

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМ. ІГОРЯ СІКОРСЬКОГО”
Факультет електроніки
Кафедра електронної інженерії

До захисту допущено

Завідувач кафедри

В. І. Тимофєєв

“ ” 20__р.

Дипломна робота

освітнього рівня «магістр»

за спеціальністю 153 мікро- та наносистемна техніка

на тему Визначення стадії структурно-функціонального дозрівання
кардіоміоцитів методами машинного навчання

Виконав студент 2 курсу, групи ДМ-11мн

Бондарев В'ячеслав Ростиславович

(прізвище, ім'я, по батькові)

Керівник к.т.н., доц., доц. каф. ЕІ, Іванько К.О.

(посада, вчене звання, науковий ступінь, прізвище та ініціали)

Рецензент к.т.н., доц. каф. МЕ, Обухова Т.Ю.

(посада, вчене звання, науковий ступінь, прізвище та ініціали)



(підпис)



(підпис)



(підпис)

Засвідчую, що у цій дипломній роботі немає
запозичень з праць інших авторів без
відповідних посилань.

Студент _____

(підпис)



Київ - 2023 року

**Національний технічний університет України
“Київський політехнічний інститут імені Ігоря Сікорського”**

Факультет електроніки
Кафедра електронної інженерії
Освітній рівень «магістр»
за спеціальністю 153 мікро- та наносистемна техніка

ЗАТВЕРДЖУЮ

Завідувач кафедри

В. І. Тимофєєв

“ ” 20__ р.

**З А В Д А Н Н Я
НА ДИПЛОМНУ РОБОТУ СТУДЕНТУ**

Бондарев В'ячеслав Ростиславович

(прізвище, ім'я, по батькові)

1. Тема роботи Визначення стадії структурно-функціонального дозрівання кардіоміоцитів методами машинного навчання

керівник роботи Іванько Катерина Олегівна, к.т.н., доц. каф. ЕІ,
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від “ 23 ” 03 2023 року № 1310-с

2. Строк подання студентом роботи _____

3. Вихідні дані до роботи _____

4. Зміст дипломної роботи (перелік питань, які потрібно розробити)

1. Кардіоміоцити. Отримання, використання та оцінка зрілості кардіоміоцитів отриманих з плюрипотентних стовбурових клітин людини.

2 Аналіз методів цифрової обробки зображень.

3 Аналіз методів машинного навчання для вирішення задачі визначення стадії структурно-функціонального дозрівання кардіоміоцитів.

4 Розробка програмного забезпечення для розпізнавання стадій дозрівання кардіоміоцитів з використанням нейронних мереж.

5 Задача класифікації для ідентифікації стадій дозрівання кардіоміоцитів

5. Перелік графічного (ілюстративного) матеріалу (із зазначенням обов'язкових креслень, плакатів, презентацій тощо)

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання _____

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів виконання дипломної роботи	Строк виконання етапів роботи	Примітка
1	Літературний огляд матеріалу стосовно дослідження кардіоміоцитів, отриманих з плюрипотентних стовбурових клітин людини.	01.03.2023	
2	Літературний огляд матеріалу стосовно систем "серце-на-чипі"	08.03.2023	
3	Опанування методів розпізнавання стадій дозрівання кардіоміоцитів за допомогою машинного навчання	01.04.2023	
4	Розробка програмного забезпечення для розпізнавання стадій дозрівання кардіоміоцитів з використанням нейронних мереж	14.04.2023	
5	Проведення експериментальних досліджень запропонованих методів	01.05.2023	

Студент

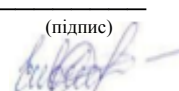


(підпис)

В.Р. Бондарев

(прізвище та ініціали)

Керівник роботи



(підпис)

К.О. Іванько

(прізвище та ініціали)

РЕФЕРАТ

Магістерська дисертація: 76 с., 5 ч., 36 рис., 34 джерел.

КАРДІОМІОЦИТ, СТОВБУРОВІ КЛІТИНИ, ЗОБРАЖЕННЯ, ОБРОБКА
ЗОБРАЖЕНЬ, МАШИННЕ НАВЧАННЯ, ЗАДАЧА МАШИННОГО НАВЧАННЯ,
КЛАСИФІКАЦІЯ, ТОЧНІСТЬ КЛАСИФІКАЦІЇ, НЕЙРОННА МЕРЕЖА,
ЗГОРТКОВА НЕЙРОННА МЕРЕЖА

Дипломна робота присвячена алгоритмам машинного навчання і використанню нейронних мереж зокрема для вирішення задачі визначення стадії структурно-функціонального дозрівання кардіоміоцитів.

Метою роботи є розробити алгоритм, що буде включати обробку зображень, підготовку моделі нейронної мережі і навчання класифікаторів для вирішення задачі визначення стадії структурно-функціонального дозрівання кардіоміоцитів.

У першому розділі було проведено огляд існуючих підходів до вирощування кардіоміоцитів з плюрипотентних стовбурових клітин людини, розглянуто стадії їх дозрівання, а також використання для лікування хвороб серця.

У другому розділі проведено аналіз існуючих методів обробки зображень і алгоритмів цих методів.

У третьому розділі проведено аналіз задачі класифікації машинного навчання і структури нейронних мереж і згорткових нейронних мереж зокрема.

У четвертому розділі розроблено алгоритм цифрової обробки зображень і підготовлена архітектура згорткової нейронної мережі для подальшого навчання.

У п'ятому розділі проведено навчання моделі і аналіз результатів.

У висновках на підставі проведеного аналізу даних і розробленого алгоритму, наведено головні результати роботи та сформульовано рекомендації щодо їх практичного застосування.

ABSTRACT

Master's dissertation: 76 pages, 5 chapters, 36 figures, 34 sources.

CARDIOMYOCYTE, STEM CELLS, IMAGING, IMAGE PROCESSING, MACHINE LEARNING, MACHINE LEARNING PROBLEM, CLASSIFICATION, CLASSIFICATION ACCURACY, NEURAL NETWORK, CONVOLUTIONAL NEURAL NETWORK

Thesis is devoted to machine learning algorithms and the use of neural networks to solve the problem of classifying the stage of structural and functional maturation of cardiomyocytes.

The purpose of the work is to develop an algorithm that includes image processing, preparation of a neural network model, and training of classifiers to solve the problem of classifying the stage of structural and functional maturation of cardiomyocytes.

In the first chapter, an overview of existing approaches to the cultivation of cardiomyocytes from human pluripotent stem cells was carried out, the stages of their maturation, as well as their use for the treatment of heart diseases, were considered.

In the second chapter, an analysis of existing image processing methods and algorithms was carried out.

In the third chapter, an analysis of the classification problem of machine learning was carried out and the structure of neural networks and convolutional neural networks were considered.

In the fourth chapter, the digital image processing algorithm was developed and the convolutional neural network architecture was prepared for further training.

In the fifth chapter, the model was trained and the results were analyzed.

In the conclusions based on the conducted data analysis and the developed algorithm, the main results of the work are given and recommendations for their practical application are formulated.

ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ.....	7
ВСТУП.....	8
1 КАРДІОМІОЦИТИ. ОТРИМАННЯ, ВИКОРИСТАННЯ ТА ОЦІНКА КАРДІОМІОЦИТІВ ОТРИМАНИХ З ПЛЮРИПОТЕНТНИХ СТОВБУРОВИХ КЛІТИН ЛЮДИНИ.....	10
1.1 Отримання кардіоміоцитів з плюрипотентних стовбурових клітин і стадії їх розвитку.....	10
1.2 Методи оцінки кардіоміоцитів отриманих з плюрипотентних стовбурових клітин.....	13
1.3 Методи використання кардіоміоцитів отриманих з плюрипотентних стовбурових клітин.....	18
1.4 Висновки до розділу 1.....	20
2 АНАЛІЗ МЕТОДІВ ЦИФРОВОЇ ОБРОБКИ ЗОБРАЖЕНЬ.....	22
2.1 Аналіз методів сегментації зображень.....	22
2.2 Аналіз методів аугментації даних.....	26
2.3 Аналіз методів обробки яскравості зображень.....	30
2.4 Висновки до розділу 2.....	33
3 АНАЛІЗ МЕТОДІВ МАШИННОГО НАВЧАННЯ ДЛЯ ВИРІШЕННЯ ЗАДАЧІ ВИЗНАЧЕННЯ СТАДІЇ СТРУКТУРНО-ФУНКЦІОНАЛЬНОГО ДОЗРІВАННЯ КАРДІОМІОЦИТІВ.....	34
3.1 Навчання з вчителем. Задачі класифікації і регресії.....	34
3.2 Нейронні мережі. Згорткова нейронна мережа.....	39
3.3 Висновки до розділу 3.....	43
4 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ РОЗПІЗНАВАННЯ СТАДІЙ ДОЗРІВАННЯ КАРДІОМІОЦИТІВ З ВИКОРИСТАННЯМ НЕЙРОННИХ МЕРЕЖ.....	44
4.1 Висновки до розділу 4.....	56

5 ЗАДАЧА КЛАСИФІКАЦІЇ ДЛЯ ІДЕНТИФІКАЦІЇ СТАДІЙ ДОЗРІВАННЯ КАРДІОМІОЦИТІВ.....	57
5.1 Висновки до розділу 5.....	65
ВИСНОВКИ.....	66
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ.....	68
Додаток А.....	72
Додаток Б.....	74

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

ВГ – вирівнювання гистограми

ВГЗЯ – вирівнювання гистограми із збереженням яскравості

ГЗМ – генеративні змагальні мережі

ГНМ – глибинні нейронні мережі

ЕСК – ембріональні стовбурові клітини

ЗНМ – згорткова нейронна мережа

ІМ – інфаркт міокарда

КМ – кардіоміоцит

МАК – модель активного контуру

МН – машинне навчання

ПОІ – полігональна області інтересу

ПСК – плюрипотентні стовбурові клітини

СН – серцева недостатність

ШНМ – штучна нейронна мережа

ВСТУП

Серцева недостатність (СН) — це багатогранний та небезпечний для життя синдром, який характеризується значною захворюваністю та смертністю, низькою функціональною здатністю та якістю життя, а також високими витратами [1]. СН страждає понад 64 мільйони людей у всьому світі [1]. Тому спроби зменшити його соціальний та економічний тягар стали головним глобальним пріоритетом охорони здоров'я.

Клітинна регенеративна терапія стала одним із найбільш перспективних варіантів лікування пацієнтів із серцевою недостатністю. На сьогоднішній день найбільш значущим джерелом кардіоміоцитів для ремускуляризації серця є кардіоміоцити вирощені з плюрипотентних стовбурових клітин (ПСК-КМ). Ремускуляризація передбачає приживлення трансплантованих клітин у міокарді реципієнта, їх проліферацію та диференціювання з подальшим електромеханічним зв'язуванням з резидентними кардіоміоцитами та збільшенням скорочувальної функції серця [2].

Проте наразі, при вирощуванні кардіоміоцитів та дослідженнях стадій їх розвитку існує декілька суттєвих обмежень. Для статистичного аналізу всієї популяції необхідно призначити стадію розвитку кожному спостережуваному зображенню кардіоміоцита відповідно до складного набору ознак об'єкта та правил класифікації. Такий підхід непрактичний і схильний до суб'єктивних помилок. Враховуючи кількість даних, необхідних для статистичного аналізу, автоматична класифікація зображень є більш доречною.

Тому в даній роботі будуть розглянуті основні методи цифрової обробки зображень і алгоритми машинного навчання, включаючи нейронні мережі в задачі визначення стадії структурно-функціонального дозрівання кардіоміоцитів. Буде розроблено алгоритм на основі найбільш ефективних методів, що буде вирішувати задачі підготовки зображень до аналізу, збільшення масиву вхідних даних, підготовки архітектури нейронної мережі і вирішення задачі класифікації для

визначення стадії структурно-функціонального дозрівання кардіоміоцитів з урахуванням основних вимог до такого алгоритму, а саме: швидкість навчання, точність навчання і стійкість до перенавчання.

1 КАРДІОМІОЦИТИ. ОТРИМАННЯ, ВИКОРИСТАННЯ ТА ОЦІНКА КАРДІОМІОЦИТІВ ОТРИМАНИХ З ПЛЮРИПОТЕНТНИХ СТОVBУРОВИХ КЛІТИН ЛЮДИНИ

Клітини серцевого м'яза, які називаються кардіоміоцитами, є скоротливими міоцитами серцевого м'яза. Клітини серцевого м'яза є скорочувальними клітинами, які забезпечують роботу серця. Кардіоміоцити містять Т-канальці, мішечки клітинної мембрани, які проходять від поверхні клітини до внутрішньої частини клітини, що допомагає підвищити ефективність скорочення. Більшість цих клітин містять лише одне ядро (деякі можуть мати два центральних ядра), на відміну від клітин скелетних м'язів, які містять багато ядер. Кожна клітина містить міофібрили, спеціалізовані білкові скоротливі волокна актину та міозину, які ковзають одне вздовж одного. Вони організовані в саркомери, основні скоротливі одиниці м'язових клітин. Правильна організація міофібрил у саркомери надає клітинам серцевого м'яза смугастого вигляду під мікроскопом, подібно до скелетних м'язів. Якщо дивитися під мікроскоп, клітини серцевого м'яза мають приблизно прямокутну форму, розміром 100–150 мкм на 30–40 мкм [3].

1.1 Отримання кардіоміоцитів з плюрипотентних стовбурових клітин і стадії їх розвитку

У багатоклітинних організмах стовбурові клітини є недиференційованими або частково диференційованими клітинами, які можуть диференціюватися в різні типи клітин і проліферувати необмежено для отримання більшої кількості однакових стовбурових клітин. Вони є найбільш раннім типом клітин у клітинній лінії [4].

Кардіоміоцити з ембріональних стовбурових клітин людини (ЕСК) і пов'язані з ними індуковані людиною плюрипотентні стовбурові клітини (ПСК) мають величезні перспективи як потенційні джерела клітин для відновлення серця. Обидва ці плюрипотентні типи стовбурових клітин мають безсумнівний кардіоміогенний потенціал, що вигідно відрізняє їх від багатьох типів дорослих стовбурових клітин, для яких здатність диференціюватись у значну кількість остаточно кардіоміоцитів є суперечливою. Крім того, як недиференційовані ЕСК, так і ПСК виявляють потужну проліферативну активність, що робить ці типи клітин особливо привабливими для застосувань, які вимагають великої кількості клітин (наприклад, заміна $\sim 1 \times 10^9$ кардіоміоцитів господаря втрачених при типовому інфаркті міокарда). Кардіоміоцити, отримані з ЕСК і ПСК, мають однозначний серцевий фенотип, виявляючи спонтанну скорочувальну активність, серцеві механізми зчеплення збудження-скорочення та експресію очікуваних саркомерних білків, іонних каналів і факторів транскрипції [5].

Процес вирощування кардіоміоцитів з плюрипотентних стовбурових клітин можна поділити на декілька етапів. Починаючи від зростання до стадії повної конфлюентності (повне покриття поверхні культуральної посудини) в недиференційованому стані в середовищі "MEF-CM" із використанням "bFGF", культури переходять в умови диференціації в середовищі "RPMI-B27" і послідовно обробляються двома факторами росту, активіном "А" (день 0) і "ВМР-4" (день 1). Через 5 днів після індукції активіном А культури, що диференціюються, вирощують у середовищі "RPMI-B27" за відсутності екзогенних культур. Спонтанна активність биття починається приблизно на 9-11 день після індукції, і культури можуть бути зібрані після 14 дня [5].

Перед обробкою активіном "А" (день 0) (рис 1.1) культури повинні сформувати компактний, 100% конфлюентний моношар, який складається з клітин із високим співвідношенням ядер і цитоплазми, але більшою нерегулярністю у формі клітин. До 5-го дня клітини в моношарі стають фазово яскравими, більш вільними та мають округлу морфологію. На цій стадії слід очікувати деяку загибель клітин, але моношар все ще повинен займати >90% площі поверхні. На

14-й день культури є набагато гетерогеннішими та містять широкі ділянки зі спонтанною активністю биття. Після диспергування та повторного посіву при низькій щільності (день 25) отримані диференційовані культури складаються з окремих і невеликих кластерів скорочувальних кардіоміоцитів з веретеноподібною або трикутною морфологією [5].

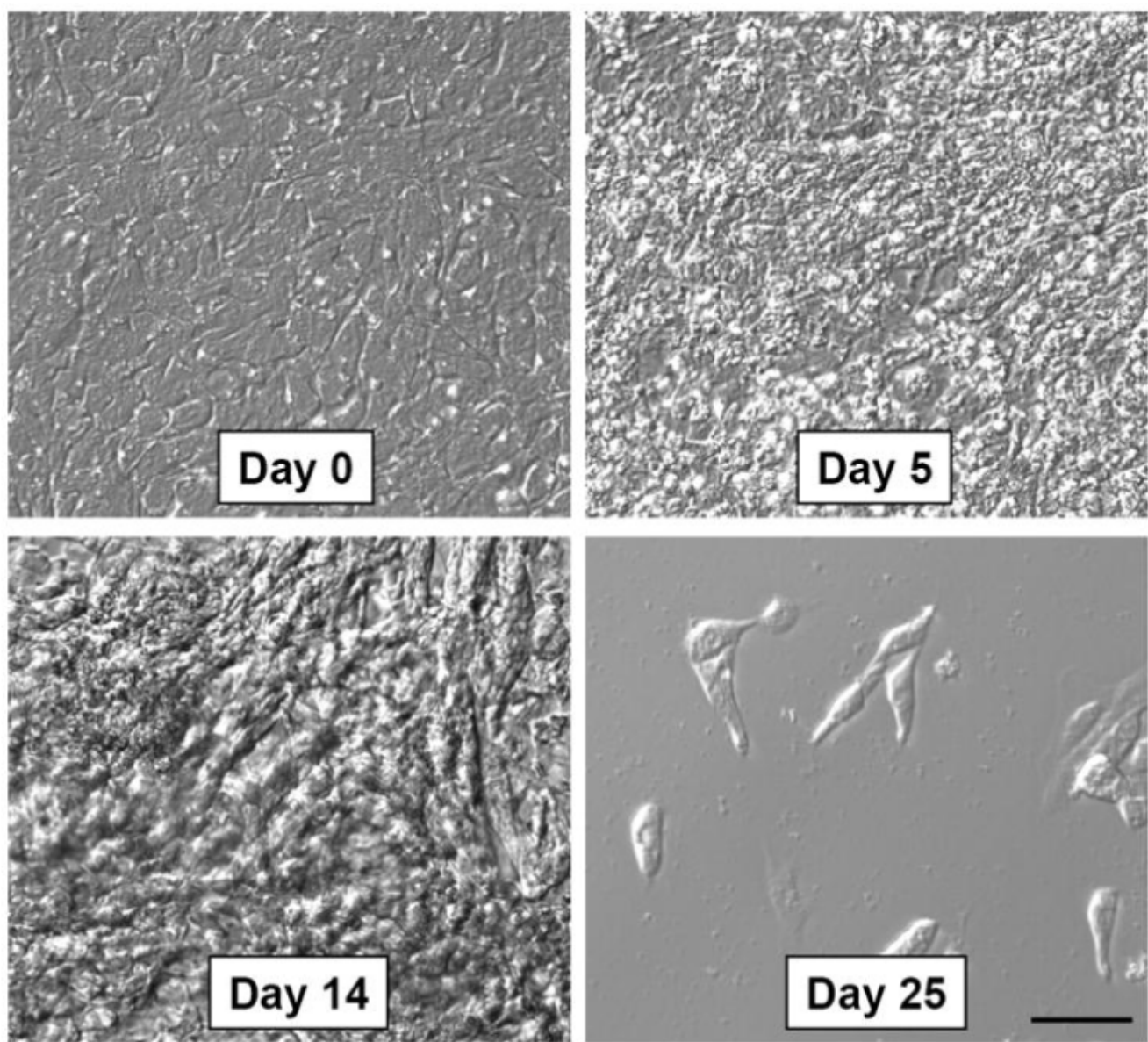


Рисунок 1.1 – Стадії вирощування кардіоміоцитів з з плюрипотентних стовбурових клітин [5]

Крім того стадії дозрівання кардіоміоцитів можна поділити на п'ять етапів за морфологічною ознакою. Кардіоміоцити без мембранних включень можна

віднести до першої стадії. Друга стадія характеризується наявністю коротких (~ 2 мкм), рівномірно розташованих перпендикулярних включень мембрани та/або окремих довгих (>10 мкм) каналців. На третій стадії присутня мережа довших трубочок (>10 мкм), як поперечних, так і поздовжніх. Складну систему поперечних каналців із явною наявністю довших (>2 мкм) поздовжніх каналців і частих ділянок, позбавлених каналців, можна класифікувати як четверту стадію. П'ята стадія, присутня у дорослих кардіоміоцитів, відрізняється складною системою поперечних каналців, з розрідженими поздовжніми структурами, які зазвичай охоплюють не більше 1-2 саркомерів (від ~ 2 мкм до ~ 4 мкм), що заповнюють всю область клітини, крім ядер [6].



