, тут рівномірне покриття вибірки навчанням надає змогу рівномірно захопити майже уся інформацію що міститься у відео. І це даватиме якісно кращі результати при роботі з іншою вибіркою. Однак маючи один датасет та проводячи навчання та перевірку тренування на ньому ж, результати оцінки роботи нейромережі будуть завищеними через мізерну різницю об'єктів, їх місцеположення в кадрі та ракурсів на навчальних та тренувальних зображеннях.

Схематичне позначення розподілення датасету методом N-ного кадру. На тренувальний та навчальних вибірках приблизно однаково добре представлені клітини усіх типів у різних частинах кадру навіть при малій пропорції навчальної вибірки порівняно до тренувальної ( $1/2$ ). Збільшивши пропорцію до  $2/3$  в тренувальну вибірку потрапили б фрейми 2, 5 та 8.

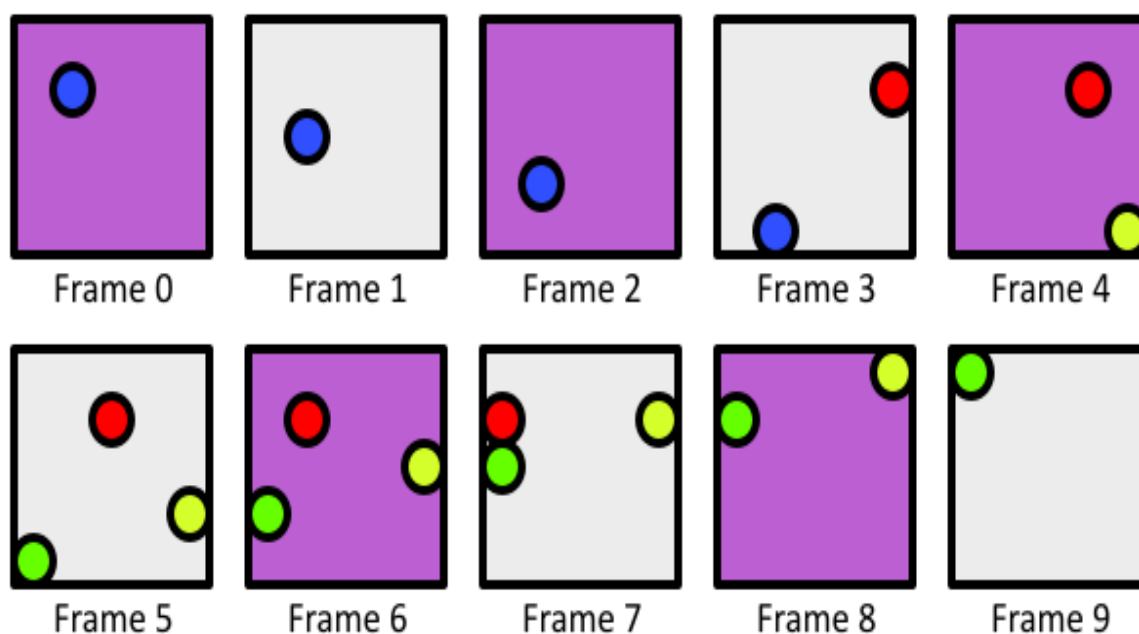


Рисунок 3.3 – Пропорційний поділ датасету

В обох випадках результати оцінювання точності сегментації були б доволі високими, але реальна ситуація з дещо інакшим датасетом могла б виявитися для системи значно важчим випробуванням аніж тренування на цьому ж датасеті.

Умовно «посереднім» між вищезгаданими двома підходами розподілення навчально-тренувальної вибірки хронологічної послідовності зображень можна визначити метод випадкового розподілення датасету на навчальну та тренувальну частини. Це дозволить відносно рівномірно розприділити датасет, залишивши місце деякій невизначеності. Розподіл схематично позначений на рисунку 3.4.

Метод випадкового розділення даватиме різні результати щоразу. Його раціонально використовувати декілька разів для отримання найбільш об'єктивної оцінки. Зі схеми видно що розподіл фреймів нагадує попередні два методи водночас, комбінуючи їх переваги.

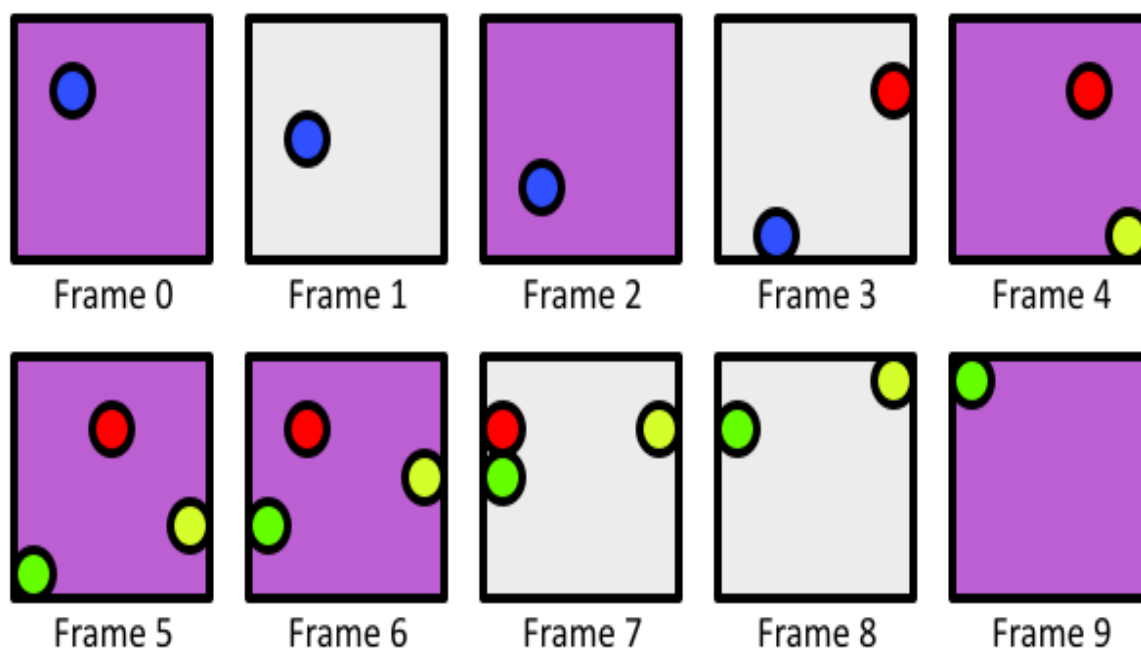


Рисунок 3.4 – Випадковий розділ датасету

Проте головними недоліками випадкового розподілу є неточність в контексті одного прогону, а задля декількох прогонів необхідно мати весь датасет підготовленим (розміченим) задля можливості використання будь-якого фрейму у якості навчального.

### 3.5 Розмітка об'єктів

Розмітка зображень для сегментації – це процес розмежування та ідентифікації окремих областей на зображенні. Основна мета сегментації полягає в тому, щоб призначити кожному пікселю на зображенні відповідну мітку або клас. Це дозволяє комп'ютерним системам розуміти та аналізувати зображення на більш високому рівні, а також допомагає у вирішенні задач розпізнавання об'єктів, відстеженні руху, сегментації частин тіла тощо. Процес розмітки проілюстровано на рисунку 3.5.

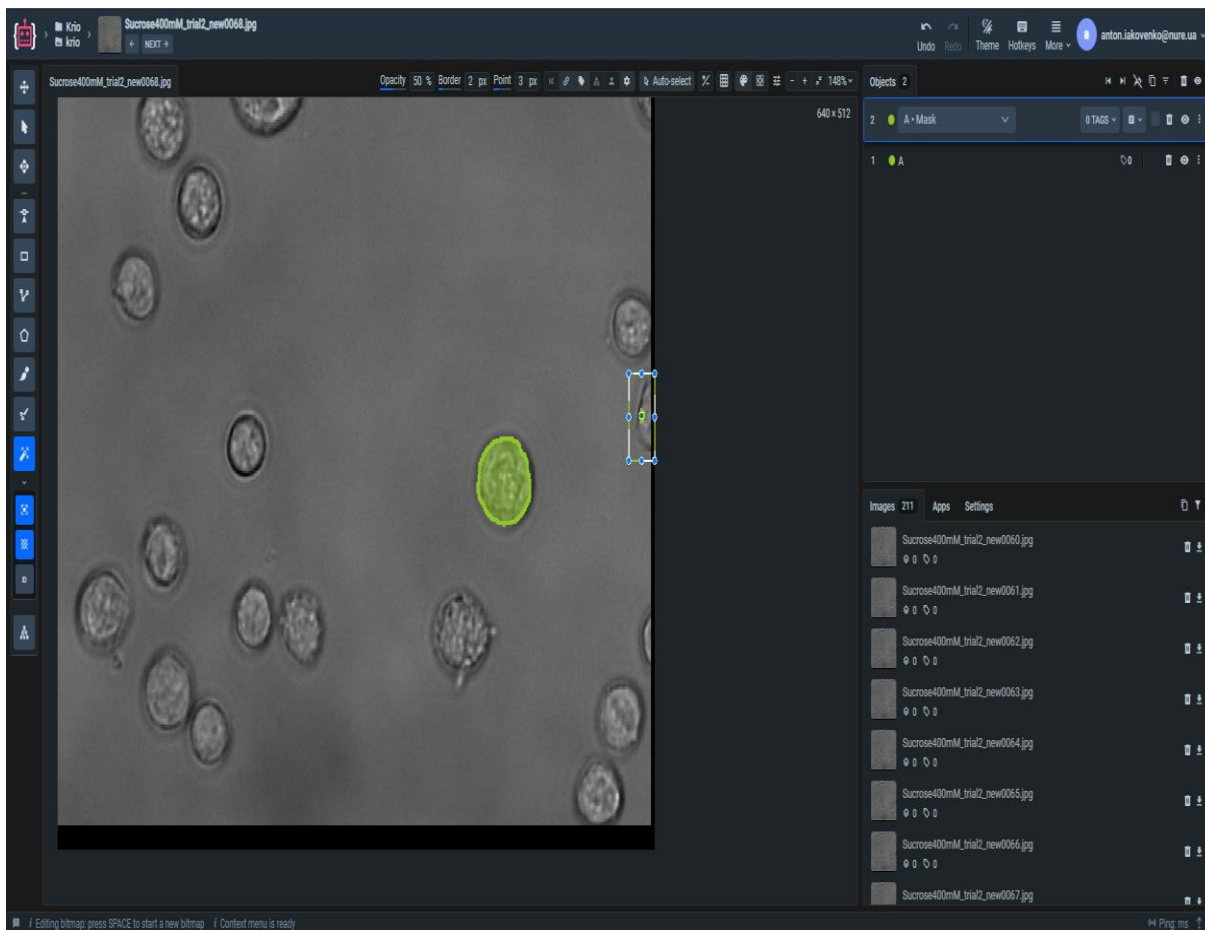


Рисунок 3.5 – Розмітка даних

Існують кілька типів розмітки для сегментації:

- піксельна розмітка (Pixel-level segmentation): кожен піксель на зображенні призначається до певного класу. Це найбільш точний тип сегментації, але вимагає значних обчислювальних ресурсів;
- регіональна розмітка (Region-based segmentation): зображення розбивається на області, які відповідають певним об'єктам або частинам зображення. Це менш точний, але менш обчислювально вимогливий підхід;
- розмітка з контуром (Outline segmentation): тільки контури об'єктів на зображенні відзначаються, а не кожен піксель всередині об'єкту. Це може бути корисним для визначення форм та розмірів об'єктів.

Розмітка зображень може бути здійснена вручну анотаторами, що вимагає багато часу та зусиль, або може бути автоматизованою за

допомогою алгоритмів комп'ютерного зору. Багато проектів з використанням нейронних мереж використовують комбінацію людської та автоматичної розмітки для створення навчальних наборів даних для тренування моделей сегментації.

Принципи присвоєння класів при розмітці зображень для сегментації залежать від конкретного завдання та потреб проекту. Нижче наведені деякі загальні принципи. Для додаткового контрасту в описі принципів наведені паралелізм іншою сферою застосування комп'ютерного зору – автомобільною детекцією ситуації на дорогах.

Класифікація за типом об'єкта: Один з основних принципів – це класифікація областей зображення за типом об'єкта, який вони представляють. Наприклад, у сценах з вулиць автомобілі можуть бути віднесені до класу «автомобіль», люди – до класу «людина», дерева – до класу «дерево» тощо. В контексті даного проекту типами об'єктів можуть виступати різні типи клітин при наявності їх різноманіття. Або ж лишити один клас при типовому наборі клітин, що спростить процес навчання і зменшить необхідний обсяг вхідних зображень.

