Київський національний університет імені Тараса Шевченка
факультет комп'ютерних наук та кібернетики

Випускна кваліфікаційна робота бакалавра
на тему:

"Розпізнавання раку молочної залози за допомогою аналізу інтерфазних ядер букального епітелію"

Студента 4 курсу
кафедри обчислювальної математики
Чан Ха Ву

Науковий керівник
професор, доктор фізико-математичних наук
Клюшин Дмитро Анатолійович

Робота заслухана на засіданні кафедри обчислювальної математики та рекомендована до захисту в ДЕК, протокол № 11 від 18 травня 2017 р.

Завідувач кафедри обчислювальної математики проф. Ляшко С.І.

Київ 2017

Анотація

У зв'язку зі стрімким зростанням кількості випадків захворювань раком молочної залози, проблема неінвазивної діагностики цієї хвороби є дуже актуальною. Новітні дослідження в області онкології [1] виявили закономірність, яка полягає в тому, що існує зв'язок між змінами у інтерфазному ядрі букального епітелію та розвитком злоякісної або доброякісної пухлини (раком молочної залози та фіброаденоматозом відповідно). В роботі досліджується цей зв'язок за допомогою технік комп'ютерного зору, фрактальної геометрії та машинного навчання.

Ця робота складається з трьох частин. У першій частині пропонується алгоритм сегментації інтерфазних ядер букального епітелію на зображенні. У другій частині використовується фрактальний аналіз інтерфазних ядер букального епітелію, щоб відрізнити здорових пацієнтів від хворих раком молочної залози або фіброаденоматозом. У третій частині використовуються методи глибинного навчання для класифікації зразків (зображень) на групи хворих раком молочної залози та хворих фіброаденоматозом.

Подяки

Хочу виразити щиру подяку своєму науковому керівнику, професору та доктору ф-м наук Клюшину Дмитрові Анатолійовичу, хто надав мені можливість займатися цим неймовірно цікавим дослідженням.

Моїй матері, яка успішно переборола рак молочної залози у 2011-му році та завжди мене підтримувала в навчанні.

Та Анні Разумовій, що допомогла мені в оформленні тексту цієї роботи.

Зміст

Анотація i

Подяка ii

1 Вступ 1
1.1 Метод дослідження . . . 1
1.2 Опис даних . . . 1

2 Попередня обробка даних 2
2.1 Сегментація ядер . . . 2
2.1.1 Зменшення шуму. Медіанна фільтрація. . . . 3
2.1.2 Відділення фону. Бінаризація Оцу. . . . 4
2.1.3 Бінарні морфологічні операції над зображеням. . . . 5
2.1.4 Трансформація дистанції . . . 7
2.1.5 Алгоритм водоподілу . . . 8
2.1.6 Кінцевий алгоритм сегментації . . . 9
2.2 Нормалізація зображень . . . 10
2.3 Фільтр на основі еліпсоїдів Петуніна . . . 11
2.3.1 Еліпсоїди Петуніна . . . 11
2.3.2 Фільтр зображень на основі еліпсоїдів Петуніна . . . 12
2.3.3 Швидкий алгоритм на основі динамічного програмування . . . 12

3 Фрактальний аналіз 15
3.1 Фрактальна характеристика . . . 15
3.1.1 Теоретичні відомості . . . 15
3.1.2 Алгоритм box-counting . . . 16
3.1.3 Лінійна регресія . . . 16
3.2 Класифікація пацієнтів . . . 18
3.2.1 Міра близькості та p-статистика . . . 18
3.2.2 Метод k найближчих сусідів. Кросвалідація. . . . 18
3.3 Результати та спостереження . . . 19

4 Глибинне навчання 21
4.1 Нейронні мережі . . . 21
4.1.1 Багатошаровий перцептрон . . . 22
4.1.2 Функція активації ReLU . . . 22
4.1.3 Згорткові нейронні мережі . . . 22
4.1.4 Алгоритм RMSProp . . . 23
4.1.5 Регуляризація нейронної мережі . . . 24
4.1.6 Функція втрат. Перехрестна ентропія. . . . 24
4.2 Класифікація за допомогою нейронних мереж . . . 25
4.2.1 Прирощення даних . . . 25
4.2.2 Проблема незбалансованості вибірки для тренування . . . 25

4.2.3 Архітектура мережі 26
4.2.4 Тренування мереж та результати 27
4.2.5 Аналіз результатів 28
4.3 Підвищення якості розпізнавання 28
4.3.1 Гибрідний класифікатор 29
4.3.2 Аналіз результатів 30

5 Висновок 31
5.1 Поради для подальших досліджень 31

A Програмний пакет 32

Bibliography 33

Розділ 1

Вступ

Незважаючи на стрімкий розвиток сучасної онкології, захворюваність на рак молочної залози як в Україні, так і в більшості розвинених країн світу продовжує зростати. Щороку у світі реєструють більше п'яти тисяч нових випадків захворювання на рак молочної залози, що становить понад 25% всіх ракових захворювань у жінок. Тому задача неінвазивної діагностики раку молочної залози є надзвичайно актуальною.

Обробка цифрових зображень давно стала складовою досліджень практично у всіх галузях науки. Робота зі зразками з будь-якої предметної області має на увазі не тільки витяг даних із зображень, але і класифікацію знімків, роботу зі складноструктурованими зразками, з неочевидними закономірностями і особливостями, часто помітними лише фахівцям у цій галузі. У медицині можливість автоматично обробляти та розпізнавати великі набори знімків мікроскопа певної тематики може значно вплинути на хід досліджень, полегшити процес роботи із зображеннями та у результаті, наприклад, прискорити виявлення хвороби та постановку діагнозу, що допоможе підібрати своєчасне і необхідне лікування.

1.1 Метод дослідження

Як показано у попередніх роботах в області онкології [1], злоякісні та доброякісні утворення певним чином впливають на інтерфазні ядра букального епітелію. Ми сподіваємось що використовуючи методи комп'ютерного зору, фрактального аналізу та машинного навчання, можна діагнозувати рак молочної залози та фіброаденоматоз (доброякісна пухлина) з певною точністю.

1.2 Опис даних

Для дослідження було використано набір зображень інтерфазних ядер букального епітелію, отриманих від здорових пацієнтів, а також хворих раком молочної залози та фіброаденоматозом, які лікувались у інституті онкології. Вхідний набір даних для складається з знімків 6752 інтерфазних ядер букального епітелію, для кожного було зроблено 3 знімки мікроскопу (сканограми оптичної щільності): без фільтру, через жовтий фільтр та через пурпурного фільтру (отже всього 20256 фотографії), взятого з 130 пацієнтів, з них 68 хворих раком, 29 здорових та 33 хворих фіброаденоматозом.

Розділ 2

Попередня обробка даних

2.1 Сегментація ядер

Мікроскопічні зображення у чистому вигляді зазвичай неприйнятні для аналізу у зв'язку з наявністю дефектів, цифрових шумів, спричинених необхідністю підвищення світлочутливості фотоматеріалів, а також сторонніх об'єктів. Отже, для подальшого аналізу вхідного набору зображень (або набору пацієнтів), необхідно зробити попередню обробку та сегментацію.

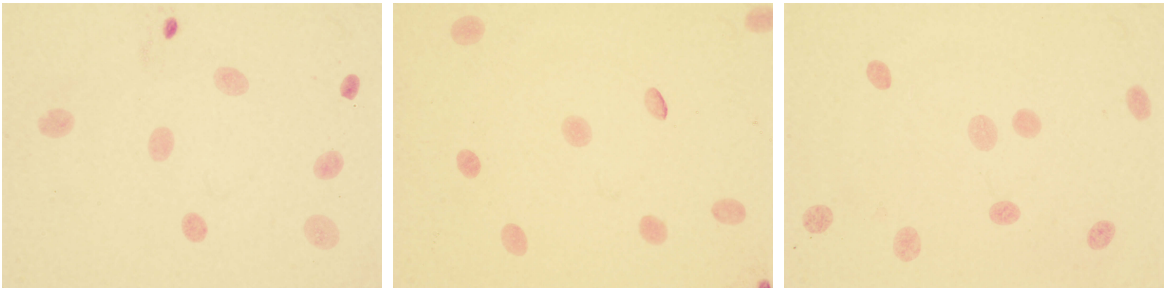
На (Мал. 2.1) зображено приклади знімків інтерфазних ядер букального епітелію. З них треба виділити окремі клітини для подальшого аналізу. Задача обробки та сегментації зображень розбивається на наступні підзадачі:

1. Зменшення шуму
2. Відділення фону
3. Виділення окремого ядра клітини

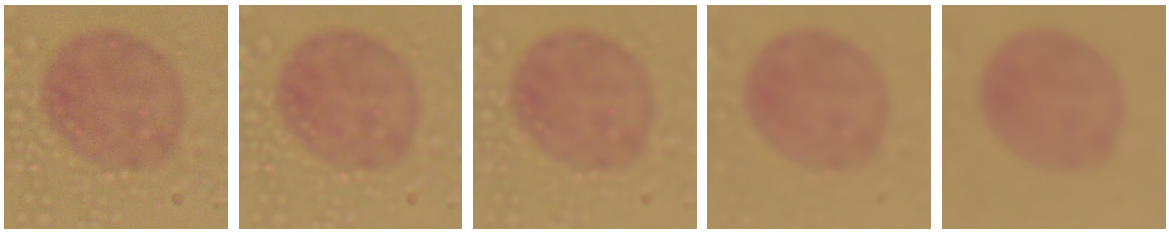
Перша та друга підзадачі стандартно розв'язуються за допомогою алгоритмів фільтрації зображень, а друга підзадача розв'язується за допомогою алгоритму бінаризації Оцу.

У загальному випадку, якщо на зображенні присутні різні об'єкти (у даному випадку – ядра різних типів клітин чи різні клітини), третя підзадача дуже складно розв'язується детермінованими алгоритмами, оскільки об'єкти можуть перекриватися, торкатися один одного та мати абсолютно різні форми, кольори та розміри. Так, наприклад, [2] використав нейронні мережі для класифікації та сегментації зображень клітин.

У нашому випадку, на досліджуваних зображеннях присутні тільки інтерфазні ядра букального епітелію. Отже, ми припускаємо, що радіуси цих ядер приблизно однакові та



Мал. 2.1: Зображення інтерфазних ядер букального епітелію, які треба сегментувати.



Оригінал | При $k = 3$ | При $k = 5$ | При $k = 9$ | При $k = 13$

Мал. 2.2: Дія медіанного фільтру на зображення.

можемо використовувати алгоритми на основі морфологічних операцій, трансформації відстані та алгоритму водоподілу, які будуть детально розглянуті далі.

2.1.1 Зменшення шуму. Медіанна фільтрація.

Для зменшення рівня шуму знімків мікроскопа будемо використовувати медіанний фільтр – один з видів цифрових фільтрів, запропонований [3], що широко застосовується в цифровій обробці сигналів та зображень. Його використання дає найкращі результати для збереження перепадів відтінків, різноманітних кордонів та локальних піків яскравості на спотворених імпульсним шумом зображеннях. Швидкий алгоритм медіанної фільтрації виглядає наступним чином:

Алгоритм 1: Медіанна фільтрація

Вхід : Зображення X розміром $m \times n$, радіус ядра (вікна фільтру) r
Вихід: Зображення Y такого ж розміру

```
Ініціалізація гістограму ядра H;
for i = 1 to m do
    for j = 1 to n do
        for k = −r to r do
            Вилучити X_{i+k,j−r−1} від H;
            Додати X_{i+k,j+r} до H;
        Y_{i,j} ← медіана(H);
```

На (Мал. 2.2) показано, як змінюється рівень шуму та чіткість зображення в залежності від розміру вікна фільтру $k = 2r + 1$. Можна побачити, що при $k = 3$, зображення мінімально позбавлене шуму, а текстура ядра майже повністю зберігається. При $k = 13$ дуже гарно виділяються контури ядра клітини та зображення зовсім позбавлене шуму, але майже повністю втрачено текстуру ядра.

Отже, на різних етапах аналізу можна використовувати медіанні фільтри з різними розмірами вікна фільтру. Наприклад, при сегментації зображень (Мал. 2.1) доречно використовувати вікна фільтру з великими розмірами, оскільки на цьому етапі нас більше цікавлять контури об'єктів, а при класифікації (здоровий, хворий раком або фіброаденоматозом) краще використовувати якомога менші вікна фільтру, оскільки нас цікавить текстура інтерфазного ядра клітини букального епітелію.

2.1.2 Відділення фону. Бінаризація Оцу.

Наступний етап обробки зображення – відділення фонових пікселів від пікселів об'єктів, наявних на зображенні (ядер клітин або сторонніх об'єктів). Для досягнення цієї мети підходить алгоритм бінаризації зображення Оцу [4].