Рисунок 1.2 – Стадії вирощування кардіоміоцитів за морфологічною ознакою [6]

1.2 Методи оцінки кардіоміоцитів отриманих з плюрипотентних стовбурових клітин

Існує декілька підходів до оцінки кардіоміоцитів з плюрипотентних стовбурових клітин. Один з них вимірювання скоротливої можливості клітин. Для цього тестові клітини покривають желатином і переносять в перфузійну камеру, встановлену на столику інвертованого мікроскопа. Клітини утримують методом перфузії культуральним середовищем при $35,5 \pm 1^\circ\text{C}$. Це процес за якого свіже середовище поповнює поживні речовини, тоді як клітинні відходи та середовище,

збіднене поживними речовинами, видаляються. Скорочення клітин оцінюють за допомогою систем виявлення країв на основі відео для безперервного вимірювання руху вибраної точки моношару ПСК [7].

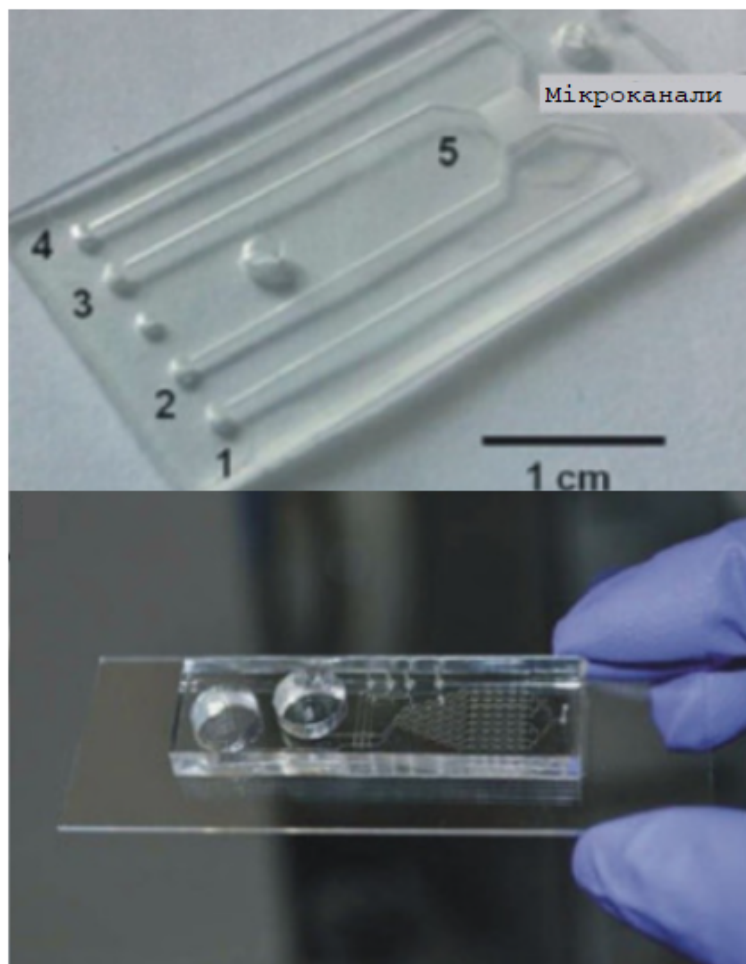


Рисунок 1.3 – Мікрофлюїдні платформи для дослідження скоротливої можливості клітин [7]

Одночасно із вимірюванням скоротливої можливості клітин вимірюють перехідні процеси Ca^{2+} . Метод базується на використанні флуоресцентного барвника, який можна використовувати для моніторингу змін у внутрішньоклітинних перехідних процесах Ca^{2+} , пов'язаних зі скороченням кардіоміоцитів. Спершу клітини інкубують в барвнику протягом 40 хв, після чого дослідження проводиться при 37°C і реєструються при збудженні 485 нм і випромінюванні 530 нм. Отримується ряд параметрів перехідних процесів.

Вимірюються кількість піків, середня амплітуда піку, середня ширина піку, середній час наростання піку, середній час спаду піку та середній інтервал між піками (рисунк 1.4) [7].

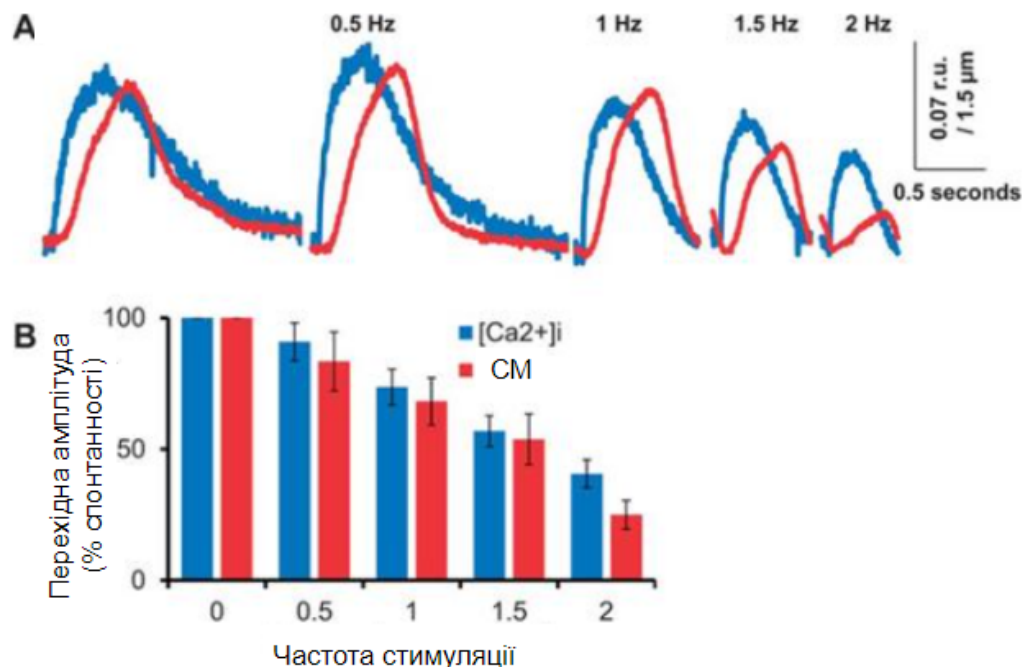


Рисунок 1.4 – Одночасне вимірювання скоротливої можливості та перехідних процесів Ca^{2+} у ПСК-КМ. Рисунок А: репрезентативний слід, що показує запис перехідних процесів Ca^{2+} (синій) і скоротливої можливості (червоний) з однієї області моношару ПСК-КМ. Сліди показують перехідні процеси в результаті спонтанного биття, а також стимуляції поля на зростаючих частотах. Рисунок В: підсумок співвідношення частота-відповідь для Ca^{2+} та скоротливої можливості [7]

Також використовуються методи оцінки стадій дозрівання кардіоміоцитів (рисунк 1.5), що є важливим параметром при вирощуванні кардіоміоцитів і ключовим фактором для правильного використання інших методів оцінки. Для цього сарколему ізольованих кардіоміоцитів фарбують за допомогою ліпофільних мембранних зондів. Зображення отримують за допомогою конфокального мікроскопа, оснащеного водяним імерсійним об'єктивом з роздільною здатністю від 58 нм до 116 нм на піксель [6].

Конфокальна мікроскопія є технікою отримання зображень для підвищення оптичної роздільної здатності та контрастності мікрофотографії за допомогою використання просторового отвору для блокування розфокусованого світла при формуванні зображення [8]. Захоплення кількох двовимірних зображень на різних глибинах в зразку дає змогу реконструювати тривимірні структури (оптичне розрізання) всередині об'єкта. Основний принцип конфокальної мікроскопії полягає в тому, що оптика освітлення та детектування фокусується на одній і тій самій дифракційно обмеженій зоні, яка переміщується над зразком для створення повного зображення на детекторі. Хоча все поле зору освітлюється під час конфокального зображення, все, що знаходиться за межами фокальної площини, мало впливає на зображення, зменшуючи ефект туманності, яка спостерігається під час стандартної світлової мікроскопії з товстими зразками з високим рівнем розсіювання [8].

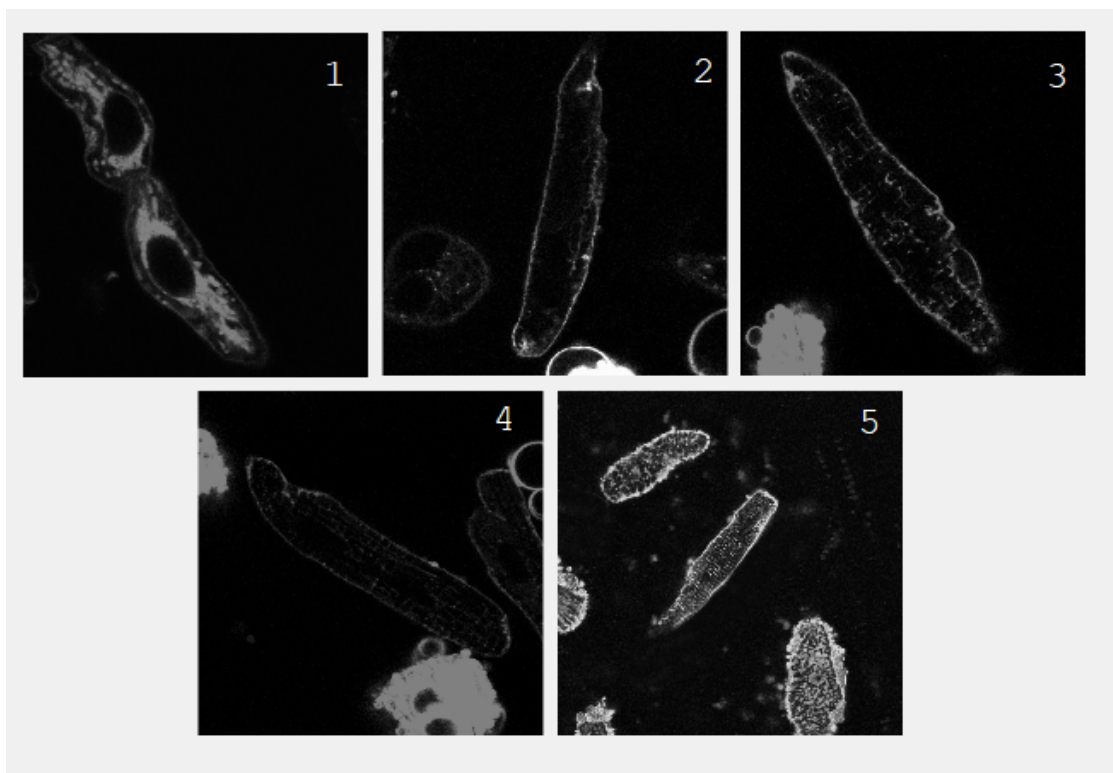


Рисунок 1.5 – Конфокальні зображення кардіоміоцитів відповідно до стадії розвитку

Світло від лазерного джерела пропускається через колімуючу оптику до змінного дихроматичного дзеркала і відбивається до лінзи об'єктива, яка фокусує промінь на певній зоні зразка. Скануючі дзеркала переміщують промінь збудження по зразку від зони до зони, щоб створити повноцінне зображення. Випромінювальна флуоресценція проходить назад через лінзу об'єктива, дихроматичне дзеркало і виявляється детектором. Точковий отвір, розміщений у площині сполученого зображення з фокальною точкою зразка, служить для відхилення розфокусованого світла, яке не вловлюється детектором [8].

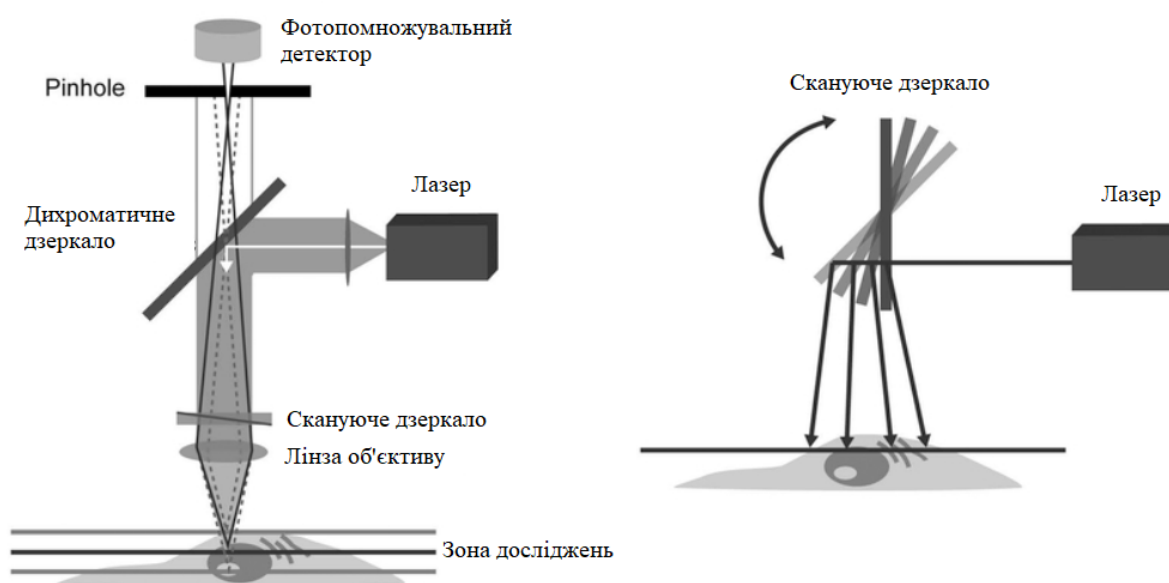


Рисунок 1.5 – Будова конфокального мікроскопа [8]

Проте наразі, намагаючись кількісно визначити складність Т-каналців у серцевих міоцитах, необхідно ідентифікувати здорові міоцити у зразку та записати високоякісні зображення. Для статистичного аналізу всієї популяції необхідно призначити стадію розвитку кожному спостережуваному зображенню кардіоміоцита відповідно до складного набору ознак об'єкта та правил класифікації. Такий підхід непрактичний і схильний до суб'єктивних помилок. Враховуючи кількість даних, необхідних для статистичного аналізу, автоматична класифікація зображень є більш доречною. Оскільки кардіоміоцити є об'єктами

високого рівня складності та демонструють значну морфологічну мінливість, автоматична класифікація ускладнена відсутністю реалізованих методів [6].

1.3 Методи використання кардіоміоцитів отриманих з плюрипотентних стовбурових клітин

Клітинна регенеративна терапія стала одним із найбільш перспективних варіантів лікування пацієнтів із серцевою недостатністю. Дослідження, проведені на тваринних моделях, показують, що ПСК-КМ можуть частково ремускуляризувати інфаркт серця та покращити скорочувальну функцію [9].

На експериментальних тваринних моделях гострого інфаркту міокарда (ІМ) повідомлялося про сприятливий вплив на функцію серця після трансплантації ПСК у місце пошкодження. Однак було підкреслено ризик утворення тератоми (пухлини, що формується із ембріональних клітин і яка складається з різних типів тканин, які є похідними різних зародкових листків, присутність яких не характерна для тих органів або анатомічних областей організму, в яких розвивається пухлина, в даному випадку для серцевого м'яза), і в одному дослідженні імплантовані клітини не інтегрувалися з тканиною хазяїна, а також були задокументовані ознаки реактивного фіброзу [9].

Caspi [9] та його колеги продемонстрували ехокардіографією виживання трансплантатів ПСК-КМ в інфарктних серцях щурів протягом 8 тижнів після трансплантації, а також покращення розмірів і функції лівого шлуночка в реципієнтів кардіоміоцитів порівняно з контрольною групою через 4 і 8 тижнів після трансплантації. Вони також показали, що трансплантація недиференційованих ПСК призвела до утворення тератом і що попередня диференціація ПСК до КМ значно зменшує утворення тератом після трансплантації в інфарктних серцях щурів [9].

Щоб підвищити вірогідність виживання трансплантата, Лафлам та його колеги [9] розробили коктейль, що сприяє виживанню, що містить позаклітинний матрикс - матригель і речовини, що впливають на виживання клітин і називаються факторами виживання (до факторів виживання можна віднести морфогени, протеоглікани, цитокіни, гормони, нейромедіатори тощо), які значно покращили вірогідність виживання ПСК-КМ в інфарктному серці щурів. Трансплантати вижили та були присутні навіть через 4 тижні після трансплантації та ремускуляризували близько 11 відсотків зони інфаркту, при цьому серця реципієнтів демонстрували збережену загальну скорочувальну функцію, регіональне потовщення стінки та розміру лівого шлуночка порівняно з контролем [9].

Однак результати різних клінічних випробувань показали, що клітинна терапія, яка передбачає пряме введення КМ, призводить до значної втрати клітин через фізичне напруження або первинну гіпоксію. Кількісне визначення динаміки продемонструвало, що більшість клітин трансплантата гине протягом кількох днів у моделі щурів. Щоб подолати цю перешкоду, трансплантація ділянок міокарда за допомогою тканинної інженерії розглядається як клітинна терапія другого покоління. Доставка клітин у тканиноподібних структурах, які зберігають клітинну адгезію, може підвищити ефективність доставки клітин і зменшити загибель клітин [9].

Найпопулярнішим підходом тканинної інженерії є посів клітин у тривимірні біорозкладні каркаси. Є надія, що каркас з часом деградує; клітини можуть проліферувати і заповнювати простір, який раніше займав полімер, що призводить до утворення тканини. Використання каркасів для тканинної інженерії сприяє регенерації міокарда. Таким чином, трансплантація ділянок міокарда за допомогою тканинної інженерії може збільшити вірогідність виживання клітин трансплантата та покращити функцію пошкодженого серця більше, ніж ізольована ін'єкція клітин, що викликає значний інтерес як клітинна терапія другого покоління для тяжкої серцевої недостатності [9].

Міокардіальні пластирі на основі каркаса мають деякі недоліки, такі як недостатня міграція клітин у попередньо виготовлені каркаси та запальні реакції через біодеградацію полімеру. Щоб подолати ці проблеми, Шімізу та інші [9] створили підхід на основі клітинних листів без каркасу. Клітинні листи включають чисті клітини без будь-яких альтернативних матеріалів і трансплантуються на пошкоджене серце безпосередньо як міокардіальний пластр. Клітини культивуються на інтелектуальній поверхні, виготовленій із термочутливого полімеру. Ця поверхня має унікальні властивості, що дозволяють спонтанно відокремлювати клітини у вигляді безперервних листів з усіма суміжними міжклітинними з'єднаннями, контактами і адгезивними білками без змін. Ця система має високу перевагу на відміну від системи культивування на основі каркасу, яка потребує його деградації, у свою чергу, впливаючи на міжклітинні з'єднання та адгезивні білки. Клітинні листи трансплантуються безпосередньо на поверхню серця без швів, і клітини можуть бути ефективно доставлені у вигляді тонких, але щільних клітинних конструкцій великої площі без втрати клітин. Трансплантовані таким чином клітини ефективно впливають на функцію міокарда після інфаркту, сприяючи не тільки неоваскуляризації, але й індукуючи значний рівень кардіоміогенезу [9].

1.4 Висновки до розділу 1

У ході виконання першого розділу були розглянуті різні методи одержання кардіоміоцитів з плюрипотентних стовбурових клітин, включаючи ембріональні стовбурові клітини та індуковані плюрипотентні стовбурові клітини. Була показана перевага використання диференційованих клітин отриманих за допомогою ПСК. Були розглянуті стадії дозрівання кардіоміоцитів і виділено п'ять стадій дозрівання.

Були проаналізовані методи, які використовуються для оцінки кардіоміоцитів, включаючи функціональні та морфологічні. До функціональних оцінок можна віднести оцінки скоротливої можливості клітин і вимірювання перехідних процесів Ca^{2+} . Не менш важливою є оцінка морфології клітин, а саме визначення стадії дозрівання, що є важливим параметром для проведення інших оцінок.

Були розглянуті можливі застосування кардіоміоцитів, отриманих з плюрипотентних стовбурових клітин, для дослідження та лікування кардіоваскулярних захворювань, таких як інфаркт міокарда та серцева недостатність і показано їх відносну ефективність.

2 АНАЛІЗ МЕТОДІВ ЦИФРОВОЇ ОБРОБКИ ЗОБРАЖЕНЬ

Використання підходу машинного навчання до задачі визначення стадії структурно-функціонального дозрівання кардіоміоцитів потребує особливої уваги до вхідного масиву зображень. Іншими словами цифрова обробка масиву вхідних зображень необхідна для якості подальшого машинного навчання. При обробці зображень для вирішення задачі визначення стадії структурно-функціонального дозрівання кардіоміоцитів постає декілька проблем. Перша з них: наявність малої кількості вхідних даних, що робить неможливим побудову якісної моделі машинного навчання і призводить до необхідності використання методів аугментації. Друга: наявність артефактів, що також негативно впливає на подальше навчання. Для вирішення цієї проблеми будуть розглянуті методи сегментації. Наостанок, покращення контрастності зображень допоможе у виділенні основних елементів зображення. Далі будуть розглянуті запропоновані методи.

