

Міністерство освіти і науки України
Західноукраїнський національний університет
Факультет комп'ютерних інформаційних технологій
Кафедра комп'ютерної інженерії

САЛУК Роман Васильович

**«Алгоритми пошуку мікроб'єктів на
гістологічних зображеннях за двома
шаблонами /**

**Micro objects searching algorithms for
histological images using two templates»**

Студент групи КІм – 21
САЛУК Роман Васильович

Науковий керівник
д.т.н., професор Березький О.М.

Тернопіль – 2020

РЕЗЮМЕ

Кваліфікаційна робота на тему «Алгоритми пошуку мікрооб'єктів на гістологічних зображеннях за двома шаблонами» зі спеціальності 123 «Комп'ютерна інженерія» написана обсягом 100 сторінок і містить 36 рисунків, 3 таблиці, 2 додатки та 51 джерело за переліком посилань.

Метою роботи є розроблення алгоритмів пошуку мікрооб'єктів за шаблонами на гістологічних зображеннях.

Методи досліджень. Для розв'язання поставлених задач у кваліфікаційній роботі використано: методи математичного аналізу та об'єктно-орієнтованого програмування.

Результати дослідження: алгоритми пошуку мікрооб'єктів за шаблонами на гістологічних зображеннях, програмний модуль пошуку мікрооб'єктів за шаблонами на гістологічних зображеннях.

Результати роботи можуть бути використані для пошуку інформативних ознак в базах даних біомедичних зображень, для синтезу штучних баз даних біомедичних зображень із заданими інформативними ознаками та в навчальному процесі.

Орієнтовні напрямки розвитку досліджень: розроблення модулів систем автоматизованої мікроскопії для пошуку зображень; створення розподілених баз даних для зберігання штучних баз даних зображень.

КЛЮЧОВІ СЛОВА: ПОШУК, ШАБЛОН, ГІСТОЛОГІЧНІ ЗОБРАЖЕННЯ, ПРОГРАМНИЙ МОДУЛЬ.

RESUME

The qualification graduation thesis on «Micro objects searching algorithms for histological images using two templates» from the specialty 123 «Computer engineering» is 100 pages long and contains 36 illustrations, 3 tables, 2 appendices, and 51 references.

The aim of the work is to develop algorithms for searching microobjects by patterns in histological images.

Research methods. To solve tasks in the master's work with the following methods: methods of mathematical analysis and object-oriented programming.

Research results: algorithms for searching microobjects by patterns on histological images, software module for searching microobjects by patterns on histological images.

The results of the work can be used to search for informative features in databases of biomedical images, to synthesize artificial databases of biomedical images with given informative features and in the educational process.

Directions of research development: development of modules for automated microscopy systems for image search; creation of distributed databases for storing individual image databases.

KEYWORDS: SEARCH, TEMPLATE, HISTOLOGICAL IMAGES, PROGRAM MODULES.

ЗМІСТ

Вступ.....	7
1 Аналіз методів і алгоритмів опрацювання зображень.....	9
1.1 Опрацювання зображень.....	9
1.2 Аналіз методів пошуку зображень	14
1.3 Аналіз гістологічних зображень	19
1.4 Постановка задач	30
1.5 Висновки до розділу1.....	31
2 Алгоритми пошуку зображень	32
2.1 Алгоритм побудови скелетонів	32
2.2 Порівняння зображень на основі скелетонів в метриці Фреше	44
2.3 Алгоритм пошуку зображень за шаблонами	50
2.4 Висновки до розділу 2.....	54
3 Програмна реалізація алгоритму пошуку мікрооб'єктів на зображеннях	55
3.1 Структура програмного модулю	55
3.2 Результати роботи алгоритму	64
3.3 Комп'ютерні експерименти	65
3.4 Висновки до розділу 3.....	72
Висновки.....	73
Список використаних джерел.....	74
Додаток А Лістинг фрагменту коду програми	80
Додаток Б Світлокопії виданих публікацій.....	93

ВСТУП

Актуальність теми. Кваліфікаційна робота виконана згідно методичних вимог [1, 2] і присвячена пошуку мікрооб'єктів за шаблонами на гістологічних зображеннях.

Для діагностування передракових і ракових станів органів людини використовуються цитологічні, гістологічні та імуногістохімічні зображення.

Гістологічне дослідження дає більш точну діагностику порівняно з цитологічним. Проте, імуногістохімічне дослідження уточнює діагностику, яку отримано після гістології [3]. На сьогодні накопичились великі бази даних біомедичних зображень, які в основному не анотовані та недостатньо описані. Тому пошук зображень із заданими діагностичними ознаками є актуальною задачею.

Мета і завдання дослідження. Метою роботи є розроблення алгоритмів пошуку мікрооб'єктів зображень за шаблонами на основі скелетонів.

Об'єкт дослідження – процес аналізу гістологічних зображень.

Предмет дослідження – методи та алгоритми пошуку зображень.

Методи досліджень. Для розв'язання поставлених задач у кваліфікаційній роботі використано: методи математичного аналізу та об'єктно-орієнтованого програмування.

Для досягнення поставленої мети потрібно розв'язати такі завдання:

- проаналізувати методів і алгоритми пошуку зображень;
- проаналізувати гістологічні зображення;
- розробити алгоритм побудови скелетонів;
- розробити алгоритм порівняння зображень на основі скелетонів в метриці Фреше;
- розробити алгоритм пошуку зображень за шаблонами;
- розробити структуру програмного модуля;
- провести комп'ютерні експерименти.

Наукова новизна отриманих результатів. Розроблено алгоритм порівняння зображень на основі скелетонів в метриці Фреше та алгоритм пошуку зображень за шаблонами.

Практичне значення отриманих результатів. Розроблено програмний модуль для пошуку мікрооб'єктів за шаблонами на гістологічних зображеннях.

Публікації та апробація кваліфікаційної роботи. Отримані результати апробовані на III Науково-практичній конференції молодих вчених і студентів «Інтелектуальні комп'ютерні системи та мережі» Західноукраїнського національного університету [4, 5].

Впровадження результатів КР. Результати роботи планується використати при виконанні госпдоговірної теми «Методи машинного навчання для кластеризації та класифікації зображень ауто- та ксеногенних тканин».

Кваліфікаційна робота складається із трьох розділів, висновків, списку використаної літератури та додатків.

У першому розділі здійснено аналіз методів опрацювання зображень, методів пошуку зображень і проведено аналіз гістологічних зображень.

У другому розділі проаналізовано алгоритми побудови скелетонів, здійснено порівняння зображень на основі скелетонів в метриці Фреше, розроблено алгоритм пошуку зображень за шаблонами.

У третьому розділі описано структуру програмного модуля, приведено результати роботи алгоритму, та проаналізовано комп'ютерні експерименти.

1 АНАЛІЗ МЕТОДІВ І АЛГОРИТМІВ ОПРАЦЮВАННЯ ЗОБРАЖЕНЬ

1.1 Опрацювання зображень

У повсякденному житті та трудовій діяльності люди часто мають справу з різними типами зображень з усіх сфер – природи, виробництва, відпочинку і дозвілля, художньої творчості тощо. Для деяких видів діяльності людини зображення стали центральними об'єктами пізнання природи явищ, їхніх причин та основними факторами для прийняття тих чи інших рішень. Це стосується, насамперед, криміналістики, у якій відбитки пальців, фотографії технічних засобів пересування та учасників злочинів, знімки місць різноманітних подій стають вирішальними свідченнями для прийняття судових рішень. У розв'язанні фундаментальних і прикладних задач астрономії, геології, екології, біології, медицини, сільського господарства та інших галузей людської діяльності використовують різні методи отримання та аналізу зображень. Аналізує зображення експерт самотійно або за допомогою програмного і математичного забезпечення, яке розробляють фахівці з різних галузей наукових знань.

Аналіз зображень переважно зосереджені на роботі з двовимірними зображеннями, а саме: перетвореннями одного зображення на інше [6, 7]. Це, наприклад, операції збільшення контрастності, виділення країв, видалення шумів або геометричні перетворення, зокрема обертання чи масштабування зображення. Ці операції передбачають, що опрацювання чи аналіз зображення здійснюють незалежно від змісту самих зображень.

Одним із найважливіших застосувань цих операцій є опрацювання зображень у медицині, де необхідно отримати інформацію із зображень для встановлення медичного діагнозу пацієнтам. Здебільшого вхідні дані одержують за допомогою мікроскопії, рентгенівського дослідження, ангиографії, ультразвукових досліджень, рентгенівської комп'ютерної томографії, магнітно-резонансної томографії. Вимірювання розмірів органів,

аномалій тощо за зображеннями формують прикладну сферу медичних досліджень [3].

Комп'ютерний зір передбачає опрацювання тривимірних сцен, спроектованих на одне або декілька зображень, наприклад, відновленням структури або іншої інформації про тривимірну сцену за одним або декількома зображеннями. Комп'ютерний зір застосовується у виробничих системах, наприклад, для візуального контролю та вимірювань. У комплексних системах управління технологія давачів зображення та методи управління пов'язані з опрацюванням потоку відеоданих для керування роботом у реальному часі апаратно або програмно.

Іншою прикладною областю комп'ютерного зору є технології. Тут інформацію отримують для підтримки виробничого процесу. Прикладом може слугувати контроль якості, коли деталі або кінцевий продукт автоматично перевіряють на наявність дефектів. Іншим прикладом є вимірювання положення та орієнтації деталей, що піднімають руки робота.

Військове застосування є, мабуть найбільшою сферою комп'ютерного зору. Очевидними прикладами є виявлення ворожих військових і транспортних засобів і системи наведення ракет. Найдосконаліші системи посилають ракету в задану область, не маючи конкретної мети, а селекція цілей здійснюється тоді, коли ракета досягає заданої області бойових дій, ґрунтуючись на одержаних даних.

Сучасне військове поняття, таке як “бойова поінформованість”, передбачає, що різні задачі, зокрема зображення, надають великий набір інформації про поле бою, яка використовується для прийняття стратегічних рішень. У цьому випадку автоматичне опрацювання даних використовується для зменшення складності або збільшення надійності отримуваної інформації.

Однією з нових сфер застосування комп'ютерного зору є автономні транспортні засоби насамперед підводні, наземні (робота, машини), повітряні. Рівень автономності змінюється від повністю автономних (безпілотних) засобів до транспортних засобів, у яких основані на комп'ютерному зорі системи

підтримують водія або пілота в різних ситуаціях. Повністю автономні транспортні засоби використовують комп'ютерний зір для навігації, тобто для отримання інформації про місце свого перебування, для створення карти навколишнього оточення для виявлення перешкод. Їх також використовують для певних завдань наприклад, для виявлення лісових пожеж. Прикладами таких систем є системи попереджувальної сигналізації про перешкоди на машинах і системи автономної посадки літаків. Деякі виробники машин демонструють системи автономного управління автомобілем, але ці технології поки що не досягають того рівня, коли їх можна запуснути у масове виробництво.

Кожна зі сфер застосування комп'ютерного зору пов'язана з низкою типових завдань, деякі приклади з яких подано нижче.

Розпізнавання образів є галуззю знань, яка використовує різні методи для отримання інформації з даних, що ґрунтуються на статистичному підході. Це класична задача комп'ютерного зору для визначення наявності в певному наборі даних заданого об'єкта незалежно від його природи. Це завдання порівняння легко вирішує людина, однак у комп'ютерному зорі ця задача поки розв'язана частково, за винятком окремих об'єктів у визначених обставинах (прості геометричні фігури, людські обличчя, друковані чи рукописні символи, автомобілі тощо).

Серед великого різноманіття проблем розпізнавання образів найважливіші такі:

- один або кілька попередньо заданих і вивчених об'єктів або класів об'єктів можуть бути розпізнані, зазвичай разом із їх двовимірним станом на зображенні або тривимірним станом у сцені;

- розпізнається індивідуальний екземпляр об'єкта, наприклад, ідентифікується певне людське обличчя, відбитки пальців або сліди автомобіля;

- перевірка даних на наявність аномалій, наприклад, виявлення неправильних клітин або тканин у медичних зображеннях. Спочатку знаходять невеликі ділянки в зображенні, які потім аналізують за допомогою дещо

складніших методів для забезпечення правильної інтерпретації;

- пошук зображень за змістом: виявлення всіх зображень у великому наборі зображень певного змісту. Зміст визначається різними способами, наприклад, подібність з конкретним зображенням або високорівневі критерії пошуку, що вводяться як текстові дані.

Наприклад, знаходження всіх зображень, на яких багато будинків, але немає автомобілів тощо;

- оцінка положення або орієнтації певного об'єкта щодо камери;
- розпізнавання символів на зображеннях друкованого або рукописного тексту, зазвичай для перетворення у текстовий формат.

Відновлення зображень – сукупність задач для видалення шуму (шум давача, розмитість у русі тощо), корекції дефектів об'єкта і фону, що виникли в процесах створення зображення та його зберігання. Простим підходом до вирішення цих завдань є фільтри нижніх або середніх частот. Складніші методи зміни зображення використовують дані про те, як мають виглядати ті чи інші ділянки зображення. У такому разі високий рівень видалення шумів досягається у ході первинного аналізу даних на наявність таких структур, як лінії або межі, а потім на основі нових отриманих даних здійснюється керування процесом фільтрації.

Попереднє опрацювання здійснюється залежно від потреб комп'ютерного зору щодо вхідних даних, наприклад, застосування попереднього перетворення кольорів зображення з одного подання до іншого. Найважливіші приклади такі:

- видалення шуму, внесеного приймачем і каналом передавання даних;
- поліпшення контрастності для виявлення корисної інформації;
- зміна масштабу для кращої структуризації об'єктів на зображенні;
- квантування кольорів для зменшення розмірів файлів зображень;
- повторна вибірка, щоб переконатись у правильності координатної системи зображення.

Виділяють деталі зображення різних рівнів складності найрізноманітнішими методами. Типовими прикладами таких деталей є:

- лінії, межі та краї;
- локальні точки, що становлять інтерес для кодування тощо.

Детектування / сегментування здійснюється, якщо потрібно розділити зображення на два сегменти: корисної та зайвої інформації [8]. Вже розроблено багато алгоритмів сегментування, які відрізняються між собою якістю частин поділеного зображення, алгоритмічною складністю, типами зображень, до яких вони найкраще адаптовані. Прикладами є:

- виділення певного набору точок;
- сегментація однієї або декількох ділянок зображення, які містять характерний об'єкт.

Опрацювання на високому рівні — це складні задачі, в яких вхідні дані, зазвичай, представляють набір, наприклад, точок або ділянку зображення, серед яких, ймовірно, певний об'єкт. Приклади:

- перевірка, чи задовольняють вони умови, які залежать від методу опрацювання та застосування зображення;
- оцінювання характерних параметрів, таких як положення або розмір об'єкта;
- класифікація виявленого об'єкта за різними категоріями.

Відновлення сцени. Методи опрацьовують два або більше зображення сцени (часто у відеопотоці). Відновлення сцени має на меті відтворити тривимірну модель сцени, а саме: віддалі між об'єктами, типи об'єктів, швидкості руху об'єктів тощо. Найпростіша модель – набір точок тривимірного простору. Складніші методи відтворюють тривимірні моделі об'єктів і простору

Пошук та класифікація. Одним з важливих застосувань аналізу зображень є їх пошук, який здійснюється у різноманітних архівах і базах даних [9, 10]. Це – спеціалізовані криміналістичні архіви, державні бази даних нерухомості, бази даних зображень у мережі Інтернет тощо. Розміри цих баз – десятки і сотні мільйонів зображень. Тому необхідні пошукові системи з високою швидкістю для порівняння індексів запиту та наявних екземплярів

даних. Окрім суто професійних потреб, пошук надає додаткові можливості отримання інформації, тобто дає змогу:

- з'ясувати походження зображення в мережі Інтернет та одержати про нього більше відомостей;
- дослідити або простежити появу зображення в мережі Інтернет;
- знайти це зображення / фотографію з вищою роздільною здатністю;
- зафіксувати інтернет-сторінки, на яких використано створене зображення;
- відшукати модифіковані або відредаговані версії зображення / фотографії.

1.2 Аналіз методів пошуку зображень

Для пошуку зображень використовують такі підходи [11, 12]:

1. Кольорові фільтри – якщо об'єкт суттєво виділяється на фоні за кольором, то можна підібрати відповідний фільтр.
2. Виділення і аналіз контурів – якщо ми знаємо, що об'єкт має форму, наприклад, кола, то можна пошукати кола на зображенні.
3. Зіставлення з шаблоном – у нас є зображення об'єкта, шукаємо в іншому зображенні області збігаються з цим зображенням об'єкта.
4. Робота з особливими точками – на зображенні з об'єктом шукаємо особливості (наприклад кути), які намагаємося зіставити з такими особливостями на іншому зображенні.
5. Методи машинного навчання – навчаємо класифікатор на зображеннях з об'єктом, деяким способом поділяємо зображення на частини, перевіряємо класифікатором кожену частину на наявність об'єкта.

Виділення та аналіз контурів. Якщо об'єкт на фоні за кольором істотно не виділяється і / або має складну розмальовку, то застосування методу колірних

фільтрів не дасть хороших результатів. В цьому випадку можна спробувати застосувати метод виділення та аналізу контурів. Для цього ми виділяємо контур на зображенні. Межі це місця різкої зміни градієнта яскравості, їх можна знайти за допомогою методу Canny [13]. Далі ми можемо перевірити виділені лінії контуру на відповідність геометричним контурам об'єкта, це можна зробити застосувавши метод Хафа (Hough Transform) [13], наприклад ми можемо пошукати в межах околу. Якщо зображення має багато дрібних деталей то аналіз контурів може бути складним. В цьому випадку можна застосувати метод зіставлення з шаблоном (template matching). Він полягає в наступному – беремо зображення з об'єктом і шукаємо на великому зображенні області, що збігаються із зображенням об'єкта. Метод зіставлення з шаблоном шукає точні співпадіння точок шаблону з точками зображення. Якщо зображення повернуто чи зі зміненим розміром щодо параметрів шаблону то цей метод працює погано. Для подолання цих обмежень застосовують методи базовані на особливих точках [14, 15]. Особлива точка (key point) – це невелика область, яка істотно виділяється на зображенні. Існує кілька методів визначення таких точок, це можуть бути кути (Harris corner detector) [16] або БЛОБ (blob) [17], тобто невеликі області однакової яскравості, досить чіткою границею, що виділяються на загальному фоні. Для особливої точки обчислюють дескриптор – характеристику особливої точки. Дескриптор обчислюють за заданим оточенням особливої точки, як напрямки градієнтів яскравості різних частин цього околу. Існує кілька методів розрахунку дескрипторів для особливих точок: SIFT, SURF, ORB і ін. Особливі точки можна застосовувати для пошуку об'єкта на зображенні. Для цього нам необхідно мати зображення шуканого об'єкта і далі виконати такі дії:

1. На зображенні з об'єктом шукаємо особливі точки об'єкта і обчислюємо їх дескриптори.

2. На зображенні теж шукаємо особливі точки і обчислюємо для них дескриптори.

3. Порівнюємо дескриптори особливих точок об'єкта та дескриптори особливих точок, знайдених на зображенні.

4. Якщо знайдено достатню кількість відповідностей то помічаємо область з відповідними точками.

Методи машинного навчання. Метод пошуку об'єктів шляхом порівняння наборів особливих точок має свої недоліки, один з них це недостатня узагальнююча здатність. Для вирішення цієї проблеми необхідно використати методи машинного. Якщо розглядати задачу пошуку об'єктів на відео, то її можна розділити на три підзадачі – виявлення, розпізнавання та супроводження:

1. Виявлення (detection) – виділяємо області на зображенні, які можуть містити цікаві для нас об'єкти.