Семантична розмітка: У цьому випадку кожний клас відображає не просто тип об'єкта, а його семантичну значущість. Наприклад, у сцені з вулиць пішоходи, велосипеди та автомобілі можуть бути об'єднані в клас «дорожній рух», але це не семантично однакові об'єкти. В контексті біомедицини це групування клітин по важливості для нинішньої задачі. Наприклад, виділення рухомих клітин на фоні нерухомих, фільтр по конкретній властивості, відстеження оклюзій, тощо.

Інстанційна розмітка: В цьому випадку кожен об'єкт на зображенні отримує власну унікальну мітку. Наприклад, якщо на зображенні є кілька автомобілів, кожен з них має власну мітку, що відрізняє його від інших автомобілів. Клітини по аналогії можуть відстежуватися у якості конкретних екземплярів, навіть при однаковій класовій та семантичній приналежності.

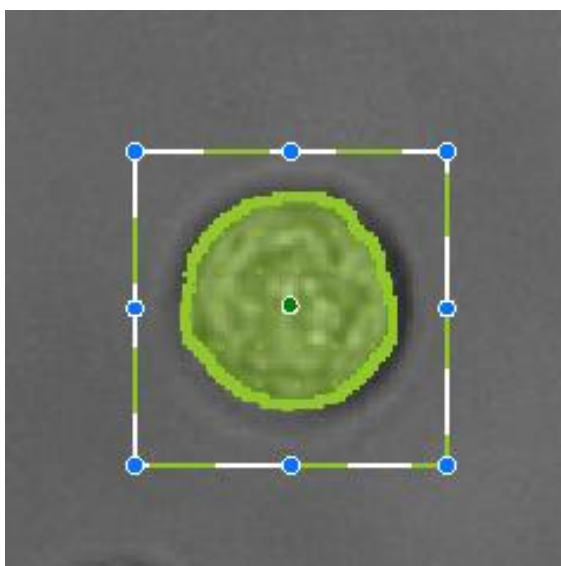
Фон та фонові класи: Для більшої точності сегментації часто вводять клас «фон», який відображає області на зображенні, які не представляють інтересу для аналізу. Додатково можуть вводитися класи для фонових об'єктів.

Маркери невизначеності: Іноді деякі області на зображенні може бути важко чітко класифікувати. У таких випадках використовують маркери невизначеності або «невідомі» класи, які вказують на те, що алгоритм сегментації не може однозначно визначити клас об'єкта.

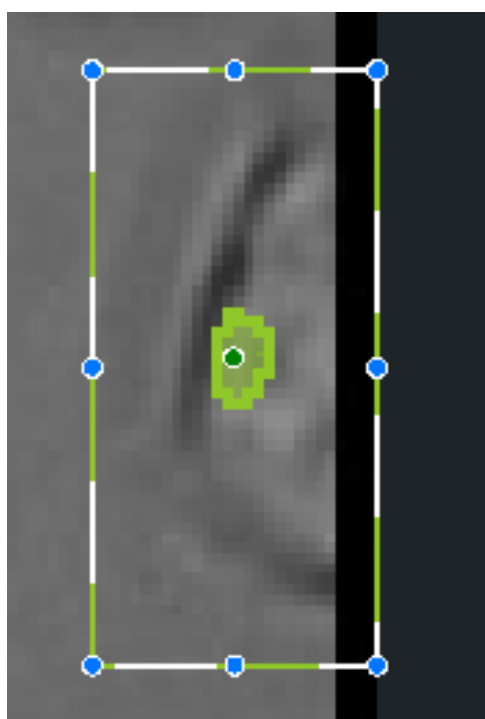
Ці принципи можуть застосовуватися окремо або в комбінації залежно від конкретного додатку та потреб проекту. Розуміння цих принципів допомагає побудувати більш точну та коректну систему сегментації зображень.

Оскільки розмітка даних проводиться вручну на відносно невеликому обсязі даних, інструментом для розмітки даних було обрано онлайн-сервіс «Supervisely». Він має досить широкий набір інструментів, в числі яких автоматичне розпізнавання контурів. Зазвичай воно працює непогано, але потребує деякої ручної корекції для отримання максимально якісних розмічених зразків даних. Об'єкти, що знаходяться в кутку кадру, на його межах та будь-яким чином перерізані сторонніми об'єктами, автоматично виділяються значно гірше.

Описана ситуація доводить недосконалість вбудованого алгоритму при роботі з обмеженим обсягом інформації (рисунок 3.6 та рисунок 3.7). На рисунку 3.6 зображені дві клітини. Одна із них має повний замкнений контур, через що її сегментувати легше. Інша має складнішу, ребристу, структуру контура і наявна в кадрі лише частково. Через це алгоритмічна сегментація стає набагато складнішою.



(a)



(б)

Рисунок 3.6 – Автоматична розмітка зображень:

(a) якісна сегментація у центрі кадру; (б) проблеми сегментації на краю кадру

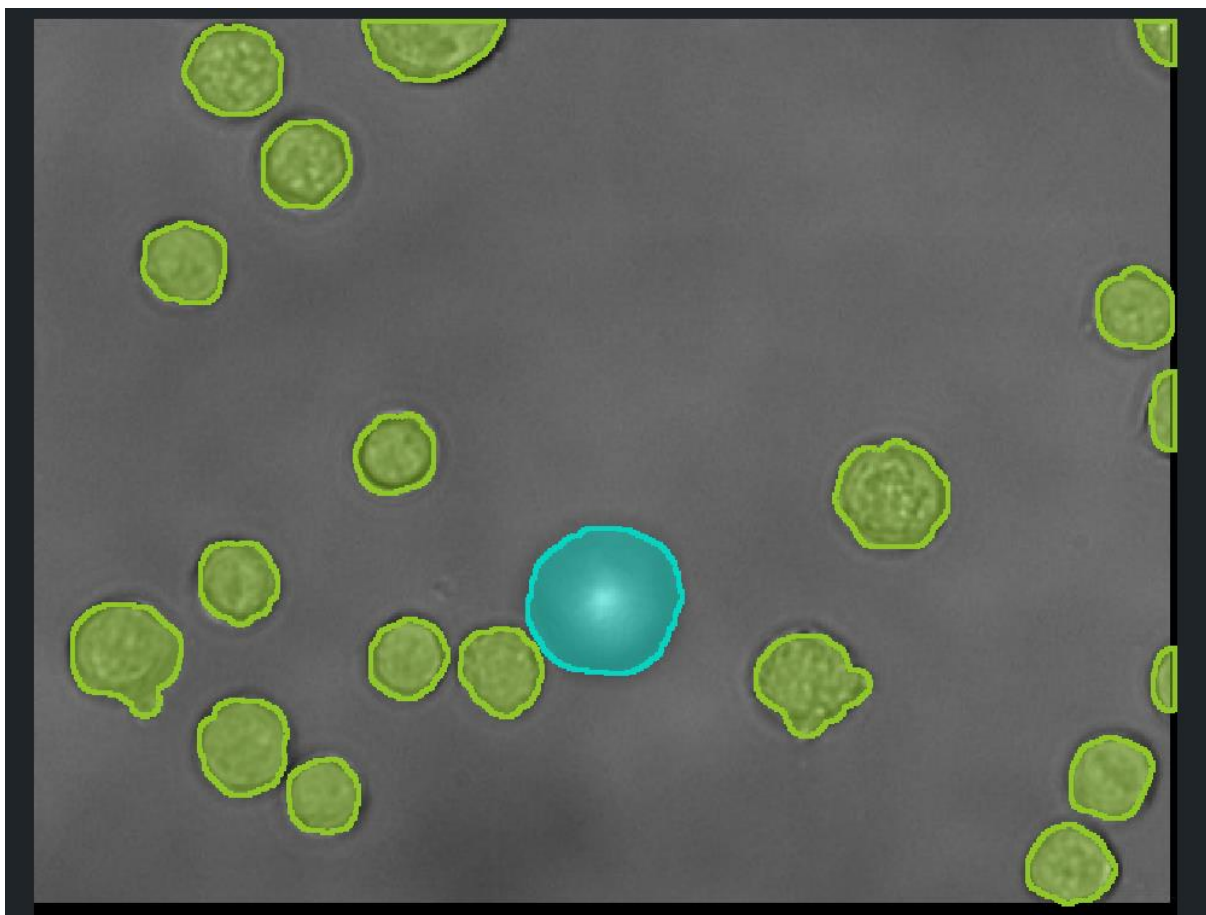


Рисунок 3.7 – Розмічені маски клітин

### 3.6 Експорт зображень

Після процесу розмітки даних, наступним кроком стане їх експорт.

Дані експортуються з середовища «Supervisely» у форматі пар зображення – маска. З метою оптимізації обчислювальних процесів, вхідні зображення подаються в одноканальному форматі. Тобто в чорно-білих тонах, де кожен піксель позначається одним байтом інформації від 0 для позначення абсолютно чорного, до 255 – абсолютно білого кольору відповідно.

У комп'ютерному зорі маски використовуються для попіксельної анотації об'єктів шляхом виділення відповідного об'єкту регіону на зображенні [18].

Маски – здебільшого бінарні. Представляючи зображення у вигляді двовимірної чи тривимірної матриці (в залежності від колірної каналності), її довжина по осі  $x$  та осі  $y$  має відповідати параметрам маски. Бінарна маска позначає пікселі, які належать до площини шуканого об'єкту позитивним значенням на фоні матриці, заповненої нулями. Таким чином позначається факт наявності об'єкту в кадрі та конкретна область зображення, де він розташований. У випадку коли існує кілька класів, треба позначити який конкретно із них описує знайдений об'єкт. В такому разі замість бінарного прапорця класової приналежності, кожному пікселю присвоюється значення від зарезервованого нуля до  $N$  включно ( $N$  – кількість класів об'єктів у даному датасеті). Такого ж ефекту можливо досягти збільшенням глибини матриці, за рахунок додавання третього бінарного виміру маски. У випадку сегментації екземплярів, який і буде застосовуватись, структура маски досить складна і поєднує обидві описані вище властивості. Тобто варіативність віднесення об'єкта до класу досягається за рахунок розширення діапазону станів пікселів у масці. А множинна детекція досягається шляхом додавання додаткової вимірної осі  $z$  до матриці маски. Така модель має здатність запобігати колізіям об'єктів і вони не зливаються у навчальних прикладах, навіть якщо віднесені до одного класу водночас. Багатокласова сегментація проілюстрована на рисунку 3.8. Процес формування її маски – на рисунку 3.9.

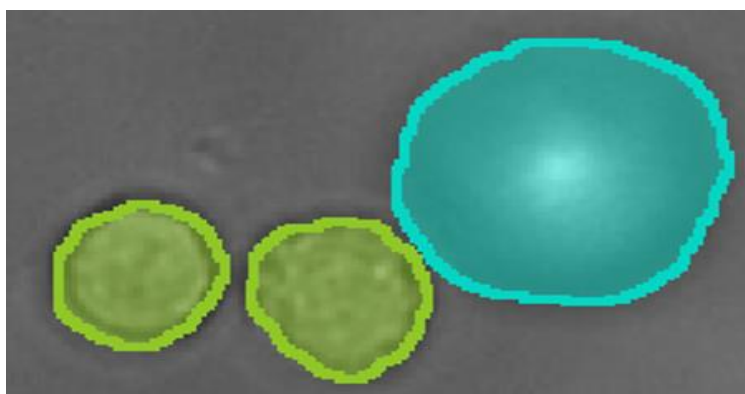


Рисунок 3.8 – Фрагмент зображення з трьома об'єктами двох різних класів



Рисунок 3.9 – Схематичний приклад формування маски для наведеного фрагменту зображення. На кожен присутній екземпляр створюється шар маски. Кожен шар може містити лише єдиний екземпляр об'єктів. І всі пікселі області, зайнятої екземпляром, позначені ідентифікатором класу.

### 3.7 Конвеєр даних

Конвеєр даних – це концепція або архітектурний підхід до обробки даних, де дані обробляються послідовно у вигляді потоку або конвеєра. Кожен етап обробки даних виконує певну операцію або аналіз, після чого результат передається до наступного етапу для подальшої обробки.

Ця концепція часто використовується в різних областях інформаційних технологій, таких як обробка даних, комп'ютерні мережі, обчислювальна архітектура та інші. Наприклад, у системах обробки даних, конвеєр може включати етапи, такі як збір даних, фільтрація, аналіз,

обробка та візуалізація. У комп'ютерних мережах, конвеєр даних може використовуватися для передачі пакетів даних через різні етапи маршрутизації та обробки.

Основна перевага конвеєра даних полягає в тому, що він дозволяє розділити процес обробки на більш прості кроки, що полегшує розробку, тестування та підтримку систем. Крім того, конвеєрна архітектура може бути ефективною для паралельної обробки даних, де різні етапи конвеєра можуть виконуватися одночасно на різних ресурсах обчислення [11].

У контексті TensorFlow, конвеєр даних також використовується для ефективною обробки даних, особливо в глибокому навчанні та інших областях машинного навчання. Основна ідея полягає в тому, щоб оптимізувати час обробки даних, шляхом попередньої обробки, упаковки та підготовки даних перед навчанням моделі.