Нехай пікселі зображення приймають значення у L рівнях сірого кольору, тобто значення з множини $\{1, 2, \ldots, L\}$, кількість пікселів зі значенням i дорівнює n_i, а кількість пікселів у зображенні – N. Треба розділити множину пікселів на два класи C_0 та C_1 (фонові пікселі та пікселі об'єктів, або навпаки) за допомогою деякого значення порогу k. До C_0 віднесемо усі пікселі зі значеннями з $\{1, 2, \ldots, k\}$, а до C_1 – усі пікселі зі значеннями з множини $\{k+1, k+2, \ldots, L\}$. Нехай ω_0 та ω_1 – частота класів C_0 та C_1 відповідно.

$$\omega_0 = \omega_0(k) = \sum_{i=0}^{k} \frac{n_i}{N}, \quad \omega_1 = \omega_1(k) = \sum_{i=k+1}^{k} \frac{n_i}{N} = 1 - \omega_0$$

оді метод Оцу полягає у пошуку порогу k, який мінімізує дисперсію в середині класу, яка визначається як зважена сума дисперсій двох класів:

$$\sigma_\omega^2(k) = \sigma_0^2(k)\omega_0(k) + \sigma_1^2(k)\omega_1(k) \to \min$$

Де $\sigma_0^2(k)$ та $\sigma_2^2(k)$ – дисперсія класів C_0 та C_1 для порогу k відповідно. У своїй роботі Оцу також показав, що мінімізація дисперсії в середині класу рівносильна максимізації дисперсії між класами:

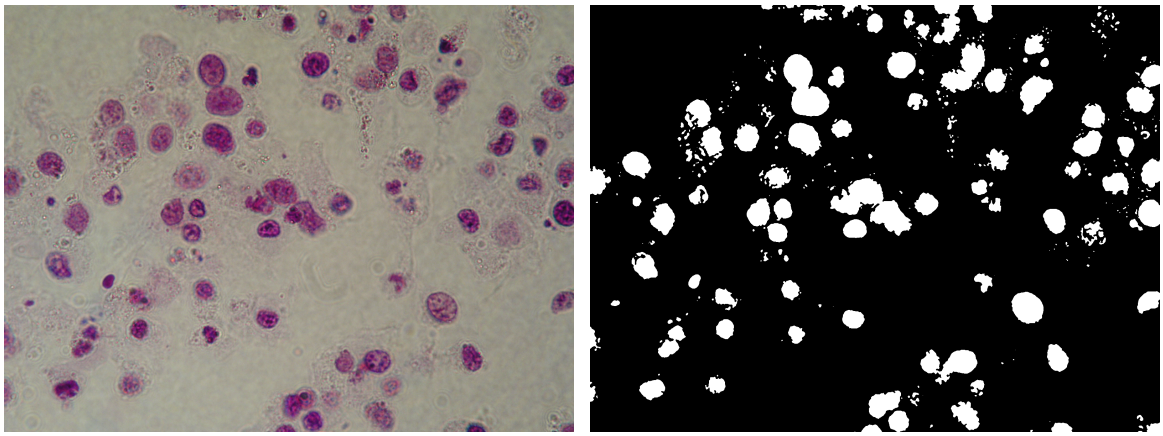
$$\sigma_b^2(k) = \omega_0(k)\omega_1(k)\left(\mu_0(k) - \mu_1(k)\right)^2 \to \max$$

Де $\mu_0(k)$ та $\mu_1(k)$ – середнє арифметичне класів C_0 та C_1 при порозі k відповідно, тобто:

$$\mu_T = \frac{\sum_{i=0}^{L} i \cdot n_i}{N}$$

$$\mu_0(k) = \frac{\sum_{i=0}^{t-1} i \cdot n_i}{N \cdot \omega_0(k)}, \quad \mu_1(t) = \frac{\mu_T - \mu_0(k) \cdot \omega_0(k)}{\omega_1(k)}.$$

Значення $\omega_0(k+1)$, $\omega_1(k+1)$, $\mu_0(k+1)$ та $\mu_1(k+1)$ достатньо легко виражаються через значення $\omega_0(k)$, $\omega_1(k)$, $\mu_0(k)$ та $\mu_1(k)$, що дозволяє швидко обчислити оптимальний поріг



Мал. 2.3: Результат бінаризації.

k. Отже, алгоритм бінаризації Оцу виглядає наступним чином:

Алгоритм 2: Бінаризація зображення Оцу

Вхід : Зображення X розміром $m \times n$
Вихід: Поріг k

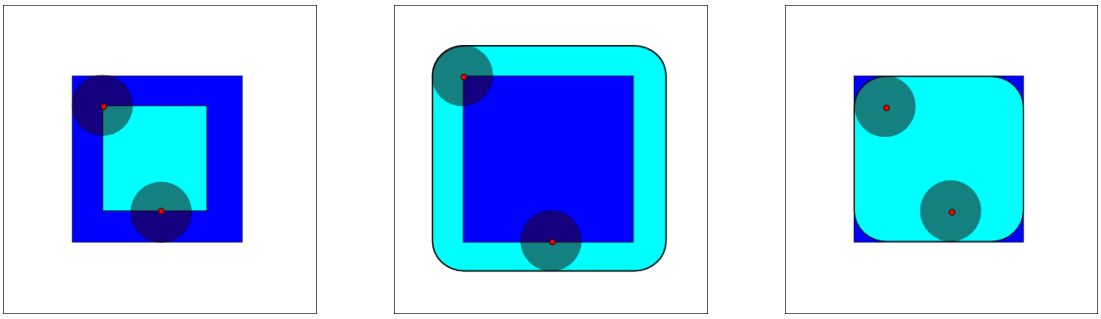
обчислити гістограму n_i зображення та частоту $N(i) = \frac{n_i}{N}$ для кожного рівня інтенсивності зображення X;
обчислити початкові значення $\omega_0(0)$, $\omega_1(0)$, $\mu_0(0)$ та $\mu_1(0)$;
поставити $\sigma^2_{\max} \leftarrow 0$;
for $k = 1$ to L do
  Оновити значення $\omega_0(k)$, $\omega_1(k)$, $\mu_0(k)$ та $\mu_1(k)$;
  Обчислити $\sigma_b^2(k) = \omega_0(k)\omega_1(k)\left(\mu_0(k) - \mu_1(k)\right)^2$;
  if $\sigma_b^2(k) > \sigma^2_{\max}$ then
    поставити $\sigma^2_{\max} \leftarrow \sigma_b^2(k)$
поставити $k \leftarrow \sigma^2_{\max}$;

На зображенні (Мал. 2.3) результат бінаризації фільтрованого медіанним фільтром зображення з розміром $k = 5$. Негладкі контури інтерфазних ядер клітин після бінаризації, а також неточності у деяких регіонах зображення нас не турбують на даному етапі, оскільки це можна виправити морфологічними операціями, які будуть детально розглянуті далі.

2.1.3 Бінарні морфологічні операції над зображеням.

У бінарній морфології, зображення розглядається як підмножина Евклідового простору $\mathbb{R}^d$ або цілочисельної сітки $\mathbb{Z}^d$, де d – розмірність простору. Операції морфології застосовуються до зображень разом з заданим структурним елементом, або ядром, яке визначає природу операції. Для обробки зображень найчастіше використовують наступний структурний елемент:

$$B = \{(-1,-1), (-1,0), (-1,1), (0,-1), (0,0), (0,1), (1,-1), (1,0), (1,1)\}.$$



Звуження $\ominus$ Розширення $\oplus$ Відкриття $\circ$

Мал. 2.4: Морфологічні операції з круглим структурним елементом над синім квадратом.

Двома основними морфологічними операціями є Erosion(звуження) і Dilation(розширення). Операція звуження зображення задається наступним чином:

$$A \ominus B = \left\{ z \in \mathbb{Z}^d | B_z \subseteq A \right\} ,$$

де A – зображення, B – структурний елемент, B_z – трансляція вектору z, тобто

$$B_z = \{b + z | b \in B\} , \quad \forall z \in \mathbb{Z}^d$$

Ідея цієї операції полягає в наступному. Ядро (структурний елемент) ковзає по зображенню. Піксель в бінаризованому зображенні (1 або 0) буде вважатися 1, тільки якщо всі пікселі під ядром рівні 1, в іншому випадку він буде зруйнований (стане рівним нулю). Таким чином, в результаті операції всі пікселі, що знаходяться на краю, будуть відкинуті в залежності від розміру ядра. Таким чином, товщина або розмір об'єкта переднього плану зменшується. Цей метод може використовуватися для видалення невеликих шумів, роз'єднання двох пов'язаних об'єктів та іншого.

Операція розширення зображення задається наступним чином:

$$A \oplus B = \bigcup_{b \in B} A_b \, .$$

Ця операція протилежна ерозії. Тут піксель бінаризованого зображення дорівнює "1", якщо принаймні один піксель під ядром дорівнює "1". Таким чином, збільшується біла область зображення або збільшується розмір об'єкта переднього плану. Зазвичай у випадках, коли потрібно видалити шум навколо об'єкта, спочатку використовують операцію звуження, а потім операцію розширення. Розширення необхідне, оскільки ерозія, крім видалення білого шуму, скорочує наш об'єкт.

На основі основних операцій звуження ($\ominus$) та розширення ($\oplus$) задаються інші операції. Операція морфологічного відкриття (opening) задається наступним чином:

$$A \circ B = (A \ominus B) \oplus B \, .$$

або

$$A \circ B = \bigcup_{B_z \subseteq A} B_z .$$

У [5] більш детально розглянуто теоретичні та практичні аспекти операції математичної морфології.

Позначимо зображення, отримане у результаті розширення бінарного зображення за 2 ітерації у (2.5) як $zoneOfinterest$. Очевидно, що всі пікселі, які дорівнюють 0 у $sureBG$ є пікселями фону. Отже, ядра треба шукати тільки серед тих точок $zoneOfinterest$, які дорівнюють 1 (позначено білим кольором). Позначимо також відкриття у (Мал. 2.5) як $threshOpening$.

2.1.4 Трансформація дистанції

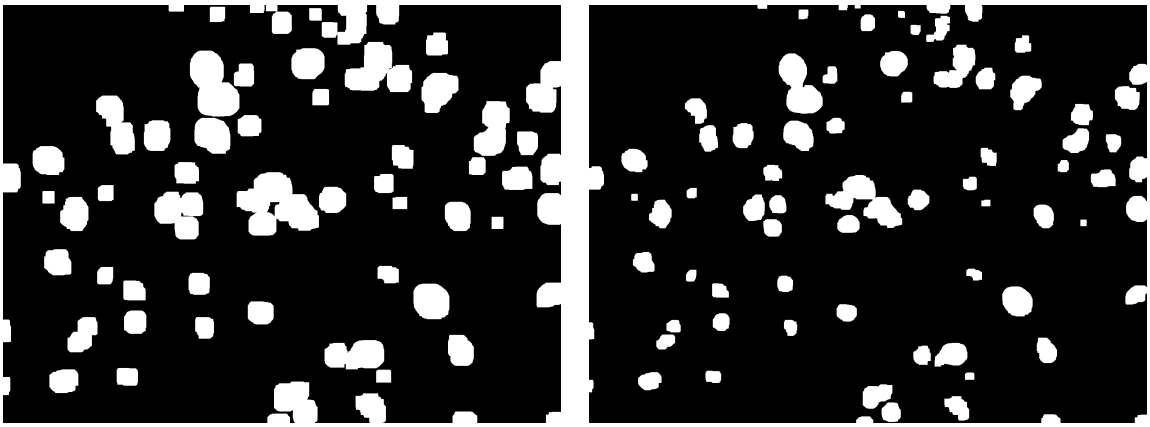
Трансформація дистанції (Distance Transformation) є також операцією математичної морфології. Кожному пікселю переднього плану (що належить E) бінарного зображення $E \subseteq \mathbb{Z}^d$ ставиться у відповідність відстань цієї точки до найближчого пікселя заднього плану (що не належить E). Для бінарних зображень найчастіше використовують Евклідову відстань, тобто для зображення $E \subseteq F = \{1, 2, \ldots, height\} \times \{1, 2, \ldots, width\}$, кожній точці $z = (z_x, z_y) \in E$ ставимо у відповідність:

$$DT(z) = \inf_{(x,y)\in F} (|x - z_x| + |y - z_y|)$$

Результат цієї операції для $E = threshOpening$ можна інтерпретувати як показник вірогідності, що піксель $z = (z_x, z_y)$ належить інтерфазному ядру клітини букального епітелію. Отже, можна задати емпіричне правило, за яким будемо приймати певні пікселі як “точно належать ядру клітини”:

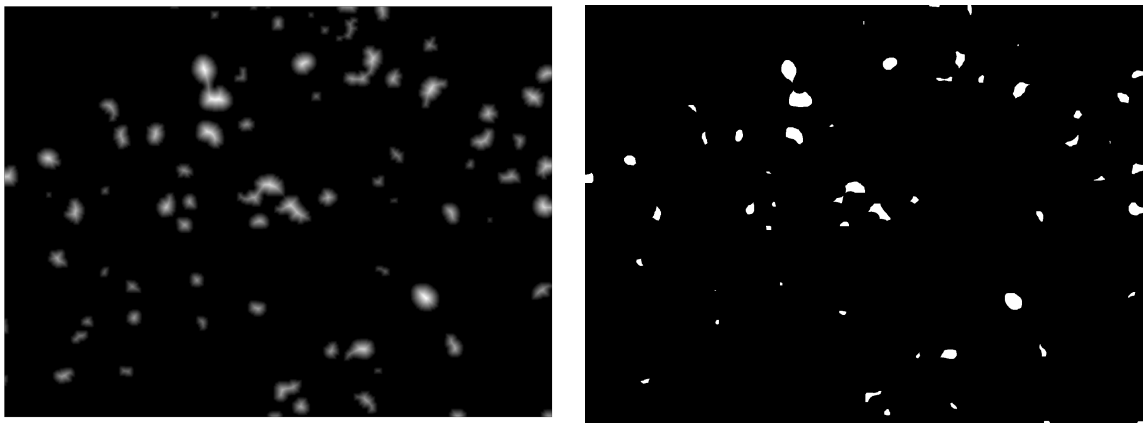
$$DT(z) \geq \alpha \cdot \sup_{s\in E} DT(s) \Rightarrow \text{“z точно належать ядру клітини”}. \tag{2.1}$$

Позначимо множину, що задовольняє цю нерівність, як $sureFG$. Множник $\alpha = 0.475$ був обраний експериментальним чином, на досліджуваних даних. Тут також використовується припущення, що усі ядра клітин мають приблизно однаковий розмір, а отже, центри цих клітин будуть знаходитись на приблизно однакому відстані від точок заднього плану. Для інших задач, дані яких зібрані за інших умов, слід обрати інший множник.