2. Розпізнавання (recognition) – уточнюємо типи знайдених об'єктів.

3. Супровід (tracking) – локалізація на наступних кадрах розпізнаних об'єктів. Машинне навчання можна застосовувати для вирішення завдань виявлення і розпізнавання, які зводяться до побудови класифікатора зображень. Класифікатор зображень складається з двох частин: метод вилучення ознак (feature extractor) і власне класифікатор. Далі ми розглянемо різні способи побудови класифікатора зображень за допомогою методів вилучення ознак HOG, BoW .

Гістограми орієнтованих градієнтів (HOG). Гістограма орієнтованих градієнтів (Histogram of Oriented Gradients, HOG) [18] – метод вилучення ознак із зображень, який дуже подібний на метод обчислення дескрипторів SIFT [19] для особливих точок, тільки обчислюємо його не для околу особливої точки, але для всього зображення. Загальна схема обчислення HOG така [18, 20]. Зображення розділяється на частини, для кожної частини будуємо гістограму напрямків градієнта яскравості, далі гістограми частин нормуються по контрасту і об'єднуються.

Мішок слів (BoW). Мішок слів (Bag of visual Words, BoW) [21] – метод вилучення ознак із зображень, який є адаптацією для зображень методу

частотного аналізу текстів (TF, term frequency) [22]. Застосування методів машинного навчання для розпізнавання об'єктів на зображенні передбачає наявність навчальної вибірки, що складається з двох (як мінімум) частин (класів):

1. Набір зображень містять об'єкт (позитивні приклади).
2. Набір зображень об'єкт не містять (негативні приклади).

Аналогічно методу TF, для того, щоб використовувати метод VoW, перше що потрібно зробити це зібрати словник. Опишемо алгоритм покроково:

1. Вибираємо метод виявлення особливих точок і метод обчислення для них дескрипторів.
2. Для кожного зображення, що входить в навчальну вибірку, визначаємо особливі точки і обчислюємо для них дескриптори.
3. Об'єднуємо подібні дескриптори в групи, тобто виконуємо кластеризацію [23, 24] множини отриманих дескрипторів.

Отримані кластери дескрипторів відіграють роль візуальних слів і складати словник для частотного аналізу зображення. Ознаки VoW вилучаються з зображення так:

1. Обраним на етапі складання словника методом, визначаємо особливі точки на зображенні і обчислюємо для них дескриптори.
2. Розбираємо знайдені дескриптори за кластерами словника.
3. Для кожного кластера словника вважаємо кількість знайдених дескрипторів.

Таким чином, вектор VoW-ознак це кількість знайдених дескрипторів в кожному кластері словника.

Ознаки Хаара. Ознака Хаара (Haar-like features) [25] обчислюється так:

1. Вибираємо прямокутну область на зображенні.
2. Розбиваємо її на кілька суміжних прямокутних частин.
3. В кожній частині підсумовуємо яскравість точок.
4. Після чого обчислюємо різницю між цими сумами.

Ця різниця і буде значенням ознаки. Для формування характеристики зображення використовують кілька різних ознак Хаара, у кожного будуть свої параметри - розмір області для обчислення ознаки, кількість частин, позиція на зображенні. На основі ознак Хаара побудований метод Віоли-Джонса [26], який часто використовують для локалізації зображення осіб на фотографіях. Він об'єднує в собі ознаки Хаара та класифікатор AdaBoost [27, 28] і метод ковзного вікна для поділу зображення на частини.

Згорткова нейронна мережа. Далі розглянемо модель класифікатора зображень під назвою згорткова нейронна мережа (convolutional neural network, CNN) [29]. Особливістю цієї моделі класифікатора є вбудований механізм вилучення ознак із зображення. Він є частиною мережі і "самоналаштовується" в процесі навчання мережі.

Локалізація об'єктів. Задачу локалізації об'єктів на зображенні можна розбити на дві підзадачі: чи є об'єкт на зображенні і яку область зображення він займає? Ця задача розв'язується за допомогою методу ковзаючого вікна (sliding window) [30], схема якого така:

1. Визначити розмір w вікна (досліджуваної області зображення).
2. Зібрати навчальний набір L з позитивних (на зображенні є об'єкт) і негативних (на зображенні немає об'єкта) прикладів, розмір навчальних картинок повинен відповідати розміру w вікна.
3. Навчити класифікатор s на отриманому наборі L .
4. Отримати зображення для дослідження, вибрати крок зсуву вікна d (по горизонталі і вертикалі) і коефіцієнт масштабування зображення s .
5. Розмістити вікно w в крайньому положенні (лівий верхній кут) на досліджуваному зображенні.
6. Виконати класифікацію поточного вікна, якщо класифікатор s визначив наявність об'єкта в поточному вікні, то помістити параметри вікна (поточний стан вікна, масштаб зображення і значення видане класифікатором) в список результатів R .

7. Якщо вікно не досягло кінцевого положення (правий нижній кут), то зсуваємо вікно на крок d і переходимо на попередній пункт, інакше переходимо на наступний пункт.

8. Якщо розмір зображення перевищує розмір w вікна, то масштабується зображення з коефіцієнтом s і переходимо на п.5.

Результатом роботи алгоритму є список R , який складається з позиції вікна, відповідних йому значення масштабу зображення і значення виходу класифікатора. Далі необхідно виконати обробку результатів, яка зводиться до приведення координат вікон до єдиного масштабу і видалення зайвих (дублікатів) вікон. Результатом роботи алгоритму є список R , який складається з позиції вікна, відповідних йому значення масштабу зображення і значення виходу класифікатора. Далі необхідно виконати обробку результатів, яка зводиться до приведення координат вікон до єдиного масштабу і видалення зайвих (дублікатів) вікон.

1.3 Аналіз гістологічних зображень

Гістологічні зображення розглянемо на прикладі диспластичних процесів раку молочної залози [6, 31-36].

Аденома – це доброякісна пухлина, яка має виражену тонку сполучнотканинну капсулу. Її долькова структура частково порушена, різниця між альвеолами та внутрішньодольковими протоками втрачена. Пухлина складається із більш однотипних мілких трубочок. Строма пухлини виражена незначно і ніби витісняється епітеліальними трубочками. Ці трубочки перебувають в стані розквіту або часткової атрофії, в залежності від гормонального статусу жінки (рисунок 1.1).

Фібroadенома є найбільш частою доброякісною пухлиною, має вид інкапсульованого вузла щільної консистенції, волокнистої будови. Вона

складається із фіброзної та залозистої тканини. Мікроскопічно фіброаденома виявляється проліферацією альвеол і внутрішньодолькових протоків з розростанням внутрішньодолькової сполучної тканини. Якщо вона оточує внутрішньодолькові протоки, то це свідчить про периканалікулярну фіброаденому (рисунок 1.2). При вростанні сполучної тканини в стінку протоків виникають їх химерні риси. Така пухлина має назву інтраканалікулярна фіброаденома.

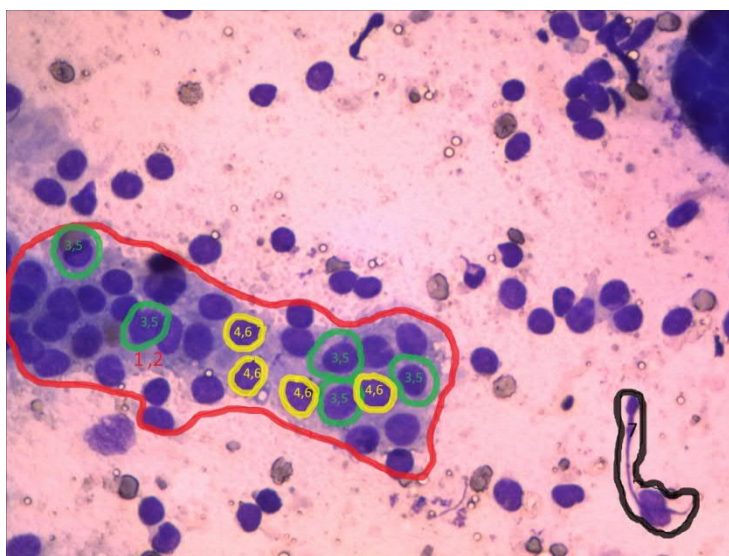


Рисунок 1.1 – Виділення якісних діагностичних ознак в цитологічних препаратах

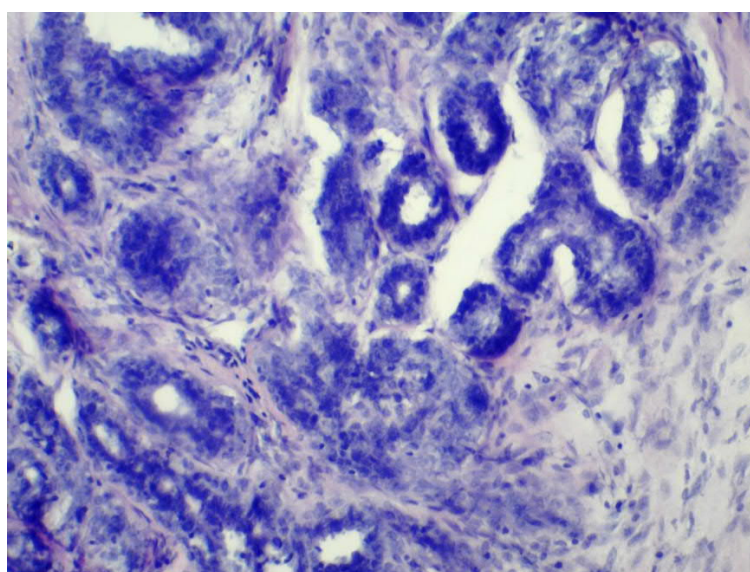


Рисунок 1.2 – Аденома. Поручена дольова структура, незначно виражена строма. Забарвлення гематоксиліном та еозином. Збільшення $\times 100$

Шаровидний щільний біловатого кольору вузол фіброаденоми розвивається як і інші дисплазії із дольок і вони зазнають змін. Спостерігається проліферація альвеол і внутрішньодолькових протоків, які перетворюються у дрібні, частково розширені трубочки. Одночасно розвивається внутрішньодолькова сполучна тканина. Ростуча сполучна тканина рихла та базофільна, даліше від протоків – груба та оксифільна (рисунок 1.3).

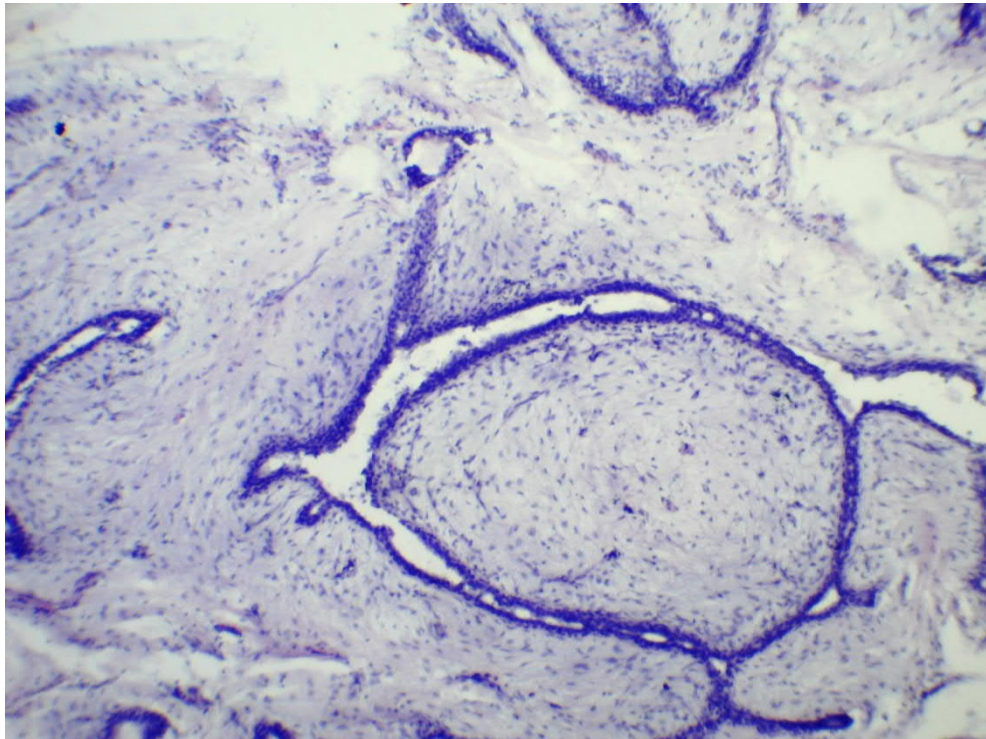


Рисунок 1.3 – Фіброаденома. Вростання сполучної тканини в стінку протоків.
Забарвлення гематоксиліном та еозином. Збільшення $\times 100$

Пекриканулярні та інтраканулярні різновиди є швидше формами, ніж стадіями аденоми. Якщо вони перетворюються одна в одну, то шляхом росту і розвитку індиферентних порівняно низькодиференційованих ділянок, а не трансформації уже готових структур (рисунок 1.4). Трубочки фіброаденоми вистелені епітелієм і міоепітелієм, які знаходяться в різних функціональних станах.

Міоепітелій представлений видовженими темними клітинами або світлими із вмістом міхурців. Часто він відсутній та іноді вистілка стає

одношаровою, представлена клітинами темного епітелію різної висоти і з різною інтенсивністю апокринової секреції (рисунок 1.5). Темний епітелій іноді піддається метаблазії в блідий епітелій. Він також може формувати своєрідні причудливі залозисті структури. З плином часу сполучна тканина гіалінізується, а замуrowаний в ній епітелій атрофується та гине. Фібroadенома трансформується у фібромu.

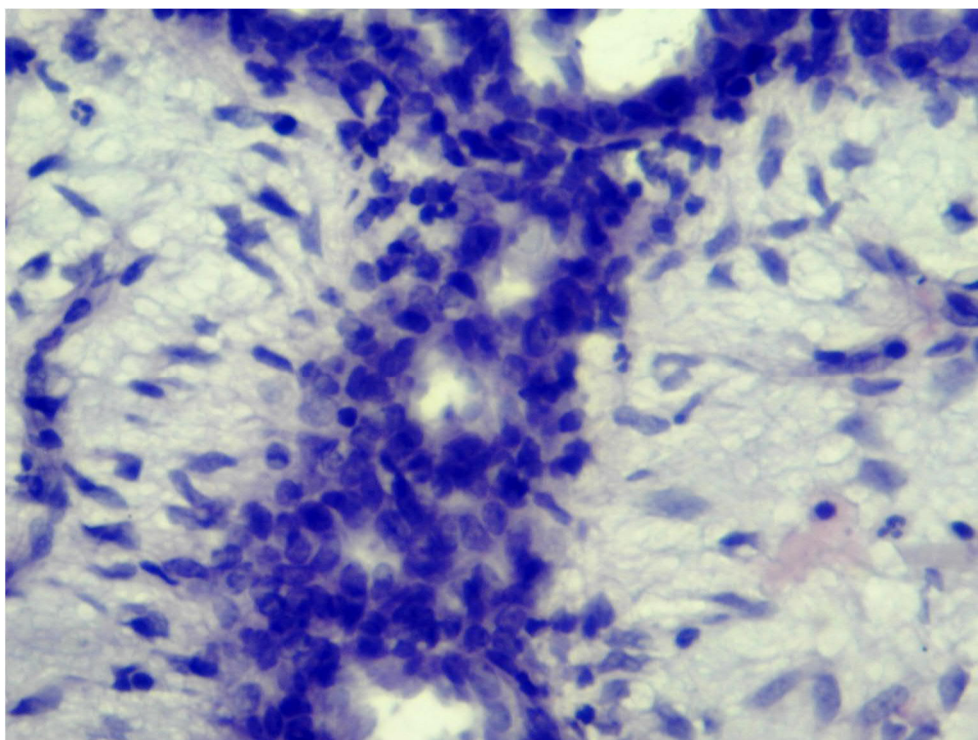


Рисунок 1.4 – Рихла базофільна сполучна тканина. Зabarвлення гематоксиліном та еозином. Збільшення $\times 200$

Філоїдна фібroadенома розвивається на тлі звичайної периканалікулярної фібroadеноми. Епітеліальні трубочки розширюються і перетворюються в кісти (рисунок 1.6).

Строма вип'ячується в просвіт кіст у вигляді масивних сосочків. Найбільш крупні кісти мають грубі сосочки. Деякі філоїдні фібroadеноми досягають надзвичайно великих розмірів, що підтверджує її назву – гігантська фібroadенома.

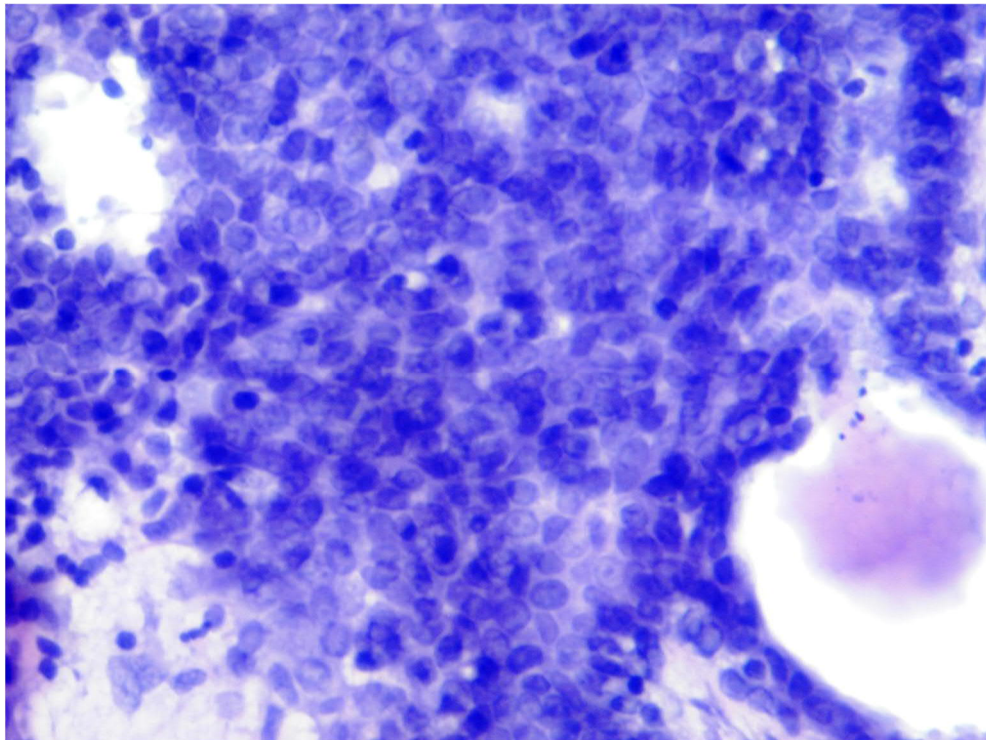


Рисунок 1.5 – Епітелій різних функціональних станів. Забарвлення гематоксиліном та еозином. Збільшення $\times 200$

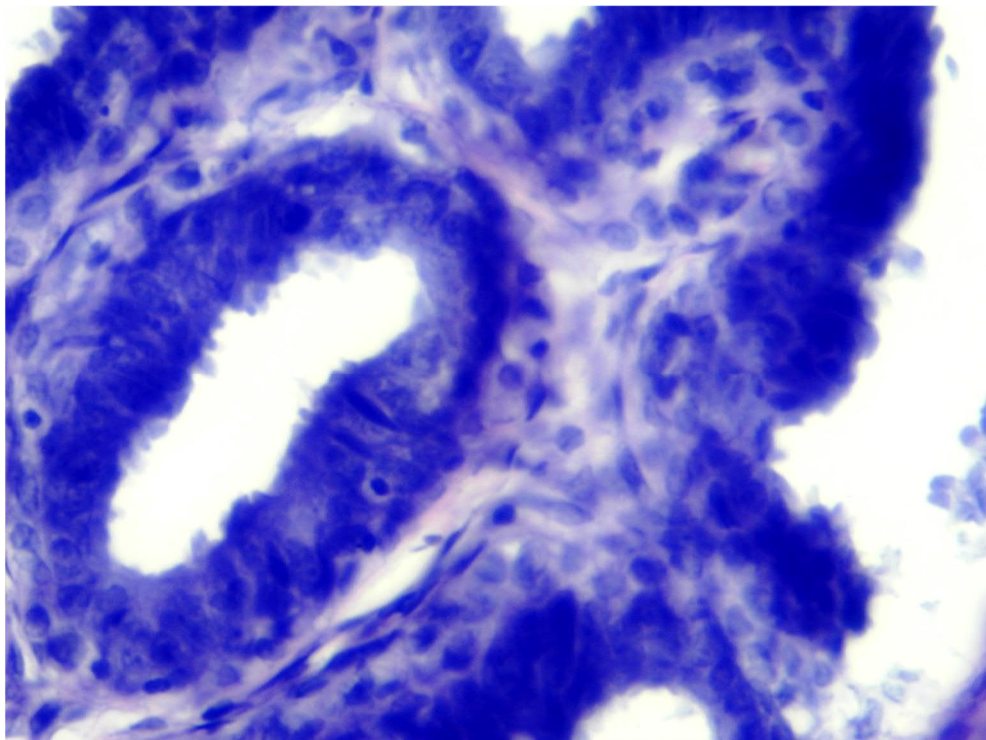


Рисунок 1.6 – Міоепітелій представлений видовженими темними клітинами або світлими із вмістом міхурців. Забарвлення гематоксиліном та еозином. Збільшення $\times 200$

Проявляється вона формуванням множинних вогнищ проліферації фібробластів строми (рисунок 1.7). Анатомічні утвори у вигляді кіст і товстих сосочків можуть бути відсутніми. Фокуси проліферації поступово зливаються один з одним, клітини розташовуються хаотично, мають велику кількість мітозів.