TensorFlow надає API під назвою `tf.data`, який дозволяє ефективно завантажувати та підготовлювати дані для навчання моделей [26]. Це включає в себе можливість завантажувати дані з різних джерел, таких як файли на диску, а також різні операції для обробки даних, які можуть бути виконані паралельно.

За допомогою `tf.data` можна виконувати операції підготовки даних, такі як нормалізація, розбиття на пакети, перемішування тощо. Це дозволяє підготувати дані безпосередньо перед їх передачею в модель для навчання.

У деяких випадках, особливо коли маємо справу з великими обсягами даних, може бути корисним завантажувати дані партіями (батчами) або навіть частково завантажувати їх за допомогою генераторів даних.

TensorFlow може автоматично використовувати багатопотоковість та розподілене обчислення для швидкого завантаження та обробки даних. Це дозволяє ефективно використовувати ресурси системи для обробки даних навіть на багатоядерних або розподілених системах.

### 3.8 Оптимізація продуктивності конвеєра даних

Графічні процесори та TPU можуть радикально скоротити час, необхідний для виконання одного кроку навчання. Для досягнення максимальної продуктивності потрібен ефективний конвеєр введення, який доставляє дані для наступного кроку до завершення поточного кроку. API `tf.data` допомагає створювати гнучкі та ефективні вхідні канали. Цей документ демонструє, як використовувати `tf.data` API для створення високопродуктивних вхідних конвеєрів TensorFlow [3].

API включає цілий набір інструментів для оптимізації процесів обробки датасетів. На рисунку 3.10 відображений таймлайн процесів отримання вхідних даних з диску в межах двох тренувальних кроків без дій, спрямованих на оптимізацію. Епоха складається із трьох фундаментальних етапів:

- відкриття файлу (якщо він іще не був відкритий раніше);
- отримання запису даних із файлу;
- використання даних для навчання.

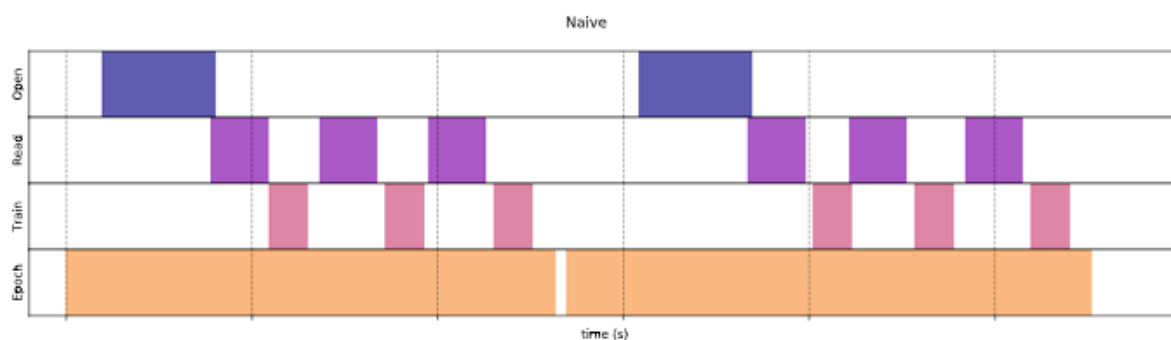


Рисунок 3.10 – Візуалізація взаємодії потоків без оптимізації

Етапи виконуються по чергово. Поки триває отримання даних чи читання файлу, модель простоює. Однак розпаралелення потоків дасть приріст у продуктивності системи.

Один із інструментів даної бібліотеки, які зможуть покращити загальну продуктивність системи – попередня вибірка. Поки триває один крок навчання, підвантажуються дані для іншого (рисунок 3.11).

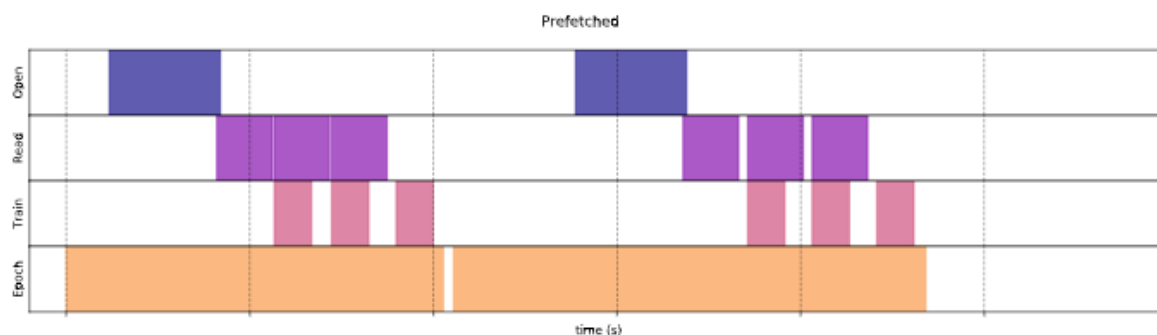


Рисунок 3.11 – Попередня вибірка

Паралельне зчитування кількох наборів даних також може пришвидшити процес їх обробки (рисунок 3.12).

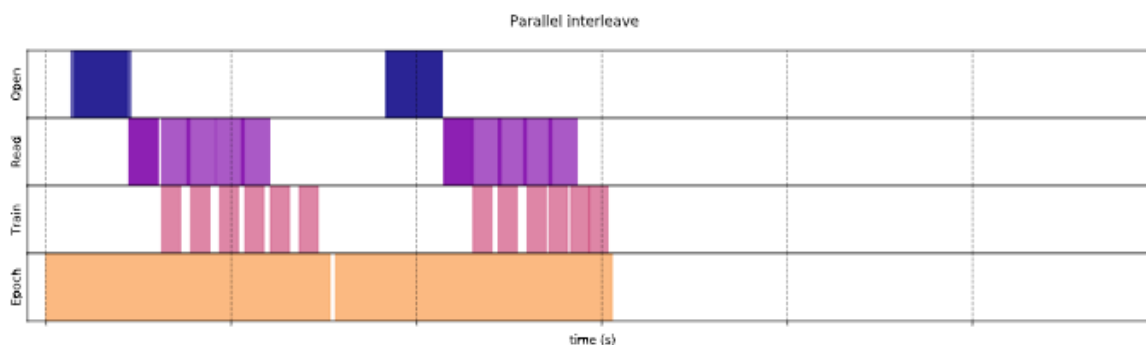


Рисунок 3.12 – Паралельне чергування

При виникненні потреби відображення оброблюваних даних (це корисний функціонал на етапі розробки системи, коли необхідне логування та деплой) також розпаралелення потоків відображення разом з іншими потоками обробки даних допоможе досягти оптимізації у роботі системи.

Діаграма потоків до того як вони були розпаралелені зображена на рисунку 3.13. Паралельні потоки ілюструються рисунком 3.14.

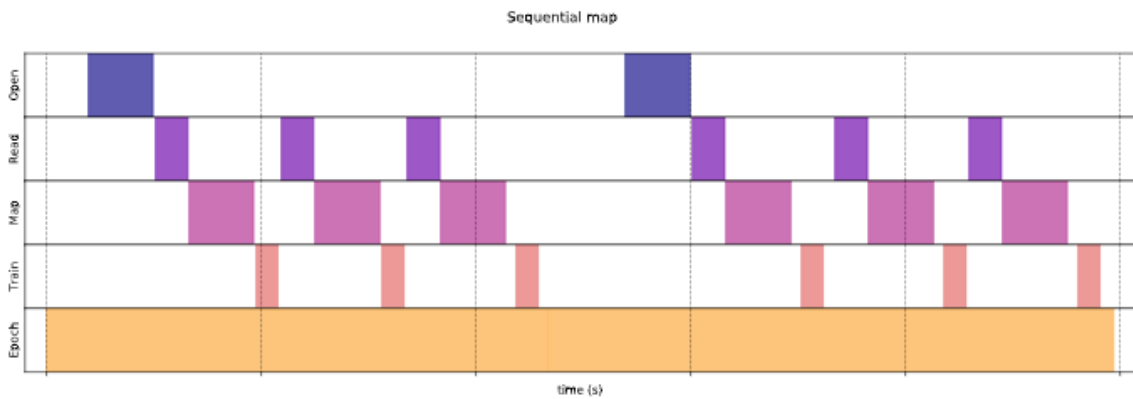


Рисунок 3.13 – Послідовні потоки

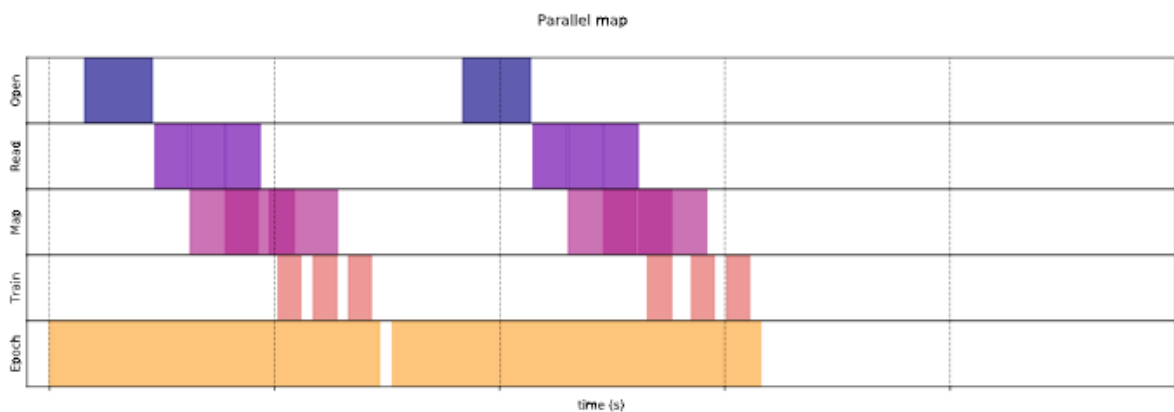


Рисунок 3.14 – Паралельні потоки

Кешування зображень дозволить повторно використовувати зображення для різних підзадач без необхідності щоразу їх завантажувати і читати, що гіпотетично може бути вузьким місцем у продуктивності системи. Діаграма кешування зображена на рисунку 3.15.

Комбінація зазначених методів оптимізації дає значний результат для економії ресурсів. Діаграма оптимізованої системи представлена рисунком 3.16.

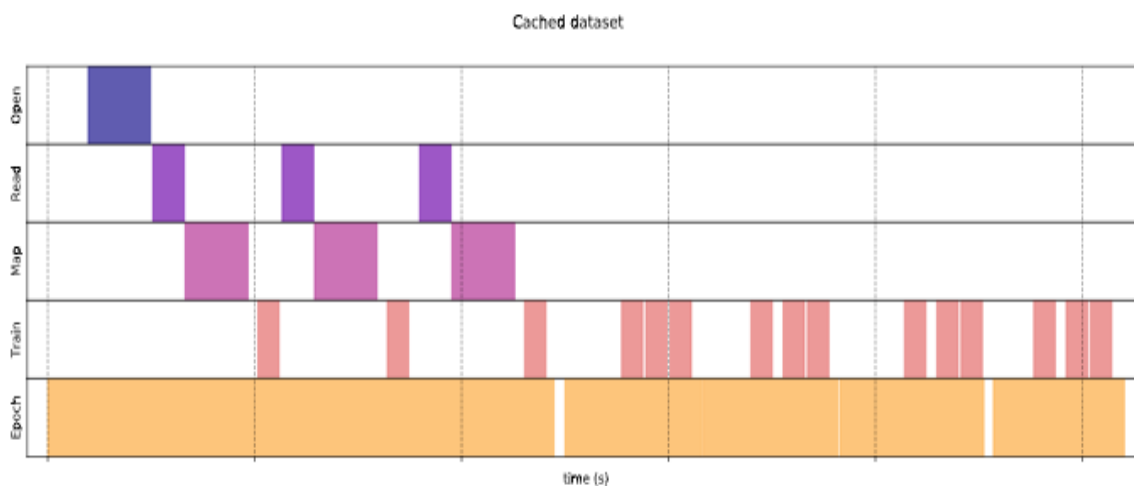


Рисунок 3.15 – Кешування

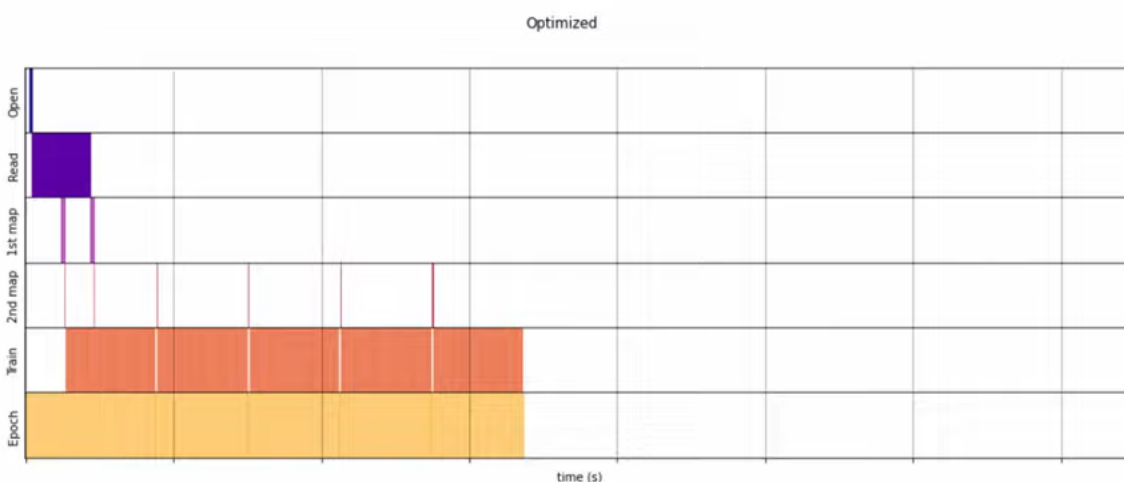


Рисунок 3.16 – Діаграма оптимізованого конвеєра даних

Ще у арсеналі доступних інструментів API є кілька направлених на векторизацію та скалярне відображення графіки. Але зважаючи на невеликі обсяги даних, розміри зображень та кількість об'єктів на них, ці методи використовуватись у ході даної роботи не будуть через відсутність оптимізації на відносно невеликих наборах даних. І, відповідно, тому не розглядаються.