2.1 Аналіз методів сегментації зображень

У зображенні лише деякі частини містять в собі корисну інформацію, яка відображається як область інтересу. Зазвичай вони відповідають зображенню в конкретній області. Їх потрібно виділити та відокремити, щоб ідентифікувати та проаналізувати об'єкт або область інтересу [10].

Сегментація зображень — це фундаментальна проблема в області комп'ютерного зору і являє собою процес поділу зображень на значущі підмножини, які відповідають поверхням або об'єктам [11]. Результатом сегментації зображення є набір сегментів, які разом покривають усе зображення,

або набір контурів, виділених із зображення. Кожен із пікселів у регіоні схожий за певною характеристикою або обчисленою властивістю [12].

Модель активного контуру (МАК) є одною з найуспішніших моделей сегментації зображення. Основна ідея МАК полягає в тому, щоб будувати криву для виділення потрібного об'єкта на основі методу мінімізації енергії. Перевага МАК для сегментації зображення полягає в тому, що вона поділяє зображення на підобласті із замкнутими та плавними межами. Одним з недоліків МАК є сегментація зображення із топологією, що зазнає переривань, або злиття з іншими топологіями. Вирішенням цього недоліку було використання диференціального рівняння в часткових похідних. Це узагальнює рівняння Ейлера–Лагранжа та розвиває алгоритм, який неявно представлений як множина функцій наближених до нуля. Видатною характерною рисою методів набору рівнів є те, що контури можуть розділятися або об'єднуються, коли змінюється топологія функції набору рівнів, тому методи набору рівнів можуть виявляти більше ніж одну межу одночасно. Ця гнучкість і зручність забезпечують засоби для автоматичної сегментації за допомогою попередньо визначеного початкового контуру [11].

Без втрати загальності, більшість МАК, заснованих на структурі набору рівнів можна класифікувати на два типи: крайові та регіональні [11].

Сегментація на основі країв базується на краях, знайдених на зображенні за допомогою різних операторів виявлення країв. Ці краї позначають місця розривів зображення в рівнях сірого, кольорі, текстурі тощо (при переході від однієї області до іншої, рівень сірого може змінюватися) [11].

Грані зазвичай асоціюються з «величиною» та «напрямом» градієнтів. Деякі детектори країв дають можливість обчислити і напрямки, і величину. Існують різні детектори країв, наприклад оператор Собеля, оператор краю Кірша, оператор краю Превітта, оператор краю Роберта тощо [10].

Оператор Собеля має перевагу в тому, що забезпечує ефект диференціації та згладжування. Оскільки похідні підсилюють шум, ефект згладжування є особливо привабливою особливістю оператора Собеля. Оператор Собеля є одним з

найкращих і простих операторів країв. Він використовує маску 3×3 . Похідні на основі масок оператора Собеля мають наступний вигляд [10]:

$$G_x(m, n) = (Z_7 + 2Z_8 + Z_9) - (Z_1 + 2Z_2 + Z_3)$$

$$G_y(m, n) = (Z_3 + 2Z_6 + Z_9) - (Z_1 + 2Z_4 + Z_7)$$

де Z — це рівні сірого пікселів, які перекриваються масками в будь-якому положенні центрального пікселя на зображенні. Потім використовується обчислення градієнта в центрі масок для знаходження величини та напрямку. Щоб отримати наступні значення, маски переміщуються до розташування наступного пікселя і алгоритм повторюється. Таким чином, після завершення алгоритму для всіх можливих позицій пікселя, результатом є градієнтне зображення такого самого розміру, як і вихідне зображення. Позиція, або розташування пікселя (m, n) оголошується як розташування краю, якщо $G(m, n)$ перевищує деякий поріг T . Карта краю надає необхідні дані для відстеження меж об'єкта на зображенні [10].

Незважаючи на простоту методу, що надає перевагу у швидкості обчислення, він має кілька суттєвих недоліків. Ці моделі не тільки дуже чутливі до шуму, але також мають проблеми із виявленням слабких меж. Крім того, результат сегментації сильно залежить від початкового розташування контуру [11].

Моделі на основі регіонів використовують статистичну інформацію зображення для побудови границь, що надає багато переваг порівняно з методами на основі країв. По-перше, вони не залежать від градієнта зображення і можуть задовільно сегментувати об'єкти зі слабкими межами. По-друге, використовуючи інформацію про глобальний регіон, тобто про регіон в цілому, а не про стан конкретних пікселів, моделі на основі регіонів, як правило, стійкі до шуму [11].

Однією з найпопулярніших моделей на основі регіону є модель Чана–Везе, яка базується на спрощеному функціоналі Мамфорда–Шаха для сегментації [11].

Нехай f позначає дане зображення в градаціях сірого в області Ω , яку необхідно сегментувати. Метод апроксимує зображення f кусково-гладкою функцією u як розв'язок задачі мінімізації [11, 13].

$$\operatorname{argmin}_{u,C} \mu\text{Length}(C) + \lambda \int_{\Omega} (f(x) - u(x))^2 dx + \int_{\Omega/C} |\nabla u(x)|^2 dx,$$

де C — крива набору країв, а u може мати розриви. Перший доданок забезпечує регулярність C , другий доданок заохочує u бути близьким до f , а третій доданок гарантує, що u є диференційованою на $\Omega \setminus C$. Наближення Мамфорда–Шаха пропонує вибрати цей набір ребер C як межу сегментації. Рішення Мамфорда–Шаха було обчислено за допомогою скороченого методу Амбросіо–Тортореллі [11, 13].

Незважаючи на те, що це природний спосіб постановки задачі сегментації, алгоритми для розв'язання загальної моделі Мамфорда–Шаха, як правило, є відносно складними і дорогими з точки зору обчислень. Для спрощення було також запропоновано кусково-неперервне формулювання

$$\operatorname{argmin}_{u,C} \mu\text{Length}(C) + \lambda \int_{\Omega} (f(x) - u(x))^2 dx,$$

де u має бути постійним на кожній зв'язній компоненті $\Omega \setminus C$. У цьому випадку C обов'язково є межею замкнутої множини, тобто C складається із замкнутих кривих. Задача мінімізації є неопуклою. Існування розв'язку кусково-сталого моделі було доведено Мамфордом і Шахом [11, 13].

Порівняно з кусково-неперервною моделлю Мамфорда–Шаха, ключовими відмінностями моделі Чана–Везе є додатковий показник, що контролює розмір замкнутої області, і подальше спрощення, яке допускає, що u може мати лише два значення

$$u(x) = c_1, \quad x \in C, \quad \text{або}$$

$$u(x) = c_2, \quad x \notin C$$

де C — межа замкнутої множини, а c_1, c_2 — значення u відповідно всередині та зовні C . Метод Чан–Везе полягає в тому, щоб знайти серед усіх u цієї форми те, яке найкраще наближає f [11, 13].

$$\begin{aligned} & \operatorname{argmin}_{c_1, c_2, C} \mu \text{Length}(C) + \nu \text{Area}(\text{inside}(C)) + \\ & + \lambda_1 \int_{\text{inside}(C)} (f(x) - c_1)^2 dx + \lambda_2 \int_{\text{outside}(C)} (f(x) - c_2)^2 dx, \end{aligned}$$

де перший член контролює регулярність, обмежуючи довжину, другий член обмежує закриту область C , щоб контролювати її розмір, третій і четвертий члени обмежують розбіжність між кусково-неперервною моделлю u та вхідним зображенням f . Знайшовши локальний мінімізатор цієї проблеми, можна отримати сегментацію як найкращу двофазну кусково-постійну апроксимацію u зображення f [11, 13].

2.2 Аналіз методів аугментації даних

Аугментація даних — це техніка штучного збільшення навчального набору даних шляхом створення модифікованих копій набору з використанням існуючих даних. Це включає внесення незначних змін до набору даних (аугментовані дані) або використання глибокого навчання для створення нових точок даних (синтетичні дані) [14].

Аугментовані дані створюються з вихідних даних із деякими незначними змінами. Можуть бути використані геометричні та кольорові просторові трансформації (перевертання, зміна розміру, кадрування, зміна яскравості, контрасту, додавання шумових складових) для збільшення розміру та різноманітності навчального набору [14].

Синтетичні дані генеруються штучно без використання вихідного набору даних. При цьому часто використовуються глибинні нейронні мережі (ГНМ) і генеративні змагальні мережі (ГЗМ) для створення синтетичних даних [14].

Можна виділити декілька технік аугментації зображення [14]:

1. Геометричні перетворення: довільна геометрична трансформація, обрізання, обертання, розтягування та масштабування зображень.
2. Трансформація колірного простору: випадкова зміна каналів кольору RGB, контрастності та яскравості.
3. Ядрові фільтри: довільно змінюють різкість або розмиття зображення.
4. Довільне стирання: видалити частину початкового зображення.
5. Змішування зображень: об'єднання кількох зображень, або частин зображень.

Проте, важливо зазначити, що в контексті вирішення задачі визначення стадії структурно-функціонального дозрівання кардіоміоцитів, необхідно обирати такі методи, що запобігають альтерації поперечного Т-канальця, що є важливим параметром для правильної класифікації. Також важливою характеристикою є контрастність зображення для виділення структури поперечного Т-канальця. Тому, враховуючи вище сказане, далі будуть розглянуті методи обертання зображень.

Обертання зображення виконується за допомогою інтерполяції методом найближчого сусіда, що дозволяє вирішити невідповідність розмірності вхідного та вихідного зображення. Цей метод є не тільки швидким, але і не вводить ніяких штучних даних в остаточний результат [15].

В алгоритмі інтерполяції найближчого сусіда яскравість пікселя P в результуючому зображенні обчислюється як яскравість його найближчого сусіда, який вираховується за відстанню між P і його сусідніми точками A , B , C і D (рисунок 2.1) [15].

На рисунку 2.1 припустимо, що (i, j) , $(i, j+1)$, $(i+1, j)$ і $(i+1, j+1)$ - це 4-сусідні точки, а їх яскравості складають відповідно $f(i, j)$, $f(i, j+1)$, $f(i+1, j)$ і $f(i+1, j+1)$. Відстань між (u, v) і (i, j) , $(i, j+1)$, $(i+1, j)$ і $(i+1, j+1)$ було обчислено і

яскравість пікселя $f(u, v)$ було встановлено як яскравість пікселя, найближчого до (u, v) [15].

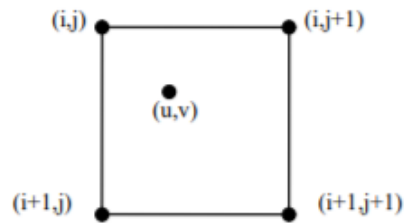


Рисунок 2.1 – Схема алгоритму інтерполяції найближчого сусіда [15]

При чому координати точки (x_1, y_1) при повороті на кут θ навколо (x_0, y_0) стають (x_2, y_2) , як показано в такому рівнянні [16]:

$$\begin{aligned} x_2 &= \cos(\theta) * (x_1 - x_0) + \sin(\theta) * (y_1 - y_0) \\ y_2 &= -\sin(\theta) * (x_1 - x_0) + \cos(\theta) * (y_1 - y_0) \end{aligned}$$

Інверсія кольорів – це фотоефект, який перетворює всі кольори на протилежні відтінки на колірній палітрі. Інверсію кольорів можна віднести до трансформацій колірного простору і в загальному визначити як

$$I_{out} = I_{max} - I_{in},$$

де I_{max} – максимальне значення яскравості пікселя в заданому типі даних.

Інверсію часто використовують разом із гамма-корекцією для зміни кольорів зображення [17].

Гамма-корекція – це нелінійна операція, що використовується для перетворення яскравості пікселів зображення. Вона відноситься до степеневих перетворень. При цьому яскравість пікселів вихідного та вхідного зображень пов'язані за наступним законом [17]:

$$r = c(s + \varepsilon)^\gamma,$$

де c , ε , γ – деякі додатні константи, s – початкове зображення, r – результуюче зображення [17].

Графіки залежностей при різних значеннях показані на рисунку 2.5. Видно, що при показниках менших за одиницю, криві таких степеневих залежностей перетворюють вузький діапазон малих значень яскравостей пікселів початкового зображення у широкий діапазон яскравостей пікселів результуючого зображення. Якщо значення степені є більшим за одиницю, то відбувається протилежний ефект: вузький діапазон високих яскравостей перетворюється у широкий діапазон яскравостей пікселів вихідного зображення [17].

Перетворення при значенні $\gamma < 1$ іноді називають кодування гамми і дана операція відповідає гамма-стисненню, і, навпаки, значення $\gamma > 1$ називається а декодування гамми і дана операція є гамма-розширенням [17].

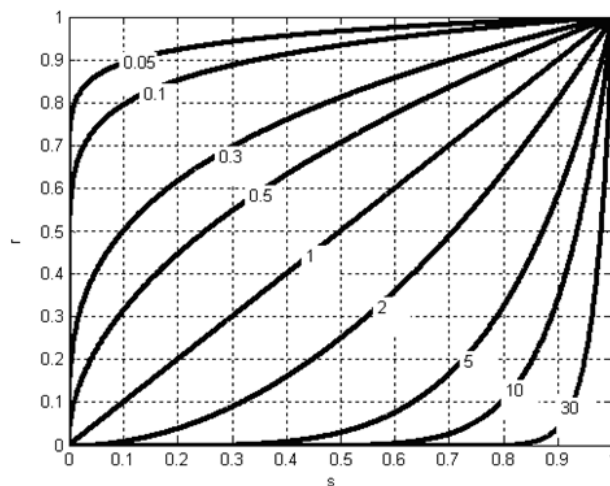


Рисунок 2.2 – Залежності між яскравостями пікселів вхідного та вихідного зображення при гамма-корекції для різних значень γ [17]

2.3 Аналіз методів обробки яскравості зображень

Контрастність, що представляє собою різницю між максимальним і мінімальним рівнями напівтонів в певній області зображення, у багатьох випадках визначає яскравість зображення. Шляхом підвищення контрастності можна покращити виділення окремих структур зображення або зменшити викривлення, що можуть виникати при отриманні зображення [18]. Покращення контрастності окремих частин кардіоміоцитів є ключовим фактором для підвищення точності вирішення задачі визначення стадії структурно-функціонального дозрівання кардіоміоцитів.

Гістограма яскравості зображення – це функція, яка показує відносну частоту виявлення пікселя певного значення яскравості в зображенні [18]. Обробка гістограми – це процес перетворення зображення шляхом зміни його гістограми. Найчастіше обробка гістограми являє собою нормалізацію, за допомогою якої гістограма зображення приводиться до якомога більш плоского вигляду. Розглянемо основні методи обробки гістограм яскравості зображень [18].

Вирівнювання гістограми (ВГ) – це метод, який генерує статистичну карту перетворення, яка змінює гістограму зображення і балансує усі значення пікселів так, щоб їх розподіл був якомога ближче до гістограми, заданої форми. ВГ дозволяє визначити та обробити області з більш низьким локальним контрастом для отримання більш високого контрасту. ВГ автоматично визначає функцію перетворення, яка оброблює початкове зображення таким чином, щоб отримати найбільш однорідну гістограму [18].

Нехай $X = \{X(i, j)\}$ позначає зображення, складене із L дискретних рівнів сірого і позначається як $X = \{X_0, X_1, \dots, X_{L-1}\}$. Для цього зображення X функція щільності ймовірності $p(X_k)$ визначається як: [18]

$$p(X_k) = \frac{n^k}{n},$$

де $K = 0, 1, \dots, L-1$, n^k представляє ту кількість разів, скільки рівень X_k присутній у вхідному зображенні X , n - загальна кількість компонент у вхідному зображенні, $p(X_k)$ асоціюється з гістограмою вхідного зображення, яка представляє кількість пікселів, які мають певну інтенсивність X_k [18].

На основі функції щільності ймовірності, результуюча функція щільності визначається як

$$c(x) = \sum_{j=0}^k p(X_k),$$

де $X_k = x$ для $k = 0, 1, \dots, L-1$, і $(X_{L-1}) = 1$ за визначенням, ВГ - це схема, яка відображає вхідне зображення в повний динамічний діапазон (X_0, X_{L-1}) з використанням результуючої функції щільності як функції перетворення [18].

Функція перетворення $f(x)$ на основі результуючої функції щільності визначається як: [18]

$$f(x) = X_0 + (X_{L-1} - X_0)c(x).$$

Тоді вихідне зображення ВГ, $Y = \{Y(i, j)\}$ можна виразити як [18]

$$Y = f(x) = f\left(\frac{X(i, j)}{\forall X(i, j) \in X}\right).$$

Головними недоліками методу ВГ є те, що він не використовує середнє значення яскравості зображення при перетворенні [18], через що яскравість і, особливо, контрастність результуючого зображення можуть погіршитися. Крім того це може призвести до появи артефактів насиченості через розтягнення одного сірого рівня монотонної області у всьому діапазоні рівнів сірого.

Вирівнювання гістограми зі збереженням яскравості (ВГЗЯ) додатково розкладає початкове зображення на два на основі середнього значення відтінків

сірого вхідного зображення. Одне з зображень представляє набір яскравостей пікселів, що менші або рівні середньому значенню яскравості, тоді як інше – набір відтінків більших середнього. ВГЗЯ вирівнює допоміжні зображення, використовуючи їх гістограми, при цьому враховуючи деякі обмеження. Перше зображення вирівнюється в діапазоні від мінімального рівня сірого до середнього значення яскравості пікселя початкового зображення, а друге зображення – від середнього до максимального рівнів сірого. Результуючі вирівняні зображення обмежені середнім значенням об'єднуються, що має ефект збереження середньої яскравості [18].

У цьому обчислювальному блоці гістограма розбивається як $(n_0, n_1, \dots, n_m)$ і $(n_m + 1, \dots, n_{L-1})$ відповідно і вихідне зображення $Y(i, j)$ може бути розраховане як

$$Y(i, j) = \{X_0 + (X_m - X_0)c_L(x) X_{m+1} + (X_{L-1} - X_{m+1})c_U(x)\},$$

що засновано на $f_L(X_L) \cup f_U(X_U)$ [18].

Метод ВГЗЯ має недолік – необхідність обчислення гістограми та середнього значення на кожному етапі обробки зображення, що вимагає значно більших обчислювальних ресурсів, ніж звичайне ВГ.

Метод адаптивного вирівнювання гістограм з обмеженою контрастністю є ефективним способом поліпшення локального контрасту зображення. Цей метод використовує кілька гістограм, кожна з яких відповідає окремій області зображення, для перерозподілу яскравості зображення. Однак, при такому підході може виникати шум, який з'являється на кордонах розділених зон [18].

Контрастно обмежене адаптивне вирівнювання є узагальненням методу адаптивного вирівнювання гістограм. В цьому методі сусідні частини сітки об'єднуються, використовуючи методи інтерполяції, щоб усунути штучно наведені кордони. Контраст може бути обмежений на однорідних ділянках, щоб уникнути посилення будь-якого шуму, що може бути присутнім на зображенні [18].

2.4 Висновки до розділу 2

В даному розділі були проаналізовані наступні способи перетворень зображень: сегментація зображень, аугментація вхідного масиву зображень і обробка яскравості зображень. Цифрова обробка масиву вхідних зображень необхідна для підвищення якості подальшого навчання моделі.

Сегментація зображень дозволяє виділити корисну інформацію та знайти зони інтересу на зображенні. Були розглянуті методи крайової та регіональної сегментації. Для подальшої роботи був обраний метод Чан–Везе, що належить до регіональних методів сегментації. Він не залежить від градієнта зображення і задовільно сегментує об'єкти зі слабкими межами і є стійким до шуму за рахунок використання інформації про глобальний регіон.

Були розглянуті методи аугментації вхідного масиву зображень. Був обраний метод обертання зображень, що запобігає альтерації поперечного Т-канальця, що є важливим параметром для правильної класифікації, а також не погіршує контрастність зображення, що є важливим параметром для виділення структури поперечного Т-канальця кардіоміоцитів.

Для обробки яскравості зображень було розглянуто метод контрастно обмеженого адаптивного вирівнювання. Метод контрастно обмеженого адаптивного вирівнювання має можливість покращити локальний контраст зображення і зменшити шуми в результуючому зображенні і тому є найбільш оптимальним.

3 АНАЛІЗ МЕТОДІВ МАШИННОГО НАВЧАННЯ ДЛЯ ВИРІШЕННЯ ЗАДАЧІ ВИЗНАЧЕННЯ СТАДІЇ СТРУКТУРНО-ФУНКЦІОНАЛЬНОГО ДОЗРІВАННЯ КАРДІОМІОЦИТІВ

Машинне навчання (МН) – це підгалузь штучного інтелекту, яка застосовує статистичні прийоми для надання комп'ютерним алгоритмам здатності «навчатися» (тобто, поступово покращувати продуктивність у певній задачі) на основі аналізу і використання даних, без того, щоб бути програмованими явно. Алгоритми машинного навчання створюють модель на основі вибіркового даних, відомих як «навчальні дані», щоб робити прогнози або рішення, не будучи явно запрограмованими на це [20].