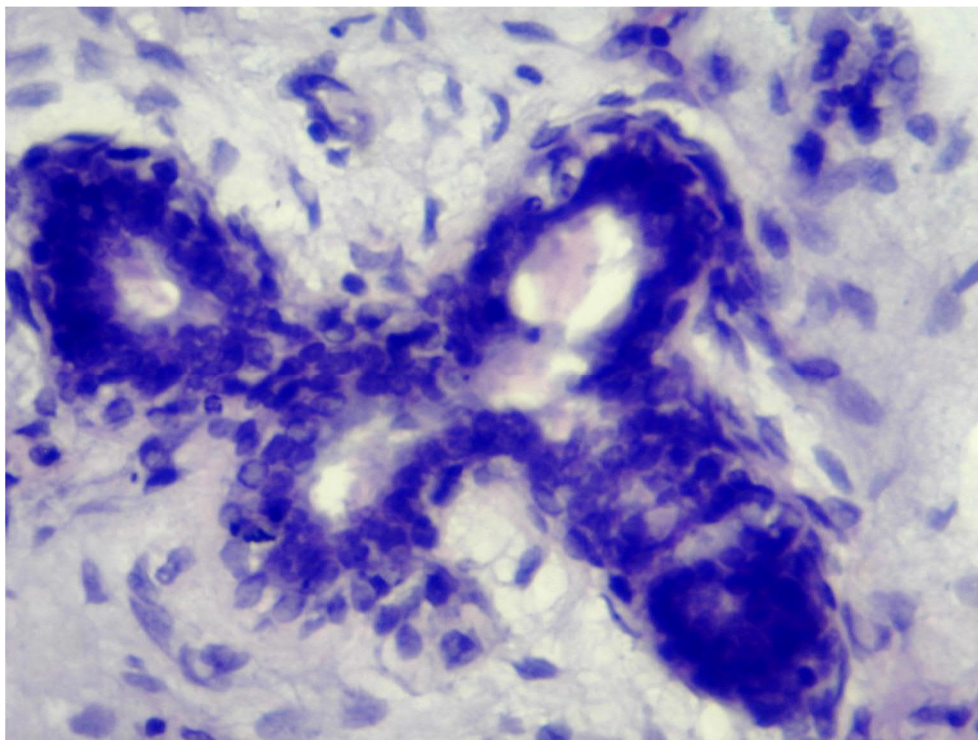


Рисунок 1.7 – Філоїдна фіброаденома. Розширення епітеліальних трубочок і утворення кіст. Забарвлення гематоксиліном та еозином. Збільшення $\times 200$

Проліферативна мастопатія. При проліферативній мастопатії міоепітеліальні клітини зазнають змін. В залежності від свого функціонального стану вони поодинокі або багаточисельні, темні і видовжені, світлі і які містять міхурцеві включення (рисунок 1.8, 1.9). Ці зміни зазнають клітини, які заходяться між базальною мембраною та секретуючим епітелієм альвеол і дрібних протоків. Фіброзуючий аденоз є виключенням. Базальна мембрана зникає і проліферуючі міоепітеліальні клітини проникають в навколишню внутрішньодолькову сполучну тканину, де стають подібними на гладком'язеві елементи (рисунок 1.10).

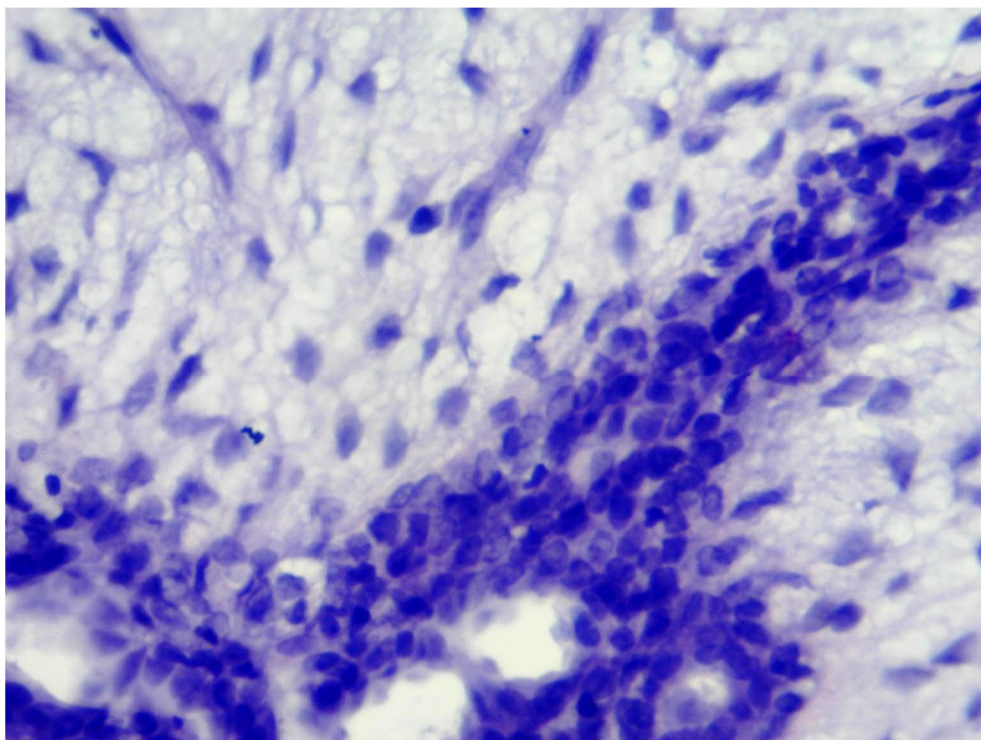


Рисунок 1.8 – Проліферація фіброblastів строми. Забарвлення гематоксиліном та еозином. Збільшення $\times 200$

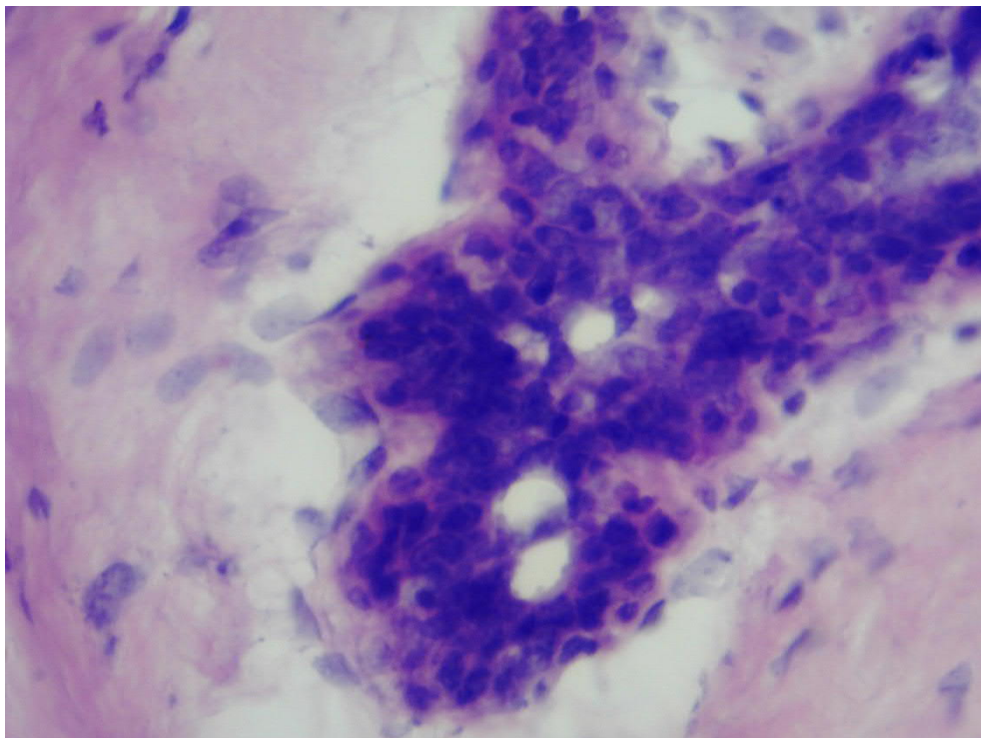


Рисунок 1.9 – Проліферативна мастопатія. Виражена проліферація епітелію залоз. Забарвлення гематоксиліном та еозином. Збільшення $\times 200$

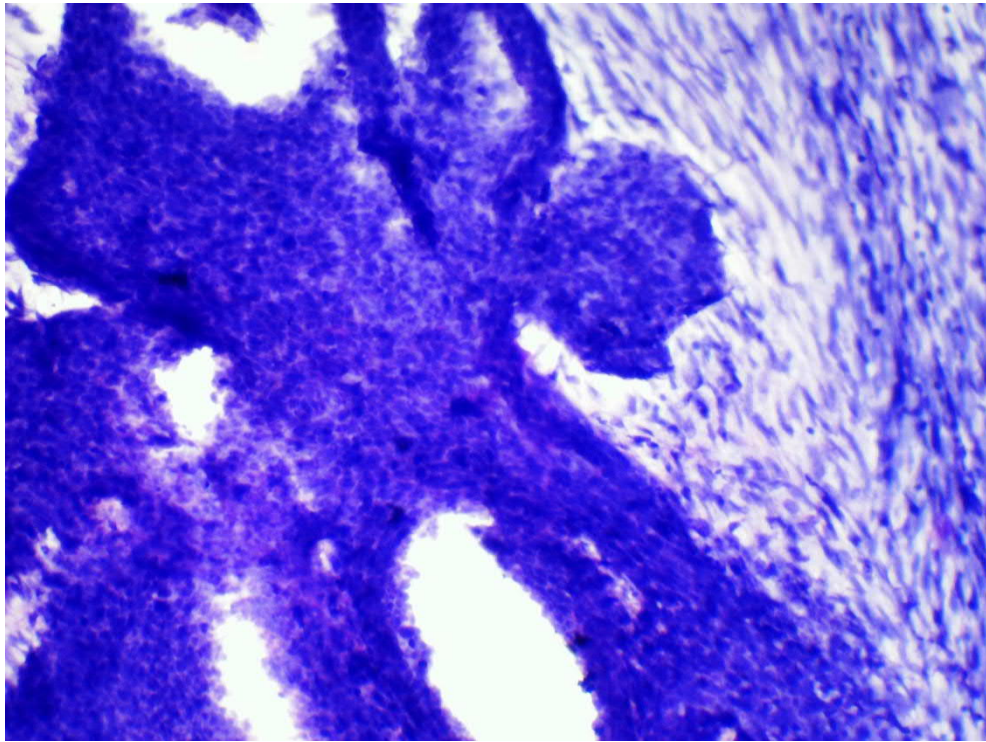


Рисунок 1.10 – Проліферативна мастопатія. Виражена проліферація епітелію залоз. Забарвлення гематоксиліном та еозином. Збільшення $\times 100$

Виникають мікроскопічні фокуси, які складаються із скупчень видовжених міоепітеліальних клітин, серед яких спостерігаються епітеліальні трубочки. Мікроскопічні фокуси мають неправильні контури або округлу форму і чіткі межі. В останньому випадку вони подібні на збільшені або змінені дольки. Між ними з'являються колагенові волокна і міоепітеліальний проліферат склерозується (рисунок 1.11).

Непроліферативна мастопатія. Мастопатія з переважанням кістозного компонента проявляється кістами, чітко відмежованими від навколишніх тканин залози, утвореними з атрофованих часточок і розширених проток залоз з фіброзними змінами інтерстиціальної тканини (рисунок 1.12). В епітелії кіст можливе виникнення проліферативних процесів з формуванням папілярних утворень.

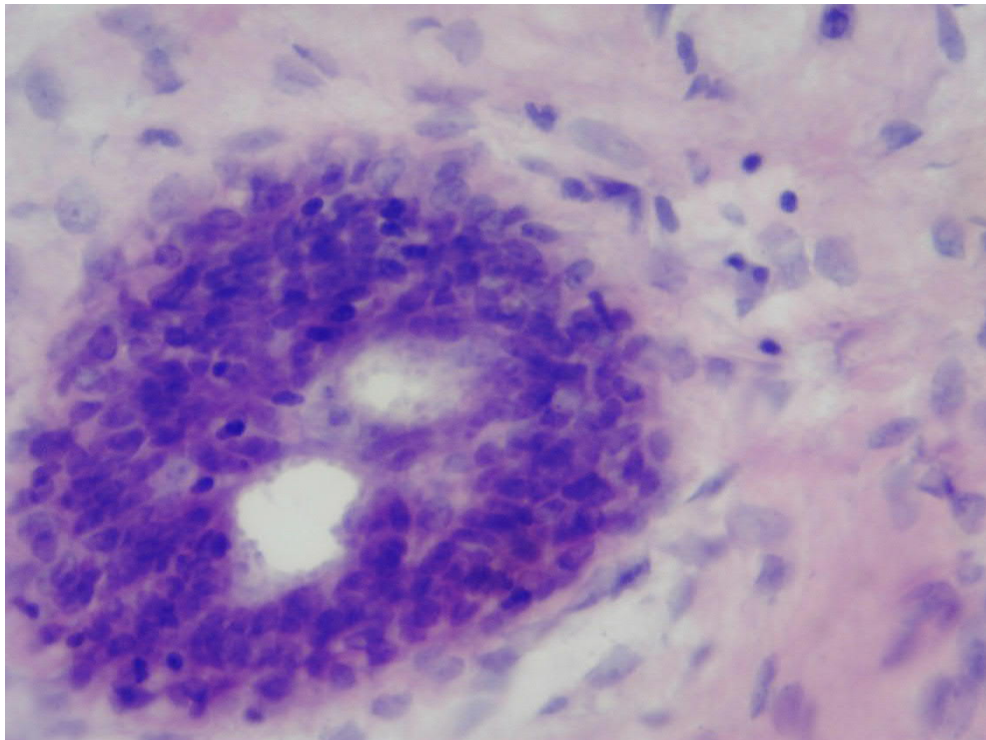


Рисунок 1.11 – Базальна мембрана зникає і проліферуючі міоепітеліальні клітини проникають в навколишню внутрішньодолькову сполучну тканину де стають схожими на гладкомязові елементи. Збарвлення гематоксиліном та еозином. Збільшення $\times 200$

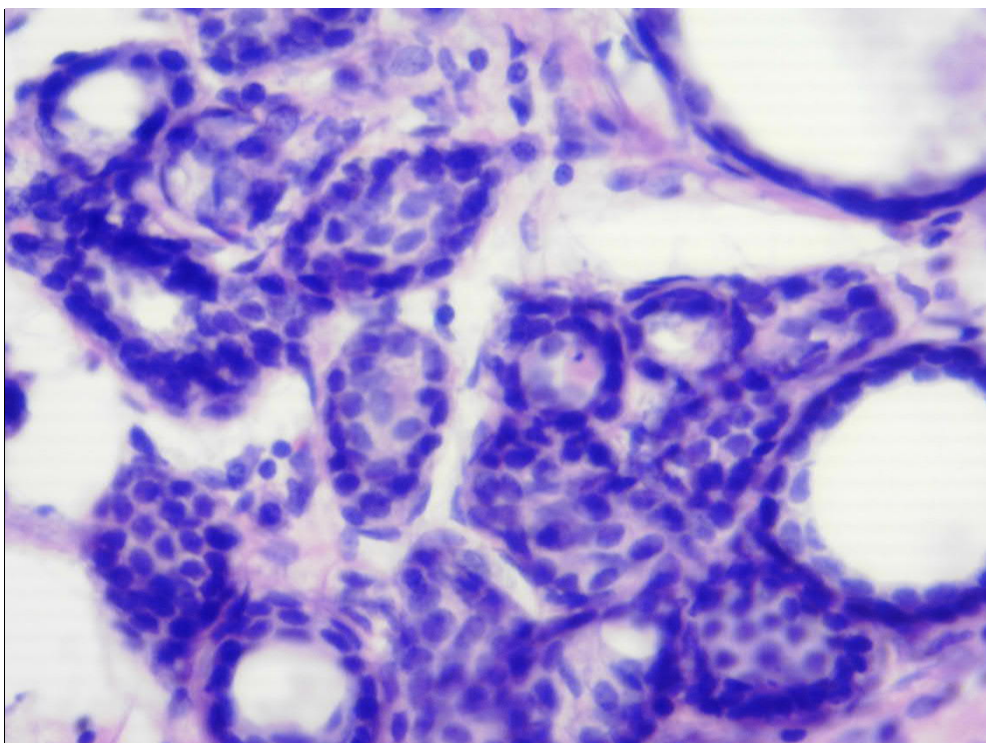


Рисунок 1.12 – Колагенові волокна і міоепітеліальний проліферат склерозуються

На основі експериментальних досліджень було вибрано основні морфологічні (гістологічні) ознаки фіброаденоми, які включають:

1. Проліферація альвеол.
2. Проліферація внутрішньодолькових протоків.
3. Наявність рихлої базофільної сполучної тканини.
4. Наявність грубої оксифільної сполучної тканини.
5. Протоки вистелені епітелієм і міоепітелієм різного функціонального стану.
6. Міоепітелій (видовжені темні клітини або світлі із шаровидними включеннями).
7. Формування химерних залозистих структур.
8. Гіаліноз сполучної тканини та атрофія епітелію.

В результаті експериментів було виділено ці якісні діагностичні ознаки в гістологічних препаратах.

В результаті аналізу морфологічних ознак сформульовано правило діагностування фіброаденоми:

ЯКЩО проліферація альвеол І проліферація внутрішньодолькових протоків І наявність рихлої базофільної сполучної тканини І наявність грубої оксифільної сполучної тканини І протоки вистелені епітелієм та міоепітелієм різного функціонального стану І міоепітелій (видовжені темні клітини або світлі із шаровидними включеннями) І формування химерних залозистих структур І гіаліноз сполучної тканини та атрофія епітелію, ТО фіброаденома.