Розширене зображення за 2 ітерації Відкриття зображення за 2 ітерації

Мал. 2.5: Морфологічні операції над бінарним зображенням (Мал. 2.3).



Трансформація дистанції $sureFG$

Мал. 2.6: Трансформація дистанції.

У (2.6) показано результат трансформації відстані та обчислення $sureFG$ над бінарним зображенням $threshOpening$, зображеним на (2.5). На $sureFG$ ядра не дотикаються, завдяки обраному параметру α, тобто наш метод сегментації коректно обробляє ситуації, коли клітини торкаються чи частково перекриваються.

Використовуючи алгоритм що запропонований у [6] який використовує так звану рекурсивну морфологію, можна обчислити трансформацію дистанції за два обходу. Для повного виявлення окремих клітин, використаємо алгоритм водоподілу.

2.1.5 Алгоритм водоподілу

Трансформація водоподілу (Watershed) розглядає одноканальне зображення як топографічну карту, де значення пікселя відображає висоту цієї точки.

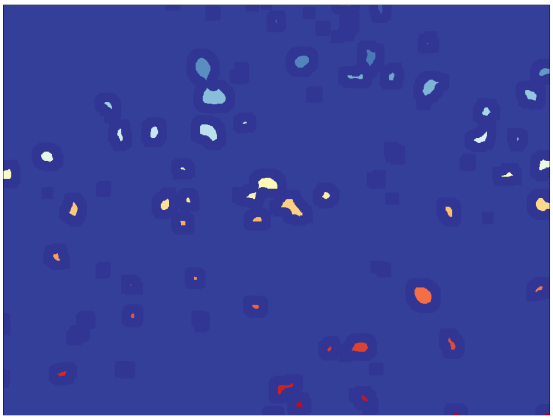
Починаючи з точок локальних мінімумів (або з заданих точок), кожна "яма" заповнюється рідиною різних кольорів. З підвищенням рівня рідини, настає момент, коли дві рідини різного кольору починають змішуватися. Щоб уникнути цього, будуються границі між регіонами цих рідин. Границі будуються, доки рівень рідини не стане вищим за найвищу точку карти. Побудовані границі є результатами сегментації.

Використаємо цю ідею для сегментації ядер клітин. Кожну зв'язну компоненту $sureFG$ пофарбуємо в певний "колір". Назвемо таку сегментацію "набором маркерів". Позначимо

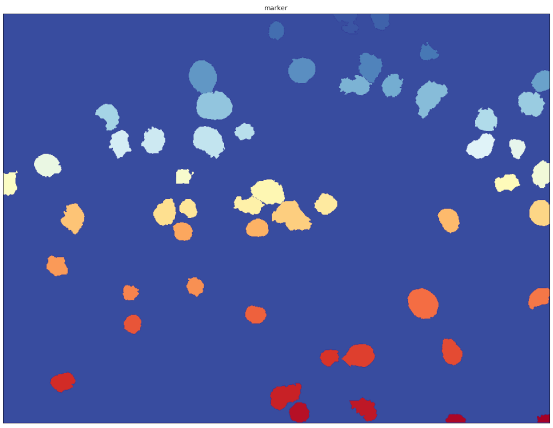
$$sureBG = E \setminus zoneOfInterest$$

Зв'язну множину пікселів заднього плану $sureBG$ також пофарбуємо в якийсь колір. Непофарбованою залишається така множина:

$$unknownRegion = zoneOfInterest \setminus sureFG$$



Маркери



Кінцевий результат сегментації

Мал. 2.7: Результат роботи алгоритму водоподілу (watershed).

Тоді алгоритм водоподілу з заданими маркерами наступний [7]:

Алгоритм 3: Алгоритм водоподілу Майера з заданими маркерами

Вхід : Зображення X розміром $m \times n$, набір маркерів
Вихід: Сегментація

0. Будемо вважати, що “затоплення” починається саме с пікселів маркерів.;
1. Сусідні пікселі кожного маркеру додаємо в чергу з пріоритетом, де рівень пріоритету залежить від величини градієнту пікселя.;
2. Розглядається піксель з найменшим рівнем пріоритету. Якщо сусідні пікселі вже пофарбовані та мають однаковий колір, то фарбуємо цей піксель у колір сусідів. Усі сусіди, які ще не пофарбовані та ще не належать до черги з пріоритетом, додаються у чергу.;
3. Повторюємо крок 2, поки черга не стане пустою.;
4. Незафарбовані пікселі є границями водоподілу.;

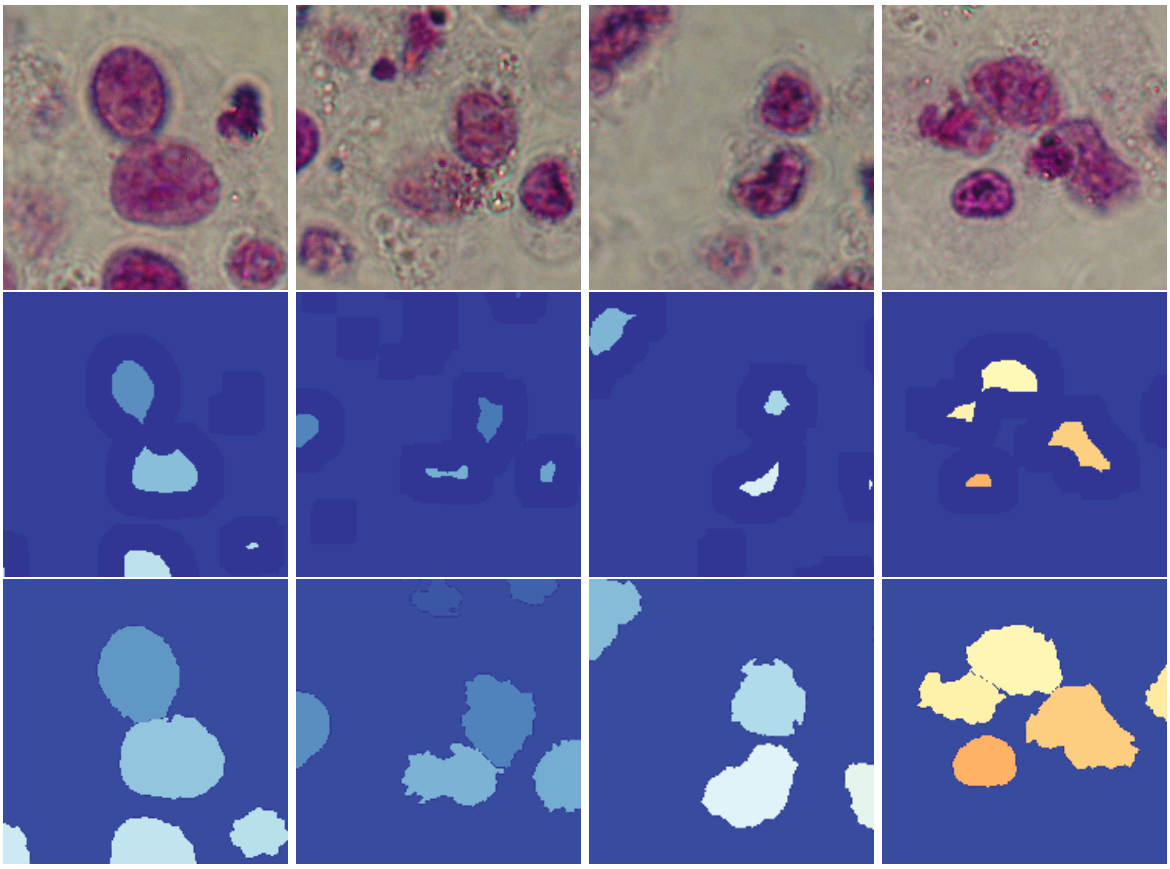
2.1.6 Кінцевий алгоритм сегментації

Отже, кінцевий алгоритм сегментації інтерфазних ядер букального епітелію виглядає наступним чином:

Алгоритм 4: Сегментація ядер

Вхід : Зображення X розміром m
Вихід: Зображення Y такого ж розміру, де різні ядра зображені різними кольорами

$gray \leftarrow$ сіре зображення X (grayscale);
$blured \leftarrow$ медіанна фільтрація $gray$ з вікном радіусом $r = 2$;
$thresh \leftarrow$ бінаризація Оцу для $blured$;
$threshOpening \leftarrow$ морфологічне відкриття $thresh$ з ядром 9×9 за 2 ітерації;
$zoneOfinterest \leftarrow$ морфологічне розширення $opening$ з ядром 9×9 за 2 ітерації;
$distTransform \leftarrow$ трансформація відстані для $threshOpening$;
$sureFG \leftarrow z$, які задовольняють формулу (2.1);
$unknownRegion \leftarrow zoneOfinterest \setminus sureFG$;
$markers \leftarrow$ зв’язні компоненти $sureFG \cup sureBG$;
Виконати алгоритм водоподілу.



Мал. 2.8: Випадки, коли ядра торкаються чи частково перекриваються.

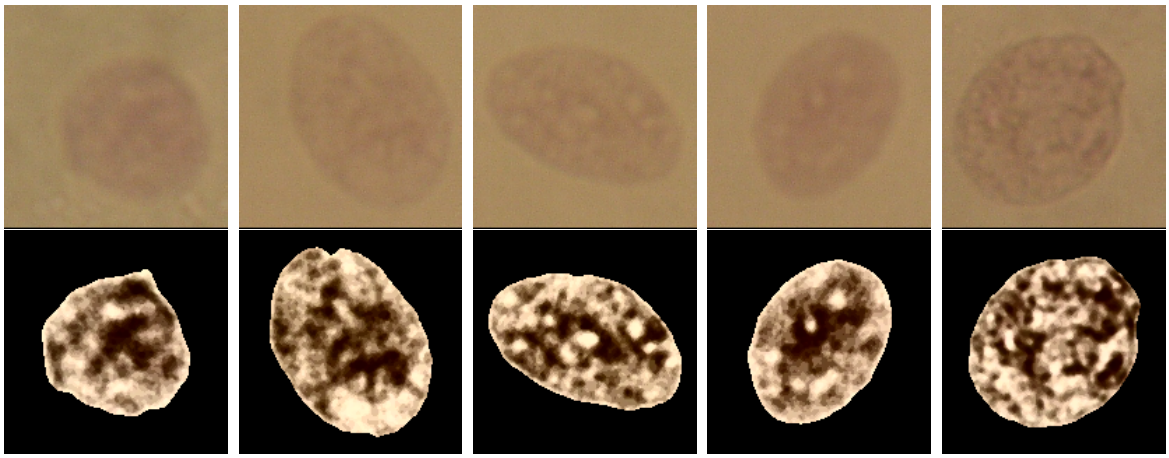
На (Мал. 2.8) бачимо, що наш алгоритм коректно сегментує також ті випадки, коли ядра клітин торкаються одне одного.

2.2 Нормалізація зображень

Для нейронної мережі важливо переконатися, що вхідні дані нормалізовані. Для випадку зображень інтерфазних ядер букального епітелію, зображення можуть бути отримані у трохи різних умовах. Це може вплинути на швидкість тренування нейронної мережі, а також на точність розпізнавання.

Задньому плану зображення присвоюємо значення 0, щоб воно не активувало нейрони у мережі. Виконуємо нормалізацію гістограми для зображення одного ядра клітини так, щоб пікселі ядра приймали значення від 0 до 255. Це також може надати прискорення тренуванню нейронної мережі, оскільки більший діапазон значень означає більшу різницю градієнтів у різних точках зображення.

Використовуючи попередні результати, видалимо задній план із зображення. Ядро клітини нормалізуємо за допомогою еквалізації гістограми, результат показано на (Мал. 2.9).



Мал. 2.9: Нормалізація зображень.

2.3 Фільтр на основі еліпсоїдів Петуніна

Згідно з роботою [8], фільтр зображень на основі використання еліпсоїдів Петуніна має багато переваг над медіанним фільтром. У випадку фільтрації зображень інтерфазних ядер клітин, цей метод краще зберігає форму текстури. Отже, його можна використовувати для попередньої обробки даних для фрактального аналізу.

2.3.1 Еліпсоїди Петуніна

Нехай маємо множину точок $M = \{x_1, x_2, \ldots, x_n\}$, де $x_i \in \mathbb{R}^m$ – множина точок у m-вимірному просторі. Знайдемо точки x_l, x_k такі, щоб

$$\sup_{x_i, x_j \in M} \|x_i - x_j\| = \|x_l - x_k\|$$

тобто відрізок $L = x_l x_k$ – діаметр множини точок M. Повернемо і перенесемо систему координат так, щоб діаметр L належав $O_{x'_1}$, де $x'_1, x'_2, \ldots, x'_n$ – координати множини M у новій системі координат.

Побудуємо найменший прямокутний паралелепіпед у $\mathbb{R}^m$, який містив би елементи множини $M' = \{x'_1, x'_2, \ldots, x'_n\}$. Стискаючи цей найменший прямокутний паралелепіпед, відобразимо його точки у гіперкуб. Знайдемо центр x_0 гіперкуба та обчислимо відстані $r_1, r_2, \ldots, r_n$ від нього до кожної точки.