Алгоритми машинного навчання можуть бути класифіковані за типом доступних навчальних даних, методами, за допомогою яких навчальні дані були зібрані, та способами оцінки алгоритму навчання [20]. Крім того, існують класичні задачі машинного навчання, які базуються на певних сценаріях машинного навчання. Проаналізуємо основні сценарії та задачі машинного навчання.

3.1 Навчання з вчителем. Задачі класифікації і регресії

Контрольоване навчання (сценарій навчання із вчителем) є одним з основних сценаріїв машинного навчання, що передбачає знаходження функції, що зіставляє вхідні дані з вихідними, використовуючи приклади пар вхід-вихід. Кожен приклад включає вхідний об'єкт (зазвичай вектор) та бажане вихідне значення (контрольний сигнал). Алгоритм навчання з вчителем аналізує ці приклади та знаходить функцію, яку можна використовувати для аналізу нових

даних [20][21]. Цей сценарій машинного навчання пов'язаний з задачами класифікації та регресії та є дуже поширеним [21].

В загальному вигляді для контрольованого навчання задається набір із N навчальних прикладів виду $\{(x_1, y_1), \dots, (x_N, y_N)\}$ такі, що x_i – це набір ознак i -того прикладу, а y_i – це контрольований сигнал (мітка класу, до якого відноситься приклад i). Навчальний алгоритм знаходить функцію $g : X \rightarrow Y$, де X – це вхідний простір, а Y – вихідний. Функція g є елементом деякого простору можливих функцій G , зазвичай званого простором гіпотез. Іноді зручно представити g за допомогою оціночної функції $f : X \times Y \rightarrow R$ такої, що g визначається як функція, що повертає значення y із найбільшою точністю [21]:

$$g(x) = \arg \arg f(x, y).$$

Нехай F позначає простір оціночних функцій [21]. Хоча G і F можуть бути будь-яким простором функцій, багато алгоритмів навчання є ймовірнісними моделями, де g приймає форму моделі умовної ймовірності $g(x) = P(y|x)$, або f приймає форму моделі спільної ймовірнісної $f(x, y) = P(x, y)$ [21].

Існують два основні підходи до вибору функцій f або g , а саме мінімізація емпіричного ризику та мінімізація структурного ризику. Перший підхід шукає функцію, яка забезпечує найкращу відповідність навчальним даним, тоді як другий включає функцію штрафу, яка контролює баланс між точністю та складністю моделі. Особливість другого підходу полягає у тому, що він приділяє увагу як ефективності моделі на навчальних даних, так і її здатності до узагальнення на нові дані [21][22].

В обох випадках передбачається, що навчальна вибірка складається з вибірки незалежних і однаково розподілених пар (x_i, y_i) . Щоб виміряти, наскільки добре функція відповідає навчальним даним, визначається функція

втрат $L: \hat{Y} \times Y \rightarrow R$ ($R \geq 0$). Для тренувальної пари (x_i, y_i) втрати при передбаченні значення $\hat{y}$ складають $L(y_i, \hat{y})$ [21].

Ризик $R(g)$ функції g визначається як очікувана втрата функції g . Його можна оцінити за даними навчання [21]:

$$R(g) = \frac{1}{N} \sum_i (y_i, g(x_i)).$$

При мінімізації емпіричного ризику алгоритм навчання з учителем шукає функцію g , яка мінімізує $R(g)$. Отже, алгоритм може бути побудований шляхом застосування алгоритму оптимізації для знаходження g [21] [22].

Коли g є умовним розподілом ймовірностей $P(y|x)$, а функція втрат – це функція негативної логарифмічної правдоподібності: $L(y_i, \hat{y}) = -\log \log P(y|x)$, тоді мінімізація емпіричного ризику еквівалентна оцінці максимальної правдоподібності [21] [22].

Коли G містить багато функцій-кандидатів або навчальна вибірка недостатньо велика, мінімізація емпіричного ризику призводить до високої дисперсії і поганого узагальнення, або ж перенавчання. Перенавчання – це явище, коли побудована модель добре пояснює приклади з навчальної вибірки, але погано працює із прикладами, які не брали участь у навчанні [21]. При перенавчанні статистична модель описує випадкову похибку або шум, замість взаємозв'язку, що лежить в основі даних. Перенавчання виникає тоді, коли модель є занадто складною, такою, що має занадто багато параметрів відносно числа спостережень. Перенавчена модель має погану передбачувальну продуктивність, оскільки вона занадто сильно реагує на другорядні відхилення в тренувальних даних (рисунок 3.1) [21]. Перенавчання стається тоді, коли модель починає «запам'ятовувати» тренувальні дані, замість того, щоби «вчитися» узагальненню з тенденції [21].

Мінімізація структурних ризиків спрямована на запобігання перенавчання за рахунок включення в оптимізацію штрафу за регуляризацію. Штраф за регуляризацію можна розглядати як реалізацію принципу “KIS” (keep it simple), який віддає перевагу більш простим функціям в порівнянні з більш складними [20].

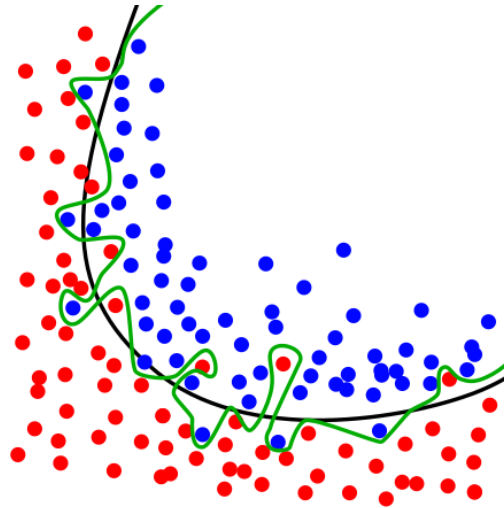


Рисунок 3.1 – Приклад перенавченої моделі машинного навчання. Зелена лінія представляє перенавчену модель, а чорна — регуляризовану. В той час як зелена лінія найкраще слідує тренувальним даним, вона занадто залежить від них, і, ймовірно, матиме вищий рівень похибки на нових небачених даних у порівнянні з чорною лінією

Для мінімізації застосовується набір штрафів, що відповідають різним визначенням складності. Наприклад, коли функція g є лінійною функцією виду

$$g(x) = \sum_{j=1}^d \beta_j x_j,$$

то штраф за регуляризацію складає $\sum_{j=1}^d \beta_j^2$, який являє собою квадрат евклідової норми вагових коефіцієнтів, також відомий як норма L_2 . Інші норми

включають норму L_1 : $\sum_{j=1}^d |\beta_j|$ і норму L_0 , яка являє собою кількість ненульових значень β_j [20]. Штраф буде позначатися як $C(g)$. Завдання оптимізації навчання з учителем полягає в тому, щоб знайти функцію g , яка мінімізує значення штрафу [22]:

$$J(g) = R(g) + \alpha C(g).$$

Значення параметра α впливає на баланс між зміщенням і дисперсією. Значення $\alpha = 0$ призводить до мінімізації емпіричного ризику з низьким рівнем зміщення та високим рівнем дисперсії, тоді як велике значення α веде до високого рівня зміщення та низького рівня дисперсії під час навчання [22].

Штраф за складність має Байєсівську інтерпретацію як негативну логарифмічну апіорну ймовірність функції g : $-\log \log P(g)$ і в цьому випадку $J(g)$ – це апостеріорна ймовірність функції g [22].

Класифікація є основною задачею машинного навчання, що використовує навчання із вчителем. Вона полягає у визначенні категорії, до якої належить певне спостереження, яке називається екземпляром, за допомогою аналізу прикладів з відомими категоріями, які називаються класами [22].

Регресійний аналіз – це статистичний метод, який використовується для вивчення зв'язку між залежною змінною ("змінна результату") та однією або більше незалежними змінними ("предиктори" або "характеристики"). Головна мета регресійного аналізу полягає у двох різних цілях. По-перше, цей метод часто використовується для прогнозування майбутніх значень залежної змінної. По-друге, у деяких випадках регресійний аналіз може допомогти встановити причинно-наслідкові зв'язки між незалежними та залежними змінними [20][21].

3.2 Нейронні мережі. Згорткова нейронна мережа

Нейронні мережі, також відомі як штучні нейронні мережі (ШНМ) є підмножиною машинного навчання та є основою алгоритмів глибокого навчання. ШНМ складаються з вузлових шарів, що містять вхідний рівень, один або більше прихованих шарів і вихідний рівень. Кожен вузол, або штучний нейрон, з'єднується з іншим і має відповідну вагу та поріг. Якщо вихід будь-якого окремого вузла перевищує вказане порогове значення, цей вузол активується, надсилаючи дані на наступний рівень мережі. В іншому випадку дані не передаються на наступний рівень мережі [23].

Кожен окремий вузол може бути представлений як модель лінійної регресії, що складається з вхідних даних, вагових коефіцієнтів, зміщення (або порогу) і вихідних даних. Із урахуванням того, що $f(x)$ може містити довільну кількість класів, функція g може бути представлена наступною softmax (узагальнення логістичної функції для багатовимірного випадку) функцією [23] [24]

$$softmax(z)_i = \frac{\exp(z_i)}{\sum_{l=1}^K \exp(z_l)},$$

де z_i представляє i -тий елемент входу в softmax, який відповідає класу i , K представляє кількість класів. Результатом є вектор, що містить ймовірності того, що x належить до кожного з класів, а також клас із найвищою ймовірністю [24].

У регресії вихід залишається аналогічним до $f(x)$, тому функція активації виводу є просто функцією ідентифікації [24].

ШНМ використовує різні функції втрат залежно від типу проблеми. Функцією втрат для класифікації є середня крос-ентропія, яку можна подати як

$$Loss(\hat{y}, y, W) = -\frac{1}{n} \sum_{i=0}^n (y_i \ln \hat{y}_i + (1 - y_i) \ln(1 - \hat{y}_i)) + \frac{\alpha}{2n} \|W\|_2^2,$$

де $\alpha \|W\|_2^2$ це термін регуляризації (він же штраф), який штрафує складні моделі і $\alpha > 0$ це невід'ємний параметр, який контролює величину штрафу [24]. Для регресії ШНМ використовує функцію середньоквадратичної помилки, яку можна записати як [24]

$$Loss(\hat{y}, y, W) = \frac{1}{2n} \sum_{i=0}^n \|\hat{y}_i - y_i\|_2^2 + \frac{\alpha}{2n} \|W\|_2^2$$

Починаючи з початкових випадкових ваг, ШНМ покращують функцію втрат шляхом багаторазового оновлення цих вагових коефіцієнтів. Після обчислення втрат зворотний перехід поширює їх із вихідного рівня на попередні шари, надаючи кожному ваговому параметру оновлене значення, призначене для зменшення втрат [24].

У градієнтному спуску градієнт $\nabla Loss_W$ втрат щодо ваг обчислюється та вираховується з W . Більш формально це виражається так:

$$W^{i+1} = W^i - \epsilon \nabla Loss_W^i,$$

де i є кроком ітерації, а ϵ — швидкість навчання зі значенням, більшим за 0. Алгоритм зупиняється, коли він досягає заданої максимальної кількості ітерацій; або коли втрати є нижчими певного невеликого числа [24].

Згорткові нейронні мережі (ЗНМ) відрізняються від інших нейронних мереж своєю високою продуктивністю в роботі з зображеннями, мовними сигналами або аудіосигналами. Вони мають три основні типи шарів, а саме: згортковий шар, шар об'єднання, рівень з повним з'єднанням. Згортковий шар є першим шаром згорткової мережі. У той час як за згортковими шарами можуть слідувати інші

згорткові шари або шари об'єднання, повністю зв'язаний рівень є останнім. З кожним шаром ЗНМ зростає у своїй складності, ідентифікуючи більші частини зображення. Попередні шари зосереджені на простих функціях, таких як кольори та краї. Коли дані зображення просуваються між шарами ЗНМ, вони починають розпізнавати більші елементи або форми об'єкта, поки нарешті не ідентифікують об'єкт [25].

Згортковий рівень є основним будівельним блоком ЗНМ, і на ньому відбувається більшість обчислень. Для цього потрібно кілька компонентів, якими є вхідні дані, фільтр і карта функцій. Припустимо, що вхідним буде кольорове зображення, яке складається з матриці пікселів у 3D. Це означає, що вхідні дані матимуть три виміри – висоту, ширину та глибину – які відповідають RGB зображенню. У нас також є детектор ознак, також відомий як ядро або фільтр, який буде переміщатися по сприйнятливим полям зображення, перевіряючи, чи присутня ознака. Цей процес відомий як згортка [25] [26].

Детектор ознак – це двовимірний масив вагових коефіцієнтів, який представляє частину зображення. Хоча вони можуть відрізнятися за розміром, розмір фільтра зазвичай є матрицею 3x3; це також визначає розмір рецептивного поля. Потім фільтр застосовується до області зображення, і розраховується скалярний добуток між вхідними пікселями та фільтром. Далі цей скалярний добуток подається у вихідний масив. Після цього фільтр поступово зміщується, повторюючи процес, доки ядро не пройде усе зображення. Остаточний результат серії скалярних добутків із входу та фільтра відомий як карта ознак, карта активації або згорнута функція [25] [26].

Після кожної операції згортки ЗНМ застосовує перетворення Rectified Linear Unit (Випрямлена лінійна функція) (ReLU) до карти ознак, вносячи нелінійність у модель [25] [26]. Випрямлена лінійна функція – це кусково-лінійна функція, яка виводить вхідний сигнал безпосередньо, якщо він додатний, інакше результатом буде нульове значення ($f(x) = \max(0, x)$). Вона стала функцією активації за замовчуванням для багатьох типів нейронних мереж, у тому числі і для ЗНМ,

оскільки модель, яка використовує її, легше навчати і вона часто досягає кращої продуктивності [26].

Як було згадано раніше, за початковим шаром згортки може слідувати інший шар згортки. Коли це відбувається, структура ЗНМ може стати ієрархічною, оскільки пізніші рівні можуть бачити ознаки виділені попередніми рівнями. Кожна окрема частина моделі складає шаблон нижчого рівня в нейронній мережі, а комбінація його частин представляє шаблон вищого рівня, створюючи ієрархію ознак у ЗНМ [25] [26].

Об'єднання шарів (субдискретизація) виконує зменшення розмірності, зменшуючи кількість параметрів у вхідних даних. Подібно до згорткового рівня, операція об'єднання переміщує фільтр по всьому входу, але відмінність полягає в тому, що цей фільтр не має вагових коефіцієнтів. Замість цього ядро застосовує функцію агрегації, заповнюючи вихідний масив. Існує два основних типи об'єднання. Об'єднання за максимумом: під час руху фільтра по входу він вибирає піксель із максимальним значенням для надсилання у вихідний масив. Об'єднання за середнім значення: під час руху фільтра по входу він обчислює середнє значення в полі аналізованого околу для надсилання у вихідний масив [25] [26].

Об'єднання за максимумом зазвичай використовується частіше, ніж середнє об'єднання. Хоча багато інформації втрачається на рівні об'єднання, це також має ряд переваг для ЗНМ. Вони допомагають зменшити складність, підвищити ефективність і обмежити ризик перенавчання [25] [26].

Згортковий рівень і рівень об'єднання шарів є частково з'єднаними рівнями, тобто значення пікселів вхідного зображення не пов'язані безпосередньо з вихідним шаром. Натомість у повнозв'язному рівні кожен вузол вихідного рівня з'єднується безпосередньо з вузлом попереднього рівня. Цей рівень виконує завдання класифікації на основі ознак, витягнутих через попередні шари та їхні фільтри. У той час як рівні згортки та об'єднання, як правило, використовують функції ReLu, повнозв'язані рівні зазвичай використовують функцію активації softmax для відповідної класифікації вхідних даних, тобто знаходження

ймовірності належення досліджуваного об'єкту до певного класу [25] [26].
Приклад побудови архітектури згорткової мережі наведено на рисунку 3.1.

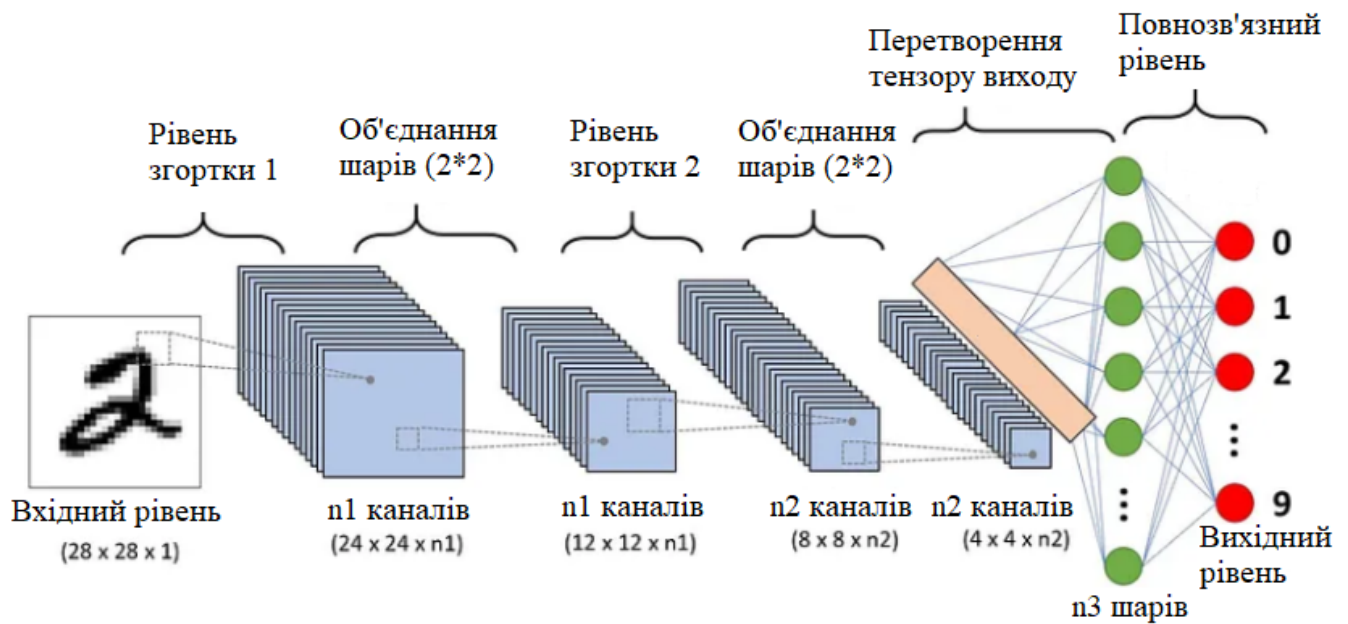


Рисунок 3.2 – Приклад побудови архітектури згорткової мережі [26]

3.3 Висновки до розділу 3

В даному розділі розглянуто підходи до машинного навчання, а саме задача навчання із вчителем, включаючи задачі класифікації та регресії.

Була розглянута структура нейронних мереж і проаналізовані можливості та області їх використання. Було встановлено, що для задачі розпізнавання стадій дозрівання кардіоміоцитів згорткова нейронна мережа через її продуктивність у роботі із зображеннями.

Були проаналізовані шари згорткової нейронної мережі разом із функціями активації та визначення втрат.

4 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ РОЗПІЗНАВАННЯ СТАДІЙ ДОЗРІВАННЯ КАРДІОМІОЦИТІВ З ВИКОРИСТАННЯМ НЕЙРОННИХ МЕРЕЖ

Головною задачею при розробці вказаного програмного забезпечення було підготувати навчальний набір зображень, а також підготувати модель нейронної мережі для її подальшого навчання.

Початковим набором даних слугував набір конфокальних мікроскопічних зображень кардіоміоцитів [27]. Однак, як було вказано раніше існує декілька обмежень до отримання зображень таким чином. Лише частина такого створеного набору даних містить зображення високої якості цілих і здорових міоцитів. Решта набору даних складається з порожніх зображень, зображень мертвих кардіоміоцитів, розфокусованих зображень та зображень фрагментів клітин. Перевагою використання обраного набору даних є те, що дані були попередньо ідентифіковані та відсортовані належним чином у п'ять класів, базуючись на морфологічних чинниках.

Таким чином загалом було доступно 18 зображень першого класу, 23 зображення другого класу, 22 зображення третього класу, 22 зображення четвертого класу і 26 зображень п'ятого класу. Всі зображення мають розмірність у 1024×1024 пікселів у сірошкальному спектрі. Для подальшого навчання було необхідно створити класово-збалансований набір даних. Є два підходи для вирішення поставленої задачі. Перший з них обрати мінімальну кількість зображень із використанням невикористаних даних для оцінки навченої моделі. Другий – використання аугментації для вирівнювання кількості зображень в кожному класі. В цьому випадку навчальний набір буде розділений на навчальну та тестову вибірки. Зважаючи на те, що збільшення кількості оригінальних зображень може покращити якість навчання, був обраний другий підхід.