### 3.9 Аугментація зображень

Аугментація даних – це процес створення нових прикладів даних шляхом застосування різноманітних трансформацій до існуючих даних. Це часто використовується у сферах машинного навчання і глибокого навчання для поліпшення роботи моделей, особливо коли набір даних обмежений.

Наприклад, у випадку обробки зображень аугментація може включати зміну розміру, обертання, зміщення, зміну контрасту, яскравості тощо. У текстових даних можна застосовувати аугментацію, додавши синоніми, випадково видаляючи або додаючи слова тощо [26].

Аугментація даних допомагає уникнути перенавчання, розширює розпізнавання і збільшує робочий діапазон моделі, дозволяючи їй краще узагальнювати знання, навчені на обмеженому наборі даних (рисунок 3.17).

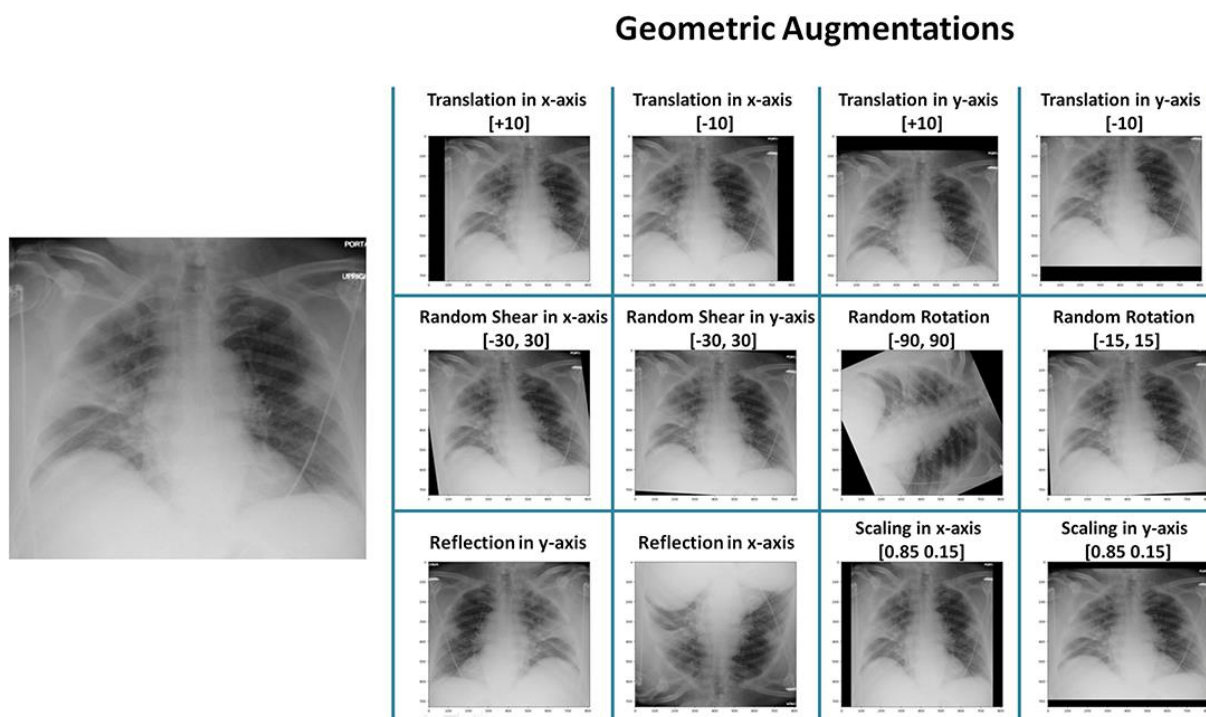


Рисунок 3.17 – Приклад аугментації зображення

Для пришвидшення процесів аугментацію також раціонально виконувати «на льоту». Приймаючи на вхід зображення та маску, система

обробляє їх, зчитуючи, змінюючи контрастність, кут нахилу, перегортаючи, обрізаючи, масштабуючи та деформуючи зображення, що в кінцевому рахунку позитивно впливає на датасет, збільшуючи варіативність вибірки.

У вирішенні задачі використовується одноканальний датасет невеликих зображень. Найбільш ефективними методами аугментації для нього видалися експериментальним шляхом кілька методів, серед яких випадкове обрізання фрагменту зображення. Цей метод в разі збільшує кількість об'єктів, що знаходяться на грані чи в кутку кадру, що надзвичайно важливо. Бо через відсутність явних ознак об'єкта, який знаходиться в кадрі частково і має перекриття межами зображення понад 60%, такі об'єкти важче впізнати, а тим паче виділити конкретний клас, контур чи екземпляр. Отже, збільшивши кількість неповних ракурсів при навчанні системи, вона краще справлятиметься з неповними даними і не втрачатиме фокус з об'єкта при виникненні найменшої оклюзії.

Також добре себе зарекомендував метод повороту зображення. Таким чином, маючи статичну клітину, нейромережа не фокусуватиметься виключно на її ознаки відносно ширини чи довжини по відношенню до ракурса.

### 3.10 Завантаження та попередня обробка даних

В першу чергу треба завантажити дані з диску. Дані є парою змінних `images` та `masks` (лістинг 3.1). Зображення мають однаковий ключ-ідентифікатор, на який ітеративно заміняється оператор-зірочка, виставлений у якості аргумента. Експортовані з інструменту для розмітки зображення та їхні маски знаходяться у різних теках у форматах «.jpg» та «.png» відповідно. Формат файлів «.jpg» є форматом вхідних даних, як універсальний та досить розповсюджений. Для масок використаний формат «.png» оскільки він має підтримку альфа-каналу. Тобто може зберігати пусті пікселі, що мають нульове значення непрозорості.

Лістинг 3.1 – Програмний код завантаження зображень і клонування датасету для аугментації

```
images = sorted(glob.glob('krio/img/*.jpg'))
masks = sorted(glob.glob('krio/masks/*.png'))
images_dataset =
tf.data.Dataset.from_tensor_slices(images)
masks_dataset = tf.data.Dataset.from_tensor_slices(masks)
dataset = tf.data.Dataset.zip((images_dataset,
masks_dataset))
dataset = dataset.map(load_images,
num_parallel_calls=tf.data.AUTOTUNE)
dataset = dataset.repeat(60)
dataset = dataset.map(augmentate_images,
num_parallel_calls=tf.data.AUTOTUNE)
```

Метод `tf.data.Dataset.from_tensor_slices` формує набір даних із завантажених зображень та масок. Далі вони об'єднуються методом `tf.data.Dataset.zip` для описаної в попередніх підрозділах оптимізації. А саме, паралельної обробки.

Після цього дані завантажуються в пам'ять і копіюються визначену кількість разів. Наступним кроком виконується аугментація даних, яка також була описана у попередньому розділі. Код аугментаційної функції наведено нижче. У ній виконуються такі трансформації як обрізання зображення до випадкової частини, змінення масштабів та віддзеркалення зображень. Комбінація функцій дублювання даних та їх аугментації дозволяє штучно збільшити датасет відносно різноманітними зображеннями у десятки разів (лістинг 3.2). А кожне навчальне зображення разом із масками є унікальним. Через випадковість трансформацій, при отриманні того ж вхідного датасету можна перетворювати його на різні дані для навчання.

## Лістинг 3.2 – Програмний код функції аугментації

```
def augmentate_images(image, masks):
    random_crop = tf.random.uniform((), 0.3, 1)
    image = tf.image.central_crop(image, random_crop)
    masks = tf.image.central_crop(masks, random_crop)
    random_flip = tf.random.uniform((), 0, 1)
    if random_flip >= 0.5:
        image = tf.image.flip_left_right(image)
        masks = tf.image.flip_left_right(masks)
    image = tf.image.resize(image, SAMPLE_SIZE)
    masks = tf.image.resize(masks, SAMPLE_SIZE)
    return image, masks
```

Штучно збільшений датасет розділяється на навчальний і тренувальний і пакетно передається на обробку (рисунок 3.18).

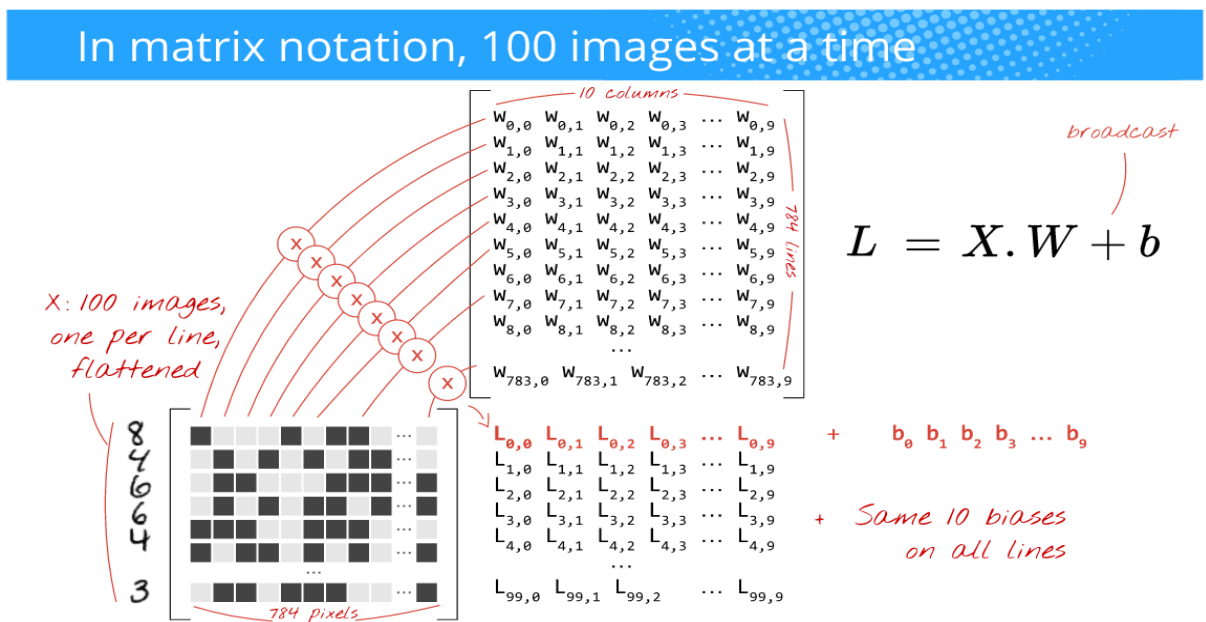


Рисунок 3.18 – Схематичне позначення подачі 100 зображень як пакету навчальних даних

Навчання моделі відбувається поступово, пакетами даних. Їх розмір є гіперпараметром, що впливає на швидкість обробки матеріалу але водночас

обмежується доступними ресурсами відеопам'яті. Розмір пакета впливає на час навчання, відсоток помилок, зміщення градієнта тощо. Немає загального емпіричного правила щодо того, який розмір партії працює найкраще. Цей гіперпараметр встановлюється експериментальним шляхом, після чого обирається найбільш ефективний. Зазвичай використовуються невеликі пакети, від 32 до 2048 зображень. Оскільки датасет досить малий, експериментальним методом було обрано розмір пакету у 8 або 16 зображень [26], [27]. Код пакетного завантаження відображений у лістингу 3.3.