На основі експериментів було вибрано основні морфологічні (гістологічні) ознаки проліферативної мастопатії, які включають:

1. Проліферація міоепітелію та ендотелію дрібних протоків.
2. Розширення міжчасточкових протоків.
3. Проліферація дрібних протоків і альвеол.
4. Незначна частина строми.
5. Відсутня базальна мембрана.
6. Проліферуючі міоепітеліальні клітини переміщуються у

внутрішньочасточкову сполучну тканину і стають подібними на гладенькі м'язи.

На основі досліджень було виділено якісні діагностичні ознаки в гістологічних препаратах.

В результаті аналізу морфологічних ознак сформульовано правило діагностування проліферативної мастопатії:

ЯКЩО проліферація міоепітелію та ендотелію дрібних проток і розширення міжчасточкових протоків і проліферація дрібних протоків і альвеол і незначна частинна строма і відсутня базальна мембрана і проліферуючі міоепітеліальні клітини переміщуються у внутрішньочасточкову сполучну тканину і стають схожі на гладенькі м'язи, ТО проліферативна мастопатія.

На основі експериментальних досліджень було вибрано основні морфологічні (гістологічні) ознаки непроліферативної мастопатії, які включають:

1. Проліферація сполучної тканини.
2. Гіаліноз сполучної тканини.
3. Атрофія залозистих ділянок з утворенням кіст.
4. Кістозно розширені протоки.

В результаті досліджень було виділено ці якісні діагностичні ознаки в гістологічних препаратах.

В результаті аналізу морфологічних ознак сформульовано правила діагностування непроліферативної мастопатії:

ЯКЩО гіаліноз сполучної тканини і проліферація сполучної тканини і (мілкі кісти із альвеол однієї дольки АБО кісти розташовуються гніздами АБО кістозно розширені протоки АБО атрофія залозистих ділянок і утворення кіст) ТО непроліферативна мастопатія.

ЯКЩО гіаліноз сполучної тканини і проліферація сполучної тканини і (мілкі кісти із альвеол однієї дольки АБО кісти розташовуються гніздами АБО кістозно розширені протоки АБО метаплазія темного епітелію в блідий (світлий) АБО утворення псевдососочків), ТО непроліферативна мастопатія.

ЯКЩО гіаліноз сполучної тканини І проліферація сполучної тканини І (мілкі кісти із альвеол однієї дольки АБО кісти розташовуються гніздами АБО наявність великої кількості сполучної тканини навколо залоз і протоків свідчить про їх атрофію АБО атрофія залозистих ділянок і утворення кіст), ТО непроліферативна мастопатія.

ЯКЩО гіаліноз сполучної тканини І проліферація сполучної тканини І (мілкі кісти із альвеол однієї дольки АБО кісти розташовуються гніздами АБО кістозно розширені протоки АБО метаплазія темного епітелію в блідий (світлий) АБО утворення псевдососочків), ТО непроліферативна мастопатія.

ЯКЩО гіаліноз сполучної тканини І проліферація сполучної тканини І (мілкі кісти із альвеол однієї дольки АБО кісти розташовуються гніздами АБО наявність великої кількості сполучної тканини навколо залоз і протоків свідчить про їх атрофію АБО атрофія залозистих ділянок і утворення кіст), ТО непроліферативна мастопатія.

1.4 Постановка задач

На основі проведеного аналізу в другого та третьому розділах необхідно розробити:

- алгоритм побудови скелетонів;
- алгоритм порівняння зображень на основі скелетонів в метриці

Фреше;

- алгоритм пошуку зображень за шаблонами;
- структуру програмного модуля;
- провести комп'ютерні експерименти.

1.5 Висновки до розділу 1

У першому розділі отримано такі результати:

- проаналізовано методи опрацювання зображень;
- здійснено аналіз методів пошуку зображень ;
- проведено аналіз гістологічних зображень.

2 АЛГОРИТМИ ПОШУКУ ЗОБРАЖЕНЬ

2.1 Алгоритм побудови скелетонів

Різноманітна геометрія граничних ліній об'єктів, мабуть, привела до людського поняття форми.

Точного однозначного визначення поняття форми немає. Для людського спілкування точне формулювання поняття форми не дуже важливе, але для алгоритмів аналізу і розпізнавання зображень строгі визначення всіх понять необхідні. Визначення форми повинно бути сформульовано у вигляді загальних математичних моделей. Адекватність цих моделей, що визначається як несуперечливість практичного досвіду людини, служить підтвердженням правильності сформульованого поняття в тих межах, в яких цей досвід є.

Абстрагуючись від конкретних фізичних властивостей об'єкта, людина зазвичай представляє його форму у вигляді суцільного просторового тіла в тривимірному просторі або у вигляді плоского об'єкта з неперервною границею в двовимірному просторі.

Одне з найбільш відповідних визначень форма – це зовнішні обриси, зовнішній вигляд предмета.

Інше визначення зовнішніх границь предмета це фігура – зовнішній обрис, форма чогось.

Найбільш поширені і доступні математичні моделі форми об'єктів реального світу дає евклідова геометрія. Геометричні фігури на площині і тіла в просторі надають спільну мову для опису форми навколишніх. Сама по собі потреба людей використовувати деякий понятійний апарат для визначення форми предметів навколишньої дійсності з'явилася, мабуть, одним з важливих джерел для створення і розвитку геометрії в давнину.

Для опису всієї множини плоских «суцільних» форм потрібно більш широке загальне визначення. З цієї точки зору прийнятною загальною математичною моделлю представляється поняття фігури [37].

Визначення 1. Жордановою кривою називається образ кола при неперервних ін'єктивних відображеннях на евклідову площину.

Визначення 2. Фігурою називається зв'язана замкнута область на площині, обмежена скінченим числом жорданових кривих, які не перетинаються.

Будемо використовувати такі позначення. Нехай \mathbb{R}^2 – евклідова площина з відстанню $d(p, q)$ між точками $p, q \in \mathbb{R}^2$.

Фігура A – це деяка множина точок на площині \mathbb{R}^2 . Границя фігури визначається як множина точок

$$\partial A = \{p: p \in \mathbb{R}^2, \forall r > 0, Sr(p) \cap A \neq \emptyset \text{ і } Sr(p) \cap \bar{A} \neq \emptyset\},$$

де $Sr(p) = \{q: q \in \mathbb{R}^2, (d(p, q) < r)\}$ – відкрите коло радіуса r з центром в точці p , $\bar{A} = \mathbb{R}^2 \setminus A$ – доповнення фігури A в просторі \mathbb{R}^2 .

Внутрішньою областю фігури називається множина $\hat{A} = A \setminus \partial A$, яка складається із внутрішніх точок фігури.

Таким чином, фігура в нашому розумінні – це зв'язана частина площини, обмежена однією або декількома замкнутими лініями. Під це визначення підходять суцільні об'єкти з кінцевим числом дірок. Тому математичну модель фігури ми виберемо в якості базової для представлення поняття форми.

Фігура є замкнута обмежена область. Як відомо, областю на евклідовій площині називається непорожня відкрита множина точок. Замкнутою областю називається замикання області, тобто замкнута множина, яка містить цю область. Іншими словами замкнута область – це об'єднання деякої області (відкритої) і всіх її граничних точок. Область називається обмеженою, якщо вона міститься цілком в деякому прямокутнику.

На рисунку 2.1 зображені замкнуті області, що представляють собою квадрат з трикутним отвором.

Теоретично фігура, як будь-яка множина точок може бути описана своєю індикаторною функцією $x(x, y)$, яка приймає значення 1 в точках, що належать множині, і 0 – в інших точках.



Рисунок 2.1 – Замкнені області: зліва – фігура, праворуч – ні

Граничне представлення фігури. Точка фігури називається граничною, якщо в будь-якій її околиці знаходяться точки, які не належать фігурі. Границею фігури називається множина всіх її граничних точок. Кожна така крива - це лінія без само перетинів, що утворює замкнутий контур. Якщо в граничній фігурі всього один такий контур, то фігура є однозв'язної, тобто в неї входять всі точки, що лежать всередині цього контуру. Якщо кількість контурів, що становлять границю, більше одного, то фігура є багатозв'язною. У цьому випадку один контур описує зовнішню границю фігури, а решта контури (один або більше) описують границі внутрішніх отворів фігури. Багатозв'язну фігуру будемо також називати фігурою з дірками.

Зауважимо, що відповідно до нашого визначенням фігурою не є ізольована точка, а також лінія.

Згідно з визначенням жорданової кривої, що є гомеоморфними образами кола, всі вхідні в неї точки можуть бути описані у вигляді множини

$$V(t) = (x(t), y(t)), \quad t \in [0, T], \quad (2.1)$$

де $x(t)$ і $y(t)$ – неперервна функція.

Умова того, що лінія (2.1) є замкнутою, є $V(0) = V(T)$. А умова відсутності само перетинів записується так: для всіх $t_1 \in (0, T), t_2 \in [0, T], t_1 \neq t_2$ має місце $V(t_1) \neq V(t_2)$.

Безпосередній опис граничних ліній фігури зовсім не обов'язково має включати в явному вигляді рівняння кривих (2.1). Наприклад, в тому випадку,

коли лінія є багатокутником, досить перерахувати його вершини V_0, V_1, \dots, V_n , де $V_0 = V_n$. Тоді явний опис границі може бути отримано у вигляді кусково-лінійної функції параметра $t \in [0, n]$, яка на напівінтервалах $t \in [i, i + 1], i = 0, \dots, n - 1$ приймає значення

$$V(t) = V_i * (i + 1 - t) + V_{i+1} * (t - i).$$

Граничний опис фігури надає багато корисної інформації для аналізу її форми. Так кількість граничних контурів визначає топологію фігури. У прикладі (рисунок 2.2) за розмірами і кількістю граничних контурів фігур можна розпізнавати кістки доміно.

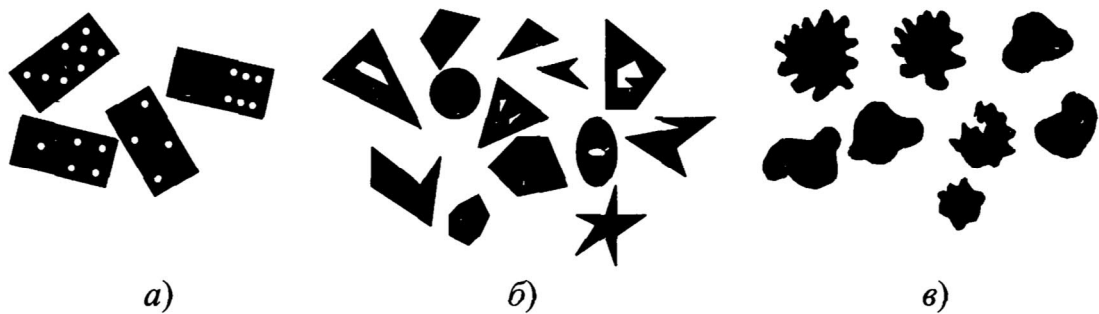


Рисунок 2.2 – Класифікація фігур за границею

Співвідношення між периметром і площею фігур, яке можна обчислити за відомою границею, дає можливість класифікувати фігури за ступенем гладкості границі.

Опис фігур за допомогою скелетонів. Опис фігури перерахуванням замкнутих ліній, що визначають контури її межі, є найбільш природним і простим. Явна опис границі фігури дає можливість вирішити багато завдань класифікації фігур, що виникають при аналізі форми зображень. Однак цей спосіб не є єдиним і не завжди достатнім для аналізу форми фігури. Існують різні практичні завдання, в яких форма об'єкта складається з окремих протяжних елементів, що мають вигляд «широких ліній», штрихів і т. п.

Прикладами є зображення тексту, фігури людини, візерунка папілярних ліній. Для аналізу форми об'єктів такого класу граничний опис є недостатньо інформативним. Наприклад, аналітичний опис трьох замкнутих контурів кордону фігури, зображеної на рисунку 2.3(г), не дозволяє побачити в цих контурах обриси літери В і відрізнити його від трьох контурів, що описують букву Ф.

А для замкнутих контурів на рисунку 2.3(а, б) дуже непросто визначити ділянки, що описують руки зображених людей. Граничне опис папілярних ліній у вигляді множини контурів на рисунку 2.3(в) не дозволяє аналізувати особливі точки цих ліній: точки злиття папілярних ліній, їх кінцеві точки і т.п. Тому для аналізу форми подібних фігур виникає необхідність вибору іншого, більш тонкого способу їх опису, який відображає як загальну структуру об'єкта, так і більш детальну структуру його елементів. на пальці людини (рисунок 2.3).

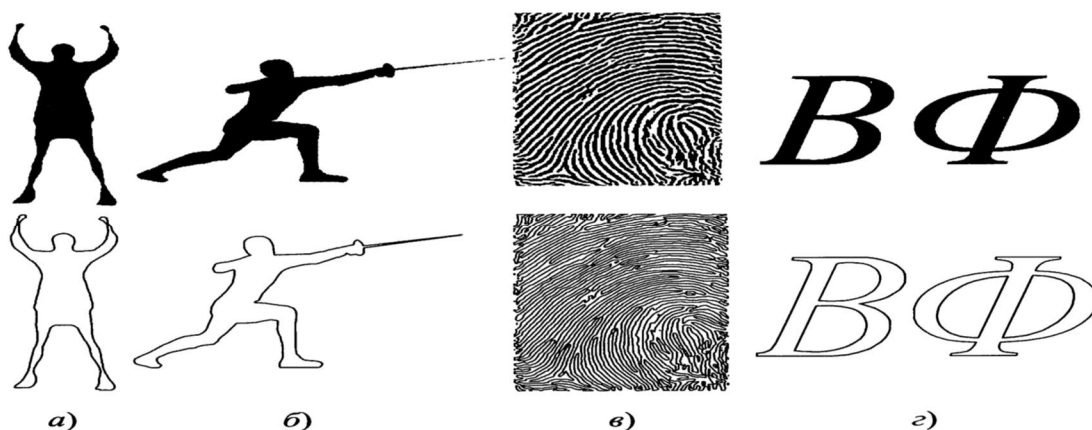


Рисунок 2.3 – Фігури і їх граничні форми

Таким чином, опис фігури у вигляді одновимірного плоского графу, так званого скелетону фігури. Існує кілька різних визначень скелетона. Ми будемо користуватися визначенням, яке ґрунтується на понятті максимального пустого кола.

Визначення 1. Порожнім колом фігури A називається замкнута множина точок $\tilde{S}_r(p) = \{q: q \in R^2, d(p, q) \leq r\}$ таке, що $\tilde{S}_r(p) \subset A$.

Це множина є коло радіуса $r \geq 0$, з центром в точці $p \in R^2$. Поряд з терміном «порожнє коло» ми будемо використовувати також як синонім термін «порожнє коло», розуміючи під ним коло, яке цілком лежить в фігурі разом зі своїми внутрішніми точками. Залежно від контексту під околom будемо розуміти або лінію, або замкнуту область, обмежену цією лінією.

Згідно з визначенням 1 граничні точки фігури не можуть лежати всередині порожнього кола. Саме в цьому сенсі він називається порожнім – він не містить в собі граничних точок фігури.

Порожній коло цілком лежить в фігурі. Окремим випадком пусого кола є будь-яка точка фігури $p \in A$, розглянута як окіл нульового радіуса $p = \tilde{S}_0(p)$. Такі порожні кола будемо називати виродженими.

Очевидно, що множина всіх порожніх кіл покриває фігуру, а їх об'єднання збігається з цією фігурою.

На множині порожніх кіл природним чином задається відношення часткового порядку: коло $\tilde{S}_r(p)$ більше круга $\tilde{S}_t(q)$, якщо $\tilde{S}_t(q) \subset \tilde{S}_r(p)$, перше коло накриває другий. Для такого відношення порядку існують максимальні елементи, так звані максимальні порожні кола.

Визначення 2. Максимальним порожнім колом називається пусте коло, яке не міститься ні в якому іншому порожньому колі.

Максимальні порожні кола фігури будемо також називати вписаними колами.

На рисунку 2.4 представлені порожні кола фігури.

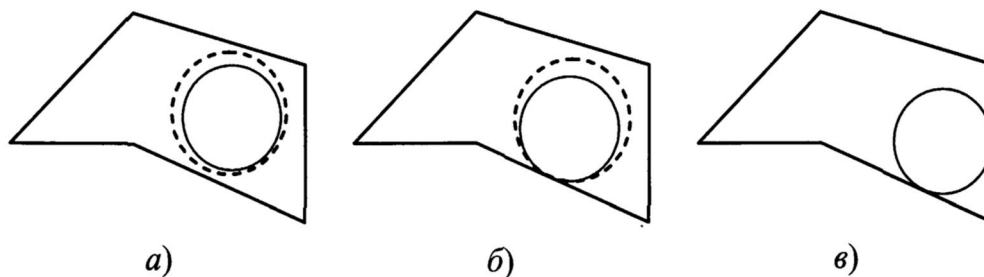


Рисунок 2.4 – Порожні кола фігури

Якщо невироджене порожнє коло не має спільних точок з границею фігури (рисунок 2.4(а)) або має всього одну спільну точку (рисунок 2.4(б)), то він не є максимальним, оскільки в цих випадках існують інші порожні кола. Тому для невиродженого максимального порожнього кола характерно дотик границі фігури не менш ніж в двох точках (рисунок 2.4(в)). Вироджені кола з центрами в опуклих кутових вершинах фігури є максимальними пустими колами, оскільки не містяться в інших порожніх колах.

На рисунку 2.5(б) представлені приклади максимальних порожніх кіл. Кола А і В – вироджені, вони мають нульовий радіус. Їх центри – це точки зламу границі фігури. Кола С і D мають з границею множини спільних точок, оскільки границя фігури в даному прикладі містить дуги кіл. Кола Е і F мають по три загальних точки з границею фігури. Решта кіл G, H, I дотикаються границі в двох точках.

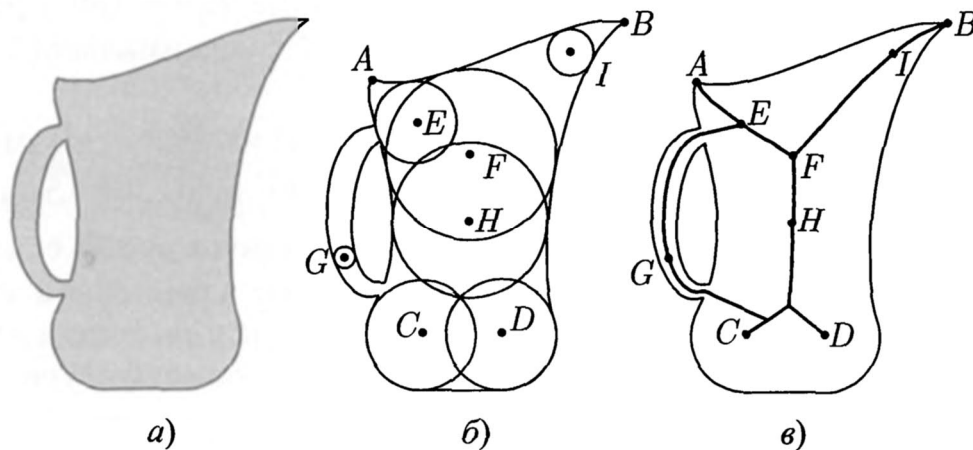


Рисунок 2.5 – Фігура і її границя (а), приклади максимальних порожніх кіл (б), скелетон фігури (в)

Визначення 3. Скелетоном фігури називається множина центрів всіх її максимальних порожніх кіл.