Знайдемо найбільше $R = \max(r_1, r_2, \ldots, r_n)$ та побудуємо гіперкулю навколо центру x_0 з радіусом R. Зробимо обернене перетворення розтягу, повороту та переносу, отримаємо еліпсоїд Петуніна у m-вимірному просторі для множини M. Система еліпсоїдів, отриманих з кіл радіусами r_i, де $i \in 1, \ldots n$ називається концентрованою системою еліпсоїдів Петуніна для множини M.

Для системи концентрованих еліпсоїдів також справжується властивості, що є наслідками роботи Хілла [9] та детально описані та доведені у [10].

2.3.2 Фільтр зображень на основі еліпсоїдів Петуніна

Нехай маємо на вході зображення, кожен піксель якого має d каналів. Зображення розглядається як відображення $M : \mathbb{Z}^{\nvDash} \to \mathbb{R}^d$, тобто кожній координаті присвоюється d-канальне значення. Для кожного пікселя зображення, розглядаємо ще декілька пікселів, координати яких знаходяться в радіусі r від цього пікселя. Побудуємо для множини значень цих пікселів концентровану систему еліпсоїдів Петуніна, та у вихідному зображенні присвоїмо пікселю з тими ж координатами значення, яке відповідає внутрішньому (або центральному) еліпсоїду.

Детальний алгоритм виглядає наступним чином:

Алгоритм 5: Фільтр Петуніна

Вхід : Зображення X розміром $w \times h$, радіус ядра (вікна фільтру) r
Вихід: Зображення Y такого ж розміру

```
for i = 1 to w do
    for j = 1 to h do
        D ← нескінченний відрізок;
        for k = −r to r do
            for l = −r to r do
                for o = −r to r do
                    for p = −r to r do
                        Якщо ‖D‖ < ‖X_{i+k,j+l} − X_{i+o,j+p}‖ то D ← X_{i+k,j+l} X_{i+o,j+p}
        Побудувати систему концентрованих еліпсоїдів Петуніна, маючи діаметр D;
        Y_{i,j} ← значення, що відповідає внутрішньому (або центральному) еліпсоїду;
```

2.3.3 Швидкий алгоритм на основі динамічного програмування

Незважаючи на переваги фільтру на основі еліпсоїдів Петуніна над медіанним фільтром (Алгоритм 5) має серйозний недолік. Так, для зображення розміром $w \times h$ та фільтру радіусом r, асимптотична оцінка швидкості цього алгоритму становить:

$$O\left(w \cdot h \cdot r^4\right).$$

Для порівняння, асимптотична оцінка швидкості швидкого алгоритму медіанної фільтрації (Алгоритм 1) або фільтра Гауса становить

$$O\left(\min\{w, h\} \cdot r^2 + w \cdot h \cdot r\right).$$

Отже, при великих розмірах ядра (наприклад, $r > 10$) та великому об'ємі зображень, фільтр зображень на основі еліпсоїдів Петуніна значно повільніше працює, ніж інші алгоритми фільтрації. Основна причина – повільний пошук діаметру.

Існує багато швидких алгоритмів пошуку діаметру множини точок у d-вимірному просторі. Так, наприклад, Har-Peled запропонував евристичний алгоритм знаходження діаметру [11]. У [12] запропоновано детермінований алгоритм, що дозволяє знайти діаметр за час $O(n \log n)$, де n – кількість точок множини. Недоліком цих алгоритмів при використанні для фільтру на основі еліпсоїдів Петуніна є те, що незважаючи на меншу асимптотику, ці алгоритми при невеликої кількості точок працюють не швидше, ніж простий

алгоритм (повний перебір) пошуку діаметру. Оскільки розміри фільтрів більших ніж 21 майже не мають практичного використання, ми не можемо використовувати ці алгоритми для прискорення процесу пошуку діаметру.

Можна помітити, що при "пересування" вікна фільтру, ми повторно виконуємо деякі обчислення. Якщо зможемо оптимальніше виконувати обчислення (без повторень), то зможемо швидше фільтрувати зображення. Автором цієї роботи пропонується алгоритм на основі динамічного програмування, що значно прискорює процес пошуку діаметру.

Опишемо структуру даних "двовимірна черга що зберігає діаметр". Ця структура представляє собою двійка (H, p), де p – координата центру вікна фільтру відносно зображення X, а H – таблиця розміром $(2r+1) \times (2r+1)$, кожен елемент якого зберігає діаметр множини точок, що відповідає пікселям, що потра- пляють у фільтр нижче чи правіше, тобто:

$$H_{i,j} = \left\| D(\{X_{y,x} | x \in [(p_x+i),(p_x+r)]_{\mathbb{Z}}, y \in [(p_y+j),(p_y+r)]_{\mathbb{Z}}\}) \right\|, \; i,j \in -r \dots r \quad (2.2)$$

де D – діаметр множини, $X_{y,x}$ – d-вимірне значення пікселю з координатами (y, x). Очевидно, що в $H_{-r,-r}$ зберігається діаметр множини точок значень пікселів, що потрапляють у вікно фільтру.

Опишемо алгоритм оновлення таблиці H при пересуванні фільтру праворуч. Алгоритм при пересуванні фільтру вниз задається аналогічним чином.

Алгоритм 6: Оновлення таблиці H при пересуванні вікна фільтру праворуч

Вхід : Зображення X, радіус ядра (вікна фільтру) r, двовимірна черга що зберігає діаметр (H, p) що відповідає вікну

Вихід: двовимірна черга що зберігає діаметр (H', p') що відповідає вікну після зміщення праворуч на 1 піксель

$H'_{r,r} \leftarrow 0$;

for $i = r-1$ downto $-r$ do

  $H'_{r,j} \leftarrow \max\left\{\max\left\{\|X_{p_y+i,p_x+r} - X_{p_y+k,p_x+r}\| \quad \forall k \in i \dots r\right\}, \quad H'_{i+1,r}\right\}$

for $j = r-1$ downto $-r$ do

  for $i = r$ downto $-r$ do

    $H'_{i,j} \leftarrow \max\left\{\max\left\{\|X_{p_y+i,p_x+j} - X_{p_y+k,p_x+r}\| \quad \forall k \in i \dots r\right\}, \quad H'_{i+1,j}, \quad H'_{i,j+1}\right\}$

Також опишемо алгоритм для обчислення (H, p), знаючи (H^{left}, p^{left}) та (H^{top}, p^{top}), де $p^{left} = (p_y, p_x - 1)$, $p^{top} = (p_y - 1, p_x)$, тобто знаючи черги, що відповідають вікнам на піксель вище чи лівіше.

Алгоритм 7: Обчислення таблиці H знаючи відповідні таблиці ліворуч та вище

Вхід : Зображення X, радіус ядра (вікна фільтру) r, двовимірні черги що зберігає діаметр (H^{left}, p^{left}) та (H^{top}, p^{top}) що відповідає вікну лівіше та вище відповідно

Вихід: двовимірна черга що зберігає діаметр (H, p)

$H_{r,r} \leftarrow 0$;

for $i = r$ downto $-r$ do

  for $j = r$ downto $-r$ do

    $H_{i,j} = \max\left\{H^{left}_{i,j+1}, \quad H^{top}_{i+1,j}, \quad H_{i,j+1}, \quad H_{i+1,j}, \quad \|X_{p_y+r,p_x+r} - X_{p_y+i,p_x+j}\|, \right\}$

Слід зазначити, що в (Алгоритм 6) та (Алгоритм 7), після кожного кроку присвоєння значення діаметру, слід також у таблиці $D_{i,j}$ зберігати точки кінців діаметру. Опишемо кінцевий алгоритм швидкої фільтрації зображення на основі системи концентрованих еліпсоїдів Петуніна:

Алгоритм 8: Швидкий алгоритм фільтрації Петуніна

Вхід : Зображення X розміром $w \times h$, радіус ядра (вікна фільтру) r
Вихід: Зображення Y такого ж розміру

Обчислити $(H_0, p_0 = (1,1))$ згідно за формулою (2.2), а також відповідний відрізок L_0;
for $i = 1$ to h do
  За (Алгоритм 6) обчислити $(H^{i,1}, p^{i,1} = (i,1))$ та відповідний відрізок $L^{i,1}$;
  Побудувати систему концентрованих еліпсоїдів Петуніна, маючи діаметр $L^{i,1}$;
  $Y_{i,1} \leftarrow$ значення, що відповідає внутрішньому (або центральному) еліпсоїду;
for $j = 1$ to w do
  За (Алгоритм 6) обчислити $(H^{1,j}, p^{1,j} = (1,j))$ та відповідний відрізок $L^{1,j}$;
  Побудувати систему концентрованих еліпсоїдів Петуніна, маючи діаметр $L^{1,j}$;
  $Y_{1,j} \leftarrow$ значення, що відповідає внутрішньому (або центральному) еліпсоїду;
for $i = 1$ to h do
  for $j = 1$ to w do
    За (Алгоритм 7) обчислити $(H^{i,j}, p^{i,j})$ та відповідний відрізок $L^{i,j}$;
    Побудувати систему концентрованих еліпсоїдів Петуніна, маючи діаметр $L^{i,j}$;
    $Y_{i,j} \leftarrow$ значення, що відповідає внутрішньому (або центральному) еліпсоїду;

Очевидно, що (Алгоритм 6) має асимптотичну оцінку $O\left(r^3\right)$, а (Алгоритм 7) має оцінку $O\left(r^2\right)$. Тоді асимптотична оцінка для (Алгоритм 8) дорівнює:

$$O\left(r^4 + r^3\left(w + h\right) + w \cdot h \cdot r^2\right)$$

Оскільки значення r значно менша ніж розмір зображення $w \times h$, то (Алгоритм 8) можна оцінити як

$$O\left(w \cdot h \cdot r^2\right)$$

Що дорівнює швидкості звичайного алгоритму медіанної фільтрації.

Розділ 3

Фрактальний аналіз

Наступним етапом цієї роботи є обчислення фрактальної розмірності для попередньо оброблених зображень. Таким чином, для кожного пацієнта буде побудовано набір фрактальних характеристик.

3.1 Фрактальна характеристика

3.1.1 Теоретичні відомості

Для визначення поняття фракталу треба спочатку навести визначення розмірності Хаусдорфа. Нехай Ω – обмежена множина у метричному просторі X.
Означення 1. Нехай $\epsilon > 0$. Не більш ніж зліченне сімейство $\{\omega_i\}_{i\in I}$ підмножин простору X будемо називати ϵ-покриттям множини Ω, якщо виконуються наступні умови:

(1) $\Omega \subset \bigcup\limits_{i\in I} \omega_i$

(2) $\forall i \in I : |\omega_i| < \epsilon$

де $|\omega_i|$ – діаметр множини ω_i.

Нехай $\alpha > 0$. Нехай $\Theta = \{\omega_i\}_{i\in I}$ – покриття множини Ω. Визначимо наступну функцію, яка в деякому плані визначає розмір цього покриття:

$$F_\alpha(\Theta) := \sum_{i\in I} |\omega_i|^\alpha$$

Позначимо через $M_\alpha^\varepsilon(\Omega)$ "мінімальний розмір" ϵ-покриття множини Ω:

$$M_\alpha^\varepsilon(\Omega) := \inf(F_\alpha(\Theta))$$

де інфімум береться по всіх ϵ-покриттях множини Ω. Очевидно, що функція $M_\alpha^\varepsilon(\Omega)$ не спадає при зменшенні ϵ, оскільки при зменшенні ϵ ми також звужуємо множину можливих ϵ-покриттів. Отже, у неї є скінченна або нескінченна границя при $\varepsilon \to 0+$:

$$M_\alpha(\Omega) = \lim_{\varepsilon\to 0+} M_\alpha^\varepsilon(\Omega)$$

Означення 2. Величину $M_\alpha(\Omega)$ називають α-мірою Хаусдорфа множини Ω.

З властивостей α-міри Хаусдорфа слід визначити те, що $M_\alpha(\Omega)$ спадає по α. Більш того для будь-якої множини Ω існує критичне значення α_0, таке, що:

1. $M_\alpha(\Omega) = 0$ для всіх $\alpha > \alpha_0$

2. $M_\alpha(\Omega) = +\infty$ для всіх $\alpha < \alpha_0$

Значення $M_{\alpha_0}(\Omega)$ може бути нульовим, скінченно додатнім або нескінченним.
Означення 3. Число α_0 називають розмірністю Хаусдорфа множини Ω

Наведемо означення фракталу (або фрактальних множин), запропоноване Мандельбротом [13]:
Означення 4. (Мандельброт) Множина називається фракталом, якщо її розмірність Хаусдорфа строго перевищує його топологічну розмірність.

3.1.2 Алгоритм box-counting

Як правило, фрактальні множини мають складну геометричну структуру, а також мають властивість самоподібності. Характеристика, що описує цю властивість, є фрактальною розмірністю [14].

При вимірюванні фрактальної розмірності різних природних і штучних об'єктів виникає ряд проблем, пов'язаних з тим, що існує кілька визначень фрактальної розмірності. Базовим поняттям є розмірність Хаусдорфа, але її обчислення часто виявляється досить непростою задачею. Тому на практиці частіше використовуються інші розмірності, що відносяться до так званого класу box-computing (або box-counting) [15]. Цей метод полягає у тому, що для довільного додатного δ обчислюється деяка функція $M_\delta(\Omega)$. Якщо $M_\delta(\Omega) \propto \delta^{-D}$, то множина Ω має фрактальну розмірність D. З пропорції випливає, що

$$\dim_B \Omega = D = \lim_{\delta \to 0} \frac{\log(M_\delta(\Omega))}{-\log(\delta)},$$

де значення $M_\delta(\Omega)$ дорівнює кількості S-вимірних кубів зі сторонами δ, необхідних для покриття множини Ω.
Означення 5. Число $\dim_B \Omega$ називають box-counting розмірністю множини Ω.