Лістинг 3.3 – Програмний код завантаження пакетів по 8 зображень

```
train_dataset = train_dataset.batch(8)
test_dataset = test_dataset.batch(8)
```

### 3.11 Спосіб обробки зображень

Вхідні зображення мають кілька сотень пікселів у висоту та в ширину. Тобто в оригінальному розмірі є достатньо великими для повноцінної цілісної обробки. Великі зображення можуть містити деталі, які важко обробити за допомогою стандартних архітектур нейромереж через обмеження пам'яті та обчислювальних ресурсів. Нижче розглянуті різні методи, які дозволяють ефективно обробляти великі зображення нейронними мережами.

Одним з найпоширеніших підходів є розбиття великого зображення на менші частини, або патчі, які можуть бути оброблені окремо. Кожен патч обробляється нейромережею, після чого результати обробки об'єднуються. Цей метод дозволяє зменшити навантаження на пам'ять та процесор, забезпечуючи при цьому високу роздільну здатність обробки.

Масштабування зображення передбачає зменшення його розміру до рівня, який може бути оброблений нейромережею. Після обробки

зменшеного зображення результати масштабуються назад до оригінального розміру. Цей метод є ефективним у випадках, коли дрібні деталі не є критично важливими для кінцевого результату.

Ієрархічна обробка використовує мультискалярний підхід, де зображення обробляється на різних рівнях масштабу. Спочатку аналізуються грубі деталі, а потім, поступово, більш детальні рівні. Це дозволяє нейромережі враховувати як глобальну структуру зображення, так і його дрібні деталі, що покращує загальну якість обробки.

Глибокі генеративні моделі, такі як автокодері (autoencoders) та генеративні змагальні мережі (GANs), також можуть бути використані для обробки великих зображень. Автокодері дозволяють зменшити розмірність зображення перед обробкою і відновити його після обробки, тоді як GANs можуть генерувати високоякісні зображення, що відповідають вхідному великому зображенню.

Контекстуальні методи, такі як рекурентні нейронні мережі (RNNs) та трансформери (Transformers), дозволяють враховувати довготривалі залежності між різними частинами зображення. Це особливо корисно при обробці великих зображень, де важливо враховувати контекст сусідніх частин зображення.

Методи обробки на основі регіонів, такі як R-CNN (Region-based Convolutional Neural Networks) та Mask R-CNN, використовують регіональні пропозиції для виділення важливих областей на зображенні. Ці методи дозволяють обробляти лише релевантні частини зображення, що зменшує обчислювальні витрати та підвищує ефективність.

Паралельна обробка включає використання розподілених обчислювальних платформ, таких як Hadoop або Spark, а також графічних процесорів (GPUs) для обробки великих зображень. Цей підхід дозволяє значно прискорити процес обробки за рахунок паралельного виконання обчислень.

Підсумовуючи властивості цих методів, найбільш підходящими для даної задачі виглядають розбиття на підзображення методом ковзаючого вікна або масштабування зображення. Перший працюватиме краще з великими зображеннями (понад 1000 пікселів у висоту і ширину) або з малими об'єктами, що займають невелику площу на зображенні і втратять впізнаваність при масштабування або навіть зникнуть з зображення зовсім. Масштабування до меншого розміру ефективніше при менших зображеннях. Коли коефіцієнт масштабу буде невисокий то втрати деталізації будуть незначні і кожне зображення після трансформації оброблятиметься стабільно швидко.

### 3.12 Згортка

Нейромережева модель побудована за принципом енкодер-декодер, за принципом обраної архітектури. Її перша частина – згорткова мережа. Вона складається із згорткових шарів пакетної нормалізації та активаційної функції Leaky ReLU (рисунок 3.19). Опційно можна використати пулінг-шари. Але було прийнято замість них використати параметр `strides` у самому згортковому шарі.

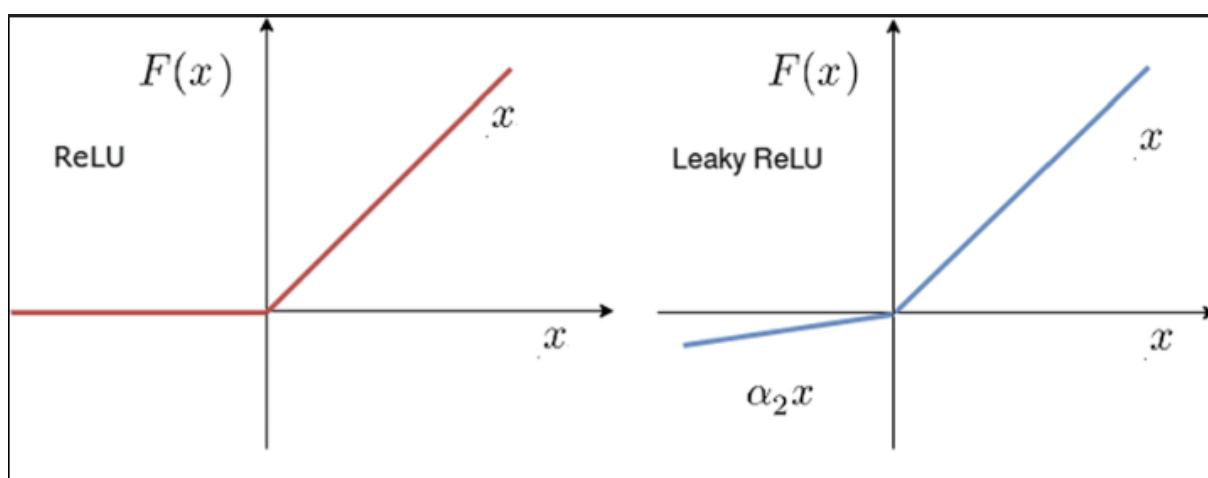


Рисунок 3.19 – Порівняння активаційних функцій

Leaky ReLU (Rectified Linear Unit) є варіантом активаційної функції ReLU, яка замість того, щоб обнуляти всі негативні значення, зберігає їх зі зменшеною величиною.

Використання Leaky ReLU в архітектурі U-Net для згортки зображень має кілька переваг, які роблять його привабливим вибором, порівняно з іншими активаційними функціями. Використання Leaky ReLU може призвести до швидшого і стабільнішого навчання глибоких нейронних мереж, оскільки вона допомагає уникнути проблеми затухання градієнта, яка часто зустрічається в дуже глибоких архітектурах [29]. У випадку задач сегментації зображень, зокрема в архітектурі U-Net, Leaky ReLU допомагає зберігати більше деталей і контексту в зображеннях завдяки своїй здатності передавати інформацію навіть через негативні значення [22].

Завдяки плавнішому розподілу градієнтів, Leaky ReLU може допомогти зменшити ризик перенавчання моделі, оскільки функція активації не обнуляє значення повністю і дозволяє модель навчатися більш узагальненим особливостям [17].

Leaky ReLU (Rectified Linear Unit) відрізняється від ReLU тим, що у Leaky ReLU негативні значення мають невеликий нахил замість нуля. Це означає, що замість обнулення всіх негативних вхідних значень, Leaky ReLU пропускає їх з певним коефіцієнтом (зазвичай невеликим, наприклад, 0.01). Такий підхід дозволяє уникнути проблеми «мертвих» нейронів, коли великі частини мережі перестають навчатися через те, що їхні градієнти стають нулями при використанні стандартного ReLU.

Пулінг-шари допомагають зменшити обчислювальну складність і зменшити кількість параметрів в моделі, зберігаючи при цьому найважливіші особливості. У порівнянні з ними, використання шару згортки (Conv2D) виконують згортку і зменшують розмірність, але можуть зберігати більше контекстної інформації завдяки навчуваним фільтрам. Хоч вони і мають трохи більшу обчислювальну складність у порівнянні з пулінг-

шарами, проте можуть зберігати більше контекстної інформації за пулінг-шари.

Друга частина нейронної мережі складається з комбінації transpose згорткових шарів, шарів пакетної нормалізації, dropout шарів та активаційної функції ReLU.

Dropout шари є інструментом для перенавчання мережі, а також загально позитивно впливають на процес її навчання. Шари batch (пакетної нормалізації) застосовуються для підвищення продуктивності нейромережі, а також загальної стабілізації її роботи.

Transpose згорткові шари, також відомі як дезгорткові шари або зворотні згорткові шари (deconvolutional layers), використовуються для підвищення просторової роздільної здатності вхідних даних.

Transpose шари є протилежністю звичайних згорткових шарів і часто застосовуються в архітектурах, де потрібно збільшити розмір зображення, наприклад, у генеративних змагальних мережах (GAN), автоенкодерах та в U-Net для сегментації зображень. Вони беруть вхідні дані з меншою роздільною здатністю і перетворюють їх у вихідні дані з більшою роздільною здатністю. Це досягається за допомогою процесу, який можна уявити як «згортку у зворотному напрямку» [22], [28].

Особливістю згорткової архітектури U-Net є застосування skip-connections. Наразі це стандартний інструмент для багатьох згорткових архітектур. Дана техніка часто застосовується у глибоких архітектурах і позитивно дається взнаки на збіжності моделі.

Завдяки міжшаровим з'єднанням інформація низького рівня спільно використовується вхідними та вихідними шарами. А в архітектурі U-Net ці з'єднання використовуються для того щоби передати ознаки, отримані в енкодері в декодер. Це допомагає відновити просторову інформацію, яка втрачається в результаті операцій згортки.

В цілому, skip-connections (рисунок 3.20) дозволяють використовувати повторно ознаки, що були раніше вже були вилучені в

декодері та стабілізують процес навчання та процес збіжності нейронної мережі.

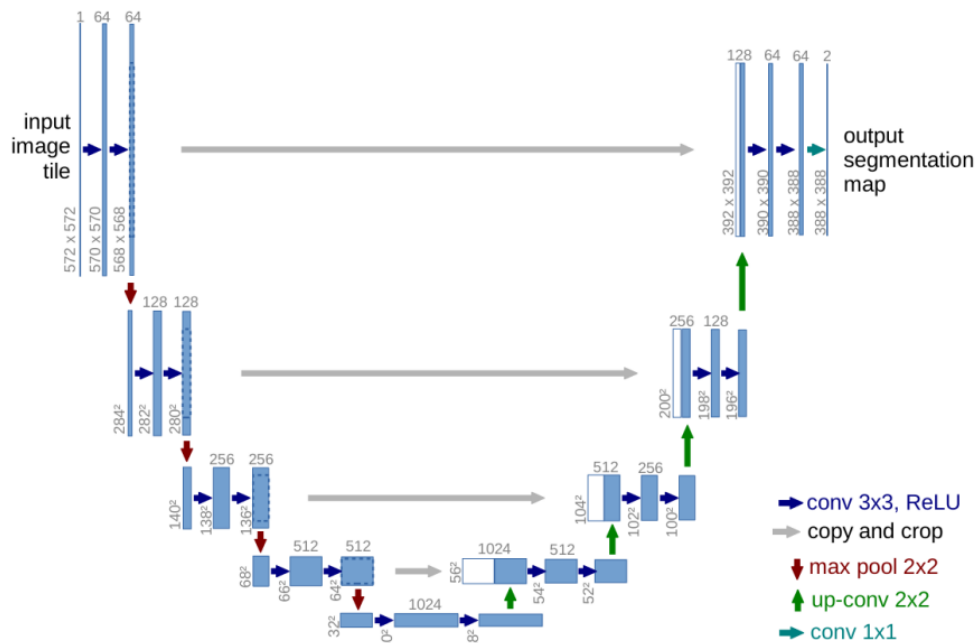


Рисунок 3.20 – Skip-connections в архітектурі U-Net

### 3.13 Реалізація моделі

Модель побудована на модулях бібліотеки «Keras». За її допомогою вдалось компактно реалізувати основну логіку нейронної мережі.

Функція: `input_layer` – задає вхідний шар нейронної мережі. А також встановлює розмір вхідних даних.

Функція `downsample_block` описує блоки, сформовані енкодером. Вона задає метод ініціалізації вагових коефіцієнтів, створює згортковий шар, додає шар пакетної нормалізації та встановлює активаційну функцію.

Функція `upsample_block` допомагає формувати декодер нейронної мережі. Тільки замість функції згортки використовується протилежна їй

функція «Conv2DTranspose». Також у цьому блоці перебачена можливість додавання Dropout шарів.

І остання функція `output_layer` – задає вихідний шар, розмірність якого відповідає кількості класів об'єктів на зображенні та використовує сигмоїдну активаційну функцію.

Опісля визначення цих функцій було створено масив `downsample_stack`, котрий представляє енкодер та `upsample_stack`, котрий представляє декодер. Така структура даних дозволить достатньо просто реалізувати `skip connections`.

За допомогою циклів блоки енкодера та декодера з'єднуються між собою та реалізуються міжшарові з'єднання через конкатенацію.

В результаті на базі Keras створюється модель, в якій вказуються вхідні та вихідні шари.