Скелетон можна розглядати як плоский граф (рисунок 2.5(в)), вершинами якого є центри максимальних порожніх кіл, що мають або одну загальну точку з границею, або три і більше загальних точки. А ребра графа є лінії, які

складаються з центрів тих порожніх кіл, які дотикаються границі фігури рівно в двох точках. З цієї причини ребра скелетона називають також серединними осями фігури.

Таким чином, скелетон фігури, як і її границя, є одновимірним образом. Але на відміну від границі цей образ має більш складну структуру – це не набір замкнутих ліній без само перетинів, а зв'язний плоский граф.

Скелетон є досить зручним і інформативним інструментом для аналізу топологічних і метричних властивостей фігури, для проходження її структури. При цьому він несе інформацію, яку досить складно було б витягти з граничного опису фігури. У прикладі на рисунку 2.6 приведені скелетони фігур, зображених на рисунку 2.3. Як неважко побачити, з скелетонів людських фігур можна зрозуміти положення кінцівок і жести. З скелета папілярного візерунка легко виділити окремі папілярні лінії, знайти точки їх з'єднання і кінцеві точки. Скелети букв В і Ф мають істотні топологічні відмінності, в той час як контурні опису їх топологічно однакові (фігури з двома дірками).

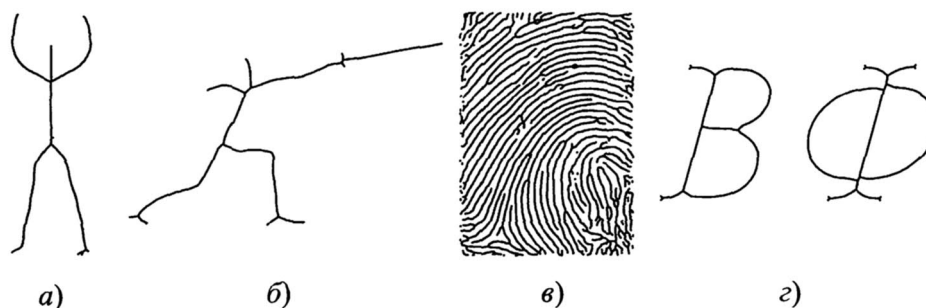


Рисунок 2.6 – Скелет фігур

Оскільки скелетон будується на основі множини кіл, вписаних в фігуру, з кожною точкою скелетона можна зв'язати «ширину» фігури в цій точці.

Визначення 4. Будемо називати радіальної функцією точки скелета величину радіуса максимального порожнього кола з центром в цій точці.

Скелетон фігури разом зі своєю радіальної функцією однозначно задає все множину максимальних порожніх кіл фігури. А об'єднання цієї множини кіл, як зазначалося вище, збігається з самою фігурою. Таким чином, скелетон з

радіальної функцією є поданням фігури, еквівалентним як її опису за допомогою індикаторного функції, так і граничного опису.

Хороші властивості скелетонів зумовили широке їх застосування для аналізу форми зображень в розпізнаванні образів, комп'ютерній графіці і візуалізації, стисненні відеоінформації і т.п.

Алгоритм «пожежа в прерії». Розглянемо деякі характерні властивості скелетона, які можуть бути покладені в основу визначення скелетона і використані в алгоритмах скелетонізації.

Часто в якості скелетона розглядають множину серединних осей фігури. Назвемо точку фігури симетричної, якщо вона має дві або більше найближчих точки на границі фігури. Множина всіх симетричних точок називається середині осями фігури.

Неважко зрозуміти, що кожна симетрична точка є центр максимального порожнього кола і тому належить скелетону. Однак, центри вироджених максимальних порожніх кіл хоч і належать скелетону, але не є симетричними, отже, не належать серединним осям. Вони і визначають відмінність між скелетоном і множиною серединних осей. Але ці точки є потрібними точками для серединних осей. Тому, якщо ними поповнити серединні осі, розглянувши замикання серединних осей, то отримані множини точок дійсно співпадуть із скелетоном. Таким чином, правильним альтернативним визначенням скелетона є наступне: скелетон фігури є замикання її серединних осей.

Ще одне визначення використовує фізичну модель, що базується на метафорі «пожежі в прерії». Побудова скелета неперервної фігури може бути наочно інтерпретовано у вигляді процесу розповсюдження вогню при «вигоранні» фігури. Вздовж усієї границі фігури одночасно загоряється вогонь і поступово спалює всю фігуру, рухаючись від границі всередину фігури. Якщо припустити, що вогонь поширюється з постійною швидкістю в усіх напрямках, то від границі всередину фігури буде переміщатися вогняний фронт (рисунок 2.7).

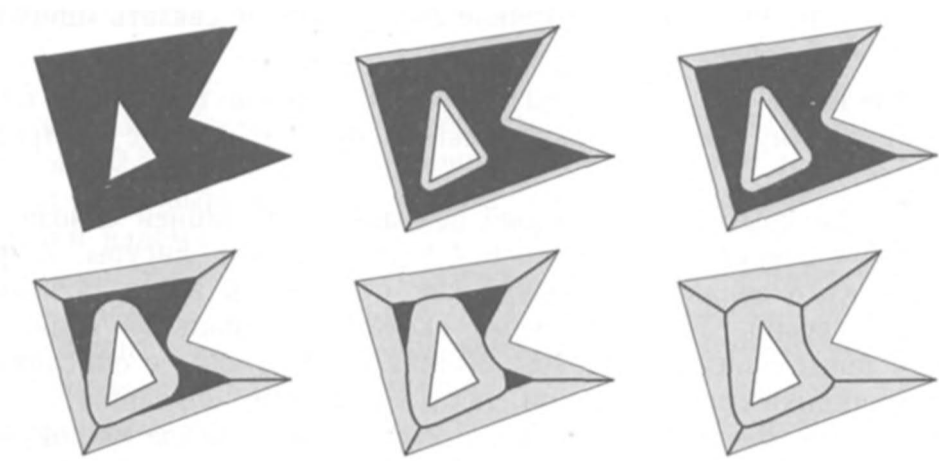


Рисунок 2.7 – «Пожежа в прерії» і утворення скелетона

Розглянемо точки «згасання вогню», тобто ті точки всередині фігури, в яких відбувається зустріч кількох (двох або більше) вогняних фронтів. Вогонь приходить в ці точки одночасно з декількох напрямків і гасне через те, що тут більше нема чому горіти. Кожна така точка, в силу постійної і однакової швидкості поширення вогню, однаково віддалена від початкового положення фронтів, тобто від граничних точок, в яких зародився вогонь. З іншого боку, ці граничні точки є найближчими до неї точками границі. Отже, точки згасання лежать на серединних осях фігури. Отже, множина точок «згасання вогню» просто збігається з середніми осями. Тому замикання множини точок згасання вогню збігається зі скелетоном фігури.

Обчислення і регуляризація скелетона. Скелетон фігури є досить зручною моделлю для описання її форми. Однак практичне використання цієї моделі має дві проблеми, що носять фундаментальний характер. Це проблема обчислення скелета і проблема регуляризації скелета.

Обчислення скелетона фігури за описом її граничних контурів є складна задача. Аналітичний розв'язок цього завдання в загальному випадку, коли контури границі описуються довільними кривими, отримати неможливо. Аналітичний розв'язок існує лише для деяких спеціальних класів кривих, що описують границі фігури. Зокрема, точне геометричне рішення можна отримати для фігури з кусково-лінійною границею. Такі фігури називаються

багатокутними. В принципі, будь-яку фігуру можна як завгодно з високою точністю апроксимувати полігоном. Однак така апроксимація породжує нові труднощі.

Побудова високоточної багатокутної апроксимації фігур на зображеннях, що обробляються в системах машинного зору, приводить до сцен, в яких кількість граничних багатокутників має порядок $k = 10^2 \div 10^3$, а загальне число вершин багатокутників – $n = 10^4 \div 10^5$. При цьому реальний час рішення завдань обробки сцен в системах машинного зору має становити десяті частки секунди. Очевидно, що в цьому випадку алгоритми, які використовують попарний перебір всіх вершин або ребер багатокутників і мають внаслідок цього обчислювальну складність $O(n^2)$, є неприйнятними.

Швидкі алгоритми побудови скелетів простих багатокутників з n вершинами мають обчислювальну складність $O(n \log n)$ в гіршому випадку [37]. Однак відомі узагальнення цих алгоритмів на випадок багатокутної фігури [37] мають обчислювальну складність $O(kn + n \log n)$, де k – кількість багатокутних дірок. Для задач, в яких k і n мають порядок 10^3 і 10^5 відповідно, така складність неприйнятна. У той же час, ефективні алгоритми побудови діаграми Вороного множини лінійних сегментів [37] в силу своєї універсальності не використовують специфічних особливостей множини відрізків, що становлять границю багатокутної фігури. Зокрема, ці алгоритми будують скелет не тільки всередині, а й поза багатокутної фігури, що є зайвою роботою.

Таким чином, виникає задача створення ефективного обчислювального алгоритму скелетонізації багатокутної фігури.

Однак практичне застосування скелетонів для аналізу форми ставить ще одну, набагато гіршу задачу, пов'язану з інтерпретацією отриманого скелетона багатокутної фігури. Дотримуючись термінології [37], це завдання будемо називати регуляризацією скелетона.

Необхідність регуляризації скелетона пов'язана з наявністю «шумових ефектів» в побудованих скелетони. Скелетон фігури дуже чутливий до

локальних властивостей її границі. Це добре помітно на прикладі фігур, Рисунки не відрізняються з точки зору їх границь, але мають різні скелети (рисунок 2.8).

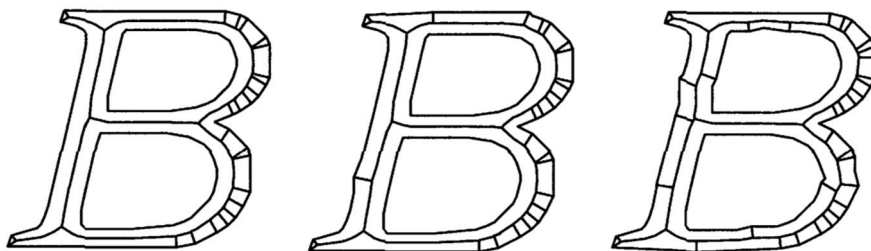


Рисунок. 2.8 – Вплив шумів на структуру скелетона

Невеликі викривлення границі приводять до значних змін топологічної структури скелетного графа. Тому, якщо для деякої форми побудувати дві різні багатокутні фігури, але апроксимувати її з однією і тією ж точністю, можна отримати значно різні скелети.

З кожної опуклою вершиною багатокутної фігури пов'язана гілка скелетона. Тому чим більше вершин в фігурі, тим більше гілок в скелеті. І більшість цих гілок – несуттєві, тому що не несуть ніякої інформації для аналізу форми фігури. Разом з тим, інтуїтивно очевидно, що подібні фігури мають і подібні скелетони. Ця подібність скелетонів виражається в наявності у них деякої суттєвої загальної частини, яка і характеризує структуру фігури. Завдання виділення цієї загальної частини називається регуляризацією скелетона. Вона зводиться до очищення отриманого скелетона від шуму. Для її розв'язку також повинні бути розроблені відповідні ефективні алгоритми.

Таким чином, практичне використання скелетонів в задачах обробки та аналізу зображень вимагає розробки ефективних алгоритмів скелетонізації багатокутних фігур і алгоритмів регуляризації скелетонів шляхом очищення їх від шумових ефектів.

2.2 Порівняння зображень на основі скелетонів в метриці Фреше

Отже, скелетон в загальному випадку для R^2 випадку є планарним графом. Після регуляризації скелетону отримаємо дерево, тобто граф без циклів [6, 38, 39].

Дерево – це зв'язний граф без циклів. Дерево називають кореневим, якщо у ньому вибрано точку (корінь). Ми розглядаємо топологічні дерева в R^n (R^2), тобто топологічні вкладення дерев-графів. Отже, топологічне дерево – це трійка (T, α, x_0) , де $T \subseteq R^n$, $x_0 \in T$ і існує граф-дерево T' та вкладення $f : T' \rightarrow R^n$, $\alpha(T') = T$, $f(x'_0) = x_0$.

На кожному топологічному дереві T існує, таким чином, дві метрики. Перша індукована з R^n (вона позначається через d) і друга – індукована з геодезійної метрики в T' відображенням α – її позначаємо через ρ_T . Як знайти $\rho_T(x, y)$? Треба взяти прообрази $\alpha^{-1}(x)$, $\alpha^{-1}(y)$ в T' та поміряти в графі T' довжину відрізка, що з'єднує ці точки.

Відображення графів називають монотонним, якщо воно монотонне на всіх відрізках, що починаються з коренів (рисунок 2.9).

Тобто, відображення $f : T \rightarrow S$ є монотонним, якщо $f(x_0) = y_0$ (корінь в S) і при віддаленні x від x_0 вздовж відрізка (у метриці ρ_T) точка $f(x)$ монотонно віддаляється від y_0 (у метриці ρ_S).

Визначимо відстань Фреше між топологічними деревами T і S так

$$d_F(T, S) = \inf \{ \sup \{ d(f(x), g(x)) \mid x \in R \} \mid f : R \rightarrow T, g : R \rightarrow S \text{ – сюр'єктивні монотонні відображення дерев} \}$$

Розглянемо відображення дерев з рисунку 2.9.

f відображає ліву частину R на T , а праву – вздовж правої частини T .

g відображає праву частину R на праву частину S , ліву частину R на ліву частину S , складаючи два кінці розвилки.

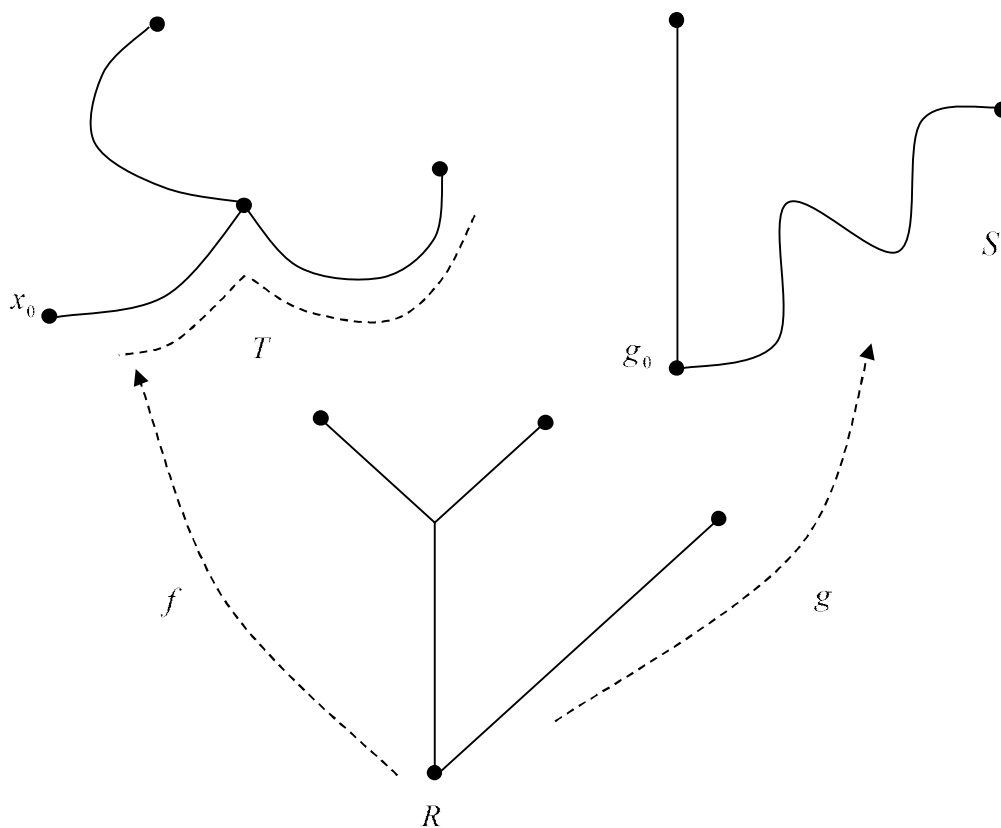


Рисунок 2.9 – Відображення дерев

Розглянемо кореневі дерева на площині R^2 . Якщо (T, t_0) – таке дерево, то для кожного $t \in T$ через $\|t\|$ позначаємо $d(t, t_0)$. Тут і далі, через d позначається геодезійна відстань всередині дерева, тобто $d(t, t_0)$ означає довжину дуги всередині T , що з'єднує t і t_0 .

Відображення корневих дерев.

Вважаємо, що $f: (S, s_0) \rightarrow (T, t_0)$ має ту властивість, що $f(s_0) = t_0$, і для кожного геодезійного сегмента J , що йде від точки t_0 маємо:

$$s, s' \in J, \|s\| \leq \|s'\| \Rightarrow \|f(s)\| \leq \|f(s')\|.$$

Для означення метрики Фреше на множині всіх вкладених дерев в R^2 позначимо через ρ евклідову метрику в R^2 .

Приймемо

$$D(T, S) = \inf \left\{ \sup \{ \rho(f(r), g(r)) \mid r \in R \} \mid f : R \rightarrow T, g : R \rightarrow S \text{ – відображення "на"} \right\}$$

Спочатку нам потрібно переконалися у коректності заданого означення. Маючи T, S , позначимо через R букет цих двох дерев, $R = T \cup S$ (тобто фактор-простір об'єднання $T \cup S$ відносно відношення еквівалентності, що отожднює точку t_0 і s_0) (рисунок 2.10).

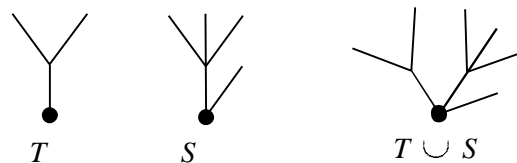


Рисунок 2.10 – Фактор-простір об'єднання $T \cup S$ відносно відношення еквівалентності

Тоді $f : R \rightarrow T$ можна взяти відображення, що деформує S в точку, а $g : R \rightarrow S$ – відповідно, як відображення, що деформує T в точку.

Теорема. Функція D є метрика.

Доведення.

1) Очевидно, що $D(T, S) \geq 0$ для кожних T, S . Очевидно також, що $D(T, T) = 0$ (для цього досить взяти за f і g тотожне відображення).

Нехай тепер $D(T, S) = 0$. Звідки нескладно вивести, що $\rho_H(T, S) = 0$ (через ρ_H позначимо метрику Гаусдорфа на площині).

Таким чином, $T = S$.

2) Симетричність функції D безпосередньо випливає з означення.

3) Нерівність трикутника.

Нехай T, S, U – дерева і $\varepsilon > 0$. Існують дерева P, R і k відображення «на» $f: P \rightarrow T$, $g: P \rightarrow S$ та $k: R \rightarrow S$, $h: R \rightarrow U$ такі, що

$$\sup\{\rho(f(p), g(p)) \mid p \in P\} \leq D(T, S) + \varepsilon,$$

$$\sup\{\rho(k(r), h(r)) \mid r \in R\} \leq D(S, U) + \varepsilon.$$

Розглянемо пулбек (pullback) дерев P і R , тобто дерево $Q = \{(p, r) \in P \times R \mid g(p) = k(r)\}$. Нехай $\alpha: Q \rightarrow P$, $\beta: Q \rightarrow R$ – проєкції.