Зауважимо, що $\dim_B \Omega$ не завжди може існувати. Зв'язок між box-counting розмірністю $\dim_B \Omega$ та розмірністю Хаусдорфа $\dim_H \Omega$ виражається наступною теоремою:
Теорема 1. $\dim_B \Omega \leq \dim_H \Omega$

3.1.3 Лінійна регресія

Як зазначено у попередньому пункті, використання алгоритму обчислення фрактальної розмірності box-counting вимагає перевірки пропорції $M_\delta(\Omega) \propto \delta^{-D}$, а отже, і обчислення регресійної прямої. Кутовий коефіцієнт цієї прямої буде дорівнювати $\dim_B \Omega$. Пряму регресії будемо шукати методом найкращого середньоквадратичного наближення – треба знайти такий елемент Φ_0, який мінімізує значення $\sum_{i=0}^{n} (\Phi_0(x_i) - y_i)^2$.

Нехай M_n – лінійна оболонка базису $\{\phi_0, \phi_1, \ldots \phi_n\}$, за яким ми шукаємо елемент найкращого середньоквадратичного наближення Φ_0. Його існування для випадку гільбертового простору H випливає за теоремою. Нехай ми шукаємо функцію найкращого середньоквадратичного наближення для функції f, $(\cdot, \cdot)$ – внутрішній добуток в даному гільбертовому просторі (у нашому випадку в якості такого простору логічно взяти простір неперервних функцій на області визначення функції f). Тоді за теоремою:

$$\forall i \in M_n \, (f - \Phi_0, \Phi) = 0.$$

Тоді

$$\forall i \in \{0, 1, \ldots n\} \, (f - \Phi_0, \phi_i) = 0.$$

Нехай

$$\Phi_0 = \sum_{i=0}^{n} c_i \phi_i,$$

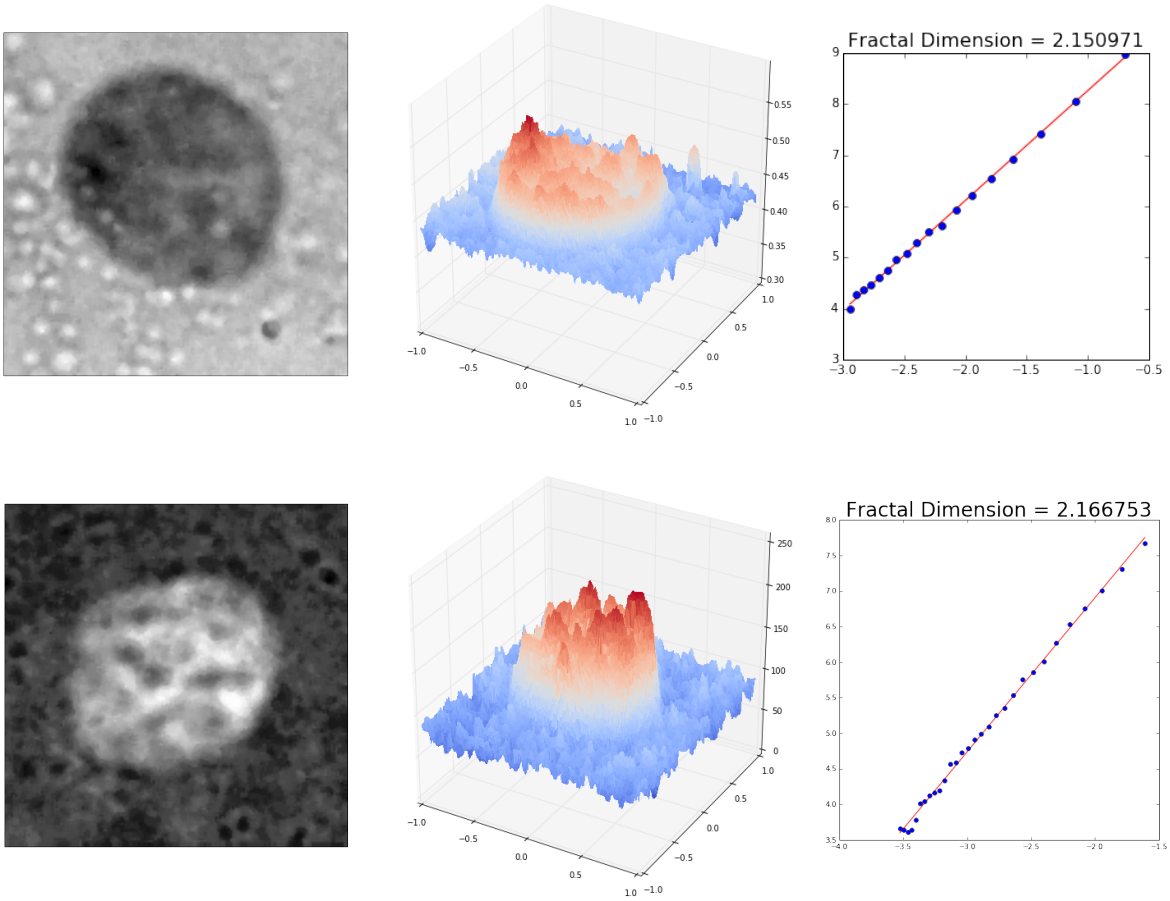
тоді

$$\forall i \in \{0, 1, \ldots n\} \left(f - \sum_{j=0}^{n} c_j \phi_j, \phi_i \right) = 0,$$

$$\sum_{j=0}^{n} c_j \left(\phi_j, \phi_i \right) = \left(f, \phi_i \right), i \in \{0, 1, \ldots n\}.$$

Таким чином ми одержали систему лінійних алгебраїчних рівнянь з матрицею, що є матрицею Грамма $G = \| \left(\phi_i, \phi_j \right) \|_{i,j \in \{0,1,\ldots n\}}$. Отже, якщо $\det(G) = 0$, то можна знайти ров'язок даної системи (в нашому випадку використовується метод Гауса), а отже і шукану функцію Φ_0.

Отже, регресійну пряму можна знайти, побудувавши елемент найкращого середньо-квадратичного наближення Φ_0, обравши базис $\{\phi_0, \phi_1\} = \{1, x\}$.



Мал. 3.1: Фрактальний аналіз зображень.

3.2 Класифікація пацієнтів

Слід помітити, що кожному пацієнту ставиться у відповідність набір (вибірка) фрактальних характеристик, розмір якої не є сталим. Отже, для порівняння набору характеристик різних пацієнтів, треба використовувати міру близькості між вибірками.

Оскільки міра близькості не є метрикою, вибір методів класифікації є дуже обмеженим. В цій роботі було використано найпростіший спосіб – метод k найближчих сусідів.

3.2.1 Міра близькості та p-статистика

Через H позначимо гіпотезу про рівность неперервних функцій розподілу $F_G(u)$ і $F_{G'}(u)$ генеральних сукупностей G і G' відповідно. Нехай вибірки $x = (x_1, x_2, \dots x_n) \in G$ і $x' = (x'_1, x'_2, \dots x'_m) \in G'$, $x_{(1)} \leq \dots \leq x_{(n)}$, $x_{(1)} \leq \dots \leq x_{(m)}$ – порядкові статистики. Припустимо що $F_G(u) = F_{G'}(u)$. Позначимо через $A_{ij}^{(k)}, k = 1, 2, \dots m$ випадкова подія, яка полягає у тому, що x' потрапляє в інтервал $\left(x_{(i)}, x_{(j)}\right)$, тобто $A_{ij}^{(k)} = \left\{x'_k \in \left(x_{(i)}, x_{(j)}\right)\right\}$. Як відомо, ймовірність цієї події обчислюється за формулою:

$$P(A_{ij}^{(k)}) = P\left\{x'_k \in \left(x_{(i)}, x_{(j)}\right)\right\} = p_{ij}^{(n)} = \frac{j-i}{n+1} = \frac{q}{n+1}, \quad q = j - i.$$

Покладемо

$$p_{ij}^{(1)} = \frac{h_{ij}^{(n)} m + \frac{1}{2}g^2 - g\sqrt{h_{ij}^{(n)}(1-h)m + \frac{1}{4}g^2}}{m + g^2},$$

$$p_{ij}^{(2)} = \frac{h_{ij}^{(n)} m + \frac{1}{2}g^2 + g\sqrt{h_{ij}^{(n)}(1-h)m + \frac{1}{4}g^2}}{m + g^2},$$

де $h_{ij}^{(n)}$ – частота події $A_{ij}^{(k)}$ в m експериментів, величина $g = 3$.

Позначимо через N кількість усіх довірчіх інтервалів $I_{ij}^{(n,m)} = \left(p_{ij}^{(1)}, p_{ij}^{(2)}\right)$ (тобто $N = n(n-1)/2$) і L – кількість інтервалів $I_{ij}^{(n,m)}$, які містять ймовірності $p_{ij}^{(n)}$. Покладемо $h^{(n,m)} = \rho(F^*, F^{*'}) = \rho(x, x') = L/N$. Оскільки $h^{(n,m)}$ – частота випадкової події $B = \left\{p_{ij}^{(n)} \in I_{ij}^{(n,m)}\right\}$, ймовірність якої дорівнює $p(B) = 1 - \beta$, тоді, вважаючи що $h^{n,m} = h^{(n)}$, $m = N$ та $g = 3$, отримаємо довірчий інтервал $I^{(n,m} = \left(p^{(1)}, p^{(2)}\right)$ для ймовірності $p(B)$. Статистика $h^{(n)}$ називається p-статистикою. Вона також є мірою близькості $\rho(x, x')$ між вибірками x та x' [16].

3.2.2 Метод k найближчих сусідів. Кросвалідація.

Використовуючи Метод k найближчих сусідів для класифікації пацієнтів, необхідно зробити припущення коректності постулату про компактність:
Припущення 1. переважна більшість об'єктів, що належать до одного класу, є ближчими один до одного, ніж до об'єктів іншого класу, і лежать в області з відносно простою межею.

Суть цього методу для класифікації елементу x полягає у тому, що з навчальної вибірки обираємо k найближчих (у розумінні міри близькості) до x елементів (характеристик пацієнтів) та відносимо x до домінантного класу з цих k елементів.

Для пошуку оптимального числа k, а також розміру навчальної вибірки R використовують метод кросвалідації. Суть цього методу заключається у тому, що для кожного значення k та R випадковим чином підбирається набір даних для навчання.

3.3 Результати та спостереження

Вхідний набір даних для дослідження складається з знімків 6751 інтерфазних ядер букального епітелія, для кожного було зроблено 3 знімки мікроскопу: без фільтру, через жовтий фільтр та через пурпурного фільтру (отже всього 20253 фотографії), взятого з 130 пацієнтів, з них 68 хворих раком, 29 здорових та 33 хворих іншою хворобою.

Назвемо групу пацієнтів хворих на рак чи на фіброаденоматоз "позитивними", а група здорових пацієнтів "негативними". Тоді, за визначенню:

$$\text{чутливість}(sensitivity) = \frac{TP}{P}$$
$$\text{специфічність}(specificity) = \frac{TN}{N}$$

де TP (true positive) – кількість коректно класифікованих позитивних прикладів, TN (true negative) – кількість коректно класифікованих негативних прикладів, P та N – кількість позитивних та негативних прикладів у вибірці відповідно.