Зображення з навчального набору можуть містити артефакти, або частини інших клітин. Крім того сірошкальний формат зображень потребує додаткової

обробки їх гістограми яскравості для покращення контрастності зображень, що може позитивно вплинути на якість подальшого навчання. Також мала кількість вхідних зображень не дозволить провести ефективне навчання. Таким чином навчальний набір даних має бути попередньо оброблений для покращення якості подальшого навчання і в загальному алгоритм цифрової обробки зображень можна поділити на наступні кроки:

1. Сегментація вхідного зображення
2. Обробка гістограми яскравості сегментованого зображення
3. Аугментація (отримання нових зображень із попередньо обробленого зображення)

Вихідний датасет зберігався як каталог, де зображення згруповані по класах, що полегшувало подальшу обробку та призначення результуючого класу кожному зображенню. Сегментація зображень виконувалася методами середовища MATLAB. Зображення зчитувалося із відповідного каталогу за допомогою функції «`imread`». Так як вхідний датасет містить сірошкальні зображення однакової розмірності та орієнтації, попередня обробка зображення не була необхідною.

Зображення виводяться за допомогою функції «`imshow`» і полігон задається за допомогою функції «`drawpolygon`». Функція «`drawpolygon`» створює об'єкт полігону, який визначає форму та положення полігональної області інтересу (ПОІ) [28]. ПОІ далі використовується для створення маски за допомогою функції «`createmask`», що використовується як початковий контур, на якому починається еволюція сегментації (рисунк 4.1). Як результат роботи вказаних функцій отримали бінарне зображення того самого розміру, що й вхідне зображення. Бінарне зображення в даному контексті можна розуміти як збереження (значення «1» бінарного зображення) яскравості пікселя вхідного зображення, або зведення його до нульового значення (значення «0» бінарного зображення). Для сегментації зображення була використана функція «`activecontour`», що реалізує ітеративний алгоритм сегментації зображення із збільшенням області [29]. В якості методу сегментації був обраний метод Чан-Везе. Функція приймає початковий контур і вхідне зображення, результатом виконання є бінарне зображення маски

сегментованого зображення. Результуюче зображення отримуємо шляхом накладання маски на вхідне зображення за правилами вказаними вище (рисунок 4.2).

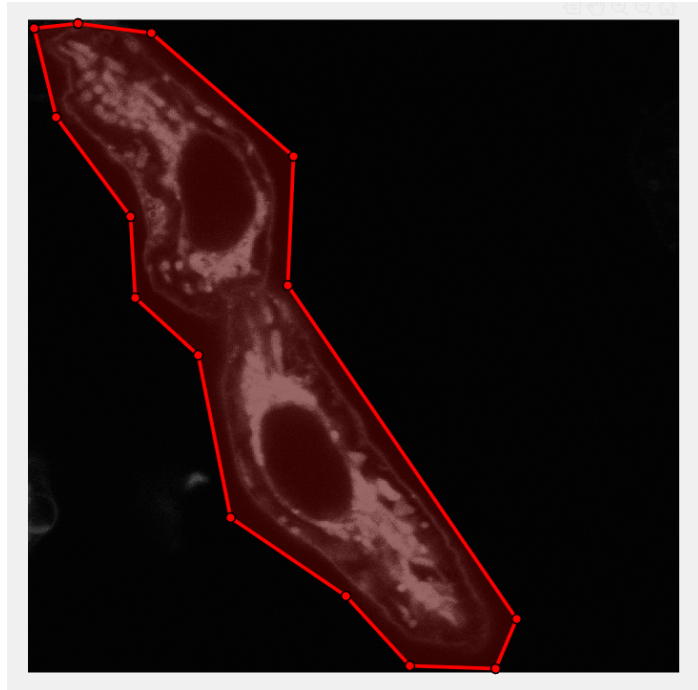


Рисунок 4.1 – Приклад вхідного зображення та ПОІ до нього

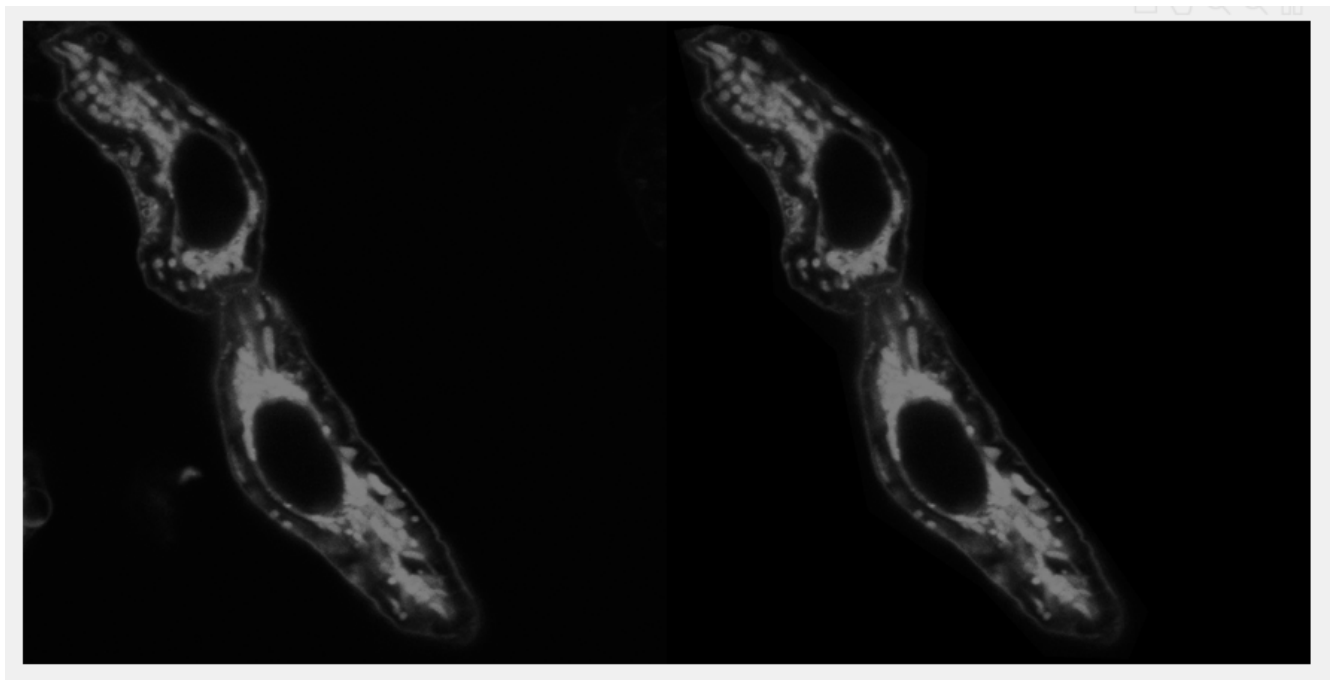


Рисунок 4.2 – Приклад вхідного (зліва) та вихідного сегментованого (зправа) зображень

Сегментовані зображення були збережені в окремий датасет зі збереженням ієрархії файлів для подальшої роботи.

Обробку гистограми яскравості зображень, а також їх аугментацію було винесено в окремий алгоритм і виконано методами PYTHON. Використання PYTHON є більш доцільним для обробки великої кількості вхідних даних за рахунок його оптимізації і наявності великої кількості гнучких бібліотек для роботи із зображеннями.

Виконується обробка усіх зображень із вказаної директорії. Зчитування зображення відбувається методами бібліотеки “cv2” і функції “imread”. Для обробки гистограми яскравості зображення була використана функція “createCLAHE” із бібліотеки “cv2”, що використовує метод контрастно обмеженого адаптивного вирівнювання і має широкий інтерфейс можливих налаштувань [30].

Параметр «clipLimit» впливає на контрастний фактор, який запобігає перенасиченню зображення зокрема в гомогенних областях [30]. Ці області характеризуються високим піком в гистограмі конкретної області зображення. Без заданої межі метод міг привести до результатів, які, в деяких випадках, гірші, ніж оригінальне зображення. Параметром «tileGridSize» задається розмір сітки на яку розбивається задане зображення [30]. При виборі розмірів слід зважати на розміри вхідного зображення (велика кількість областей на малому зображенні можуть погіршити контрастність і спричинити появу зайвого шуму на границях областей) і поставлених вимог щодо часу виконання перетворення (збільшення кількості областей призводить до збільшення часу виконання перетворення).

Результати обробки яскравості зображення та його гистограма в порівнянні із вихідним зображенням наведені на рисунках 4.3 та 4.4.

Зображення з обробленою гистограмою яскравості використовується для отримання нових зображень, або ж аугментації. Були обрані методи повороту та віддзеркалення зображення, що запобігають альтерації поперечного Т-канальця, що є важливим параметром для правильної класифікації, а також не погіршують контрастність зображення.

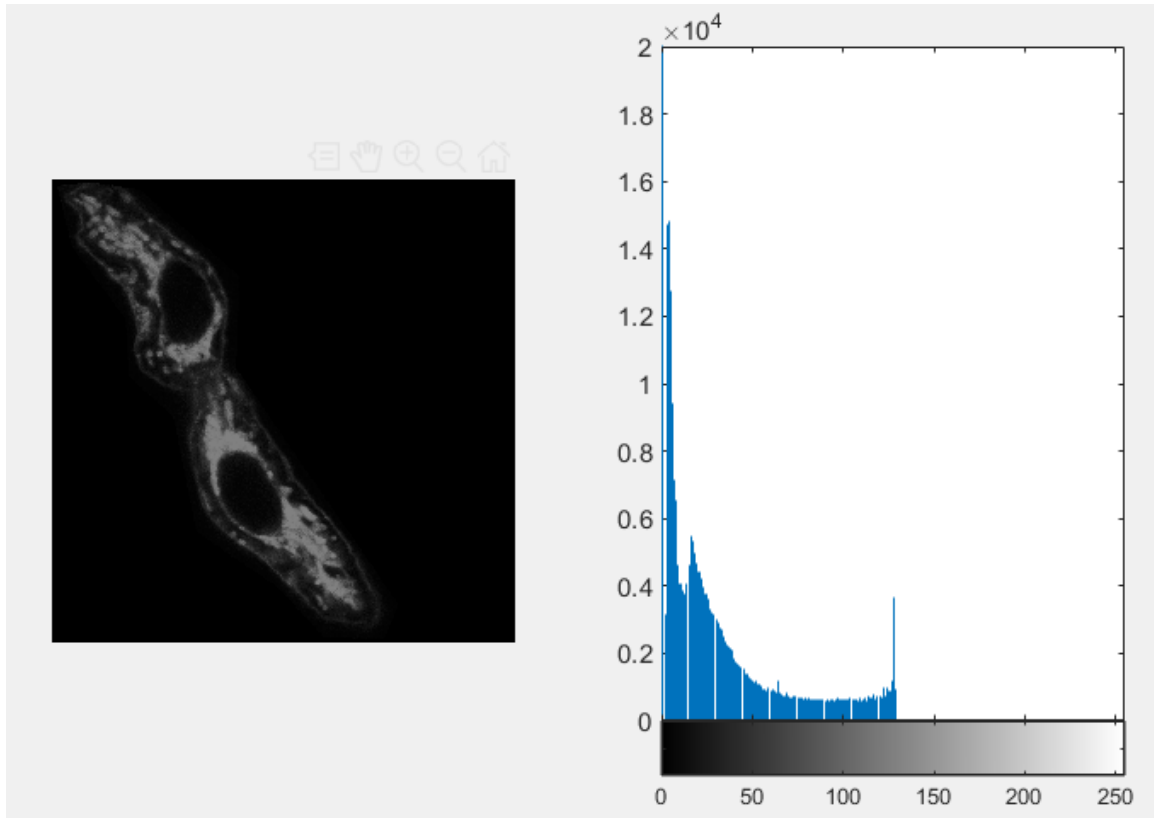


Рисунок 4.3 – Сірошкальне вихідне зображення і його гістограма яскравості

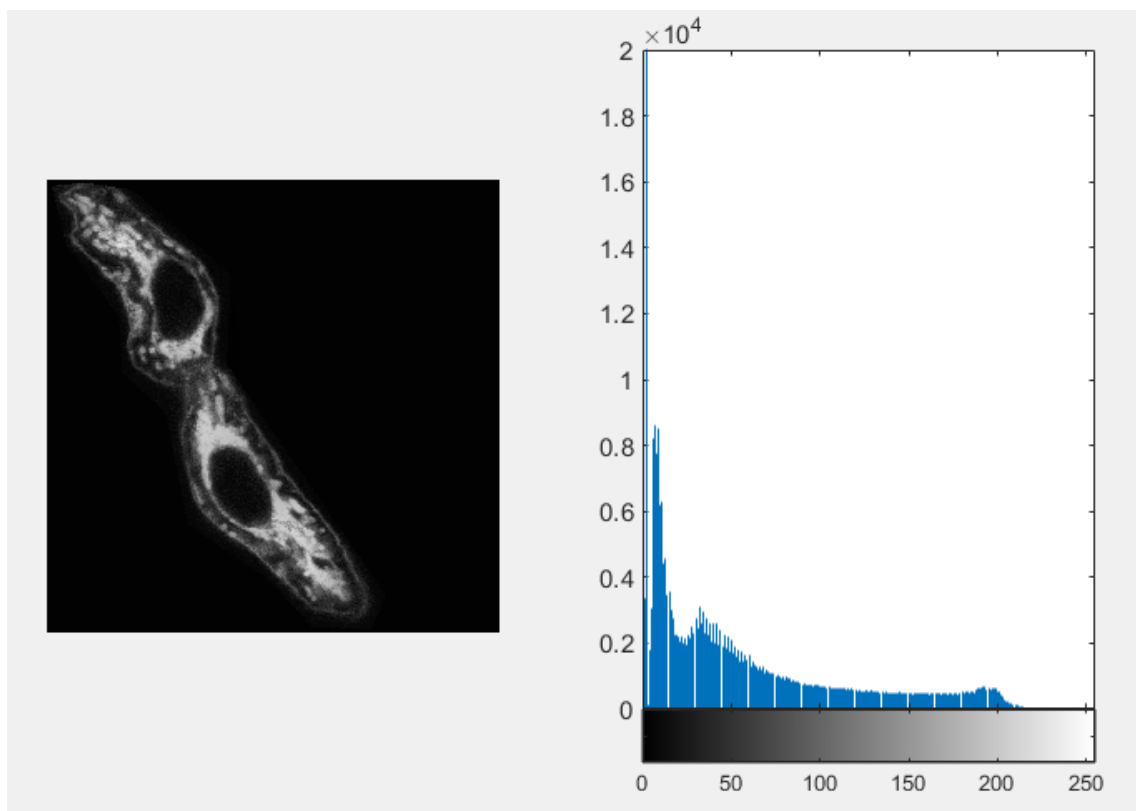


Рисунок 4.4 – Оброблене зображення і його гістограма яскравості

Обертання зображення відбувалося методами функції “rotate” бібліотеки “imutils”, що має більш зручний інтерфейс для функціональності наданої бібліотекою “cv2”. Обертання відбувалося навколо центру зображення із кроком в десять градусів. Приклад обертання зображення наведено на рисунку 4.5.

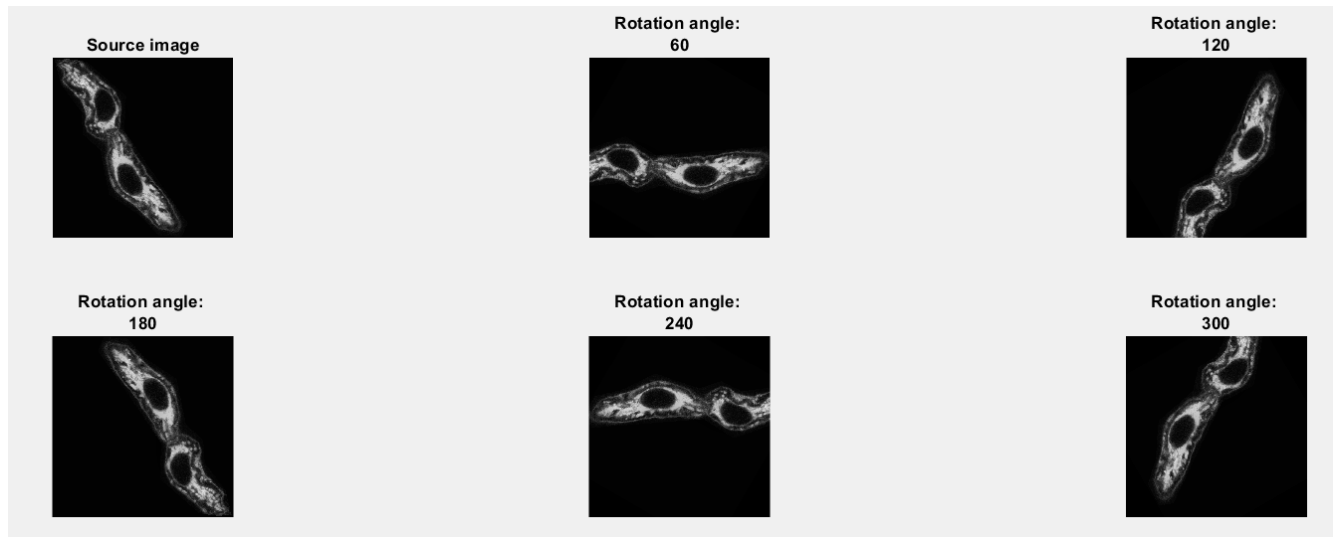


Рисунок 4.5 – Приклад обертання вхідного зображення

Відображення зображень відбувалося навколо центральної вертикальної осі за допомогою функції “flip” із бібліотеки “cv2” (рисунок 4.6).

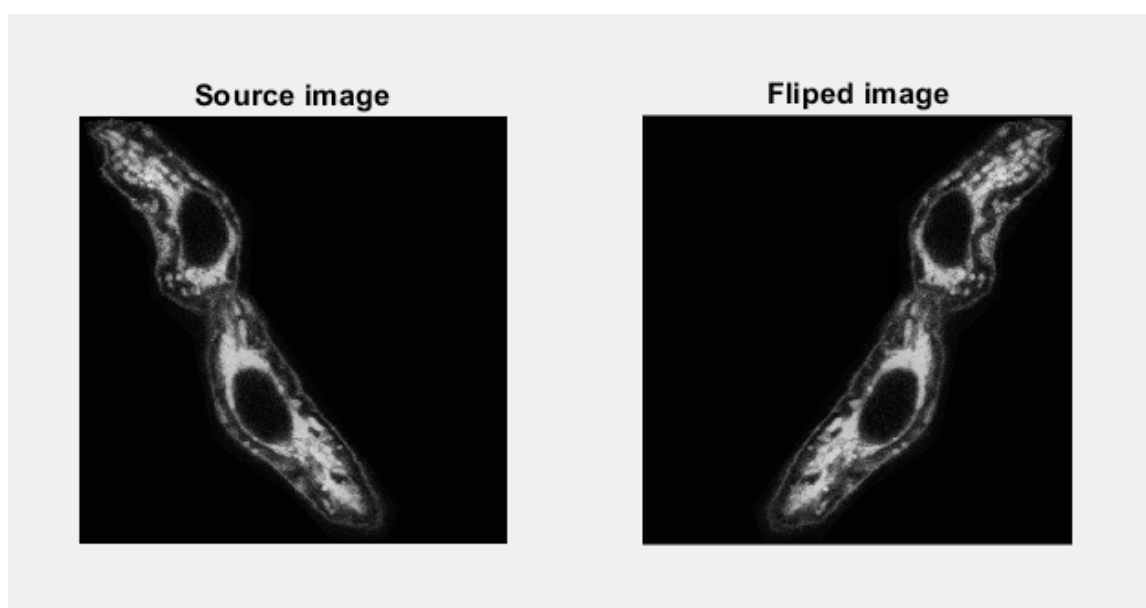


Рисунок 4.6 – Приклад віддзеркалення вхідного зображення

Можна бачити, що в якості основних методів аугментації були обрані геометричні перетворення зображень, що запобігають альтерації поперечного Т-канальця, що є важливим параметром в подальшій класифікації. Натомість для вирівнювання кількості зображень в кожному класі було необхідно обрати метод не пов'язаний з геометричними перетвореннями, що дозволяє збільшити диференціацію даних. В якості такого методу була обрана інверсія кольорів із додатковим використанням гамма-корекції, що відноситься до трансформацій колірного простору і також не змінює структуру кардіоміоцитів. Інверсія виконувалась за законом:

$$I_{out} = 255 - I_{in},$$

де 255 – максимальне значення яскравості пікселя в системі uint8 (тип даних для роботи із зображеннями і збереження значень, що відповідають яскравості пікселів в діапазоні від 0 до 255 і належить до бібліотеки “numpy” [31]).