Тоді для кожного $(p, r) \in Q$ маємо

$$\begin{aligned} \rho(f\alpha(p, r), h\beta(p, r)) &= \rho(f(p), h(r)) \\ &\leq \rho(f(p), g\alpha(p, r)) + \rho(k\beta(p, r), h(r)) = \\ &\rho(f(p), g(p)) + \rho(k(r), h(r)) \leq D(T, S) + D(S, U). \end{aligned}$$

В результаті скелетизації отримуємо скелетон Sk зображення, який є деревом. Отже, скелетон $Sk = G = (V, E)$ є неорієнтованим графом, де $V = \{v_0, v_1, \dots, v_k\}$ – множина вершин, $E = \{u_0, u_1, \dots, u_m\}$ – множина ребер. Для неорієнтованого графу ребро є множиною $\{u, v\}$, де $u, v \in V$, $u \neq v$.

Для заданого графу $G = (V, E)$ дорога (маршрут) довжини k від вершини u до вершини u' є послідовність вершин $\{v_0, v_1, \dots, v_k\}$ таких що $u = v_0$, $u' = v_k$, $(v_{i-1}, v_i) \in E$ для $i = 1, 2, \dots, k$. Шлях включає вершини v_0, v_1, \dots, v_k і ребра (v_0, v_1) , (v_1, v_2) , ..., (v_{k-1}, v_k) . Якщо є шлях P від вершини u до вершини u' то говорять, що вершина u' досяжна із вершини u по шляху P , тобто $u \xrightarrow{P} u'$. Шлях називають простим, якщо всі вершини шляху різні.

Дерева графу бувають без виділеного кореня і з виділеним коренем, тобто вільні дерева. Для них справедлива така теорема.

Теорема [40]. Нехай $G = (V, E)$ неорієнтований граф. Тоді справедливо:

1. G є вільне дерево.
2. Дві будь-які вершини G з'єднуються за допомогою єдиного простого шляху.
3. G є зв'язний граф, але при видаленні з E будь-якого ребра перестає бути таким.
4. G є зв'язний граф, $|E| = |V| - 1$.
5. G є ациклічний граф, $|E| = |V| - 1$.
6. G – ациклічний граф, але при додаванні будь-якого ребра в E отримуємо граф, який включає цикл.

Ми будемо використовувати дерева з коренем. Дерево з коренем – це вільне дерево, в якому виділена одна вершина, яка називається коренем. Представимо дерево $G = (V, E)$ за допомогою матриці суміжності A . Вершини повинні бути пронумеровані і впорядковані числами $1, 2, \dots, |V|$. Тоді матриця суміжності $A = (a_{ij})$, розміром $|V| \times |V|$ така, що

$$a_{ij} = \begin{cases} 1, & \text{якщо } (i, j) \in E, \\ 0, & \text{в іншому випадку.} \end{cases}$$

Матриця суміжності A для неорієнтованого графу рівна транспонованій матриці A^T . Тому будемо розглядати тільки елементи матриці, які знаходяться на головній діагоналі і вище, тобто субматрицю A^* .

Алгоритм переходу від субматриці A^* графу до множини його шляхів.

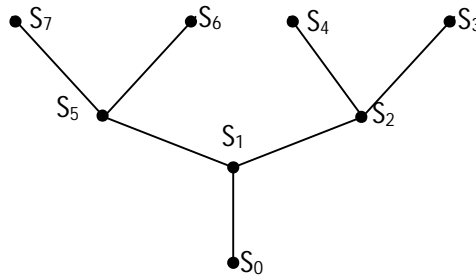
1. Формуємо субматрицю A^* на основі матриці інцидентності A графу G .
2. На її основі формуємо масив кінцевих вершин $M = \{m_1, m_2, \dots, m_k\}$.
3. Використовуючи субматрицю A^* та масив кінцевих вершин M знаходимо множину шляхів $P = \{p_1, p_2, \dots, p_k\}$, кожен з яких є підмножиною $p_i = \{v_0, v_r, \dots, v_i\}$ $i = \overline{1, k}$, де k – кількість кінцевих вершин.

Нехай задана субматриця $A^* = \begin{pmatrix} a_{01} & a_{02} & \dots & a_{0k} & \dots & a_{0m} \\ & a_{11} & \dots & a_{1k} & \dots & a_{1m} \\ & & O & M & & \\ & & & a_{kk} & M & M \\ & & & & O & \\ & & & & & a_{mm} \end{pmatrix}, a_{ij} \quad i = \overline{0, m},$

$j = \overline{0, m}$ описують елементи субматриці.

Множина шляхів $P = \{p_1, p_2, \dots, p_k\}$, де $p_1 = \{v_0, v_1, \dots, v_o\}$, $p_2 = \{v_0, v_1, \dots, v_m\}$, $\dots, p_k = \{v_0, v_1, \dots, v_n\}$.

Наприклад, нехай задано дерево $G_1 = (V_1, E_1)$.



$$A^* = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ & & 0 & 1 & 1 & 0 & 0 & 0 \\ & & & 0 & 0 & 0 & 0 & 0 \\ & & & & 0 & 0 & 0 & 0 \\ & & & & & 0 & 1 & 1 \\ & & & & & & 0 & 0 \\ & & & & & & & 0 \end{pmatrix} \begin{matrix} S_0 \\ S_1 \\ S_2 \\ S_3 \\ S_4 \\ S_5 \\ S_6 \\ S_7 \end{matrix}$$

Тоді множина шляхів для графу $G_1 = (V_1, E_1)$ дорівнює $P = \{\{S_0 \rightarrow S_1 \rightarrow S_2 \rightarrow S_3\}, \{S_0 \rightarrow S_1 \rightarrow S_2 \rightarrow S_4\}, \{S_0 \rightarrow S_1 \rightarrow S_5 \rightarrow S_6\}, \{S_0 \rightarrow S_1 \rightarrow S_5 \rightarrow S_7\}\}$

2.3 Алгоритм пошуку зображень за шаблонами

Розробимо паралельний алгоритм пошуку об'єктів на зображенні.

Пошук за шаблоном на декількох зображеннях є незалежною задачею, тому доцільно застосовувати процес розпаралелення [41]. Ярусно-паралельну форму представлення розпаралелення процесу пошуку елементів за шаблоном наведено на рисунку 2.11.

На першому етапі відбувається завантаження шуканого шаблону в пам'ять комп'ютера. На наступному проходить розгалуження процесу обробки зображень, оскільки дані етапи є незалежними і відповідно їх можна розподілити на окремі потоки. В результаті роботи на виході отримується нове зображення і якщо на ньому присутні подібні ядра клітин, то вони будуть відповідно виділені. Надалі лікар – діагност матиме можливість детальніше звернути увагу на ті цитологічні зображення, на яких знайдено відповідні області.

Алгоритм пошуку мікрооб'єктів зі шаблоном складається з таких кроків:

- 1) Виділення контурів ядер клітин за алгоритмом, розробленим у статті [42].
- 2) Ідентифікація трьох точок на контурах ядер клітин і обчислення афінних коефіцієнтів [43].

Коефіцієнти афінних перетворень, що описують зміну форми і розміщення початкового ядра отримуються із системи рівнянь:

$$\begin{bmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{bmatrix} \cdot \begin{bmatrix} a \\ b \\ e \end{bmatrix} = \begin{bmatrix} \tilde{x}_1 \\ \tilde{x}_2 \\ \tilde{x}_3 \end{bmatrix},$$

$$\begin{bmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{bmatrix} \cdot \begin{bmatrix} c \\ d \\ f \end{bmatrix} = \begin{bmatrix} \tilde{y}_1 \\ \tilde{y}_2 \\ \tilde{y}_3 \end{bmatrix}.$$

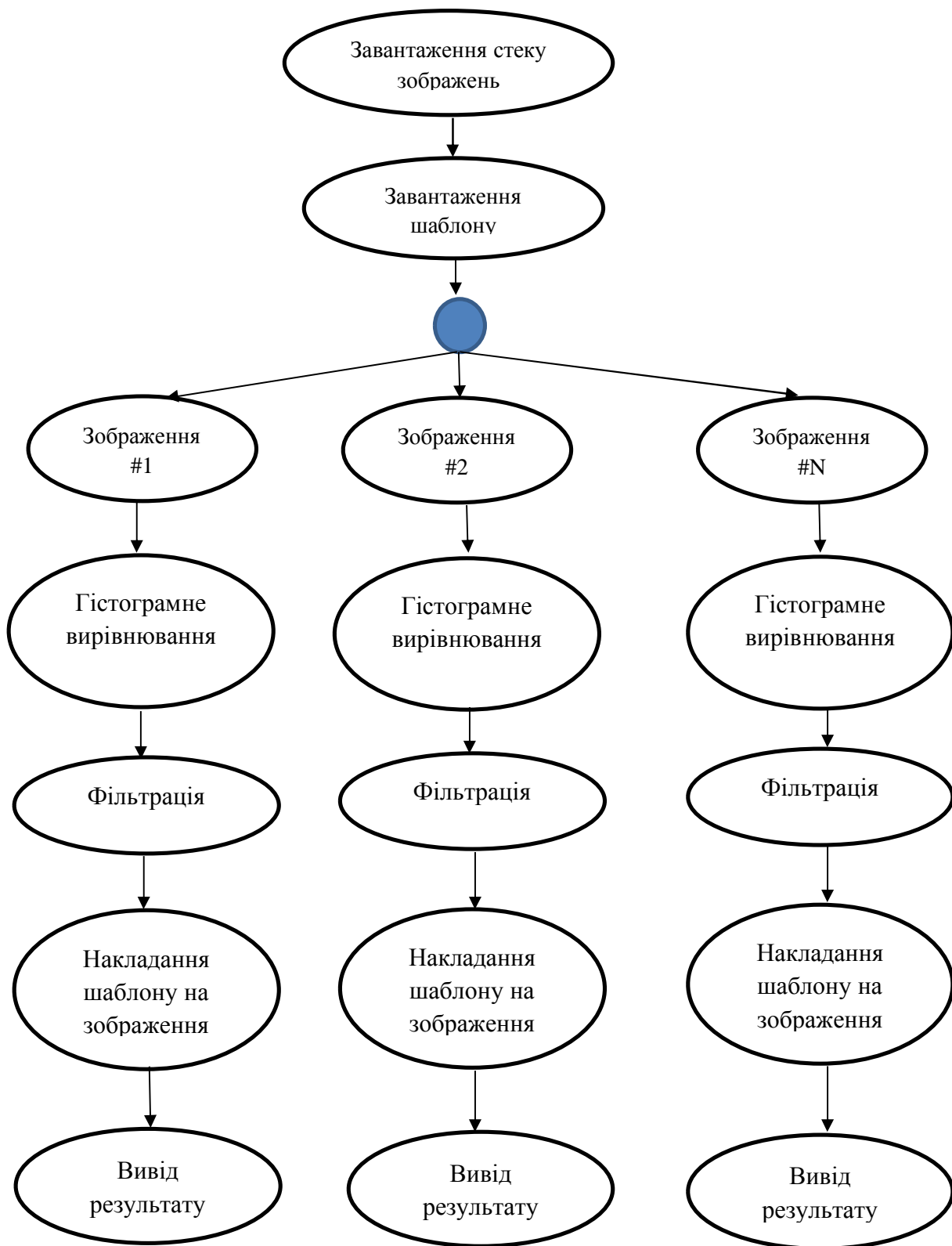


Рисунок 2.11 – Ярусно-паралельна форма представлення розпаралелення процесу пошуку елементів за шаблоном

$(\tilde{x}_1, \tilde{y}_1), (\tilde{x}_2, \tilde{y}_2), (\tilde{x}_3, \tilde{y}_3)$ – точки, що належать другому ядру;

a, b, c, d, e, f – коефіцієнти афінного перетворення, що відображають перше ядро в друге.

Невідомі коефіцієнти визначаються так:

$$a = \frac{y_1(\tilde{x}_3 - \tilde{x}_2) + y_2(\tilde{x}_1 - \tilde{x}_3) + y_3(\tilde{x}_2 - \tilde{x}_1)}{y_1(x_3 - x_2) + y_2(x_1 - x_3) + y_3(x_2 - x_1)},$$

$$b = \frac{x_1(\tilde{x}_2 - \tilde{x}_3) + x_2(\tilde{x}_3 - \tilde{x}_1) + x_3(\tilde{x}_1 - \tilde{x}_2)}{y_1(x_3 - x_2) + y_2(x_1 - x_3) + y_3(x_2 - x_1)},$$

$$e = \frac{x_1(y_2\tilde{x}_3 - y_3\tilde{x}_2) + x_2(y_3\tilde{x}_1 - y_1\tilde{x}_3) + x_3(y_1\tilde{x}_2 - y_2\tilde{x}_1)}{y_1(x_3 - x_2) + y_2(x_1 - x_3) + y_3(x_2 - x_1)},$$

$$c = \frac{y_1(\tilde{y}_3 - \tilde{y}_2) + y_2(\tilde{y}_1 - \tilde{y}_3) + y_3(\tilde{y}_2 - \tilde{y}_1)}{y_1(x_3 - x_2) + y_2(x_1 - x_3) + y_3(x_2 - x_1)},$$

$$d = \frac{x_1(\tilde{y}_2 - \tilde{y}_3) + x_2(\tilde{y}_3 - \tilde{y}_1) + x_3(\tilde{y}_1 - \tilde{y}_2)}{y_1(x_3 - x_2) + y_2(x_1 - x_3) + y_3(x_2 - x_1)},$$

$$f = \frac{x_1(y_2\tilde{y}_3 - y_3\tilde{y}_2) + x_2(y_3\tilde{y}_1 - y_1\tilde{y}_3) + x_3(y_1\tilde{y}_2 - y_2\tilde{y}_1)}{y_1(x_3 - x_2) + y_2(x_1 - x_3) + y_3(x_2 - x_1)}.$$

3) Пересування шаблону по усіх можливих позиціях та розрахунок для кожної позиції «подібності» між шаблоном і областю на зображенні.

4) Обчислення значення найбільшої подібності здійснюється згідно формули:

$$cor = \frac{\sum_{i=0}^{N-1} (x_i - \bar{x}) * (y_i - \bar{y})}{\sqrt{\sum_{i=0}^{N-1} (x_i - \bar{x})^2 \sum_{i=0}^{N-1} (y_i - \bar{y})^2}},$$

де x_i - значення пікселя шаблонного зображення,

\bar{x} - середнє значення пікселів в шаблонному зображенні,

y_i - значення пікселя вхідного зображення,

\bar{y} - середнє значення пікселів вхідного зображення,

N – загальна кількість пікселів.

Результати роботи програмного модулю для пошуку об'єкту на 20 вхідних зображеннях розміром 3000 x 3000 пікселів наведено на рисунку 2.12.

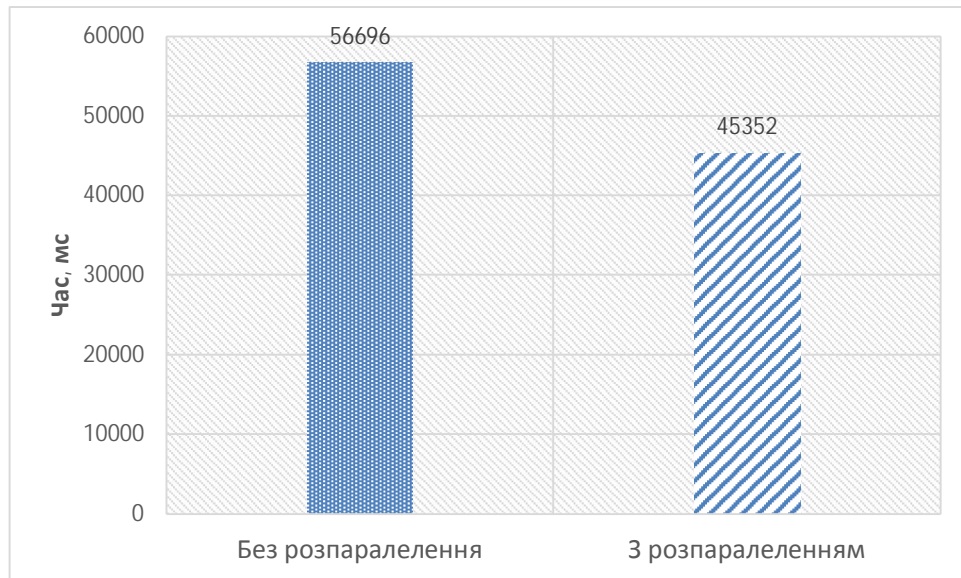


Рисунок 2.12 – Час пошуку об'єктів за шаблоном для стеку з 20 зображень

Як видно з рисунку 2.12 час роботи програмного модулю з пошуку об'єктів за шаблоном на 20 широкоформатних зображеннях близько на 20% менший, ніж без використання розпаралелення. Було проведено 5 спроб для кожного випадку і так, як час роботи програм був різний, на графіку відображено середній час для кожного випадку. Даний результат підтверджує актуальність задачі розпаралелення процесів опрацювання зображень.

2.4 Висновки до розділу 2

У другому розділі отримано такі результати:

- розроблено алгоритм побудови скелетонів;
- розроблено алгоритм порівняння зображень на основі скелетонів в метриці Фреше.
- розроблено алгоритм пошуку зображень за шаблонами.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ АЛГОРИТМУ ПОШУКУ МІКРООБ'ЄКТІВ НА ЗОБРАЖЕННЯХ

3.1 Структура програмного модулю

Розробка правильної структури програмного модулю відіграє важливе значення в процесі проектування великої програмної системи і гарантує, що розроблена система матиме можливості для зручного редагування структури та її розширення.

Розроблена інформаційна система, як і будь – яка інша має складатись із мінімум трьох базових функціональні складових:

- модуль зберігання даних;
- модуль обробки даних;
- модуль графічного інтерфейсу користувачем.

При правильному проектуванню усі три модулі повинні бути незалежними між собою.

Враховуючи специфіку розробленого програмного забезпечення та безпосередньо самої задачі було обрано підхід до розробки програмного забезпечення для десктопної версії, тобто розроблена програма буде завантажуватись на локальному комп'ютері користувача. Оскільки обробка гістологічних та цитологічних зображень вимагає використання великої кількості оперативної пам'яті, то використання клієнт – серверної архітектури у даному випадку не є доцільною. Використання великої кількості оперативної пам'яті зумовлена перш за все великим обсягом зображень, що може становити в середньому 25 мегабайт. Також потрібно врахувати, що в процесі виконання програма потрібно буде здійснювати значну кількість операцій над зображеннями, наприклад фільтрацію, сегментацію тощо. Виконання даних операцій забирає значну кількість ресурсів обчислювального пристрою.

Структуру розробленого програмного модулю наведено на рисунку 3.1.