Нехай $\Omega = R/N$, де R – розмір навчальної вибірки, N – розмір усієї вибірки. Залежність середнього значення (після кросвалідації) чутливості від Ω та кількості сусідів, аналізуючи червону компоненту зображень:

		Кількість сусідів					
		6	7	8	9	10	12
Ω	0.5	87.43%	81.42%	88.86%	83.71%	87.42%	85.43%
	0.6	83.93%	83.93%	86.78%	83.93%	87.14%	84.29%
	0.7	85.23%	88.10%	89.05%	90.00%	87.62%	90.00%
	0.8	87.14%	84.29%	90.00%	89.29%	86.43%	93.57%
	0.9	82.86%	82.86%	85.71%	84.29%	91.42%	84.26%

Залежність середнього значення (після кросвалідації) специфічності від Ω та кількості сусідів, аналізуючи червону компоненту зображень:

		Кількість сусідів					
		6	7	8	9	10	12
Ω	0.5	64.00%	74.00%	63.33%	74.67%	62.00%	63.33%
	0.6	65.83%	70.83%	65.33%	69.17%	64.17%	68.33%
	0.7	68.89%	72.22%	58.89%	71.11%	73.33%	68.89%
	0.8	71.67%	65.00%	68.33%	66.67%	73.33%	70.00%
	0.9	83.33%	83.33%	53.33%	73.33%	66.67%	66.67%

Залежність середнього значення (після кросвалідації) чутливості від Ω та кількості сусідів, аналізуючи зелену компоненту зображень:

		Кількість сусідів					
		6	7	8	9	10	12
Ω	0.5	90.86%	85.14%	89.71%	86.86%	88.29%	87.71%
	0.6	87.50%	83.57%	91.07%	83.57%	90.36%	88.93%
	0.7	81.90%	85.71%	90.95%	87.62%	89.04%	83.81%
	0.8	81.43%	83.57%	88.57%	80.00%	89.29%	91.43%
	0.9	84.29%	87.14%	88.57%	87.14%	85.71%	82.86%

Залежність середнього значення (після кросвалідації) специфічності від Ω та кількості сусідів, аналізуючи зелену компоненту зображень:

		Кількість сусідів					
		6	7	8	9	10	12
Ω	0.5	63.33%	71.33%	67.33%	67.33%	66.67%	60.67%
	0.6	61.67%	69.17%	71.67%	70.00%	67.50%	65.00%
	0.7	71.11%	70.00%	61.11%	77.78%	66.67%	70.00%
	0.8	71.67%	75.00%	70.00%	76.67%	63.33%	71.67%
	0.9	60.00%	63.33%	70.00%	76.67%	73.33%	70.00%

Залежність середнього значення (після кросвалідації) чутливості від Ω та кількості сусідів, аналізуючи синю компоненту зображень:

		Кількість сусідів					
		6	7	8	9	10	12
Ω	0.5	88.86%	80.29%	86.86%	85.71%	89.71%	86.57%
	0.6	84.29%	84.64%	87.14%	86.43%	90.36%	86.43%
	0.7	88.57%	85.71%	91.90%	80.95%	90.00%	89.53%
	0.8	84.29%	80.71%	89.29%	83.57%	90.71%	89.29%
	0.9	82.86%	81.43%	91.43%	77.14%	92.80%	81.43%

Залежність середнього значення (після кросвалідації) специфічності від Ω та кількості сусідів, аналізуючи синю компоненту зображень:

		Кількість сусідів					
		6	7	8	9	10	12
Ω	0.5	62.67%	74.67%	66.67%	69.33%	61.33%	62.00%
	0.6	66.67%	73.33%	65.00%	69.17%	64.17%	67.50%
	0.7	60.00%	68.89%	62.22%	71.11%	61.11%	60.00%
	0.8	55.00%	76.67%	65.00%	66.67%	61.67%	68.33%
	0.9	76.67%	86.67%	70.00%	73.33%	70.00%	53.33%

Аналізуючи червону чи зелену компоненту зображень, можна отримати досить гарні результати чутливості та специфічності класифікатора (більші за 80%). Нажаль, цей метод не дозволяє виявити відмінність між ядрами клітин пацієнтів з фіброаденоматозом та раком молочної залози.

Розділ 4

Глибинне навчання

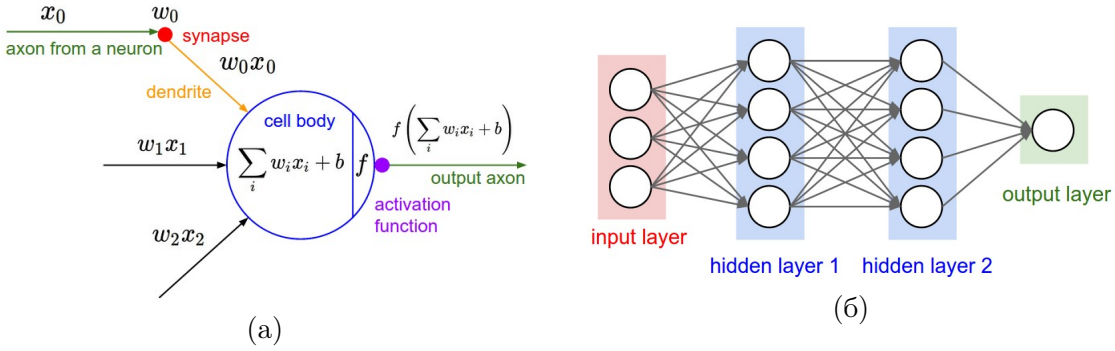
Як зазначено у попередньому розділі (Розділ 3), фрактальний аналіз текстури інтерфазного ядра букального епітелію і взагалі будь-який алгоритм класифікації, який використовує характеристики текстури та контуру ядра клітини, добре розпізнають клас здорових пацієнтів від пацієнтів, хворих раком молочної залози чи фіброаденоматозом, але майже не можуть розділяти останні два класи між собою. Це пояснюється тим, що ці хвороби викликають дуже схожі гормональні реакції, що призводить до схожих змін у ядрах ДНК клітин букального епітелію.

Цей розділ буде присвячено задачі класифікації тільки класів хворих раком та фіброаденоматозом. За допомогою технік глибинного машинного навчання, ми намагаємося аналізувати та використовувати локальні ознаки та особливості текстури поверхні ядра клітини, які не можна характеризувати одним числом.

4.1 Нейронні мережі

Штучна нейронна мережа – це математична модель, яка побудована за принципом функціонування біологічних нейронних мереж. Штучні нейронні мережі широко використовують для задач машинного навчання, розпізнавання образів, дискримінантного аналізу, кластерування тощо.

Нейронну мережу можна інтерпретувати як граф. У підручнику [17], нейронні мережі описуються як граф потоку сигналу (signal-flow graph), де кожен нейрон (або вершина у графі) є окремою одиницею обчислення. Схема одного штучного нейрону зображена на (Мал. 4.1 (а)).



Мал. 4.1: Структура одного нейрону (а) та схема багатошарових перцептронів як граф потоку сигналів (б).

Кожен нейрон мережі має справу лише з сигналами, які він періодично отримує, і сигналами, які він періодично надсилає іншим нейронам. і тим не менш, будучи з'єднаними в достатньо велику мережу з керованою взаємодією, такі прості нейрони разом здатні моделювати складні функції, виявляти складні залежності між вхідними й вихідними даними, а також здійснювати узагальнення.

4.1.1 Багатошаровий перцептрон

Штучний багатошаровий перцептрон – це така архітектура нейронних мереж, у якій групи нейронів формують "шари", в яких кожний нейрон з одного шару зв'язана з усіма нейронами сусіднього шару. Отже, сигнали на вихід одного шару нейронів є сигналами на вхід нейронів наступного шару. На (Мал. 4.1 (б)) зображено схему багатошарового перцептрону. Такі шари нейронів також називають "повністю зв'язаними" (Fully connected layers).

4.1.2 Функція активації ReLU

У цій роботі було використано функцію зрізаних лінійних вузлів, або ReLU, для активації нейронів замість сигмоїдальних функцій чи гіперболічного тангенсу. Функція активації ReLU (Rectified Linear Unit), яка задається наступним чином:

$$f(x) = \max(0, x)$$

має багато переваг над сигмоїдальними функціями чи гіперболічним тангенсом при використанні в якості функції активації:

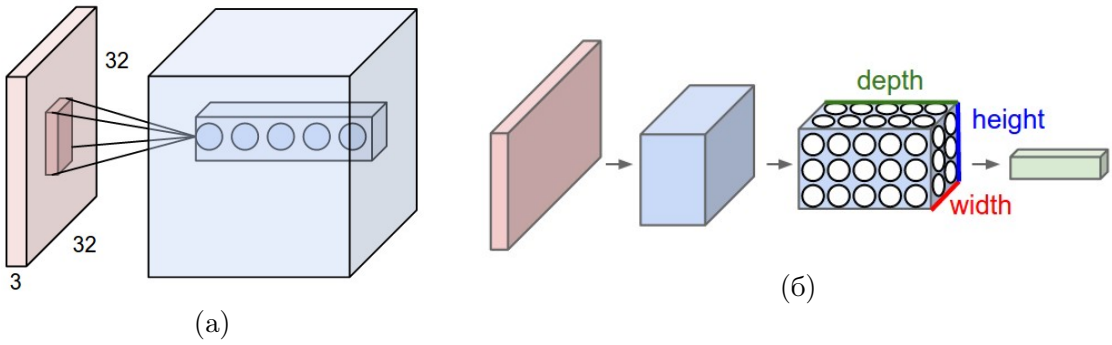
- В роботі [18], було доведено що така функція активації може прискорити збіжність спуску за градієнтом до 6 разів у порівнянні з сигмоїдальними функціями чи гіперболічними тангенсами.
- Сигмоїдальні функції та гіперболічні тангенси використовують складні операції, такі як зведення у степінь.

Детальний аналіз переваг та недоліків цієї функції, а також позбавлення недоліків за допомогою модифікованої версії "Leaky ReLU", описаний в роботі [19].

4.1.3 Згорткові нейронні мережі

Згорткові нейронні мережі – це мережі, які містять згорткові шари нейронів, уперше запропонований [20]. Згортковий шар нейронів містить фільтри – набір ваг. При подачі даних на вхід, вихідний сигнал обчислюється як згортка фільтру з локальним регіоном зображення. На (Мал. 4.2 (а)) показано схему першого згорткового шару, де кожен нейрон локально зв'язаний з пікселями зображення та "бачить" усі канали. Кількість шарів нейронів у згортковому шару (зображено кружечками) дорівнює кількості фільтрів згорткового шару, тобто для різних шарів нейронів використовується різні фільтри.

Згорткові нейронні шари використовують для виявлення локальних особливостей зображення. Ці шари також дозволяють мережі бути більш стійкою до трансляції, тобто надають деяку інваріантність. Тому, будемо використовувати згорткові шари для побудови класифікатора між хворими раком та фіброаденоматозом.



Мал. 4.2: На (а) схематично зображено згортковий шар. На (б) зображено схему згорткової мережі.

Разом із згортковими нейронними шарами також будемо використовувати підвибіркові шари (pooling layers), які значно зменшують обсяг вхідних даних на наступні шари мережі.

4.1.4 Алгоритм RMSProp

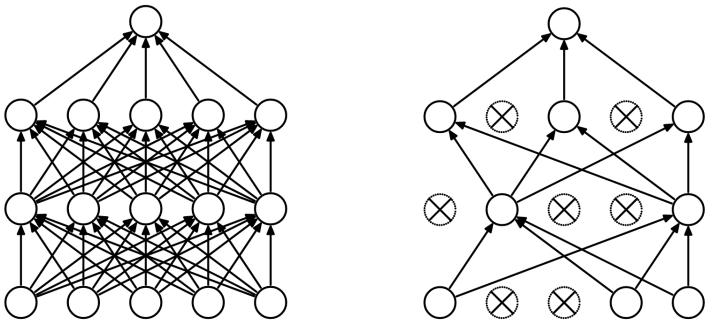
Після обчислення градієнтів за допомогою алгоритму зворотного поширення помилок (backpropagation), ці градієнти використовуються для оновлення параметрів (ваг) нейронної мережі. Зазначимо, що оптимізація алгоритму оновлення нейронних мереж є на момент написання цієї роботи дуже активною сферою досліджень.

Алгоритм RMSProp [21] є дуже ефективним, але ще не опублікований метод адаптивного оновлювання параметрів. Він налаштовує та регулює алгоритм Adagrad [22] таким чином щоб зменшити різкість монотонного зменшення швидкості навчання. На відміну від Adagrad, цей алгоритм використовує так звані "ковзаючі" середньоквадратичні значення градієнтів:

$$cache = decay_rate \cdot cache + (1 - decay_rate) \cdot dx^2$$
$$x = x - \frac{learning_rate \cdot dx}{\sqrt{cache} + eps}$$

де $decay_rate$ це гіперпараметр, якому зазвичай присвоюють значення з множини чисел $\{0.9, 0.99, 0.999\}$. Отже, метод RMSProp також модулює швидкість навчання кожного параметру з урахуванням величин його градієнтів, що має вигідний ефект вирівнювання. Але, на відміну від Adagrad, величини оновлень не зменшуються монотонно.

У цій роботі використовується саме алгоритм RMSProp. На практиці більш стандартним методом для оновлення параметрів є алгоритм Adam (adaptive moment estimation) [23], який дуже схожий на RMSProp та у деяких випадках є трохи ефективнішим ніж RMSProp, але цей алгоритм є більш громіздким та потребує трохи більше часу для обчислень. Повний алгоритм Adam також включає механізм корекції зміщень. Альтернативою RMSProp та Adam є також комбінація алгоритмів SGD та Nesterov Momentum.



(а) До Dropout (б) Після Dropout

Мал. 4.3: Візуалізація методу Dropout з ймовірністю $p = 0.5$.

4.1.5 Регуляризація нейронної мережі

Процес регуляризації нейронної мережі – це процес уникнення перенавчання (overfitting) мережі. існує кілька способів контролювання тренування нейронної мережі, щоб максимальним чином уникнути перенавчання.

У цій роботі використовується метод Dropout – простий та дуже ефективний метод регуляризації, запропонований [24]. При тренуванні, кожен нейрон є активним за ймовірністю p (що є гіперпараметром мережі).

На (Мал. 4.3) схематично зображено принцип роботи методу Dropout. На практиці також часто використовують алгоритми $L1$- та $L2$- регуляризації нейронних мереж [25], разом з методом Dropout.

4.1.6 Функція втрат. Перехрестна ентропія.

Вибір функції втрат є одним з найважливіших кроків при побудові моделі класифікатора. Вдало обрана функція втрат може значно покращити результат та швидкість тренування класифікатора.

На практиці при тренуванні класифікаторів на основі нейронних мереж найчастіше використовують перехресну ентропію (categorical crossentropy):

$$L_i = -\log\left(\frac{e^{f_{y_i}}}{\sum_j e^{f_j}}\right)$$

яка для випадку задач класифікації дає кращі результати, ніж середньоквадратична функція втрат (mean square error) та функція помилки класифікації (classification error). У цій роботі використовується саме ця функція оцінювання втрат. Слід також зазначити, що при тренуванні моделі та при класифікації чи обчисленні можна використовувати різні оцінки втрат.

Мал. 4.4: Прирощення даних на прикладі одного зображення.