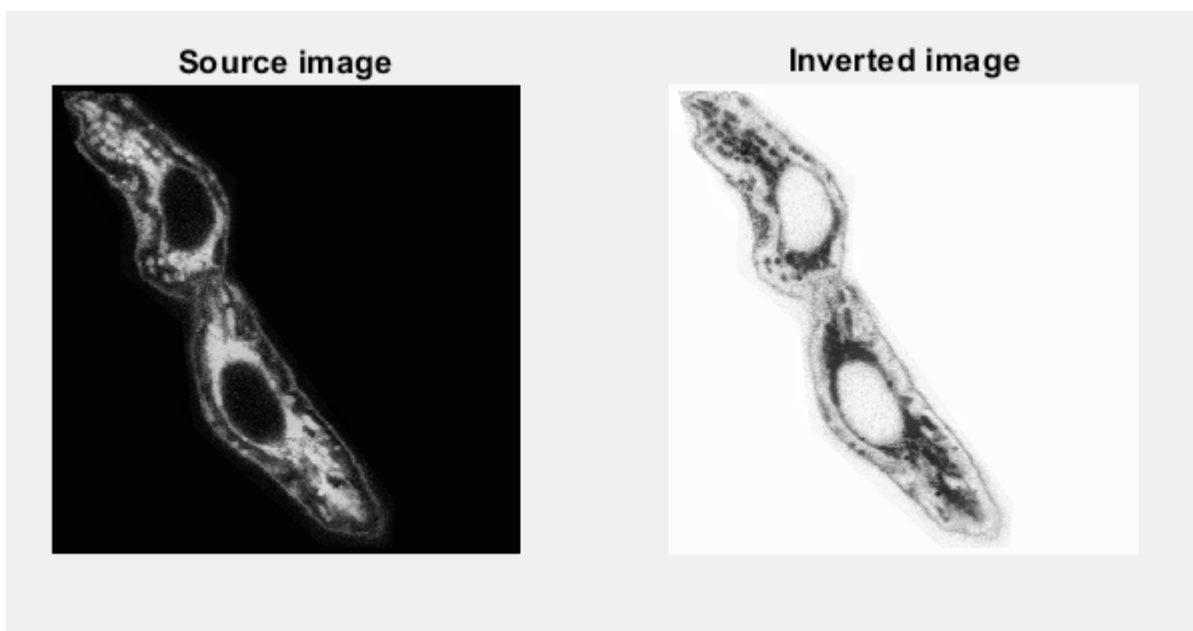


Рисунок 4.7 – Приклад інвертування вхідного зображення

Зображення із обробленою гистограмою яскравості, а також нові зображення, отримані після аугментації були збережені в окремий датасет зі збереженням ієрархії файлів для подальшого навчання нейронної мережі. Причому зображення були розділені на набори навчальних та тестових даних. Розбиття відбувалося за допомогою використання функції “train_test_split” бібліотеки із бібліотеки “sklearn.model_selection” [32]. Параметр “test_size”, що встановлює співвідношення тренувальних даних до тестових було обрано рівним 0.2, що відповідає 80% тренувальних даних порівняно із валідаційними.

Підготовка і побудова архітектури нейронної мережі була виконана в середовищі PYTHON із використанням відкритого API (Application Programming Interface) бібліотеки “tensorflow.keras” [33]. Був обраний підхід із побудовою ієрархічної структури ЗНМ із поступовим збільшенням кількості фільтрів, що значно покращує продуктивність моделі.

Вхідний шар нейронної мережі встановили методом “Input” бібліотеки “tensorflow.keras”. Параметр “shape”, що визначає розмірність вхідного зображення було встановлено рівним матриці [1024; 1024; 1], що відповідає зображенням у сірошкальному спектрі. На вхід моделі подаються зображення із підготовленого датасету.

Для додавання рівня згортки до структури нейронної була використана функція “Conv2D”, що являє собою двовимірний шар згортки і створює ядро згортки, що допомагає створити тензор виходів (об’єкт векторного простору кінцевої розмірності) [33]. Розмірність фільтру була обрана рівною матриці [3; 3], у той час як кількість фільтрів може змінюватися, як було зазначено вище. Також додатковому параметру “padding” було встановлено значення “same”, що дозволяє зберегти просторові розміри об’єму таким чином, щоб розмір вихідного об’єму відповідав розміру вхідного об’єму [33]. Функцією активації виступає виправлена лінійна функція (ReLU), що реалізується методом із аналогічною назвою бібліотеки “tensorflow.keras”.

Додатковим рівнем на кожному шарі ієрархічної структури був обраний рівень пакетної нормалізації, що стандартизує кожну партію входів до певного

рівня, іншими словами виконує нормалізацію вихідного вектору параметрів. Це дозволяє стабілізувати процес навчання та значно скоротити кількість циклів навчання, необхідних для створення мережі. Рівень пакетної нормалізації реалізується методом “BatchNormalization” бібліотеки “tensorflow.keras” [33].

Таким чином кожен шар ієрархічної структури згорткової нейронної мережі може бути представлений схематично так, як це показано на рисунку 4.8.

Рівень об’єднання шарів реалізується методом “MaxPooling2D”, що виконує об’єднання шарів у двовимірному просторі. Розмірність фільтра була обрана рівною матриці [2; 2]. Параметр “strides”, що визначає крок переміщення вікна об’єднання для кожного кроку об’єднання, було обрано аналогічним до розмірності фільтра [33].

Повнозв’язаний рівень був реалізований поєднанням трьох підрівнів. Перетворення багатовимірного вхідного тензора на одновимірний тензор виконувалося методом “Flatten” бібліотеки “tensorflow.keras” [33]. Даний рівень немає параметрів для навчання, але змінює форму тензора, що необхідно для подальшого застосування повнозв’язного нейронного шару, що реалізується методом “Dense” [32]. Вхідним параметром слугує кількість класів для класифікації, що у нашому випадку дорівнює п’яти. Активація відбувалася “Softmax” функцією, що реалізується методом з аналогічною назвою. Реалізована архітектура схематично зображена на рисунку 4.9.

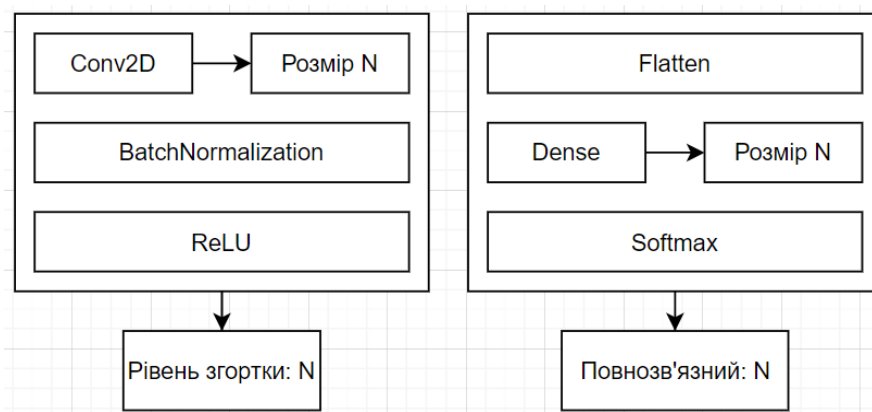


Рисунок 4.8 – Схематичне представлення шару ієрархічної структури згорткової нейронної мережі і повнозв’язаного шару

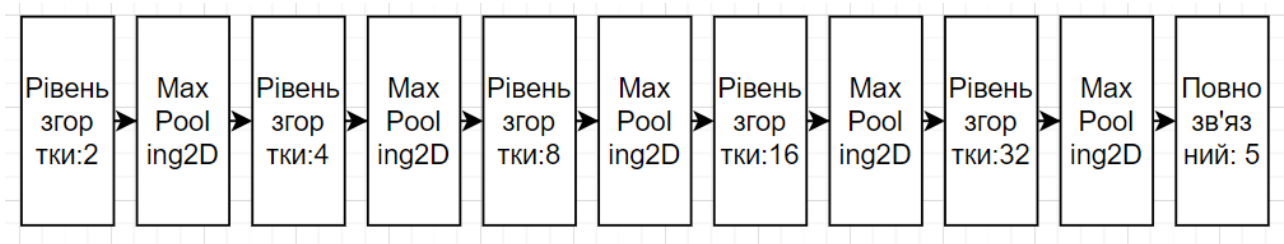


Рисунок 4.9 – Повна архітектура нейронної мережі

Однією з головних переваг побудованої глибокої мережі є те, що вона може представляти складні та комплексні функції, що гарно співвідноситься із вирішенням задачі класифікації стадії дозрівання кардіоміоцитів. Процес навчання моделі буде детально описаний у наступному розділі, проте слід вже зараз зазначити, що навчання відбувалося не на повному наборі даних, а поступово, використовуючи менший набір зображень через технічні обмеження. Це призводить до того, що певний набір даних може мати більшу, або меншу кількість корисних ознак порівнюючи з іншими. В результаті можна було спостерігати ефект деградації моделі для певного набору зображень (рисунок 4.10).

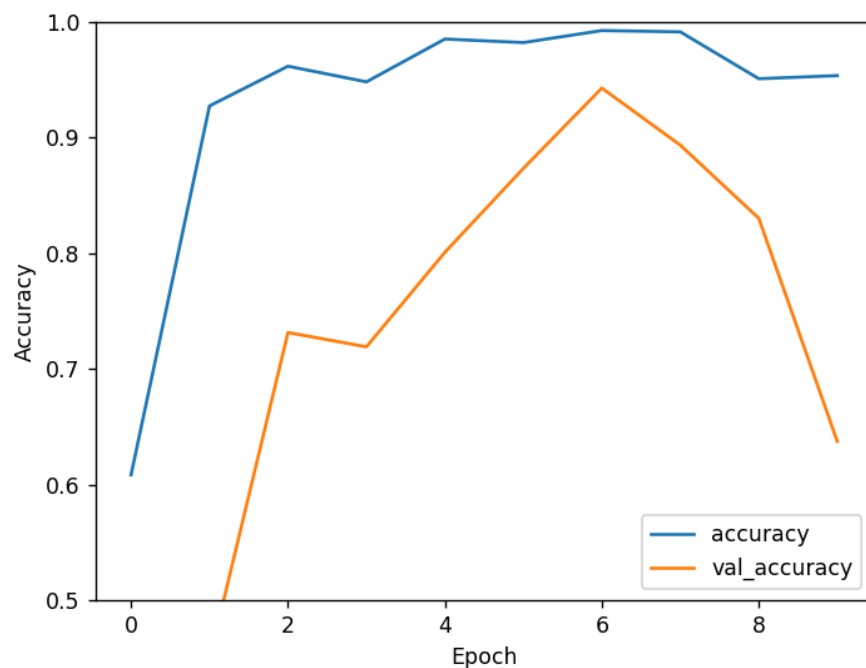


Рисунок 4.10 – Ефект деградації моделі ЗНМ при навчанні

На рисунку 4.10 можна бачити залежність точності моделі при навчанні (accuracy) і оцінка моделі на нових даних (val_accuracy) в залежності від різних епох (ітерації) навчання. Ефект деградації моделі ЗНМ проявляється як різке зменшення точності класифікації нових даних за тієї ж точності при навчанні.

Цей ефект пов'язаний з тим, що глибокі мережі часто мають градієнтний сигнал, який швидко доходить до нуля, що робить спуск градієнта непомірно повільним. Точніше кажучи, під час градієнтного спуску, при поверненні від останнього шару назад до першого, відбувається множення на вагову матрицю на кожному кроці. Якщо градієнти малі через велику кількість множень, градієнт може експоненціально швидко зменшитися до нуля (або, у рідкісних випадках, експоненціально швидко зростати та «вибухати», приймаючи дуже великі значення) [33]. Як результат маємо погіршення продуктивності моделі при використанні більш глибокої моделі, ніж це необхідно на певному етапі. Вирішенням даної проблеми може бути використання залишкового блоку, який схематично зображено на рисунку 4.11.

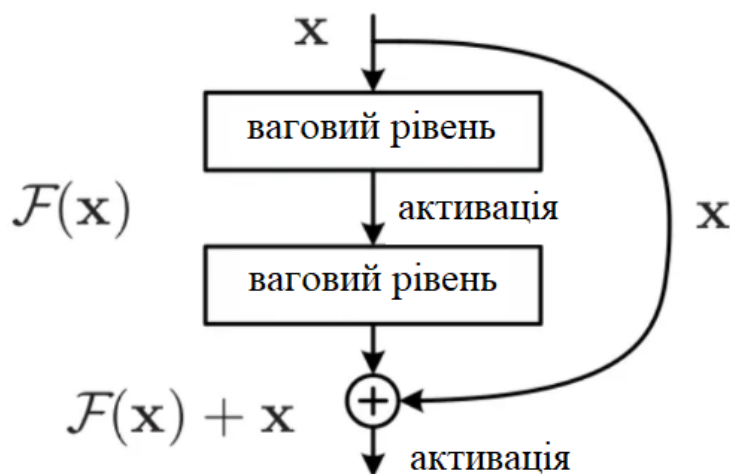


Рисунок 4.11 – Схематичне зображення залишкового блоку [34]

Ідея полягає в тому, що замість того, щоб дозволяти рівням вивчати базове відображення, дозволяти мережі відповідати залишковому відображенню. Отже, замість початкового відображення $H(x)$, нехай мережа відповідає $F(x)$: $F(x) = H(x) - x$, що відповідає $H(x)$: $H(x) = F(x) + x$. Підхід полягає в тому, щоб пропустити

з'єднання, що зменшує вірогідність впливу некорисного слою на градієнтний сигнал. Додавання додаткових, або нових шарів не зашкодить продуктивності моделі, оскільки регуляризація пропускати їх, якщо ці шари не будуть корисними. Якщо додаткові/нові шари були корисними, навіть за наявності регуляризації, ваги або ядра шарів будуть відмінними від нуля, а продуктивність моделі (точність класифікації) може дещо збільшитися. Таким чином, шляхом додавання нових шарів із використанням залишкового блоку може гарантувати, що продуктивність моделі (точність класифікації) не зменшиться, але може трохи збільшитися [34].

Після використання залишкового блоку разом із більш глибокими шарами, отримали наступну архітектуру моделі (рисунок 4.12).

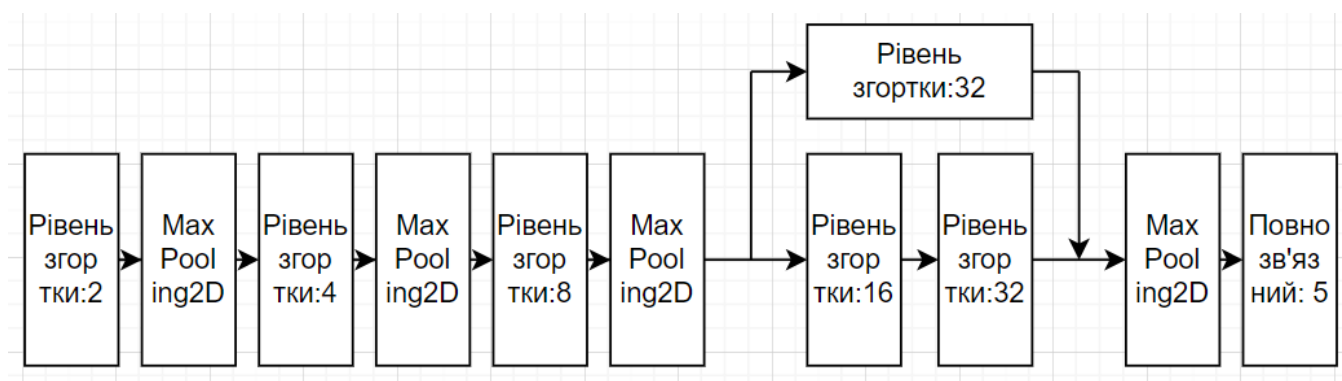


Рисунок 4.12 – Архітектура запропонованої нейронної мережі

Результуюча модель будувалася методом “Model” бібліотеки “tensorflow.keras” [33]. Параметру “inputs”, що задає вхідну точку для навчання моделі було задано значення вхідного рівня, натомість параметру “outputs” задається побудована архітектура згорткової нейронної мережі. Результатом виконання є об’єкт моделі нейронної мережі із набором функцій для її компіляції і навчання.

Компіляція моделі відбувалася викликом функції “compile”. В якості методу оптимізації був обраний метод Адама, що є розширенням стохастичного градієнтного спуску [33]. Функцією оцінки втрат було обрано функцію оцінки втрат кросентропії між мітками та прогнозами, що реалізується методом

“SparseCategoricalCrossentropy” [33]. Для оцінки ефективності моделі був обраний параметр точності, що визначається відношенням правильно класифікованих тестових даних до загальної їх кількості. Результат компіляції разом із структурою моделі наведено на рисунку А.1. Скомпільована модель була збережена для подальшого навчання.

4.1 Висновки до розділу 4

В даному розділі був розроблений алгоритм цифрової обробки зображень, а також підготовлена модель ЗНМ для подальшого навчання.

Сегментація зображень відбувалася в середовищі MATLAB і дозволила прибрати артефакти із зображень, що могли зашкодити подальшому навчання.

Основний алгоритм цифрової обробки зображень, що включає в себе обробку гістограми яскравості зображень і їх аугментацію був реалізований з використанням мови програмування Python та доступних бібліотек функцій для обробки зображень. Обробка гістограми яскравості була виконана методом контрастно обмеженого адаптивного вирівнювання і дозволила покращити контрастність зображень для покращення ефективності навчання моделі. Аугментація даних дозволила отримати нові зображення для проведення ефективного навчання, а також отримати класово-збалансований набір даних. Із початкового набору даних, що містить 18 зображень першого класу, 23 зображення другого класу, 22 зображення третього класу, 22 зображення четвертого класу і 26 зображень п'ятого класу, було отримано датасет, що включає 1036 зображень на кожний клас.

Було побудовано архітектуру згорткової нейронної мережі з ієрархічною структурою і використанням залишкового блоку. Модель було скомпільовано і збережено для подальшого навчання.

5 ЗАДАЧА КЛАСИФІКАЦІЇ ДЛЯ ІДЕНТИФІКАЦІЇ СТАДІЙ ДОЗРІВАННЯ КАРДІОМІОЦИТІВ

Як вже зазначалося раніше, навчання відбувалася у кілька етапів, що було пов'язано із обмеженими можливостями використовуваної обчислювальної системи. Для цього було введено поняття спліт фактору, тобто розбиття повного масиву вхідних даних на менші частини. Частина зображень для кожного класу завантажувалася для використання у подальшому навчанні. Кожному зображенню призначався клас, до якого належить дане зображення. Навчання відбувалося в п'ять етапів, які були названі ітераціями.

Модель, архітектура якої була побудована в минулому розділі, містить функцію “fit”, що виконує навчання та донавчання даної моделі. Вхідними параметрами для неї слугують набори тренувальних та валідаційних зображень, а також кількість епох навчання, яка була встановлена рівною п'яти. Таким чином, маючи п'ять ітерацій навчання, загальна кількість епох навчання складає 25 із різними наборами зображень.

Розбиття масиву вхідних зображень на набори тренувальних та валідаційних зображень відбувалася за допомогою використання функції “train_test_split” бібліотеки із бібліотеки “sklearn.model_selection” [32]. Параметр “test_size”, що встановлює співвідношення тренувальних даних до тестових було обрано рівним 0.2, що відповідає 80% тренувальних даних порівняно із валідаційними.

Для оцінки якості навчання моделі були обрані параметри “accuracy”, що відповідає точності класифікації тренувальних зображень і “val_accuracy”, що відповідає точності класифікації валідаційних даних.

Результати навчання моделі відповідно до ітерації наведено на рисунку 5.1, а також на рисунках із Додатку Б. Можна також показати еволюцію точності класифікації першої епохи навчання відповідно до ітерації (рисунок 5.2).