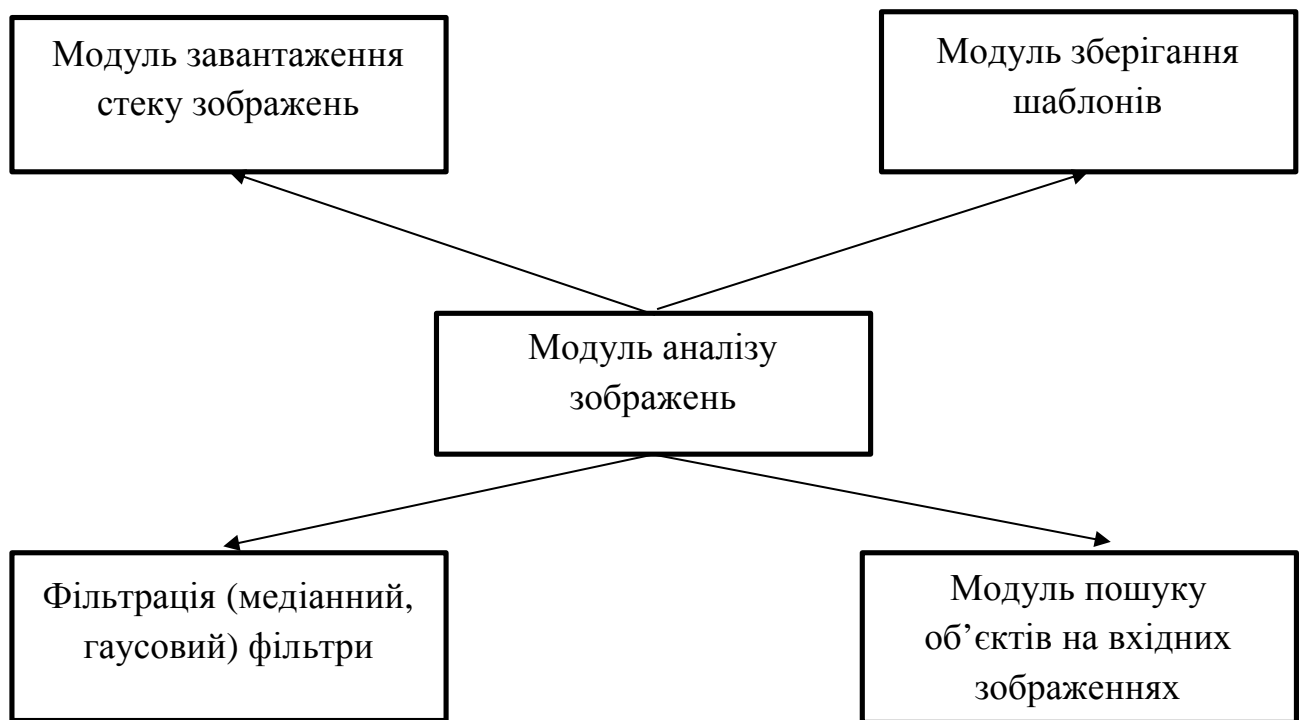


Рисунок 3.1 – Структура модуля аналізу зображень

Модуль завантаження у пам'ять отримує файли вхідних зображень на яких буде проводитися пошук об'єктів, та завантажує їх у базу даних програми. Завантаження зображень проводиться як із внутрішнього сховища комп'ютера так із зовнішніх носіїв чи мережі Інтернет. Способи форматування задають структуру даних і відрізняються один від одного. Для того, щоб комп'ютери і програми могли читати і обробляти дані, структури файлів повинні відповідати певним правилам. У сучасній машинній графіці використовуються десятки спеціалізованих форматів даних [44]. Найпростіший формат — BMP (BitMaP, тобто бітова карта). Найповажнішим серед растрових форматів є TIFF (Tagged Image File Format), тобто структурований формат файлу зображення, і саме йому віддають перевагу професіонали. Для скорочення витрат на графіку було розроблено спеціальні форми стиснення файлів. Найвідоміші з них:

JPG — базується на першому міжнародному стандарті для збереження зображень із деякою втратою якості JPEG (Joint Photographic Expert Group);

GIF — формат обміну графікою (Graphic Interchange Format), призначений для малюнків з чіткими кольорами та контурами, і економія досягається частково за рахунок мінімізації палітри;

PNG (Portable Network Graphic) — також орієнтований на малюнки з чіткими лініями, але не накладає обмежень на розміри палітри і базується на досконаліших загальнодоступних алгоритмах стиснення даних.

Після завантаження вхідних зображень відбувається їхня попередня обробка а саме фільтрація медіанним або гаусовим фільтром. Небажаний ефект від завад усувається фільтрацією.

За основу для проектування архітектури розробленої САМ взято шаблон проектування MVC. Даний шаблон передбачає поділ систем на три частини: модель даних, вид (інтерфейс користувача) і модуль керування. Використовується для відокремлення даних (моделі) від інтерфейсу користувача (вигляду) так, щоб змінити інтерфейс користувача мінімально вплинули на роботу з даними, а зміни в моделі даних могли здійснюватися без змін інтерфейсу користувача. Діаграму пакетів наведено на рисунку 3.2.

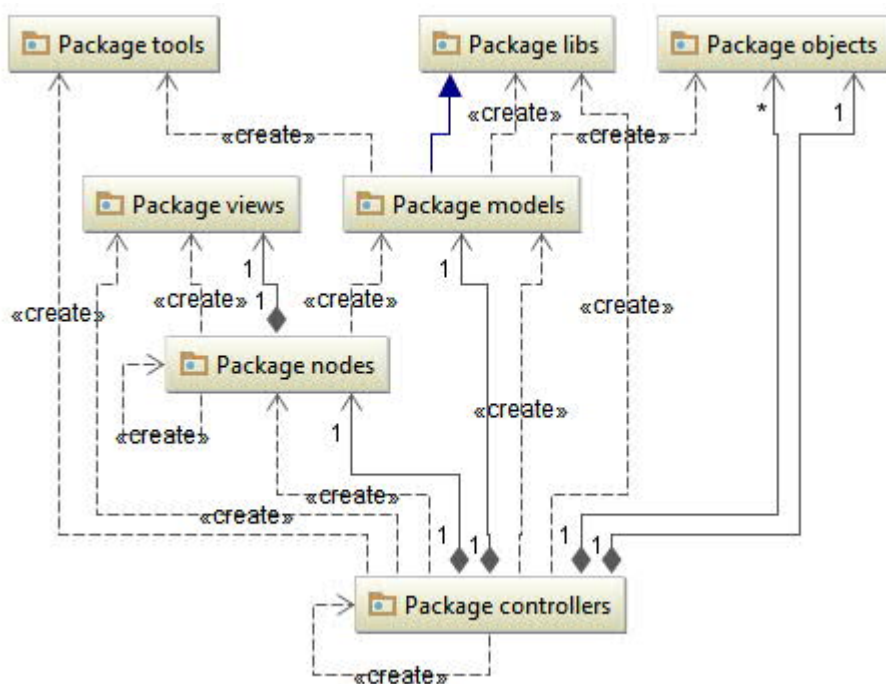


Рисунок 3.2 – Діаграма пакетів проекту

Пакет «Controllers» є ключовим у системі та керує компонентами, отримує сигнали у вигляді реакції на дії користувача (натискання кнопки, ввід даних в текстове поле, вибір елемента зі списку) і передає дані у модель.

Структуру класу «Template», який реалізований для пошуку об'єктів на зображенні наведено на рисунку 3.3.



Template		
f	VVV	int
m	run()	void
m	run1(String, String, String)	void
m	main(String[])	void

Рисунок 3.3 – Структура класу «Template»

Метод main () відрізняється від всіх інших методів тим, що є, як правило, точкою входу в програму. Цей метод викликається віртуальною машиною Java [45-47]. Як тільки закінчується виконання методу main (), так відразу ж завершується, тим самим, робота самої програми.

Метод main (), так і будь-який інший метод, повинен бути обов'язково вкладений в клас. Після компіляції клас, що містить метод main (), запускається на виконання командою:

```
java Назва_Класу
```

Приклад коду підключення усіх необхідних бібліотек до проекту пошуку зображення за шаблоном:

```
import org.opencv.core.Core;  
import org.opencv.core.Core.MinMaxLocResult;
```



```

import org.opencv.core.CvType;
import org.opencv.core.Mat;
import org.opencv.core.Point;
import org.opencv.core.Scalar;
import org.opencv.highgui.Highgui;
import org.opencv.imgproc.Imgproc;

```

Приклад програмного коду для оголошення функції `run()`, що здійснює безпосередньо пошук об'єкта за шаблоном:

```

public void run(String inFile, String templateFile, String
outFile) {

                                Mat img = Highgui.imread(inFile);
                                Mat templ =
Highgui.imread(templateFile);

                                // / Create the result matrix
                                int result_cols = img.cols() -
templ.cols() + 1;
                                int result_rows = img.rows() -
templ.rows() + 1;
                                Mat result = new Mat(result_rows,
result_cols, CvType.CV_32FC1);
...
}

```

У даному випадку

`String inFile` – змінна, що відповідає збереження шляху до файлу вхідного зображення, на яку відбувається пошук за шаблоном.

`String templateFile` – змінна, що відповідає за збереження шляху до файлу – шаблону.

`String outFile` – змінна, стрічкового типу, що відповідає за шлях, за яким буде збережено вихідне (результуюче) зображення.

Стрічка `Highgui.imread(inFile)` здійснює зчитування зображення за шляхом вказаним у змінній `inFile`. В результаті в оперативну пам'ять заноситься зображення у форматі `Mat`, що є типовим форматом бібліотеки `OpenCV` [48]. Для своєї роботи, `OpenCV` використовує кілька різних типів даних. Типи даних починаються з великої «С», а їхні конструктори (які насправді - функції) мають збігатися з назвою даних, тільки їх назву замість великий «С» починається з

маленької «с». (Деякі типи даних (наприклад, CvScalar) мають кілька конструкторів.). У функцію для зчитування зображення можна передати наступні параметри:

`CV_LOAD_IMAGE_ANYDEPTH` - повертає 16-бітове / 32-бітове зображення, коли вхід має відповідну глибину, в іншому випадку перетворить його в 8 біт.

`CV_LOAD_IMAGE_COLOR` - завжди конвертувати зображення в кольорове.

`CV_LOAD_IMAGE_GRAYSCALE` - завжди конвертуйте зображення у відтінки сірого.

Для реалізації пошуку зображення за шаблоном використовують вбудовано функцію:

```
// / Do the Matching and Normalize
                                Imgproc.matchTemplate(img,          templ,
result, Imgproc.TM_CCOEFF);
```

Коли перетворення вибирається з меню Transform, перетворення зв'язується на AffineTransform at:

```
public void setTrans(int transIndex) {
    // Sets the AffineTransform.
    switch ( transIndex ) {
    case 0 :
        at.setToIdentity();
        at.translate(w/2, h/2);
        break;
    case 1 :
        at.rotate(Math.toRadians(45));
        break;
    case 2 :
        at.scale(0.5, 0.5);
        break;
    case 3 :
        at.shear(0.5, 0.0);
        break;
    }
}
```

Перш, ніж вивести на екран форму, відповідну пунктам меню, додаток спочатку отримує поточний перетворення від Graphics2D об'єкт:

```
AffineTransform saveXform = g2.getTransform();
```

Це перетворення буде відновлено Graphics2D після рендеринга. Після отримання поточного перетворення, іншого AffineTransform, toCenterAt, створюється, що причини формують, щоб бути представленими в центрі панелі. at AffineTransform зв'язується на toCenterAt:

```
AffineTransform toCenterAt = new AffineTransform();  
toCenterAt.concatenate(at);  
toCenterAt.translate(-(r.width/2), -(r.height/2));
```

Афінний перетворення координат задається двома основними конструкторами класу AffineTransform:

```
AffineTransform(double a, double b, double c,  
    double d, double e, double f)  
AffineTransform(float a, float b, float c, float d, float e,  
    float f)
```

Для перевірки результатів слід використовувати функцію minMaxLoc, яка знаходиться всередині класу Core. Метод повертає примірник MinMaxLocResult, і він має багато варіантів всередині.

Приклад програмного коду для конвертування зображень у різні формати:

```
package model.converters;  
  
import java.awt.Color;  
import java.util.HashMap;  
  
import model.ImageMatrix;  
  
/**  
 * Implementación de un converter que toma como feature el color.  
 * Recibe como parámetro el espacio de color a utilizar. */
```

```

*/
public class ColorImageConverter extends ImageConverter {

    /**
     * Crea una nueva instancia sobre una imagen determinada. En
    el
     * segundo parámetro recibe un conjunto de parámetros, entre
    los
     * cuales debe estar "colorSpace", que como valores posibles
     * admite: RGB, HSB y GRAYSCALE.
     *
     * @param image Imagen a convertir.
     * @param params Conjunto de parámetros. El único que será
    consultado
     * es "colorSpace" para determinar el espacio de color.
     */
    public ColorImageConverter(ImageMatrix image, HashMap<String,
    String> params) {
        super(image, params);
    }

    @Override
    protected void createFeature(int i, int j, int[] feature) {
        Color c = new Color(image.getPixels()[i][j]);
        String colorSpace = params.get("colorSpace");
        if (colorSpace == null ||
colorSpace.equalsIgnoreCase("RGB")) {
            feature[0] = c.getRed();
            feature[1] = c.getGreen();
            feature[2] = c.getBlue();
        } else if (colorSpace.equalsIgnoreCase("HSB")) {
            float[] hsbvals = new float[3];
            hsbvals = Color.RGBtoHSB(c.getRed(), c.getGreen(),
c.getBlue(),
                hsbvals);
            feature[0] = (int) (hsbvals[0] * 255);
            feature[1] = (int) (hsbvals[1] * 255);
            feature[2] = (int) (hsbvals[2] * 255);
        } else {
            feature[0] = (int)(0.3*c.getRed() +
0.59*c.getGreen() + 0.11 * c.getBlue());
        }
    }

    @Override

```

```

protected int getDepth() {
    String colorSpace = params.get("colorSpace");
    if (colorSpace != null &&
colorSpace.equalsIgnoreCase("GRAYSCALE")) {
        return 1;
    } else {
        return 3;
    }
}
}
}

```

Для налаштування бібліотеки openCV в середовищі розробки IntelliJ IDEA. Приклад підключення бібліотеки openCV в IntelliJIDEA наведено на рисунку 3.4.

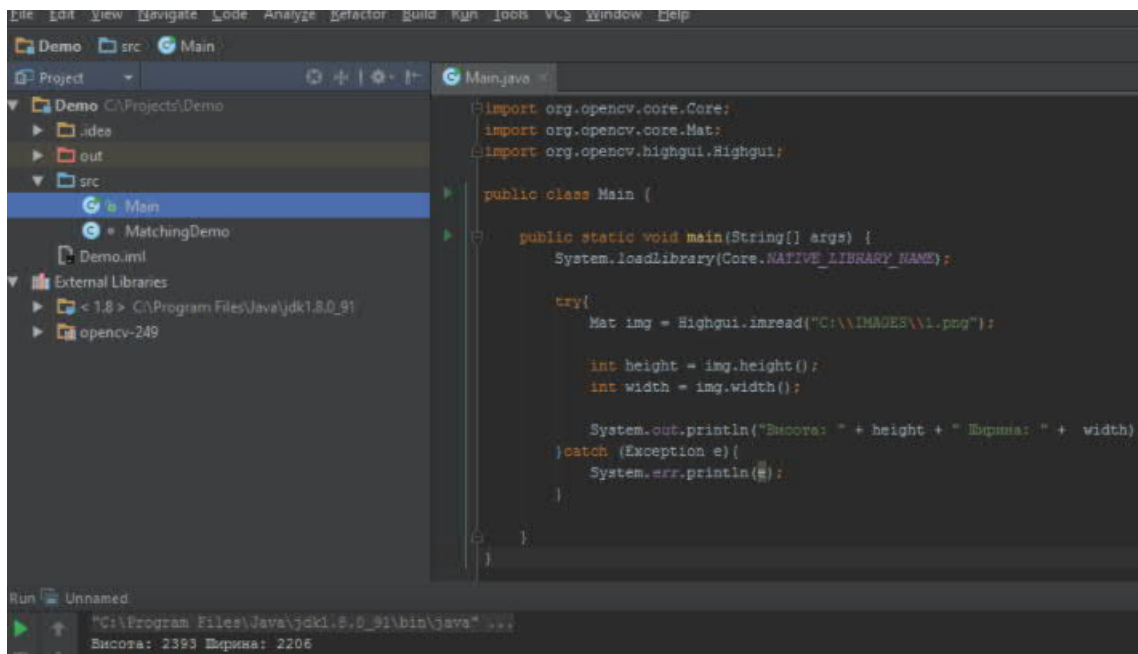


Рисунок 3.4 – Підключення бібліотеки openCV в IntelliJIDEA

На наступному етапі необхідно налаштувати VM Options в IntelliJIDEA. Приклад налаштування наведено на рисунку 3.5.

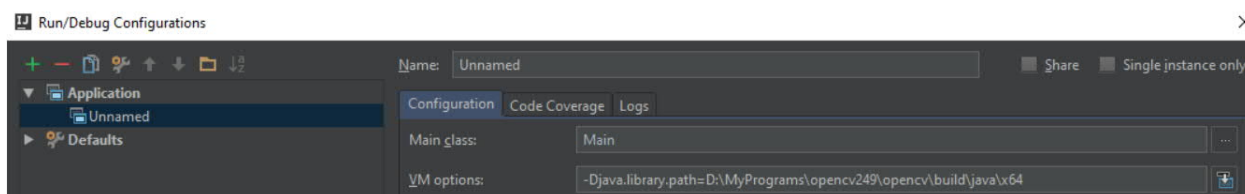


Рисунок 3.5 – Налаштування VM Options в IntelliJIDEA

Системні вимоги до комп'ютера наведено у таблиці 3.1.

Таблиця 3.1 – Системні вимоги до ПК

Найменування	Параметри
ЦП	2xXeon 3.0 ГГц
ОЗП	2 Гб
Об'єм жорсткого диску	145 Гб
Операційна система	Windows 8/10 Linux/FreeBSD
Java	8.0 +
OpenCV	2 +

3.2 Результати роботи алгоритму

Для тестування роботи розробленого програмного модуля було оброблено цитологічні зображення із бази даних «ВРСІ2100» [49]. Вибірку зображень поділено на такі класи:

- цито – фібро – кістозна мастопатія;
- цито – кістозна мастопатія;
- цито – непроліферативна – фібромастопатія;
- цито – непроліферативна мастопатія.

Графічний інтерфейс програми для пошуку елемента за шаблоном наведено на рисунку 3.6.

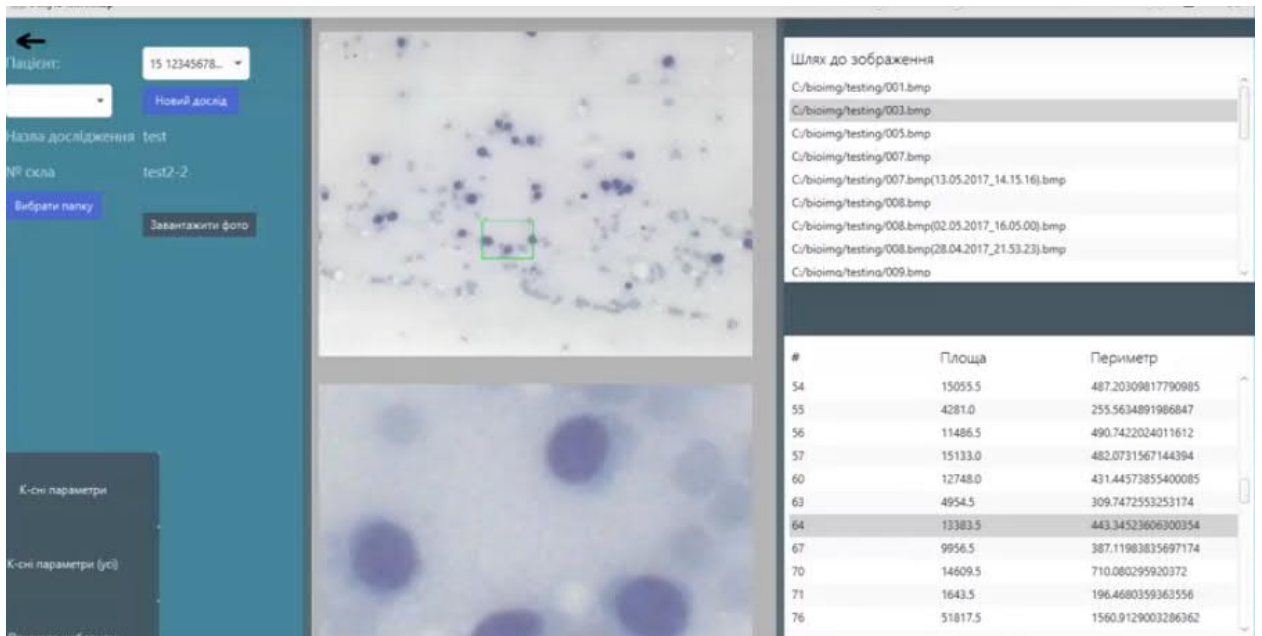


Рисунок 3.6 – Графічний інтерфейс програми для пошуку елемента за шаблоном

3.3 Комп'ютерні експерименти

Для проведення експериментів було підготовлено вхідне зображення (результат гістологічних та цитологічних досліджень), яке включає в себе велику кількість ядер клітин. Ядра клітини – є метою дослідження. Окрім вхідного зображення присутні також два шаблони (одиночні ядра клітин) [50-51]. Приклади шаблонних зображень наведено на рисунку 3.7.