4.2 Класифікація за допомогою нейронних мереж

4.2.1 Прирощення даних

Як вказано у розділі 1, для дослідження маємо тільки набір даних який містить 3403 зображень інтерфазних ядер букального епітелію, взятих з 68 пацієнтів хворих раком молочної залози, та 1741 зображень ядер взятих з 33 пацієнтів хворих фіброаденоматозом. Така кількість є дуже малою для тренування глибоких згорткових нейронних мереж. Отже, будемо штучно збільшувати кількість даних для тренування.

При подачі в нейронну мережу випадковим чином зробимо декілька з наступних перетворень над зображеннями:

- Перевернути зображення відносно горизонтальної осі
- Перевернути зображення відносно вертикальної осі
- Повернути зображення на випадковий кут від 0 до 2π, при чому пікселі, що до перетворення знаходилися за межами входу, присвоюються значенню сусідніх пікселів.

4.2.2 Проблема незбалансованості вибірки для тренування

Для тренування було обрано випадковим чином 80% від всього набору даних. Це становить 1392 зображень взятих з 33 пацієнтів хворих фіброаденоматозом та 2722 зображень з 68 пацієнтів хворих раком молочної залози. Отже, в набір даних для валідації входять інші зображення ядер букального епітелію тих самих пацієнтів. Маємо незбалансованість вибірки для тренування, де кількість даних з одного класу (з класу хворих раком) майже вдвічі перевищує кількість даних з іншого класу (хворих фіброаденоматозом).

Незбалансованість набору даних для тренування може значно негативно вплинути на навчання класифікатора [26]. Слід зазначити, що набір даних для валідації теж є незбалансованим. Так, після тренування моделі на незбалансованій вибірці, автором цієї роботи було отримано класифікатор, точність розпізнавання якого на обох наборах даних для тренування і тестування становила більш ніж 65%. Подальший аналіз результатів виявив, що отримана модель має чутливість, близьку до 100% та специфічність, близьку до 0%, тобто модель у більшості випадків просто класифікувала зображення як “належить хворого раком”. Отже, необхідно знайти спосіб боротьби з проблемою незбалансованості набору даних для тренування.

Існує багато методів для боротьби з цією проблемою. Найбільш поширеними на практиці є методи зміни об'єму даних (subsampling methods) та методи навчання з вагами (cost-sensitive methods) [27], де різним класам присвоюється деяке значення "критичності" помилки. Було вирішено використовувати перший метод.

Отже, після зміни об'єму даних, кількість зображень для тренування становить 1392 з хворих фіброаденоматозом та 1632 з хворих раком молочної залози, а кількість зображень для валідації становить 349 зображень з хворих фіброаденоматозом та 1771 зображень з хворих раком молочної залози.

4.2.3 Архітектура мережі

Для зручного опису архітектури мереж, використаємо наступні позначення:

- Позначимо повністю зв'язані шари нейронів (fully-connected layers) як xFC, де x – кількість нейронів шару. Наприклад, якщо повністю зв'язаний шар має 128 нейронів, то позначаємо $128FC$.
- Згорткові нейронні шари (convolutional layers), де фільтри шару мають розміри $x \times x$, позначимо як $yCONVx$, де y – кількість фільтрів. Наприклад, $64CONV3$.
- Підвибіркові шари (max-pooling layers) позначимо як MPx, де x – розмір ядра фільтру. Наприклад, $MP2$.
- Вхідні (input) та вихідні (output) дані позначимо як I та O відповідно.
- Домовимось, що сигнал протікає з шару, що знаходиться лівіше у записі, до шару, що знаходиться правіше. Сусідні шари нейронів з'єднуємо символом "$\rightarrow$". Повна архітектура мережі так: $I \rightarrow 32CONV3 \rightarrow MP2 \rightarrow FC16 \rightarrow O$.

Автором було розглянуто багато різних архітектур нейронної мережі. Окрім простих відносно неглибоких архітектур згорткової нейронної мережі, які схожі на AlexNet, також були розглянуті мережі, подібні до VGG [28].

ARCH-0 $I \rightarrow 32CONV3 \rightarrow MP2 \rightarrow 32CONV3 \rightarrow MP2 \rightarrow 64CONV5 \rightarrow MP2 \rightarrow FC32 \rightarrow FC32 \rightarrow O$

ARCH-1 $I \rightarrow 32CONV3 \rightarrow MP2 \rightarrow 48CONV3 \rightarrow MP2 \rightarrow 64CONV3 \rightarrow MP2 \rightarrow 72CONV3 \rightarrow MP2 \rightarrow FC64 \rightarrow FC64 \rightarrow O$

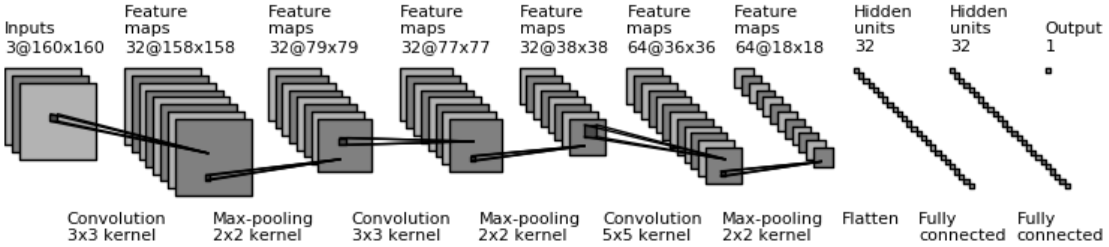
ARCH-2 $I \rightarrow 64CONV5 \rightarrow 48CONV3 \rightarrow MP2 \rightarrow FC32 \rightarrow FC16 \rightarrow O$

ARCH-3 $I \rightarrow 64CONV3 \rightarrow MP2 \rightarrow 64CONV3 \rightarrow MP2 \rightarrow 96CONV3 \rightarrow MP2 \rightarrow 96CONV3 \rightarrow MP2 \rightarrow FC48 \rightarrow FC48 \rightarrow O$

VGG-9 $I \rightarrow 32CONV3 \rightarrow 32CONV3 \rightarrow MP2 \rightarrow 48CONV3 \rightarrow 48CONV3 \rightarrow MP2 \rightarrow 64CONV5 \rightarrow 64CONV5 \rightarrow MP2 \rightarrow FC32 \rightarrow FC32 \rightarrow O$

VGG-11 $I \rightarrow 48CONV3 \rightarrow 48CONV3 \rightarrow MP2 \rightarrow 64CONV3 \rightarrow 64CONV3 \rightarrow MP2 \rightarrow 72CONV3 \rightarrow 72CONV3 \rightarrow MP2 \rightarrow 96CONV3 \rightarrow 96CONV3 \rightarrow MP2 \rightarrow FC96 \rightarrow FC96 \rightarrow O$

VGG-13 $I \rightarrow 32CONV3 \rightarrow 32CONV3 \rightarrow MP2 \rightarrow 64CONV3 \rightarrow 64CONV3 \rightarrow MP2 \rightarrow 128CONV3 \rightarrow 128CONV3 \rightarrow MP2 \rightarrow 256CONV3 \rightarrow 256CONV3 \rightarrow MP2 \rightarrow 256CONV3 \rightarrow 256CONV3 \rightarrow MP2 \rightarrow FC128 \rightarrow FC128 \rightarrow O$



Мал. 4.5: Архітектура найкращої мережі.

Для кожного шару нейронів цих архітектур було використано ReLU в якості функції активації, а для кінцевого шару O з одного нейрону (якщо на виході 0 то зображення класифікується як належним до хворого раком молочної залози, а якщо 1 – то класифікується як належним до хворого фіброаденоматозом) було використано функцію активації SoftMax. З метою регулювання, для кожного повністю зв'язаного шару було використано алгоритм Dropout. В якості алгоритму задання швидкості навчання (тобто оновлення параметрів) використовується RMSProp.

4.2.4 Тренування мереж та результати

Слід зазначити, що вибір архітектури обмежувався потужністю доступних автору обчислювальних ресурсів. У розпорядженні автора на момент дослідження був комп'ютер з процесором Intel Core i7, комп'ютер з процесором Intel Core i3, а також відеокарта Nvidia Geforce GT 520M. Отже, використання більш глибоких нейронних мереж, або мереж з більшою кількістю вільних параметрів, не було можливим.

Кожна з перерахованих вище мереж тренувалася по декілька (для ARCH-0 – декілька десятків) разів, кожен раз нейронна мережа ініціалізувалася випадковим набором ваг (параметрів) та навчалася протягом 300 епох. Кожна епоха – це цикл, в якому нейронна мережа навчається на даних, які по обсягу дорівнюють об'єму тренувальної вибірки. Неповторність вхідних даних для тренування гарантує алгоритм прирощення даних. Також були використані різні значення для Dropout, від 0.2 до 0.5.

Сумарно навчання вищеописаних мереж проводилось протягом трьох тижнів. Найбільш повільними для навчання є моделі VGG-9, VGG-11 та VGG-13. У середньому на навчання цих моделей протягом 300 епох пішло більш ніж 48 годин процесорного часу.

Найбільш вдалою є архітектура ARCH-0, яка схематично зображена на (Мал. 4.5). Нижче у таблиці приведені найкращі результати класифікації окремого зображення інтерфазного ядра клітини букального епітелію цих мереж.

Архітектура	Чутливість	Специфічність	Точність
ARCH-0	59.17%	67.33%	62.93%
ARCH-1	55.74%	57.59%	56.60%
ARCH-2	67.24%	53.58%	60.95%
ARCH-3	56.72%	58.45%	57.52%
VGG-9	55.25%	59.89%	57.39%
VGG-11	58.43%	58.45%	58.44%
VGG-13	56.23%	55.59%	55.94%

Найкращі результати за точністю дає мережа ARCH-0. Надалі будемо детальніше аналізувати її результати та намагатися побудувати класифікатор з кращими результатами на основі цієї моделі.

Можна розглядати кожне зображення ядра клітини певного пацієнта не окремо, а у сукупності з іншими зображеннями клітин цього пацієнта, тобто класифікувати не належність окремого ядра букального епітелію до класу хвороби, а належність самого пацієнта до цих класів. Наприклад, якщо більшість зображень ядер клітин цього пацієнта класифікується належними до класу "хворі раком", то віднесемо цього пацієнта до класу "хворий раком" (аналогічно з фіброаденоматозом). Тоді отримаємо наступні результати:

Архітектура	Чутливість	Специфічність	Точність
ARCH-0	58.82%	69.70%	62.38%

що незначно відрізняються від результатів, отриманих при класифікації окремих клітин.

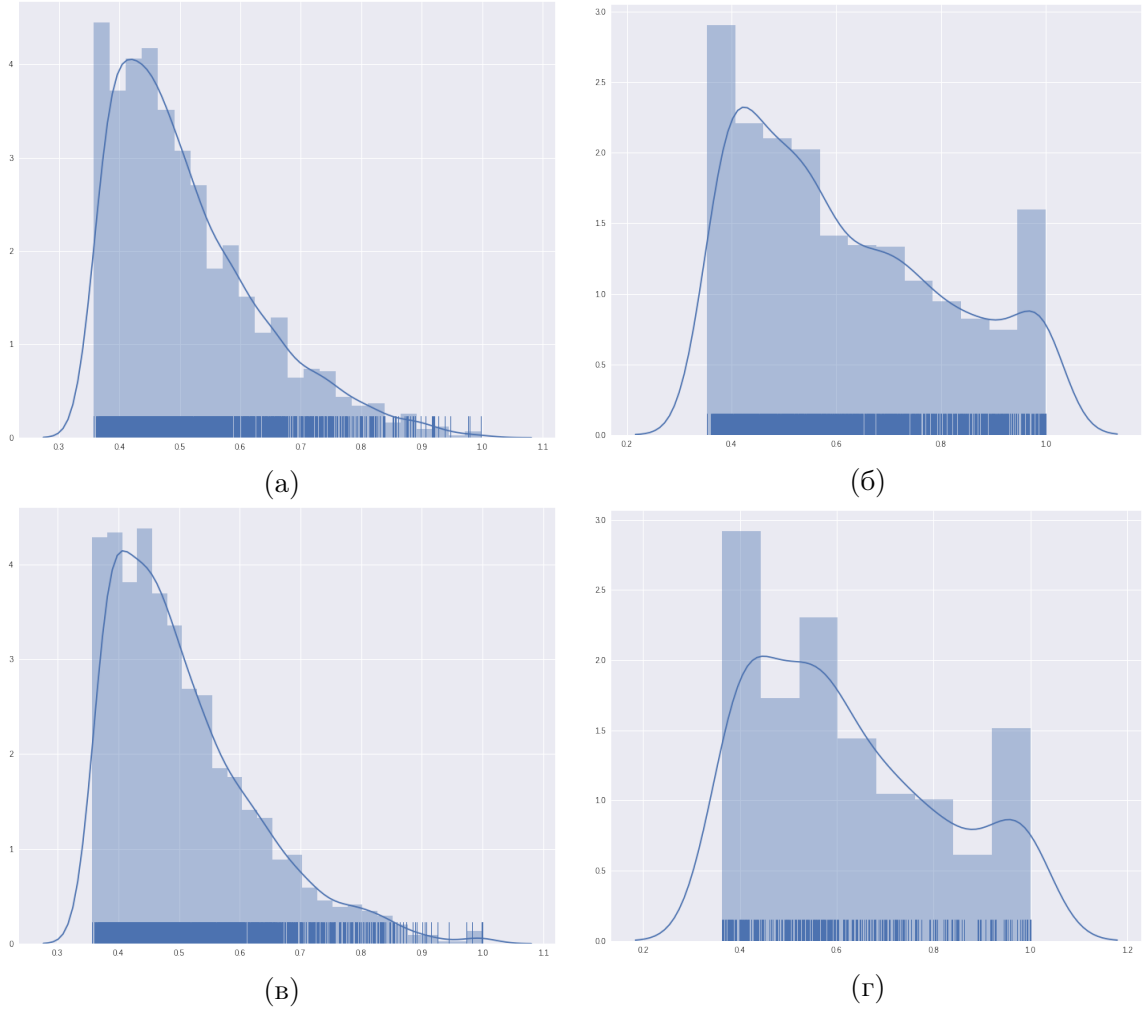
4.2.5 Аналіз результатів

Помітимо, що мережа з відносно простою архітектурою ARCH-0 дає кращі результати ніж VGG-подібні мережі. Це можна пояснити тим, що VGG-подібні глибокі архітектури містять у декілька десятків разів більше вільних параметрів, а отже, потребують більше ніж 300 епох для навчання. При тренуванні моделі VGG-11, наприклад, спостерігалося дуже вільне спадання значення функції втрат. Це свідчить про те, що, можливо, треба було обирати інші гіперпараметри для мережі, такі як *learning_rate* для алгоритму RMSProp, або використовувати менший *batch_size* (обсяг даних при тренуванні, після якого обчислюється градієнт та виконується одне оновлення параметрів).