### 3.14 Навчання нейронної мережі

Навчання відбувається скомпільованою мережею з визначеними метриками у наступному підрозділі. У якості алгоритму оптимізації використаний «adam». Стандартні гіперпараметри, задані фреймворком, є досить оптимальними і в кінцевому варіанті вони були залишені без змін.

Також експериментальним шляхом було обрано необхідну кількість епох для отримання прийнятної сегментації. У наступному підрозділі наведена статистика оцінок роботи нейронної мережі на різних епохах. Глобально такий параметр як кількість епох може варіюватися в залежності від конкретного датасету, зважаючи на кількість об'єктів, кількість навчальних зображень, їх якість, наявність колізій об'єктів, їх класів, відмінностей об'єктів різних класів, кінцевої мети сегментації, тощо.

На датасеті, який використовується у якості навчального для даної задачі, мінімальною необхідною кількістю епох виявилось значення у 20 ітерацій.

### 3.15 Метрики та функції втрат

Алгоритми глибокого навчання використовують метод стохастичного градієнтного спуску. Це метод для знаходження мінімуму або максимуму цільової функції. Цільова функція є математичним представленням ефективності роботи нейромережі. Для вирішення задачі сегментації можуть застосовуватися різноманітні функції або їх комбінації. Одними з найбільш популярних є `binary cross entropy` та `dice`.

Якщо нейронна мережа видає відповідь, близьку до правильної, тоді значення функції втрат буде малим. В протилежному випадку, значення цільової функції буде великим. Коректний вибір функції втрат впливає на швидкість та якість навчання моделі. В даній моделі застосовується комбінація двох функцій втрат, названих у попередньому абзаці: `binary cross entropy` та `dice`.

Функція `binary cross entropy` дає високу збіжність моделі. Навчання проходить більш стабільно при збалансованому наборі даних. А функція `dice` є досить результативною в задачах сегментації. Але за своєю внутрішньою структурою не може забезпечити якісну збіжність. Такий вибір допомагає системі впоратися зі слабко збалансованим набором даних та досягти прийнятних результатів навчання.

Для оцінки точності роботи нейронної мережі використовується коефіцієнт `dice = 0.3`.

Далі визначена функція `dice_multiclass_metric`. Вона приймає на вхід два аргументи. Один із них – відповідь нейромережі. Другий – результат, який очікувано має вийти. Метрика дає оцінку точності результатів роботи нейронної мережі. Зображення зчитуються багатоканальні маски, на кожен із визначених класів. І за допомогою циклу шукається середнє значення коефіцієнта по кожному екземпляру об'єкта, що присутній в кадрі. Функція втрат, відповідно віднімає отриману оцінку від константи – одиниці.

Також визначається гібридна функція втрат: `dice binary cross-entropy multiclass loss`. Вона складається з суми двох функцій: стандартної функції крос-ентропії з бібліотеки Tensorflow (рисунок 3.22), і функції втрат `dice`, за допомогою коефіцієнта 0.3 зменшеного (рисунок 3.21).

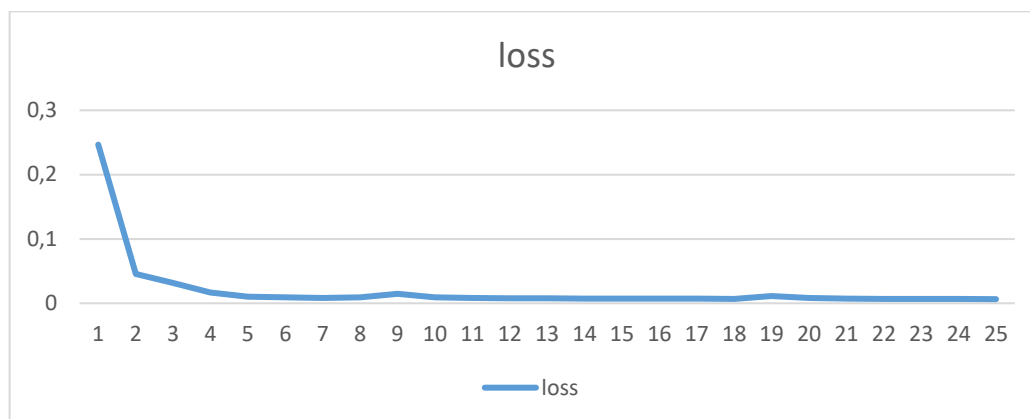


Рисунок 3.21 – Графік функції втрат

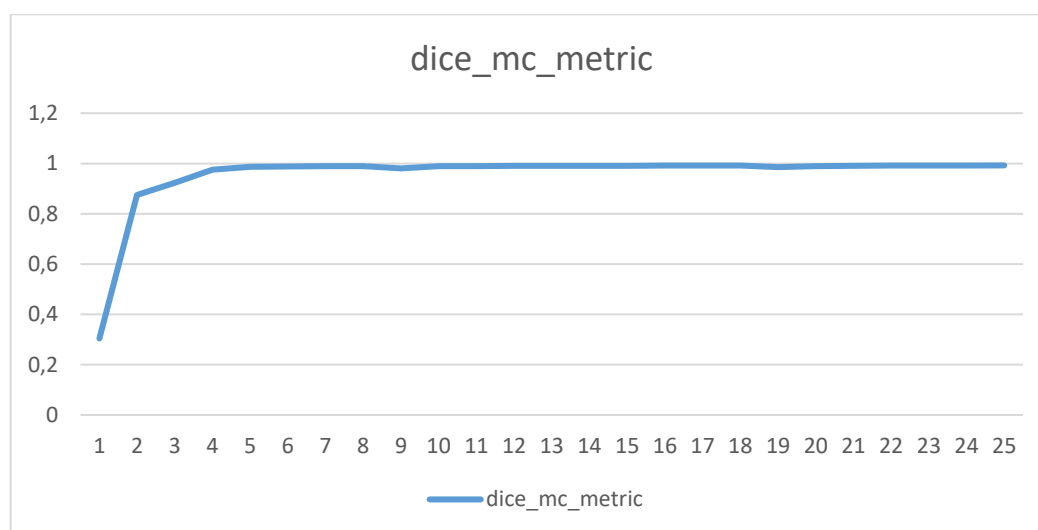


Рисунок 3.22 – Графік точності роботи нейромережі

Після навчання моделі на тестових даних з 25-ма епохами були отримані результати зазначених метрик. З коливань графіків можна оцінити що необхідний мінімум епох для відносно якісного навчання і оцінки складає близько 20. Функція втрат `val_loss` (рисунок 3.23) має суттєві коливання на 3 та 8 епохах. Далі коливання стають менш значними. З

графіка функції `val_dice_mc_metric` (рисунок 3.24) наочно видно відсутність стабільності до 20-ї епохи. Після 25 епох навчання обидві функції втрат мають незначні відхилення у результатах. Оцінка знаходиться в межах 0.6% та 1%.

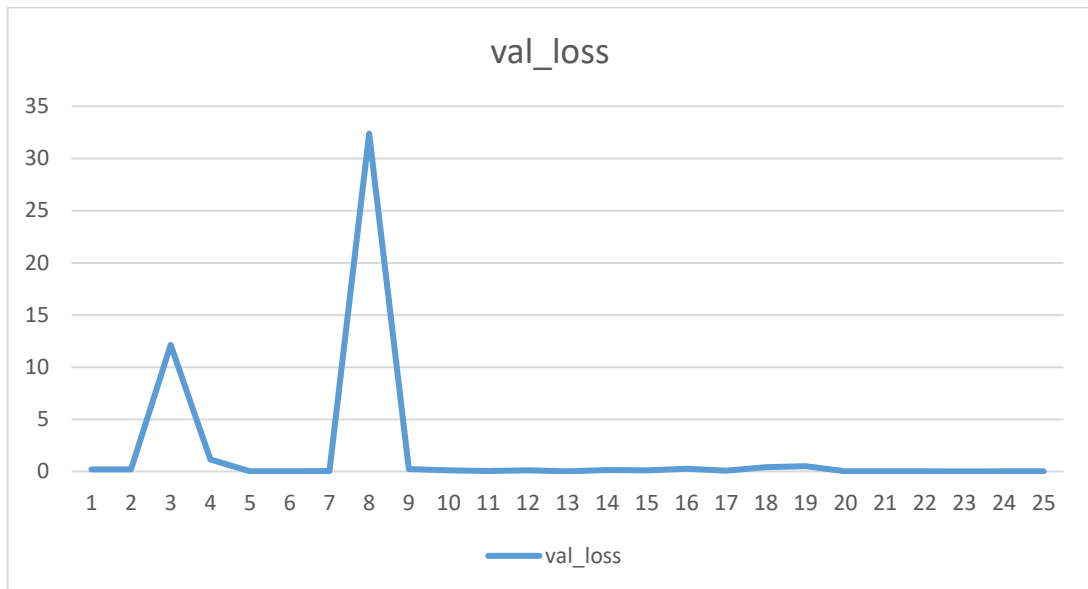


Рисунок 3.23 – Графік функції втрат при валідації

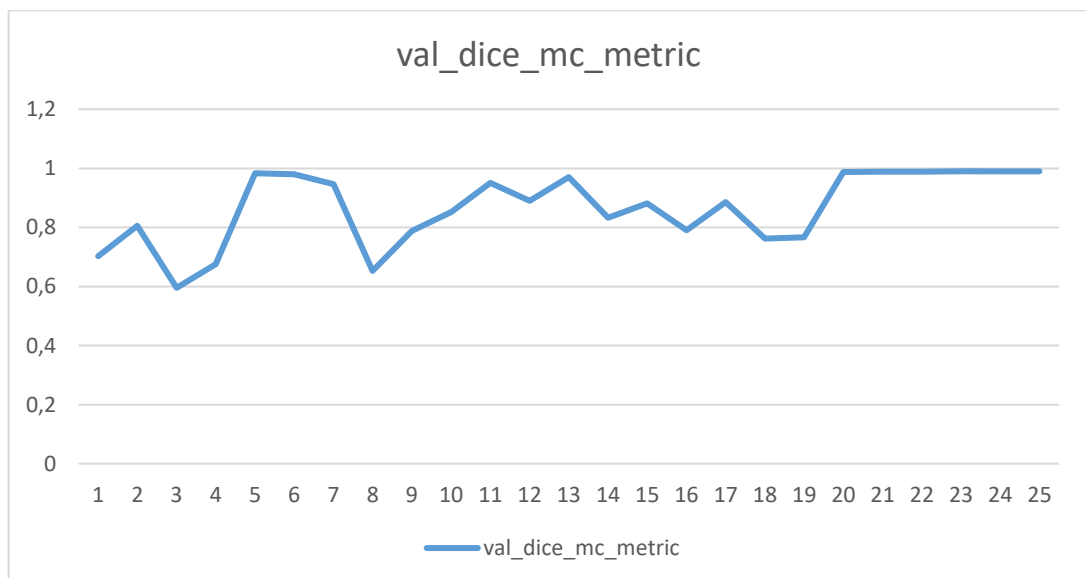


Рисунок 3.24 – Графік точності роботи нейромережі при валідації

### 3.16 Проблеми детекції

Можна оцінити що нейронна мережа, яка побудована у ході виконання даної роботи, виконує свою основну функцію. Нижче приведено декілька зображень з візуалізацією результатів детекції клітин. У них достатньо чітко розпізнані у кадрі клітини, виокремлені їх контури. Попри це, виокремлюються декілька проблем з розпізнаванням об'єктів.

По-перше, наявна проблема розпізнавання кутових об'єктів. На рисунку 3.25 видно неякісну детекцію клітини, розташованої в лівому верхньому куті. В кадрі наявна лише чверть контуру та площі об'єкта. Мережа коректно фіксує факт наявності об'єкту на його місці (рисунок 3.25). Проте відсутність помітних перепадів контрастності, властивим краю клітини, хибно змушує систему двічі виокремлювати видимий контур. А основна видима частина клітини проігнорована.