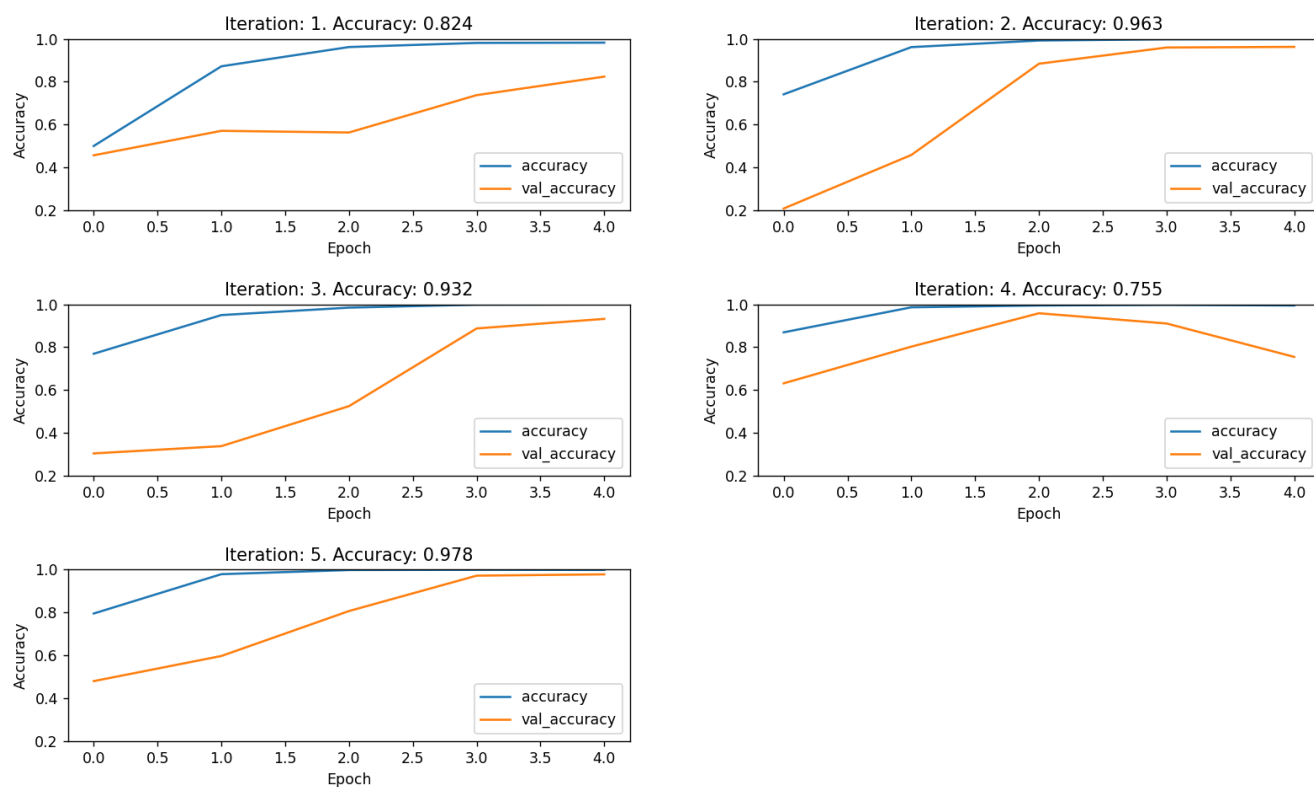


Рисунок 5.1 – Результати навчання моделі відповідно до ітерації

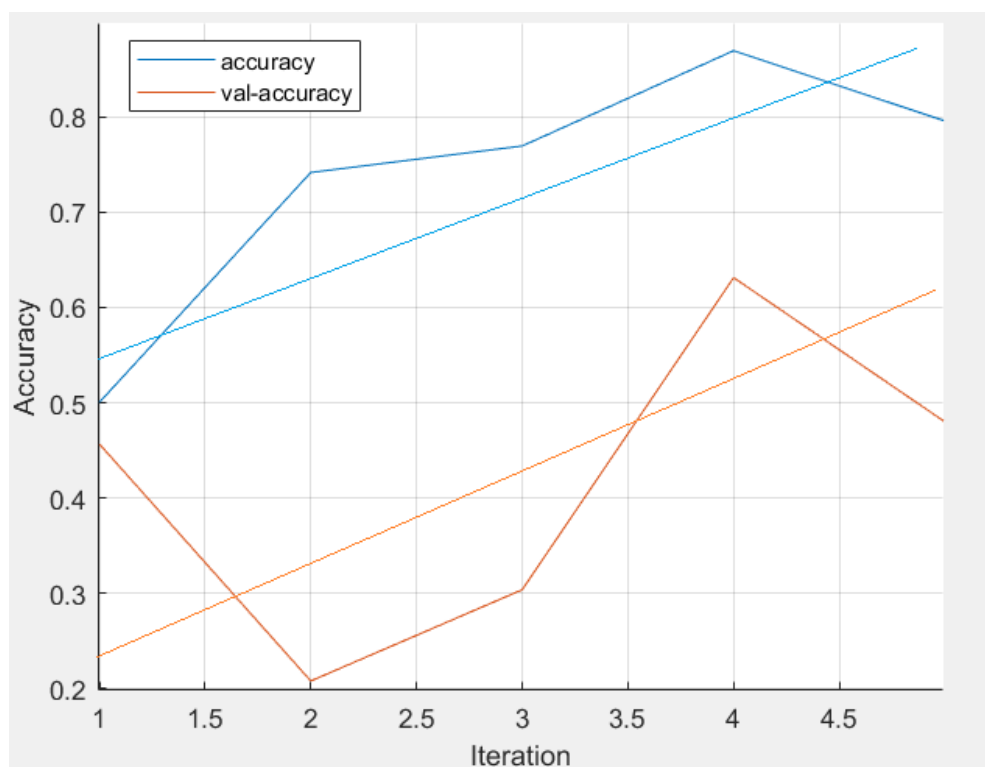


Рисунок 5.2 – Еволюція точності класифікації першої епохи навчання відповідно до ітерації

Можна проаналізувати теплові карти різних рівнів згорткових шарів та виділити параметри, що впливають на класифікацію стадії дозрівання кардіоміоцитів.

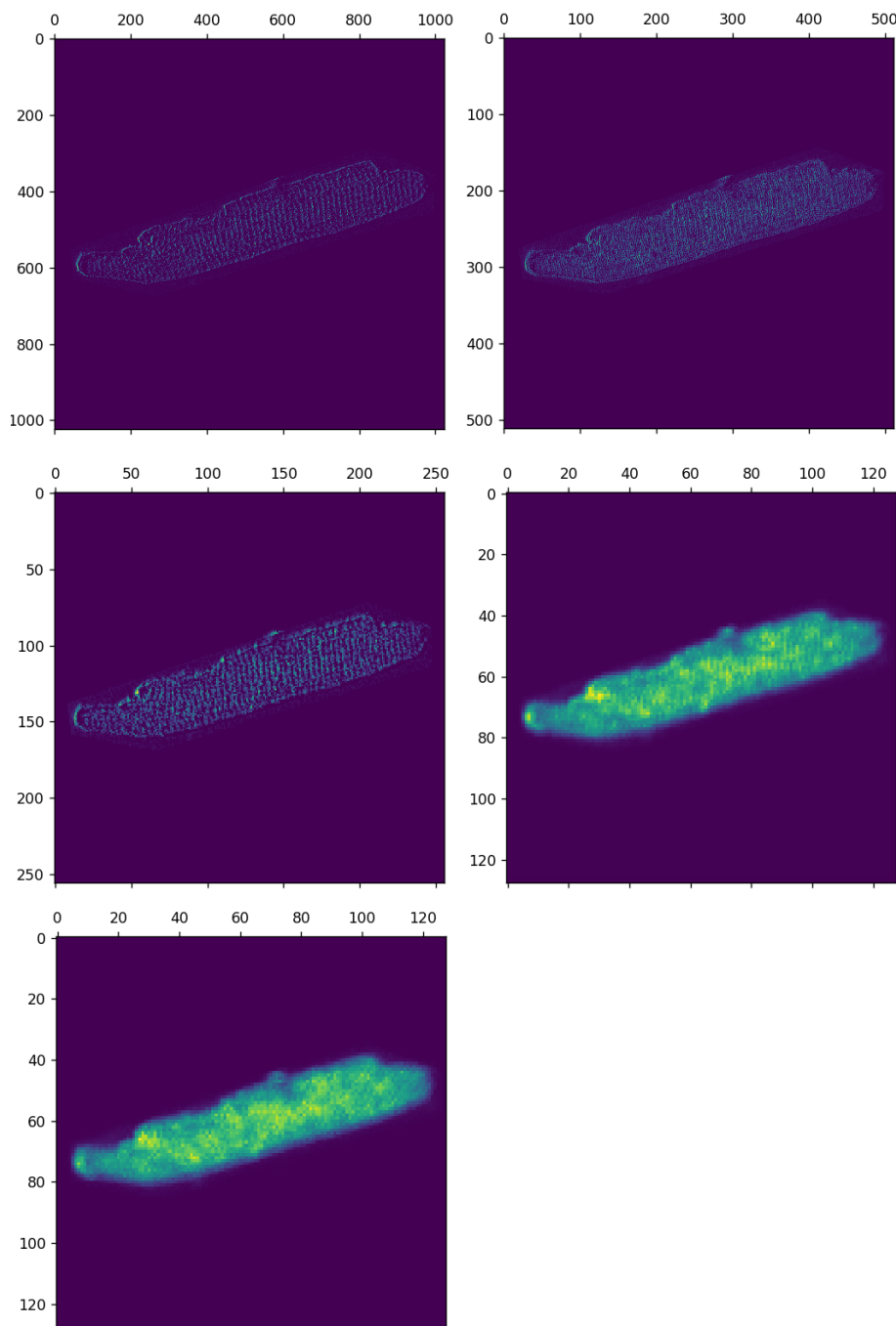


Рисунок 5.3 – Еволюція теплових карт згорткових шарів ЗНМ

Побудова теплової карти відбувалася методами бібліотеки “tensorflow.keras” [33]. Нова модель будувалася на основі вхідного та вихідного

рівнів оригінальної моделі за тою відмінністю, що нова модель має додатковий вихідний параметр, що являє собою результат роботи відповідного згорткового рівня, що знаходився за допомогою методу “get_layer”. Теплова карта будувалася на основі класифікованого зображення із використанням функції “gradient”. Для побудови зображення теплової карти, вона нормалізувалася і перетворювалася за законом:

$$I_{out} = \max(0, I_{in})$$

Можна бачити, що перші шари ЗНМ виділяють форму та структуру кардіоміоцитів, в той час як більш глибокі шари збільшують вагові коефіцієнти факторів, що підкреслюють структуру поперечних Т-каналців кардіоміоциту. Цей ефект можна підкреслити шляхом накладання теплової карти на початкове зображення кардіоміоциту. Також можна бачити зменшення розмірності теплової карти, що є результатом роботи слою зменшення дискретизації.

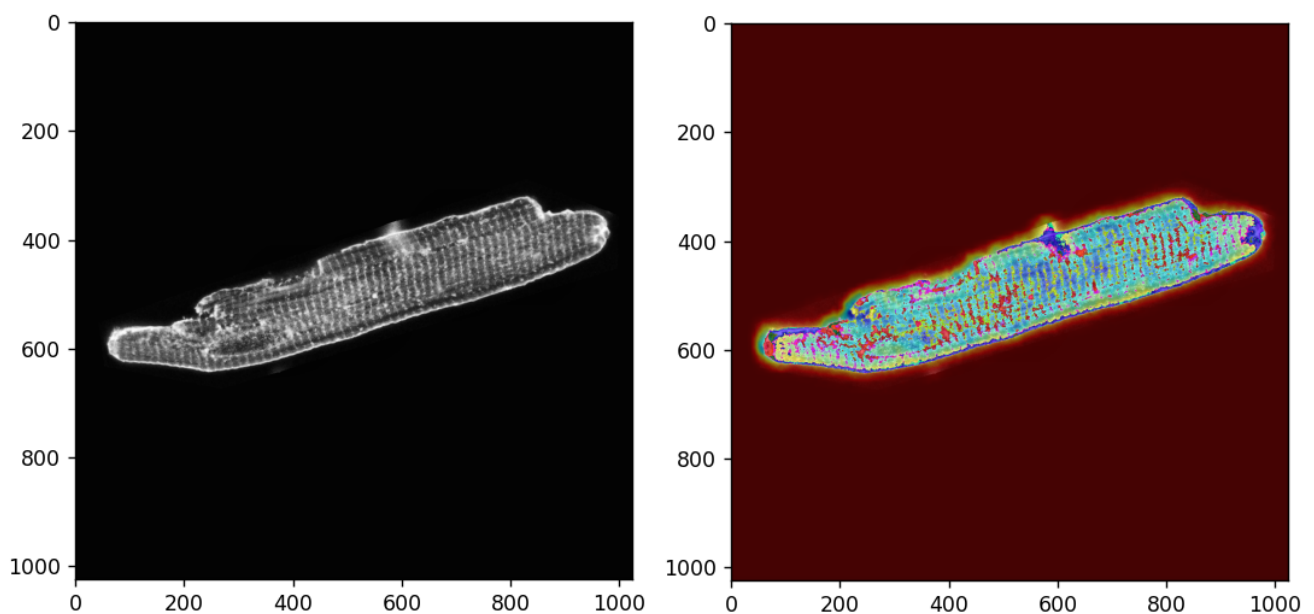


Рисунок 5.4 – Результат накладання теплової карти останнього згорткового рівня на початкове зображення кардіоміоциту

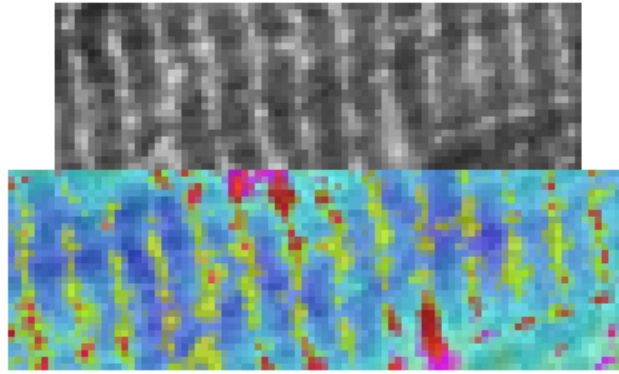


Рисунок 5.5 – Виділення поперечної структури Т-канальців кардіоміоциту

Аналізуючи отримані результати можна бачити, що точність класифікації першої епохи навчання збільшується відповідно до ітерації. Отже, ознаки виділенні з попереднього етапу навчання використовуються та оновлюються при донавчанні з новим набором даних. Результуюча точність навчання склала 0,978. Проте тут слід зазначити, що відсоток аугментованих даних в тренувальному наборі є досить високим, що призводить нас до того, що валідаційні дані можуть містити ознаки, або частини тренувальних зображень, що впливає на оцінку точності навчання. Тому для більш точного аналізу продуктивності навченої ЗНМ слід використовувати нові зображення, що попередньо не використовувалися при навчанні, або валідації.

Тестові дані були підготовленні в результаті виконання минулого розділу. Для оцінки якості навчання використовувалося два підходи. Перший із використанням функції “predict”, що повертає масив вірогідностей належання зображення до кожного класу. В якості результату класифікації обирався клас із найбільшою вірогідністю. Знаючи приналежність кожного зображення певному класу і результати класифікації, можна побудувати та проаналізувати матрицю помилок навчання. Другим методом виступає використання функції “evaluate”, що виконую класифікацію для набору зображень та повертає результуючу точність класифікації (рисунок 5.6).

```

Confusion Matrix
[[126  54  34  28  18]
 [ 23 127  52  42  16]
 [  6  15 130  87  22]
 [ 11  1  23 188  37]
 [  0  0   4  13 243]]
accuracy: 62.62%

```

Рисунок 5.6 – Матриця помилок і результуюча точність класифікації

Можна бачити, що точність класифікації зменшилася порівняно з попередніми оцінками, що може свідчити про великий відсоток аугментованих даних у навчальному наборі. Враховуючи той факт, що велика кількість тренувальних даних необхідна для побудови ефективної ЗНМ, отримуємо необхідність у подальших дослідженнях ПСК-КМ із виділенням більшої кількості оригінальних зображень кардіоміоцитів різних стадій дозрівання. Це дозволить зменшити відсоток аугментованих даних у навчальному наборі і підвищити точність класифікації нових зображень.

Також можемо бачити великий відсоток взаємних помилок між першим та другим, а також третім і четвертим класами. Це може свідчити про недостатню диференційованість вказаних стадій розвитку та про помилки початкової класифікації. Точність класифікації першого і другого класів склала 48 і 49 відсотків відповідно, причому 29 відсотків помилок вказаних класів є взаємними. Для третього класу точність класифікації склала 50 відсотків, для четвертого – 72 відсотки. Кількість взаємних помилок складає 54 відсотки від загальної кількості помилок вказаних класів. Точність класифікації для п'ятого класу склала 93 відсотки. Тобто можна бачити велику різницю між точністю класифікації ранніх та пізніх стадій дозрівання кардіоміоцитів із високою точністю класифікації пізніх стадій. Це пов'язано із виділеними параметрами класифікації (структурою поперечних Т-каналців). В той час як структура кардіоміоцитів із пізньою стадією дозрівання є ярко вираженою, для ранніх стадій структура каналців є слабо-диференційованою, або хаотичною.

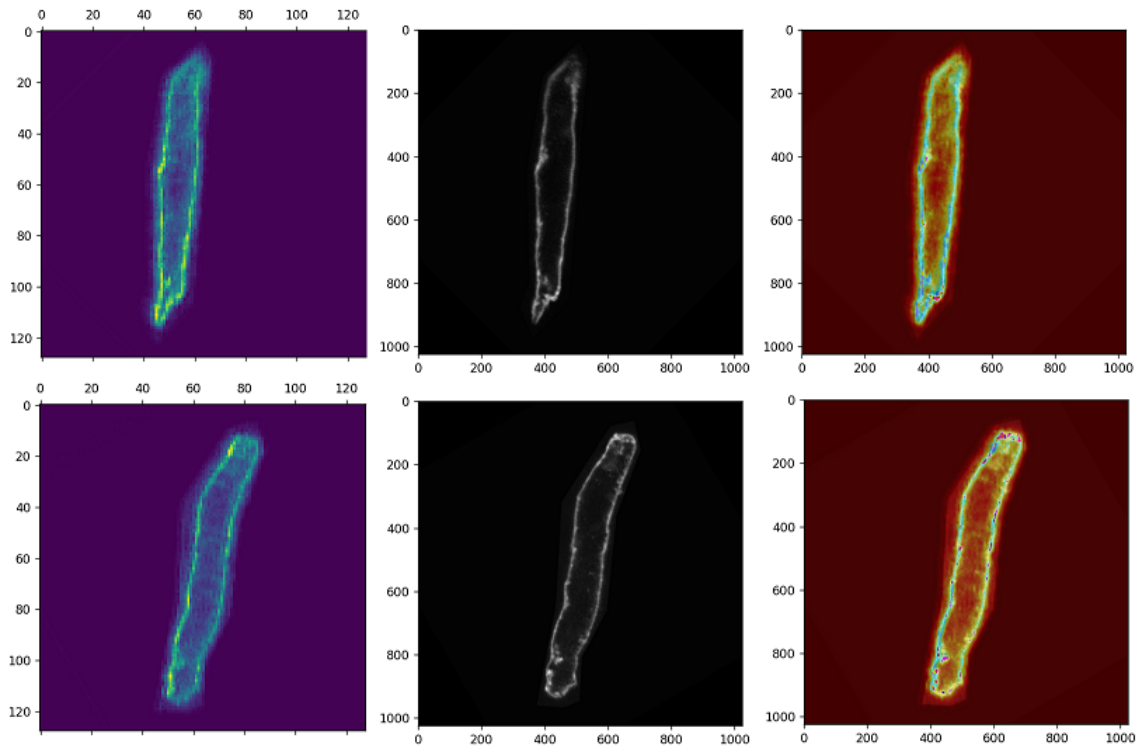


Рисунок 5.7 – Приклад взаємних помилок між першим (зверху) та другим (знизу) класами

Якщо звернути увагу на приклади зображень, що відносяться до першого та другого класів та класифіковані із взаємними помилками (рис 5.7) можна бачити слабку диференціацію зображень кардіоміоцитів, а також фокусування теплової карти на границях кардіоміоцитів, а не на структурі поперечних Т-каналців. Це пов'язано із їх відсутністю, або ранньою стадією їх розвитку. Це також пояснює загальну велику кількість помилок для вказаних класів. Аналогічну ситуацію можна бачити із зображеннями третього та четвертого класів (рис 5.8). Тут структура поперечних Т-каналців є слабо виділеною через слабкі границі і хаотичну структуру. Натомість можна бачити сильну взаємну диференціацію пар першого та другого класів і третього та четвертого класів.

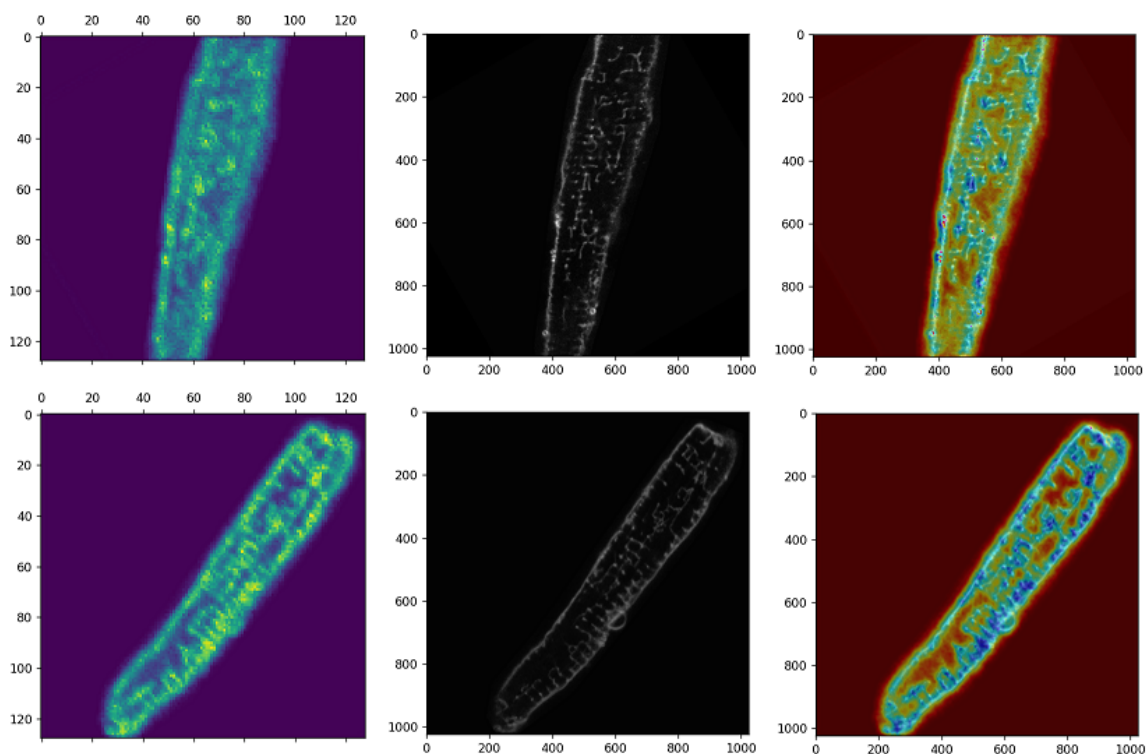


Рисунок 5.8 – Приклад взаємних помилок між третім (зверху) та четвертим класами (знизу)

Якщо виконати об'єднання класів, що демонструють велику кількість взаємних помилок, а саме визначити перший та другий класи як ті, що мають початкову стадію розвитку, а третій і четвертий – перехідну, то маємо наступну матрицю помилок.

```
Confusion Matrix
[[330 156  34]
 [ 33 428  59]
 [  0  17 243]]
```

Рисунок 5.9 – Матриця помилок після об'єднання класів

Після об'єднання класів результуюча точність складає 77%. Для подальшого покращення точності класифікації слід розвивати та покращувати архітектуру ЗНМ із використанням більш глибоких шарів, що потребує додаткових обчислювальних можливостей.