4.3 Підвищення якості розпізнавання

Надалі будемо будувати більш точний класифікатор на основі отриманої моделі нейронної мережі ARCH-0. Щоб переконатися у тому, що ця модель дійсно генералізується на невідомі дані (ті, які не було використано при тренуванні), розглянемо розподіл вихідних значень мережі для обох наборів: тренування та валідації, що показані на (Мал. 4.6). Можна побачити, що отримана мережа добре генералізується на вибірці для валідації.

Отже, припускаючи, що розподіл вихідних значень мережі для пацієнтів з однаковою хворобою (рак молочної залози чи фіброаденоматоз) буде також приблизно однаковими, замість простого підходу, що описаний у попередньому підрозділі, можна розглядати нейронну мережу не як бінарний класифікатор, а як генератор ознак, тобто кожному зображенню ставити у відповідність деяке число (ознаку). Отже, до кожного пацієнта (набору зображень) можна застосовувати статистичні тести.



Мал. 4.6: Розподіл значення виходу мережі на зображеннях ядер с вибірки для тренування, що належить пацієнтам хворі на рак (а) та на фіброаденоматоз (б). На (в) та (г) зображено розподіл виходу мережі на зображеннях ядер з вибірки для тестування що належить пацієнтам хворі на рак та фіброаденоматоз відповідно.

4.3.1 Гибрідний класифікатор

В якості критерію еквівалентності генеральних сукупностей було використано критерій Колмогорова-Смірнова, що є одним з найефективніших непараметричних критеріїв для порівняння вибірок. Альтернативно можна використовувати p-статистику, що детально описана у попередньому розділі.

Нехай can_trainX – сукупність зображень інтерфазних ядер букального епітелію у вибірці для тренування, взятих з пацієнтів хворих раком, а fib_trainX – сукупність зображень ядер у вибірці для тренування взятих з пацієнтів хворих фіброаденоматозом.

При класифікації деякого пацієнта $P = \{X^1, X^2, \dots X_n\}$ обчислюється:

$$can_train = \text{ARCH-0}(can_trainX) = \{\text{ARCH-0}(X)\ \forall X \in can_trainX\}$$

$$fib_train = \text{ARCH-0}(fib_trainX) = \{\text{ARCH-0}(X)\ \forall X \in fib_trainX\}$$

$$M = \text{ARCH-0}(P) = \{\text{ARCH-0}(X^i)\ \forall i \in 1 \dots n\}$$

де ARCH-0(X) – значення виходу нейронної мережі після обробки зображення X. Потім, обчислюється p-значення статистики Колмогорова-Смірнова спочатку для can_train та M, а потім для fib_train та M. Ці значення використаємо як показних схожості цих вибірок.

4.3.2 Аналіз результатів

У таблиці показано результати класифікації пацієнтів як сукупність ядер клітин, отримані за підходом, що описаний вище.

Архітектура	Чутливість	Специфічність	Точність
ARCH-0	70.59%	63.64%	68.32%

Слід зазначити, що в нашому випадку у зв'язку зі зміною обсягу даних для навчання спостерігається значна незбалансованість класів у наборі даних для валідації, а саме: 1771 зображень ядер, взятих з 68 пацієнтів хворих раком, та 349 – з хворих фіброаденоматозом. Отже, кожний пацієнт, хворий раком, має у середньому 26 зображень клітин ядер у наборі даних для валідації, та в той же час кожен пацієнт, що хворий фіброаденоматозом, має у середньому 10 зображень клітин ядер у вибірці для валідації. Саме тому спостерігається така різниця між чутливістю та специфічністю моделі після валідації. Можна висловити припущення, що маючи достатню кількість зображень для класифікації конкретного пацієнта, можна досягти чутливості та специфічності більші за 71%.

Перегрупуємо дані за пацієнтами, залишаючи мережу ARCH-0 тою самою. Отже, для тестування маємо окремих пацієнтів, для кожного з яких доступно по 50 клітин для валідації. Припускаючи, що розподіл вихідних значень мережі ARCH-0 однаковий для всіх пацієнтів однакової категорії (з однаковою хворобою) та використовуючи метод, описаний вище, отримуємо наступні результати (чутливість, специфічність та точність у середньому після кросвалідації):

Архітектура	Чутливість	Специфічність	Точність
ARCH-0	71.32%	78.95%	73.81%

Отже, підхід з використанням згорткових нейронних мереж та статистичних методів має дуже перспективні результати для подальшого дослідження.

Розділ 5

Висновок

У цій роботі було використано методи комп'ютерного зору для сегментації зображень інтерфазних ядер букального епітелію, досліджено взаємозв'язок між характеристиками зображень цих ядер та наявністю пухлин в молочній залозі пацієнтів за допомогою фрактальних показників і статистичних методів та запропоновано метод класифікації пацієнтів хворих на рак та на фіброаденоматоз за цими зображеннями на основі новітніх методів машинного навчання. Використаний метод сегментації зображень виявився дуже точним для цієї конкретно поставленої задачі, а запропонований швидкий алгоритм для фільтру зображень на основі еліпсоїдів Петуніна є не менш ефективним за часом, ніж звичайний алгоритм для медіанної фільтрації. Отримано дужі точні результати розпізнавання здорових пацієнтів від хворих раком молочної залози чи фіброаденоматозом. Запропонований метод класифікації пацієнтів хворих на рак та на фіброаденоматоз за допомогою глибинного навчання та статистичних критеріїв дає дуже перспективні результати.

5.1 Поради для подальших досліджень

Відмінність між локальними ознаками на зображеннях інтерфазних ядер букального епітелію у пацієнтів з раком молочної залози та фіброаденоматозом, на відміну від зв'язку між характеристиками цих ядер та наявністю пухлин в молочной залозі пацієнтів [1], є ще мало дослідженою. Подальші дослідження у цьому напрямку є дуже перспективними. Автор цієї роботи робить припущення, що використовуючи більш ефективні методи глибинного навчання, можна отримати результати, що будуть набагато точнішими за отримані.

Як зазначено у розділі 4, використання відносно неглибоких архітектур у цій роботі зумовлене відсутністю доступу до більш потужних обчислювальних ресурсів на момент написання цієї роботи. Більш того, новітні дослідження в області глибинного навчання пропонують методи для значного прискорення процесу тренування згорткової нейронної мережі. Так, наприклад, у роботах [29] та [30] пропонуються методи прискорення нейронних мереж при обчисленнях та навчанні у декілька разів без значної втрати точності. Такі прискорення є доречними при використанні для таких глибоких архітектур як VGG-16 [28], AlexNet [18] та NIN [31]. Рекомендується також обрати більш оптимальні гіперпараметри, наприклад, початкові ваги чи коефіцієнти для RMSProp чи інших алгоритмів оновлення параметрів, а також використовувати методи навчання з вагами [27] чи інші методи для боротьби з проблемою незбалансованості вибірки для тренування. Крім цього, щоб отримати більш точні результати, необхідний більший обсяг даних з кращою якістю для аналізу.

Додаток А

Програмний пакет

В процесі виконання цієї роботи для кожного розділу було створено комплекс програм. Детальний опис, інструкцію щодо встановлення та код програм можна знайти за посиланням на git-репозиторій: https://github.com/chankhavu/CancerDetection.

Bibliography

[1] N. Boroday et al. “Analysis of malignancy-associated DNA changes in interphase nuclei of buccal epithelium in persons with breast diseases”. In: Experimental Oncology 26.2 (2004), pp. 158–160.

[2] Weidi Xie, J. Alison Noble, and Andrew Zisserman. “Microscopy Cell Counting with Fully Convolutional Regression Networks”. In: Department of Engineering Science, University of Oxford, UK (2015).

[3] S. Perreault and P. Hébert. “Median filtering in constant time”. In: IEEE Transactions on Image Processing 16.9 (2007), 2389–2394.

[4] Nobuyuki Otsu. “A threshold selection method from gray-level histograms”. In: Automatica 11 (285–296 1975), pp. 23–27.

[5] Jean Serra. Image Analysis and Mathematical Morphology. Academic Press Inc., 1983.

[6] A. Rosenfeld and J. Pfaltz. “Distance Functions in Digital Pictures”. In: Pattern Recognition 1 (1968), pp. 33–61.

[7] R. Barnes, C. Lehman, and D. Mulla. “Priority-flood: An optimal depression-filling and watershed-labeling algorithm for digital elevation models”. In: Computers and Geosciences 62 (), pp. 117–127.

[8] М. В. Присяжная. “Фильтрация изображений с использованием эллипсов Петунина”. In: Журнал обчислювальної та прикладної математики 1.115 (2014).

[9] B. M. Hill. “Posterior Distribution of Percentiles: Bayes’ Theorem for Sampling from a Population.” In: Journal of the American Statistical Association 63.322 (1968), p. 677.

[10] С. И. Ляшко, Д. А. Клюшин, and В. В. Алексеенко. “Многомерное ранжирование с помощью эллиптического пилинга”. In: Кибернетика и системный анализ 4 (2013).

[11] Sariel Har-Peled. “A practical approach for computing the diameter of a point set”. In: Proceedings of the seventeenth annual symposium on Computational geometry (2001).

[12] G. Malandain and J.-D. Boissonnat. “Computing the diameter of a point set”. In: International Journal of Computational Geometry and Applications 12.6 (2002), pp. 498–509.

[13] Benoît Mandelbrot. The Fractal Geometry of Nature. W. H. Freeman and Co., 1982.

[14] F. Richard Voss. “Fundamental algorithms for computer graphics”. In: Springer, 1985. Chap. Random fractal forgeries, pp. 805–835.

[15] Ю. В. Шуплецов and Н. Б. Ампилова. “Алгоритм вычисления размерности Минковского для полутоновых изображений”. In: Известия Российского Государственного Педагогического Университета Им. АИ Герцена 165 (2014).

[16] Д. А. Клюшин and Ю. И. Петунин. “Непараметрический критерий эквивалентности генеральных совокупностей, основанный на мере близости между выборками”. In: Український математичний журнал 5 (2003), 147–163.

[17] Simon Haykin. Neural Networks and Learning Machines. 3rd ed. Pearson Education, 2009.

[18] A. Krizhevsky, I. Sutskever, and G. E. Hinton. “Imagenet classification with deep convolutional neural networks”. In: Advances in neural information processing systems (2012), 1097–1105.

[19] He Kaiming et al. “Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification”. In: (2015).

[20] Yann LeCun et al. In: Proceedings of the IEEE 86.11 (1998), pp. 2278–2324.

[21] T. Tieleman and G. Hinton. “RMSProp: Divide the gradient by a running average of its recent magnitude”. In: COURSERA: Neural Networks for Machine Learning (2012).

[22] J. Duchi, E. Hazan, and Y. Singer. “Adaptive Subgradient Methods for Online Learning and Stochastic Optimization”. In: 12 (2011), pp. 2121–2159.

[23] P. Kingma Diederik and Ba Jimmy. “Adam: A Method for Stochastic Optimization”. In: International Conference for Learning Representations (2015).

[24] N. Srivastava et al. “Dropout: a simple way to prevent neural networks from overfitting.” In: Journal of Machine Learning Research 15.1 (2014), 1929–1958.

[25] Andrew Y. Ng. “Feature selection, L_1 vs L_2 regularization, and rotational invariance”. In: International Conference on Machine Learning (2004), p. 58.

[26] Paulina Hensman and David Masko. “The Impact of Imbalanced Training Data for Convolutional Neural Networks”. PhD thesis. KTH Royak Institute of Technology, CSC School, 2015.

[27] Haibo He. “Learning from Imbalanced Data”. In: IEEE Transactions on knowledge and data engineering 21.9 (2009).

[28] K. Simonyan and A. Zisserman. “Very deep convolutional networks for large-scale image recognition”. In: International Conference for Learning Representations (2014).

[29] Michael Figurnov, Dmitry P. Vetrov, and Pushmeet Kohli. “PerforatedCNNs: Acceleration through Elimination of Redundant Convolutions”. In: Neural Information Processing Systems (2015).

[30] Xiangyu Zhang et al. “Accelerating Very Deep Convolutional Networks for Classification and Detection”. In: arXiv preprint arxiv:1505.06798 (2015).

[31] Min Lin, Qiang Chen, and Shuicheng Yan. “Network In Network”. In: International Conference for Learning Representations (2014).