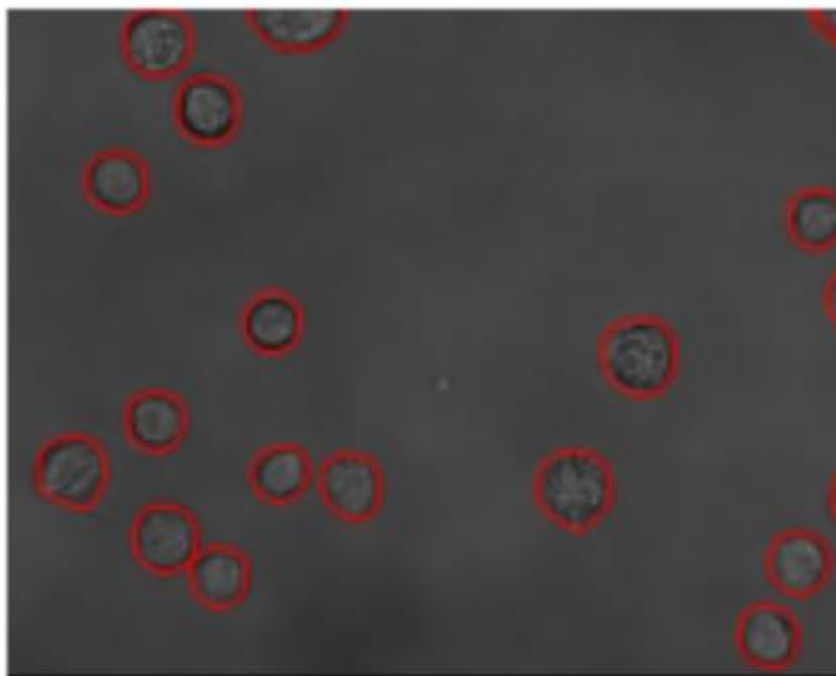


Рисунок 3.25 – Неякісна детекція клітини

Також варто зауважити що в навчальному наборі об'єктів, розташованих саме в кутку було відносно мало. При збільшенні відсоткового відношення об'єктів з кутовим розташуванням, порівняно з об'єктами, які видно повністю або мають переріз лише однією межею кадру, природньо дещо підвищиться якість розпізнавання. Однак забезпечення таких ексклюзивних вхідних датасетів на практиці є досить непростим. При тому приріст якості виявиться непропорційно малим і все ще недостатнім.

Рисунок 3.26 у нижній лівій чверті демонструє типову проблему у задачах сегментації – об'єднання контурів двох і більше близько розташованих об'єктів в один. Або об'єктів що мають візуальний перетин контурів.

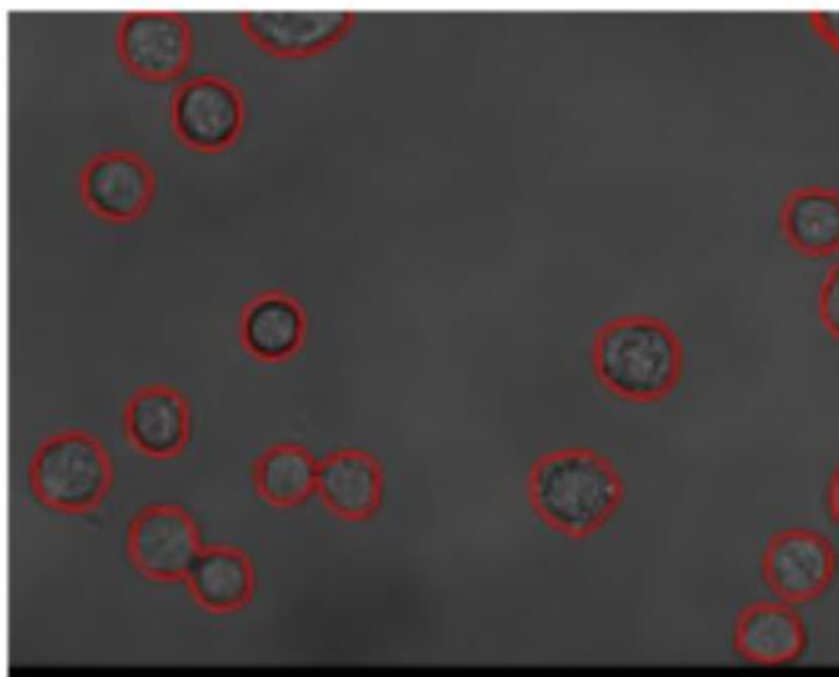


Рисунок 3.26 – Об'єднання контурів суміжних об'єктів

На рисунках 3.27 та 3.28 присутні іще два типи проблем. У правому верхньому куті обох зображень присутня клітина нетипової структури. Вона візуально має відмінні риси від основної маси клітин у навчальному датасеті. Відмінні пропорції та ознаки і порівняно висока контрастність

контура у порівнянні з центром об'єкта. Нестандартні параметри стали завадою у коректній сегментації об'єкта. Замість якісної сегментації цілого вийшла випадкова сегментація його фрагментів, що значно варіюється від кадру до кадру. На рисунку 3.28 взагалі виникла колізія між двома потенційно проблемними об'єктами (нетиповою клітиною і клітиною з частковим кутовим розміщенням в кадрі). Результат виявився абсолютно невалідним. Остання проблема сегментації, яка виникла у результаті роботи нейронної мережі – виокремлення шуму в об'єкти. На кадрі відносно рівномірне тло і шум майже відсутній. Тож глобальних проблем це не створює та не заважає процесу сегментації. Проте через відсутність повноцінного набору інформації про об'єкти в кутках та на межах кадру, це місце не лише для потенційних проблем виокремлення коректної області на зображенні, що відповідає площі шуканого об'єкта, а й потенційно вразлива частина кадру до хибних спрацювань на шум чи неоднорідне забарвлення тла.

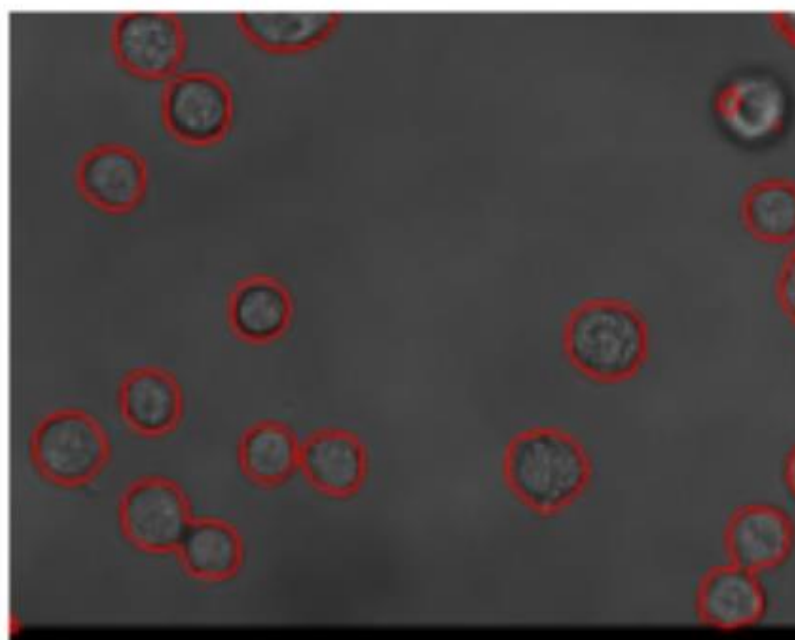


Рисунок 3.27 – Проблеми сегментації

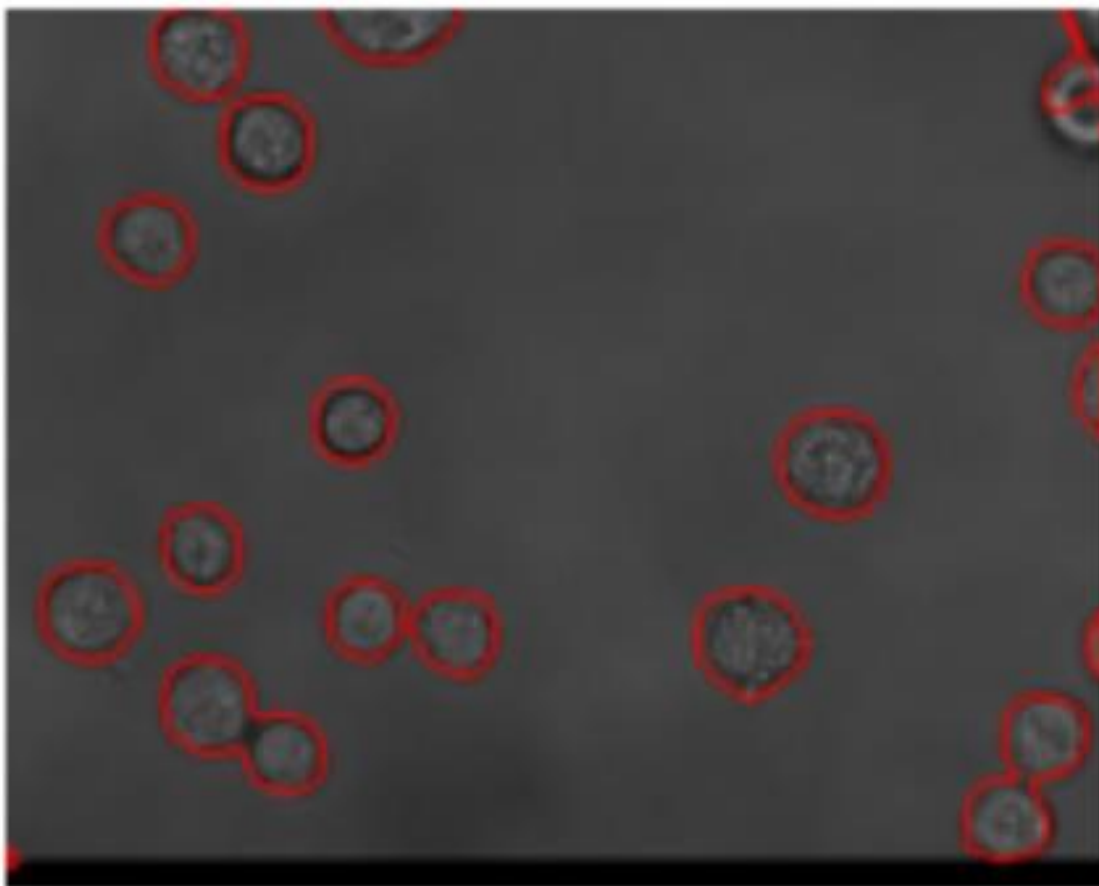


Рисунок 3.28 – Проблеми сегментації

Вирішити окреслені проблеми можна за допомогою алгоритмів, описаних у наступному підрозділі.

### 3.17 Алгоритми обробки зображень

Методи додаткової алгоритмічної обробки в сегментації зображень використовуються для поліпшення якості результатів, усунення артефактів та зменшення помилок, які можуть виникати при сегментації нейронною мережею.

Отже, підсумовуючи проблеми, наявні після сегментації нейромережею, їх можна виділити чотири:

- некоректна сегментація об'єктів у кутку зображення;

- злиття контурів двох об'єктів, які знаходяться близько один до одного;
- фальшиві спрацювання на фоновий шум;
- некоректна детекція об'єктів з розмитими межами.

Для вирішення вищезазначених проблем можуть використовуватись різні алгоритми додаткової обробки. Це методи, які базуються на кластеризації, морфологічній обробці, зменшенні кількості відтінків та інші методи з комп'ютерного зору.

### 3.17.1 Морфологічна обробка

Один із таких способів – морфологічна обробка. Морфологічні операції можуть бути використані для очищення сегментованих зображень від шуму та відокремлення злитих об'єктів. Цей метод може мати сенс у використанні як і перед обробкою нейромережею, так і після неї.

Морфологічна обробка зображень є потужним інструментом для аналізу і обробки двійкових (бінарних) та градаційних зображень. Вона заснована на застосуванні морфологічних операторів, які використовують структурний елемент для обробки зображення. Основними операціями морфологічної обробки є ерозія, дилатація, відкриття (opening) та закриття (closing).

Ерозія видаляє пікселі з меж об'єктів, зменшуючи їх розмір. Вона корисна для видалення дрібних шумових артефактів та відокремлення об'єктів, які знаходяться близько один до одного. Якщо структурний елемент є квадратом  $3 \times 3$ , ерозія замінює кожен піксель в зображенні мінімумом значень пікселів у його околі.

Дилатація – зворотній процес. Вона додає пікселі до меж об'єктів, збільшуючи їх розмір. Цей процес допомагає закрити невеликі прогалини в об'єктах та об'єднати близькі об'єкти. Дилатація замінює кожен піксель в зображенні максимумом значень пікселів у його околі.

Відкриття це комбінація ерозії, за якою слідує дилатація. Операція видаляє дрібні об'єкти і шум, зберігаючи при цьому розмір та форму основних об'єктів. Використовується для очищення зображення від дрібного шуму.

Закриття, відповідно, зворотній процес – комбінація дилатації, за якою слідує ерозія. Закриття закриває дрібні прогалини і заповнює отвори всередині об'єктів. Використовується для з'єднання близьких об'єктів і заповнення дрібних дірок.

Для задачі сегментації клітин на зображеннях з мікроскопа, морфологічна обробка має кілька ключових переваг:

- видалення шуму: морфологічні операції допомагають ефективно видалити дрібний шум, який може вплинути на точність сегментації. Це особливо важливо для мікроскопічних зображень, де присутні артефакти або невеликі шумові точки;

- розділення злитих об'єктів: використання ерозії і дилатації дозволяє відокремити об'єкти, які злилися, і зменшити ризик їх об'єднання в один об'єкт;

- покращення контурів об'єктів: відкриття і закриття можуть покращити форму і контури об'єктів, роблячи їх більш чіткими і легшими для подальшого аналізу.