5.1 Висновки до розділу 5

У ході виконання даного розділу було виконано навчання моделі ЗНМ із попередньо визначеною архітектурою. Навчання відбувалося у п'ять ітерацій. Було показано еволюцію навчання моделі і наведені теплові карти для різних згорткових рівнів. Було показано процес виділення структури поперечних Т-канальців кардіоміоциту.

Було виконано оцінку точності класифікації з використанням нових зображень. Результиуюча точність склала 62.62%. Була проаналізована матриця помилок і зображення класів, що демонструють велику кількість взаємних помилок. На основі аналізу було виконано об'єднання класів. Зображення першого та другого класів були віднесені до початкової стадії розвитку, а третього і четвертого – до перехідної. Після об'єднання точність складає 77%.

На основі результатів навчання та їх аналізу були визначені наступні кроки направлені на покращення результатів.

ВИСНОВКИ

У ході виконання дипломної роботи були розглянуті основні методи обробки зображень і основні підходи до машинного навчання включаючи побудову нейронних мереж в задачі визначення стадії структурно-функціонального дозрівання кардіоміоцитів.

Були розглянуті методи вирощування кардіоміоцитів з плюрипотентних стовбурових клітин, а також основні напрямки їх використання. Також були виділені стадії розвитку кардіоміоцитів за морфологічною ознакою.

Сегментація зображень відбувалася в середовищі MATLAB і дозволила прибрати артефакти із зображень, що могли зашкодити подальшому навчання. В якості методу сегментації був обраний метод Чан–Везе, що належить до регіональних методів сегментації. Він не залежить від градієнта зображення і задовільно сегментує об'єкти зі слабкими межами і є стійким до шуму за рахунок використання інформації про глобальний регіон. Алгоритм цифрової обробки зображень також включає в себе обробку гістограми яскравості зображень і їх аугментацію, що були реалізовані методами “PYTHON”. Для обробки гістограми яскравості зображень був використаний метод контрастно обмеженого адаптивного вирівнювання, що має можливість краще покращити локальний контраст зображення і зменшити шуми в результуючому зображенні. Для аугментації використовувалися методи, що запобігають альтерації поперечного Т-канальця, що є важливим параметром для правильної класифікації, а також не погіршують контрастність зображення, що є важливим параметром для виділення структури поперечного Т-канальця, а саме методи обертання та віддзеркалення зображення. Для виконання вирівнювання кількості зображень в кожному класі був використаний метод інвертування кольорів, що також не впливає на структуру поперечного Т-канальця і не сильно впливає на диференціацію аугментованих даних. Результуюча кількість навчальних даних склала 1036 на кожную стадію дозрівання кардіоміоцитів.

Для вирішення задачі класифікації стадії структурно-функціонального дозрівання кардіоміоцитів була використана згорткова нейронна мережа, що має чудову продуктивність при роботі з зображеннями. Було побудовано архітектуру згорткової нейронної мережі з ієрархічною структурою і використанням залишкового блоку, що дозволив зменшити деграційні тенденції при використанні глибокої нейронної мережі.

Навчання згорткової нейронної мережі відбувалося у п'ять ітерацій. Було показано еволюцію навчання моделі і теплові карти різних рівнів згорткових слоїв. Було виконано оцінку точності класифікації з використанням невикористаних у навчанні зображень. Результируюча точність склала 62.62%. Була проаналізована матриця помилок і зображення класів, що демонструють велику кількість взаємних помилок. На основі аналізу було виконано об'єднання класів. Зображення першого та другого класів були віднесені до початкової стадії розвитку, а третього і четвертого – до перехідної. Після об'єднання точність складає 77%.

За результатами проведеного аналізу було вказано на необхідність подальших досліджень у сфері вирощування кардіоміоцитів з плюрипотентних стовбурових клітин, щоб виділити більше оригінальних зображень кардіоміоцитів і визначити більш точні параметри класифікації. Такі висновки були зроблені на основі різниці між точністю класифікації зображень при навчанні і на нових даних, що пов'язано із великою кількістю аугментованих зображень порівняно із оригінальними. Також є необхідність в подальшому розвитку архітектури ЗНМ.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Global burden of heart failure: a comprehensive and updated review of epidemiology – Режим доступу до ресурсу: <https://pubmed.ncbi.nlm.nih.gov/35150240/>
2. Kupatt C, Horstkotte J, Vlastos GA, Pfosser A, Lebherz C, Semisch M, et al. Embryonic endothelial progenitor cells expressing a broad range of pro-angiogenic and remodeling factors enhance vascularization and tissue recovery in acute and chronic ischemia. FASEB J. 2005;19:1576–8.
3. M., Bers, D. (2001). Excitation-contraction coupling and cardiac contractile force (2nd ed.). Dordrecht: Kluwer Academic Publishers. ISBN 978-0792371588. OCLC 47659382.
4. Atala A, Lanza R (2012). Handbook of Stem Cells. Academic Press. p. 452. ISBN 978-0-12-385943-3.
5. Methods for the Derivation and Use of Cardiomyocytes from Human Pluripotent Stem Cells – Режим доступу до ресурсу: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4045647/>
6. Automatic assessment of the cardiomyocyte development stages from confocal microscopy images using deep convolutional networks – Режим доступу до ресурсу: <https://pubmed.ncbi.nlm.nih.gov/31145728/>
7. Assessment of Cardiomyocyte Contraction in Human-Induced Pluripotent Stem Cell-Derived Cardiomyocytes – Режим доступу до ресурсу: <https://academic.oup.com/toxsci/article/144/2/227/1716062?login=false>
8. Confocal Microscopy: Principles and Modern Practices – Режим доступу до ресурсу: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6961134/#:~:text=The%20basic%20principle%20of%20confocal,complete%20image%20on%20the%20detector.>

9. Pluripotent stem cells for cardiac regeneration: Overview of recent advances & emerging trends – Режим доступу до ресурсу: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3657850/>
10. Edge Based Image Segmentation Technique for Detection and Estimation of the Bladder Wall Thickness / B. Padmapriya , T. Kesavamurthi, H. Wassim Feroze // Dept. of Biomedical Engineering, PSG College of Technology, Coimbatore – 641004, India // Dept. of Electronics and Communication Engineering, PSG College of Technology, Coimbatore – 641004
11. A local region-based Chan–Vese model for image segmentation / Shigang Liu, Yali Peng // School of Computer Science, Shaanxi Normal University, Xi'an 710062, China // School of Electronics and Information Engineering, Xi'an Jiaotong University, Xi'an 710049, China
12. Nielsen, Frank; Nock, Richard (2003). "On region merging: The statistical soundness of fast sorting, with applications". 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings. Vol. 2. IEEE. pp. II:19–26. doi:10.1109/CVPR.2003.1211447. ISBN 0-7695-1900-8.
13. Chan-Vese Segmentation – Режим доступу до ресурсу: https://www.researchgate.net/publication/274120746_Chan-Vese_Segmentation
14. A Complete Guide to Data Augmentation – Режим доступу до ресурсу: <https://www.datacamp.com/tutorial/complete-guide-data-augmentation>
15. Han, D. Comparison of Commonly Used Image Interpolation Methods / Dianyuan Han // The 2nd International Conference on Computer Science and Electronics Engineering: proceedings. – 2013. – 4 p.
16. Heterogeneous Computing with OpenCL – Режим доступу до ресурсу: <https://www.sciencedirect.com/book/9780124058941/heterogeneous-computing-with-opencl>
17. Біомедичні електронні системи: конспект лекцій з Розділу 3 «Інформаційне забезпечення функціонування та виробництва біомедичних електронних систем» для студ. спец. 8.05080102 – фізична та біомедична

електроніка / Уклад.: А.О. Попов, О.С. Коваленко, Н.Г. Іванушкіна, К.О. Іванько, Є.С. Карплюк. – К., КПІ, 2015. – 49 с.

18. Гістограма яскравості зображення. – Режим доступу до ресурсу: https://studbooks.net/2250867/informatika/gistogramma_yarkosti_izobrazheniya

19. Vij, K. Enhancement of Images Using Histogram Processing Techniques / Komal Vij, Yaduvir Singh // Int. J. Comp. Tech. Appl. – Vol 2 (2). – 309-313. – ISSN:2229-6093

20. Mohri, M. Foundations of Machine Learning / Mehryar Mohri, Afshin Rostamizadeh, Ameet Talwalkar. – The MIT Press Cambridge, Massachusetts London, England, 2012. – 427 p. – ISBN 978-0-262-01825-8

21. Alpaydm, E. Introduction to Machine Learning Second Edition / Ethem Alpaydm. – The MIT Press Cambridge, Massachusetts London, England, 2010. – 579 p. – ISBN 978-0-262-01243-0

22. Vapnik V.N. The Nature of Statistical Learning Theory Second Edition / Vladimir N. Vapnik. – Springer-Verlag New York, Inc., 2000. – 334 p. – ISBN 0-387-98780-0

23. What is a neural network – Режим доступу до ресурсу: <https://www.ibm.com/topics/neural-networks>

24. Neural network models – Режим доступу до ресурсу: https://scikit-learn.org/stable/modules/neural_networks_supervised.html

25. Convolutional neural networks – Режим доступу до ресурсу: <https://www.ibm.com/topics/convolutional-neural-networks>

26. A Comprehensive Guide to Convolutional Neural Networks – Режим доступу до ресурсу: <https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53>

27. Confocal microscopy images of cardiomyocyte development stages – Режим доступу до ресурсу: <https://ieee-dataport.org/documents/confocal-microscopy-images-cardiomyocyte-development-stages>

28. Функція “drawpolygon” – Режим доступу до ресурсу:
<https://nl.mathworks.com/help/images/ref/drawpolygon.html>
29. функція “activecontour” – Режим доступу до ресурсу:
<https://nl.mathworks.com/help/images/ref/activecontour.html>
30. OpenCV Histogram Equalization and Adaptive Histogram Equalization (CLAHE) – Режим доступу до ресурсу:
<https://pyimagesearch.com/2021/02/01/opencv-histogram-equalization-and-adaptive-histogram-equalization-clahe/>
31. NumPy documentation – Режим доступу до ресурсу:
<https://numpy.org/doc/stable/>
32. sklearn.model_selection.train_test_split – Режим доступу до ресурсу:
https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html
33. Module: tf.keras – Режим доступу до ресурсу:
https://www.tensorflow.org/api_docs/python/tf/keras
34. Understanding and implementation of Residual Networks – Режим доступу до ресурсу:
<https://medium.com/analytics-vidhya/understanding-and-implementation-of-residual-networks-resnets-b80f9a507b9c>

Додаток А

Підготовка моделі згорткової нейронної мережі для подальшого навчання

Layer (type)	Output Shape	Param #	Connected to
input_1 (InputLayer)	[(None, 1024, 1024, 1)]	0	[]
conv2d (Conv2D)	(None, 1024, 1024, 2)	20	['input_1[0][0]']
batch_normalization (BatchNormalization)	(None, 1024, 1024, 2)	8	['conv2d[0][0]']
re_lu (ReLU)	(None, 1024, 1024, 2)	0	['batch_normalization[0][0]']
max_pooling2d (MaxPooling2D)	(None, 512, 512, 2)	0	['re_lu[0][0]']
conv2d_1 (Conv2D)	(None, 512, 512, 4)	76	['max_pooling2d[0][0]']
batch_normalization_1 (BatchNormalization)	(None, 512, 512, 4)	16	['conv2d_1[0][0]']
re_lu_1 (ReLU)	(None, 512, 512, 4)	0	['batch_normalization_1[0][0]']
max_pooling2d_1 (MaxPooling2D)	(None, 256, 256, 4)	0	['re_lu_1[0][0]']
conv2d_2 (Conv2D)	(None, 256, 256, 8)	296	['max_pooling2d_1[0][0]']
batch_normalization_2 (BatchNormalization)	(None, 256, 256, 8)	32	['conv2d_2[0][0]']
re_lu_2 (ReLU)	(None, 256, 256, 8)	0	['batch_normalization_2[0][0]']
max_pooling2d_2 (MaxPooling2D)	(None, 128, 128, 8)	0	['re_lu_2[0][0]']
conv2d_3 (Conv2D)	(None, 128, 128, 16)	1168	['max_pooling2d_2[0][0]']
batch_normalization_3 (BatchNormalization)	(None, 128, 128, 16)	64	['conv2d_3[0][0]']
re_lu_3 (ReLU)	(None, 128, 128, 16)	0	['batch_normalization_3[0][0]']
conv2d_4 (Conv2D)	(None, 128, 128, 32)	4640	['re_lu_3[0][0]']

Рисунок 1 – Результат компіляції та структура моделі ЗНМ

conv2d_5 (Conv2D)	(None, 128, 128, 32 2336)	['max_pooling2d_2[0][0]']
batch_normalization_4 (Batch Normalization)	(None, 128, 128, 32 128)	['conv2d_4[0][0]']
batch_normalization_5 (Batch Normalization)	(None, 128, 128, 32 128)	['conv2d_5[0][0]']
add (Add)	(None, 128, 128, 32 0)	['batch_normalization_4[0][0]', 'batch_normalization_5[0][0]']
re_lu_4 (ReLU)	(None, 128, 128, 32 0)	['add[0][0]']
max_pooling2d_3 (MaxPooling2D)	(None, 64, 64, 32) 0	['re_lu_4[0][0]']
flatten (Flatten)	(None, 131072) 0	['max_pooling2d_3[0][0]']
dense (Dense)	(None, 6) 786438	['flatten[0][0]']
softmax (Softmax)	(None, 6) 0	['dense[0][0]']
=====		
Total params: 795,350		
Trainable params: 795,162		
Non-trainable params: 188		

Рисунок 1 – Результат компіляції та структура моделі ЗНМ

Додаток Б

Результати навчання моделі ЗНМ.

```

Learning model. Iteration: 1
Epoch 1/5
81/81 [=====] - 1494s 18s/step - loss: 9.0163 - accuracy: 0.5004 - val_loss: 8.7244 - val_accuracy: 0.4568
Epoch 2/5
81/81 [=====] - 1476s 18s/step - loss: 0.4760 - accuracy: 0.8723 - val_loss: 3.4363 - val_accuracy: 0.5710
Epoch 3/5
81/81 [=====] - 1479s 18s/step - loss: 0.1152 - accuracy: 0.9622 - val_loss: 3.1336 - val_accuracy: 0.5633
Epoch 4/5
81/81 [=====] - 1477s 18s/step - loss: 0.0566 - accuracy: 0.9815 - val_loss: 1.6256 - val_accuracy: 0.7377
Epoch 5/5
81/81 [=====] - 1511s 19s/step - loss: 0.0546 - accuracy: 0.9826 - val_loss: 0.7400 - val_accuracy: 0.8241
21/21 - 118s - loss: 0.7400 - accuracy: 0.8241 - 118s/epoch - 6s/step

```

Рисунок 1 – Результати навчання моделі ЗНМ першої ітерації

```

Learning model. Iteration: 2
Epoch 1/5
81/81 [=====] - 1528s 19s/step - loss: 1.6098 - accuracy: 0.7415 - val_loss: 27.2182 - val_accuracy: 0.2083
Epoch 2/5
81/81 [=====] - 1491s 18s/step - loss: 0.1267 - accuracy: 0.9618 - val_loss: 5.8514 - val_accuracy: 0.4583
Epoch 3/5
81/81 [=====] - 1584s 20s/step - loss: 0.0228 - accuracy: 0.9927 - val_loss: 0.3129 - val_accuracy: 0.8843
Epoch 4/5
81/81 [=====] - 1525s 19s/step - loss: 0.0062 - accuracy: 0.9985 - val_loss: 0.1165 - val_accuracy: 0.9599
Epoch 5/5
81/81 [=====] - 1645s 20s/step - loss: 0.0038 - accuracy: 0.9992 - val_loss: 0.0835 - val_accuracy: 0.9630
21/21 - 118s - loss: 0.0835 - accuracy: 0.9630 - 118s/epoch - 6s/step

```

Рисунок 2 – Результати навчання моделі ЗНМ другої ітерації

```

Learning model. Iteration: 3
Epoch 1/5
81/81 [=====] - 1501s 18s/step - loss: 0.8894 - accuracy: 0.7693 - val_loss: 12.7803 - val_accuracy: 0.3040
Epoch 2/5
81/81 [=====] - 1547s 19s/step - loss: 0.1552 - accuracy: 0.9498 - val_loss: 7.0334 - val_accuracy: 0.3380
Epoch 3/5
81/81 [=====] - 2062s 26s/step - loss: 0.0576 - accuracy: 0.9846 - val_loss: 1.4126 - val_accuracy: 0.5247
Epoch 4/5
81/81 [=====] - 2088s 26s/step - loss: 0.0161 - accuracy: 0.9985 - val_loss: 0.2447 - val_accuracy: 0.8873
Epoch 5/5
81/81 [=====] - 2173s 27s/step - loss: 0.0091 - accuracy: 0.9996 - val_loss: 0.1743 - val_accuracy: 0.9321
21/21 - 117s - loss: 0.1743 - accuracy: 0.9321 - 117s/epoch - 6s/step

```

Рисунок 3 – Результати навчання моделі ЗНМ третьої ітерації

```

Learning model. Iteration: 4
Epoch 1/5
81/81 [=====] - 1505s 19s/step - loss: 0.4234 - accuracy: 0.8692 - val_loss: 2.0380 - val_accuracy: 0.6312
Epoch 2/5
81/81 [=====] - 1528s 19s/step - loss: 0.0551 - accuracy: 0.9861 - val_loss: 0.6301 - val_accuracy: 0.8025
Epoch 3/5
81/81 [=====] - 1498s 19s/step - loss: 0.0213 - accuracy: 0.9946 - val_loss: 0.1103 - val_accuracy: 0.9583
Epoch 4/5
81/81 [=====] - 1497s 19s/step - loss: 0.0099 - accuracy: 0.9977 - val_loss: 0.2448 - val_accuracy: 0.9105
Epoch 5/5
81/81 [=====] - 1508s 19s/step - loss: 0.0185 - accuracy: 0.9946 - val_loss: 0.5513 - val_accuracy: 0.7546
21/21 - 123s - loss: 0.5513 - accuracy: 0.7546 - 123s/epoch - 6s/step

```

Рисунок 4 – Результати навчання моделі ЗНМ четвертої ітерації

```

Learning model. Iteration: 5
Epoch 1/5
81/81 [=====] - 1513s 19s/step - loss: 0.6792 - accuracy: 0.7955 - val_loss: 6.4171 - val_accuracy: 0.4799
Epoch 2/5
81/81 [=====] - 1880s 23s/step - loss: 0.0747 - accuracy: 0.9792 - val_loss: 2.2028 - val_accuracy: 0.5972
Epoch 3/5
81/81 [=====] - 1618s 20s/step - loss: 0.0162 - accuracy: 0.9985 - val_loss: 0.7882 - val_accuracy: 0.8071
Epoch 4/5
81/81 [=====] - 1870s 23s/step - loss: 0.0075 - accuracy: 0.9996 - val_loss: 0.0688 - val_accuracy: 0.9722
Epoch 5/5
81/81 [=====] - 1689s 21s/step - loss: 0.0091 - accuracy: 0.9988 - val_loss: 0.0670 - val_accuracy: 0.9784
21/21 - 116s - loss: 0.0670 - accuracy: 0.9784 - 116s/epoch - 6s/step

```

Рисунок 5 – Результати навчання моделі ЗНМ п'ятої ітерації