### 3.17.2 Алгоритм вододілу

Алгоритм вододілу (watershed algorithm) є потужним методом для сегментації зображень, особливо ефективним при відокремленні об'єктів, які знаходяться близько один до одного або зливаються. Назва методу походить від аналогії з гідрологічним вододілом, де вода, що стікає з різних вершин, зливається в різні басейни. Цей принцип застосовується до зображень для розділення злитих областей на окремі об'єкти.

Основні етапи алгоритму вододілу – це попередня обробка (застосування фільтра Гауса для зменшення шуму та порашення сегментації), підсилення контурів об'єкта, маркування (визначення регіонів, де знаходяться об'єкти), виконання самого алгоритму і оновлення маркерів злиттям та розділенням регіонів на основі мінімальних значень градієнту або відстані. Алгоритм вододілу може включати в себе морфологічну обробку, описану раніше.

Алгоритм вододілу є дуже ефективним для розділення злитих об'єктів, особливо коли об'єкти мають чітко виражені контури. Використання маркерів дозволяє контролювати процес сегментації і уникати надмірного сегментування.

Для задачі сегментації клітин на зображеннях з мікроскопа, де часто присутні злиті клітини, алгоритм вододілу є дуже доречним. В поєднанні з морфологічною обробкою, він дозволяє ефективно розділяти та ідентифікувати окремі клітини.

### 3.17.3 Кластеризація методом $k$ -середніх

Кластеризація методом  $k$ -середніх є популярним алгоритмом машинного навчання для групування схожих об'єктів у кластери на основі їх характеристик. В контексті сегментації зображень, цей метод можна використовувати для групування пікселів на основі їх інтенсивності або інших ознак.

Його особливістю є необхідність заздалегідь визначити кількість кластерів, на які буде розбито дані. Це важливий крок, який може вимагати деякої апробації або використання методів для вибору оптимального  $k$ . Тобто для роботи даного алгоритму уже повинна бути чітко визначена кількість об'єктів, до яких будуть групуватися пікселі. Він може бути використаний як алгоритм постобробки для уникнення ситуацій коли об'єкт розпізнано лише частково.

Після автоматичного визначення кількості кластерів потрібно здійснити випадковий вибір  $k$  початкових центроїдів з набору даних (зображення). Кожен об'єкт в наборі даних призначається до найближчого центроїду на основі обраної метрики відстані, зазвичай евклідової. Далі відбувається оновлення центроїдів. Обчислення нових центроїдів як середнє значення об'єктів, призначених до кожного кластера. Призначення кластерів та оновлення центроїдів відбувається циклічно доки центроїди не перестають змінюватися. Або до певного порога ітерацій.

Алгоритм добре працює лише для сферичних кластерів з рівною дисперсією, що робить його ефективним при роботі з клітинами. Він має лінійну складність, що робить його швидким. Алгоритм може працювати некоректно при невірно визначеній кількості кластерів або наявності шуму на зображеннях. Але якщо ці проблеми вирішено, він значно підсилює якість сегментації.

#### 3.17.4 Зменшення кількості відтінків

Зменшення кількості відтінків – досить давня задача у комп'ютерній обробці зображень. Вона має різні алгоритмічні шляхи рішення. В тому числі, зменшити кількість відтінків можна використовуючи метод  $k$ -середніх, використавши його для створення кластерів у колірній гамі зображення, групуючи пікселі за схожістю кольорів.

Ще один варіант – алгоритм октавної декомпозиції (Octree Quantization). Він полягає у використанні структури октавного дерева для ієрархічного зменшення кількості кольорів. Спершу будується октавне дерево, де кожен вузол представляє колір. Потім відбувається злиття вузлів, починаючи з найменш важливих, до досягнення бажаної кількості кольорів.

Також зменшити кількість відтінків можливо за допомогою алгоритму Median Cut. Його суть у розділенні кольорового простору на підмножини, які мінімізують колірний розкид. Спершу необхідно визначити межі

кожного кольорового каналу. Далі розрізати простір вздовж осей з найбільшим розкидом значень кольорів. І повторювати процес доки не буде досягнута необхідна кількість кольорів.

І останнім з розглянутих алгоритмів є NeuQuant. Він використовує нейронну мережу для адаптивного зменшення кількості кольорів. У процесі нейронна мережа вчиться кодувати кольори зображення, після чого пікселі перетворюються на найближчі кольори з обмеженої палітри, яка визначається мережею.

## ВИСНОВКИ

В ході виконання кваліфікаційної роботи були розроблені загальні алгоритми сегментації біомедичних зображень клітин на мікроскопічних знімках. Один із акцентів був зроблений на створенні варіативної загальної стратегії та набору досліджень щодо сегментації зображень, яка має рекомендаційний характер і може бути оперативно переформована у підходящий алгоритм, в залежності від цільового набору даних.

Також акцент був поставлений на можливості роботи, заощаджуючи апаратні ресурси, і маючи невеликий обсяг вхідних даних аби мати можливість швидко адаптувати алгоритм під потреби конкретного датасету і отримати швидкі результати з мінімальними часовими і ресурсними витратами. Основу рішення складала нейронна мережа, побудована за архітектурою U-net. Вона була обрана як найбільш підходяща для даної задачі у рамках аналізу архітектур. Для навчання нейронної був підготовлений датасет, який є хронологічною нарізкою кадрів з відео. Були сформульовані правила для формування навчального і тестового набору даних. Навчальні дані були розмічені. Далі була проведена попередня обробка даних, включаючи аугментацію. Була побудована нейронна мережа і навчена на підготовлених даних.

Побудована нейронна мережа досить ефективно може впоратися з задачею сегментації клітин. Однак в ході тестування її роботи були виявлені такі проблеми як: злиття контурів об'єктів, фальшиве спрацьовування на шум, некоректна детекція нестандартних об'єктів та тих що знаходяться на межі зображення. Для вирішення виявлених проблем сегментації були розроблені комбінації алгоритмічних рішень на основі морфологічної обробки, алгоритму вододілу, кластеризації методом k-середніх, а також групуванням кольорів за відтінками. Механізм обробки зображень був вдосконалений комбінованим рішенням нейромережі та алгоритмів сегментації.

**ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ**

1. Application of artificial neural networks for analysis of ice recrystallization process for cryopreservation / M. Tymkovych et al. *8th european medical and biological engineering conference*. Cham, 2020. P. 102–111.
2. Batenburg K. J., Sijbers J. Adaptive thresholding of tomograms by projection distance minimization. *Pattern recognition*. 2009. Vol. 42, no. 10. P. 2297–2305.
3. Better performance with the tf.data API. TensorFlow Core. *TensorFlow*. URL: [https://www.tensorflow.org/guide/data\\_performance](https://www.tensorflow.org/guide/data_performance) (date of access: 08.06.2024)
4. Biomedical image segmentation using enhanced U-NET / R. Hariharan et al. *Recent trends in computational intelligence and its application*. London, 2023. P. 446–453. URL: <https://www.taylorfrancis.com/chapters/edit/10.1201/9781003388913-59/biomedical-image-segmentation-using-enhanced-net-hariharan-mohan-krunal-randive-toonesan-sundar-abi> (date of access: 09.06.2024).
5. Bowyer K., Flynn P., Kasturi R. The 20th anniversary of the IEEE Transactions on pattern analysis and machine intelligence. *IEEE transactions on pattern analysis and machine intelligence*. 2000. Vol. 22, no. 1. P. 1–3.
6. Davies E. R. *Machine vision: theory, algorithms, practicalities*. 3rd ed. Amsterdam : Elsevier, 2005. 934 p.
7. Emotional expressions reconsidered: challenges to inferring emotion from human facial movements / L. F. Barrett et al. *Psychological science in the public interest*. 2019. Vol. 20, no. 1. P. 1–68.
8. Ethylene glycol improves cryopreservation of cell-seeded electrospun scaffolds in cryobags : thesis / O. H. Avrunin et al. 2020. URL: <https://openarchive.nure.ua/entities/publication/8d210812-ff65-4a59-866f-90a538d3e06b> (date of access: 09.06.2024).

9. Experimental observations of the topology of convolutional neural network activations / E. Purvine et al. *Proceedings of the AAAI conference on artificial intelligence*. 2023. Vol. 37, no. 8. P. 9470–9479.
10. Henikoff J., Shapiro L. G. Interesting patterns for model-based machine vision. *[1990] third international conference on computer vision*, Osaka, Japan. URL: <https://ieeexplore.ieee.org/document/139590/> (date of access: 09.06.2024).
11. Hennessy J. L., Patterson D. A. Computer organization and design MIPS edition: the hardware/software interface (the morgan kaufmann series in computer architecture and design). Morgan Kaufmann, 2013. 800 p.
12. Ice crystals microscopic images segmentation based on active contours / M. Tymkovich et al. *2019 IEEE 39th international conference on electronics and nanotechnology (ELNANO)*, Kyiv, Ukraine, 16–18 April 2019. 2019.
13. Image Segmentation with U-Net. *Analytics Vidhya*. URL: <https://www.analyticsvidhya.com/blog/2022/10/image-segmentation-with-u-net/> (date of access: 08.06.2024).
14. Jain A. K., Murty M. N., Flynn P. J. Data clustering. *ACM computing surveys*. 1999. Vol. 31, no. 3. P. 264–323.
15. Krizhevsky A., Sutskever I., Hinton G. E. ImageNet classification with deep convolutional neural networks. *Communications of the ACM*. 2017. Vol. 60, no. 6. P. 84–90.
16. LeCun Y. Deep learning & convolutional networks. *2015 IEEE hot chips 27 symposium (HCS)*, Cupertino, CA, USA, 22–25 August 2015. 2015.
17. Long J., Shelhamer E., Darrell T. Fully convolutional networks for semantic segmentation. *2015 IEEE conference on computer vision and pattern recognition (CVPR)*, Boston, MA, USA, 7–12 June 2015. 2015.
18. Mudadla S. How does mask generation in computer vision involve creating binary images to highlight specific... *Medium*. URL: <https://medium.com/@sujathamudadla1213/how-does-mask-generation-in->

[computer-vision-involve-creating-binary-images-to-highlight-specific-c8aee934cdf3#:~:text=In%20computer%20vision,%20masks%20are,masks%20is%20called%20mask%20generation](https://openarchive.nure.ua/entities/publication/a43bf844-f473-41d8-9b9e-38e104683185) (date of access: 08.06.2024).

19. Multiscale quantitative analysis of microscopic images of ice crystals : thesis / M. Y. Tymkovych et al. 2019. URL: <https://openarchive.nure.ua/entities/publication/a43bf844-f473-41d8-9b9e-38e104683185> (date of access: 09.06.2024).

20. Pham D. L., Xu C., Prince J. L. Current methods in medical image segmentation. *Annual review of biomedical engineering*. 2000. Vol. 2, no. 1. P. 315–337.

21. Possibilities of automated image processing at optical capillaroscopy / A. Kovalova et al. *Optical fibers and their applications 2020*, Bialowieza, Poland, 27–31 January 2020 / ed. by R. S. Romaniuk, J. Dorosz. 2020.

22. Ronneberger O., Fischer P., Brox T. U-Net: convolutional networks for biomedical image segmentation. *Lecture notes in computer science*. Cham, 2015. P. 234–241.

23. Sahu B. The evolution of deeplab for semantic segmentation. *Medium*. URL: <https://towardsdatascience.com/the-evolution-of-deeplab-for-semantic-segmentation-95082b025571> (date of access: 08.06.2024).

24. Shi J., Chen T., Cai W. Adaptive algorithm for edge detection in medical image processing. *IS&T/SPIE's symposium on electronic imaging: science & technology*, San Jose, CA / ed. by T. Wilson, C. J. Cogswell. 1995.

25. Srivastava S., Sharma N., Singh S. K. Image analysis and understanding techniques for breast cancer detection from digital mammograms. *Research developments in computer vision and image processing*. 2014. P. 123–148.

26. Tf.data: Build TensorFlow input pipelines | TensorFlow Core. *TensorFlow*. URL: <https://www.tensorflow.org/guide/data> (date of access: 08.06.2024).

27. Tomashenko N., Khokhlov Y., Estève Y. On the use of gaussian mixture model framework to improve speaker adaptation of deep neural network acoustic models. *Interspeech 2016*.

28. Vincent L., Soille P. Watersheds in digital spaces: an efficient algorithm based on immersion simulations. *IEEE transactions on pattern analysis and machine intelligence*. 1991. Vol. 13, no. 6. P. 583–598.

29. What is a batch in TensorFlow?. *Stack Overflow*. URL: <https://stackoverflow.com/questions/41175401/what-is-a-batch-in-tensorflow> (date of access: 08.06.2024).

30. Zunair H., Ben Hamza A. Sharp U-Net: depthwise convolutional network for biomedical image segmentation. *Computers in biology and medicine*. 2021. Vol. 136. P. 104699.