Приклад вхідного зображення наведено на рисунку 3.8.



а) Шаблон А



б) Шаблон Б

Рисунок 3.7 – Приклади шаблонних зображень

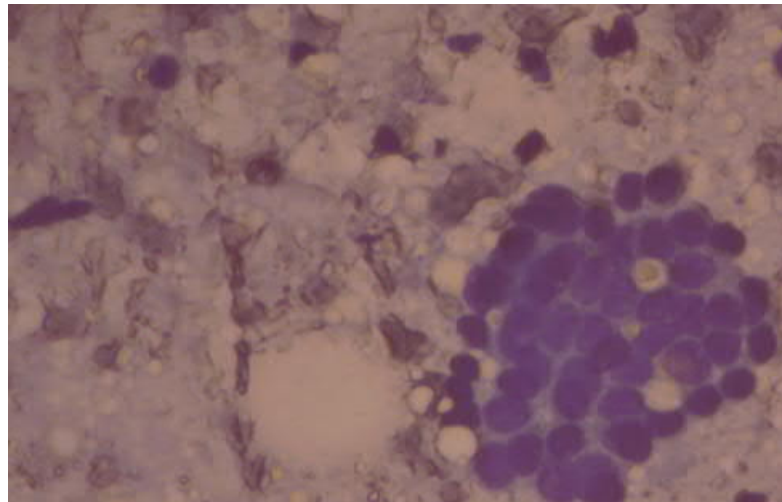


Рисунок 3.8 – Приклад вхідного зображення для дослідження

Приклади шаблонів для пошуку об'єктів на цитологічному зображенні наведено на рисунку 3.9.

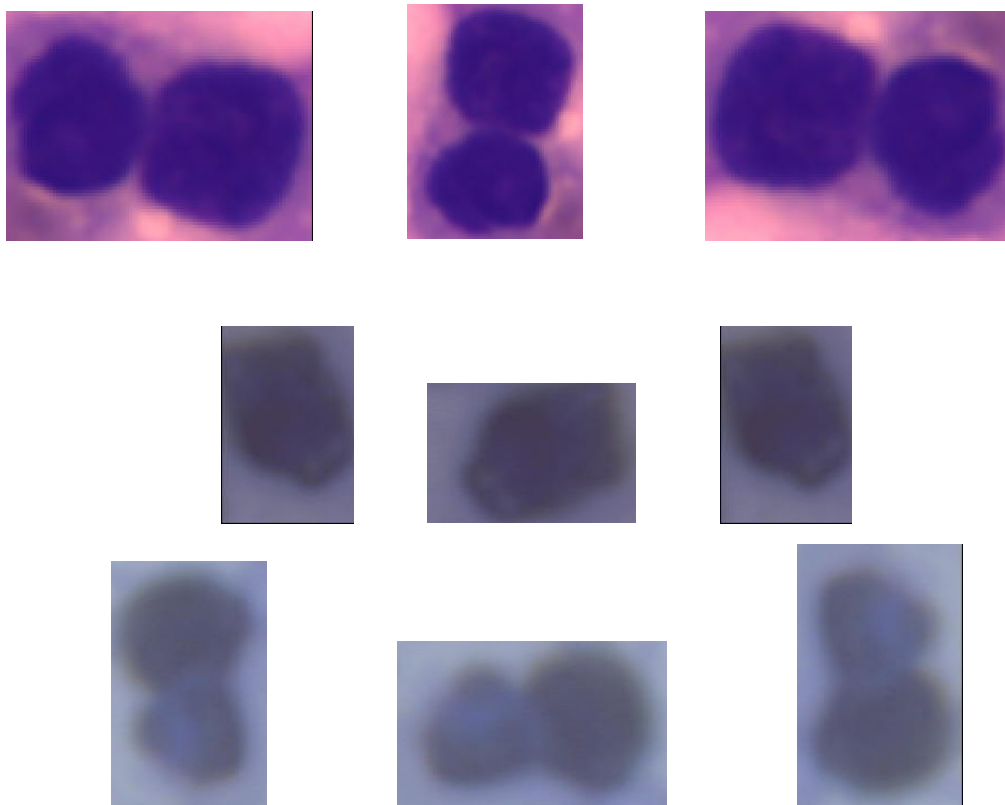
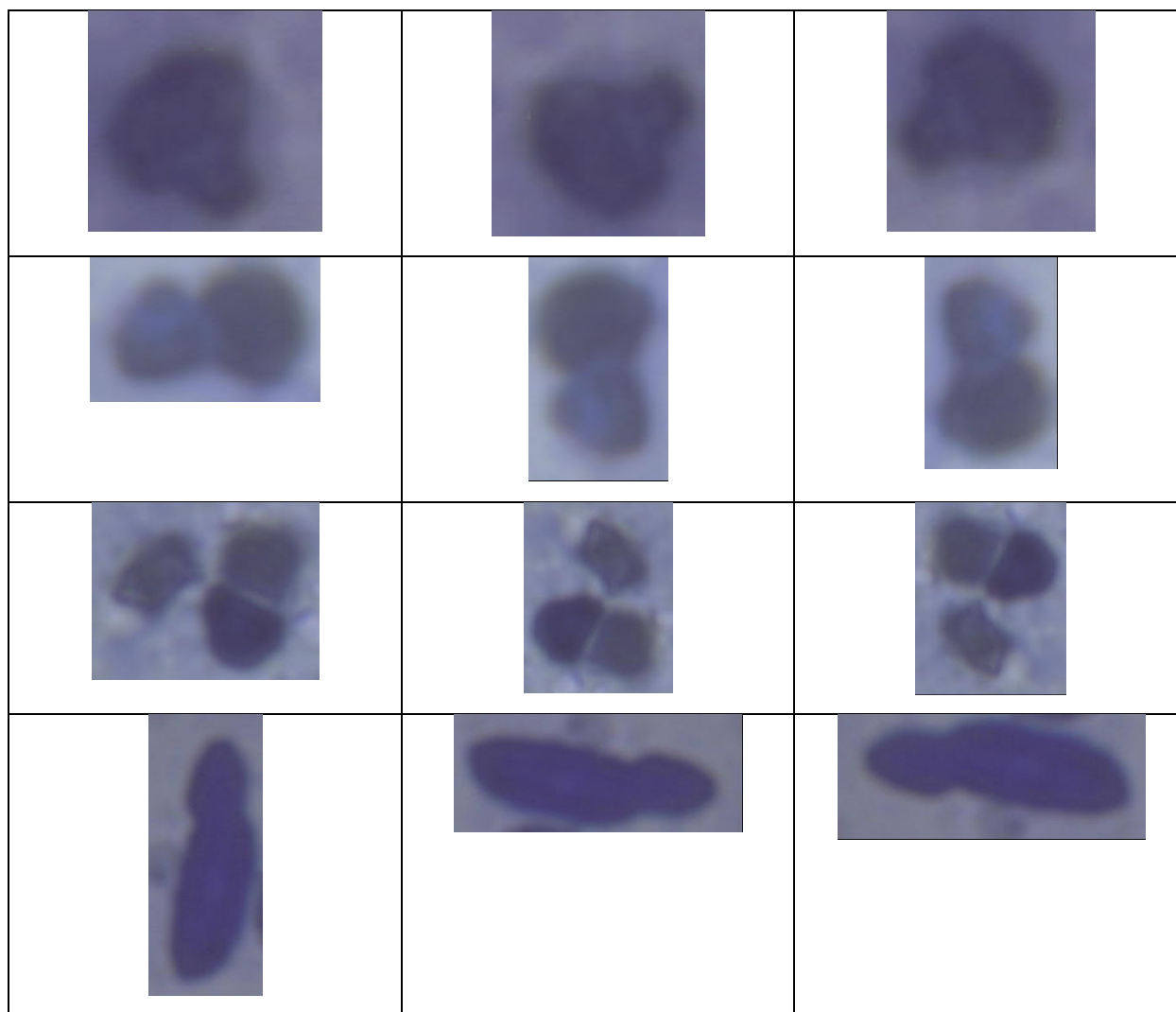


Рисунок 3.9 – Приклади шаблонів

Приклади виділених шаблонів ядер клітин наведено у таблиці 3.2.

Таблиця 3.2 – Приклади шаблонних зображень та їх модифікації



Вибірка — це множина об'єктів або подій, вибраних за допомогою визначеної процедури з генеральної сукупності для участі в дослідженні. Зазвичай, обсяг генеральної сукупності дуже великий, що робить прийняття до уваги всіх членів непрактичним або неможливим.

Навчальна вибірка є деякою підмножиною досліджуваної сукупності, яка називається «генеральна сукупність». На основі вивчення навчальної вибірки необхідно зробити висновки про генеральну сукупність, причому потрібно знати міру достовірності цих висновків. Якщо навчальна вибірка включає всі об'єкти генеральної сукупності, тобто вони збігаються, то достовірність висновків буде найбільш високою (за всіх інших рівних умов). Якщо ж навчальна вибірка дуже мала, то навряд чи на її основі можуть бути зроблені достовірні висновки про генеральну сукупність, оскільки в цьому випадку в

навчальну вибірку можуть навіть не входити приклади об'єктів всіх або переважної більшості класів.

Програмний продукт розроблено на основі мови програмування JAVA. Java — об'єктно-орієнтована мова програмування, випущена 1995 року компанією «Sun Microsystems» як основний компонент платформи Java. З 2009 року мовою займається компанія «Oracle», яка того року придбала «Sun Microsystems». В офіційній реалізації Java-програми компілюються у байт-код, який при виконанні інтерпретується віртуальною машиною для конкретної платформи.

«Oracle» надає компілятор Java та віртуальну машину Java, які задовольняють специфікації Java Community Process, під ліцензією GNU General Public License.

У створенні мови програмування Java було п'ять початкових цілей:

- Синтаксис мови повинен бути «простим, об'єктно-орієнтовним та звичним».
- Реалізація має бути «безвідмовною та безпечною».
- Повинна зберегтися «незалежність від архітектури та переносність».
- Висока продуктивність виконання.

Мова має бути «інтерпретованою, багатонитевою, із динамічним зв'язуванням модулів».

OpenCV — бібліотека функцій та алгоритмів комп'ютерного зору, обробки зображень і чисельних алгоритмів загального призначення з відкритим кодом. Бібліотека надає засоби для обробки і аналізу вмісту зображень, у тому числі розпізнавання об'єктів на фотографіях (наприклад, осіб і фігур людей, тексту тощо), відстежування руху об'єктів, перетворення зображень, застосування методів машинного навчання і виявлення загальних елементів на різних зображеннях.

Бібліотека розроблена Intel і нині підтримується Willow Garage[en] та Itseez. Сирцевий код бібліотеки написаний мовою C++ і поширюється під ліцензією BSD. Біндинги підготовлені для різних мов програмування, таких як

Python, Java, Ruby, Matlab, Lua та інших. Може вільно використовуватися в академічних та комерційних цілях.

Детектування об'єктів застосовується у багатьох галузях. Адже, можливість автоматичного пошуку об'єктів значно спрощує обробку зображення чи відео. В процесі роботи шаблонне зображення послідовно накладається на досліджуване зображення і між ними вираховується кореляція. Результат кореляції заноситься у вихідне зображення. Наприклад, маємо шаблон для пошуку.

Точність пошуку об'єктів на зображенні за одним та за двома шаблонами наведено у таблиці 3.3.

Таблиця 3.3 – Точність пошуку об'єктів на зображенні

Номер експеримент (зображення)	Один шаблон	Два шаблони
1	+	+
2	-	+
3	-	-
4	+	+
5	+	+
6	+	+
7	-	+
8	+	+
9	-	-
10	+	+

У даному випадку знак «+» означає, що шуканий об'єкт був знайдений на вхідному зображенні, знак «-» - відповідно у випадку, коли шуканий об'єкт не був знайдений на вхідному зображенні.

Порівняння часу роботи алгоритму пошуку об'єктів на зображенні за одним та двома шаблонами наведено на рисунку 3.10.

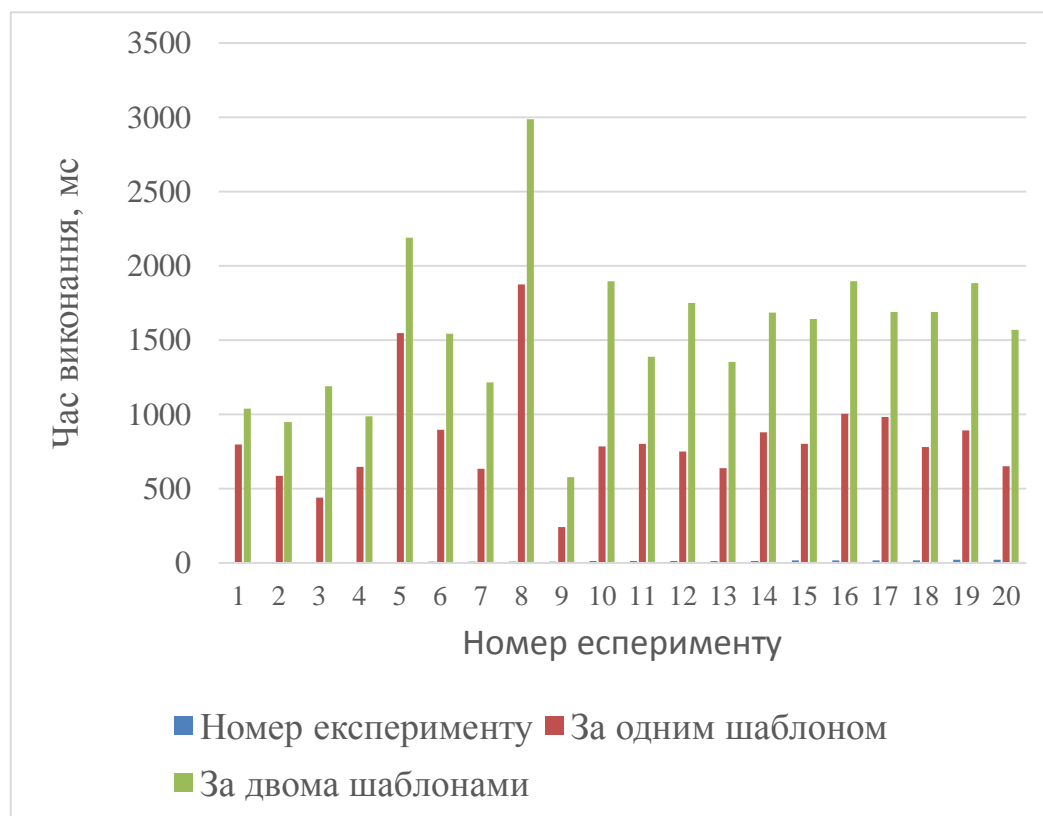


Рисунок 3.10 – Порівняння часу роботи алгоритму пошуку об'єктів на зображенні

У даному випадку наведено результат роботи програми пошуку об'єктів за шаблоном для випадку одного або двох шаблонів. Виходячи із результатів наведених у графіку можна зробити висновок, що час роботи програми пошуку за двома шаблонами практично вдвічі вищий за час роботи із одним шаблоном. Однак, час роботи навіть більше 1 секунди не є критичним для даної предметної області і даний параметр не є визначальний.

Навчальна вибірка для досліджень складає 200 зображень та 400 шаблонів. Точність пошуку об'єктів на зображенні у графічній формі наведено на рисунку 3.11.

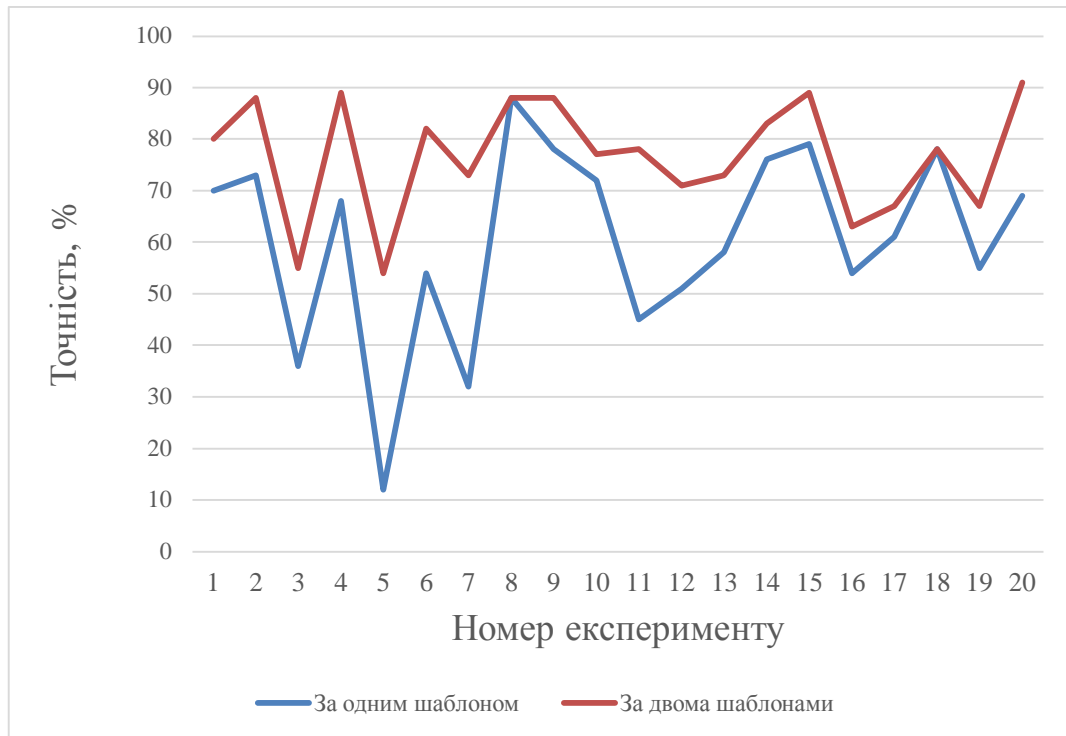
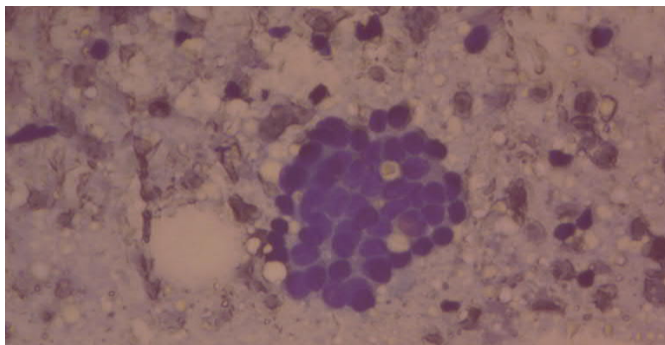


Рисунок 3.11 – Точність пошуку об’єктів на зображенні у графічній формі

В результаті аналізу даних, зображених на рисунку 3.11 можна стверджувати, що точність пошуку об’єктів на зображенні за двома шаблонами є вищою у порівнянні із одним шаблоном.

На рисунку 3.12 наведено приклад вхідного зображення та шаблонного зображення. Результат пошуку об’єкту наведено на рисунку 3.12(в).

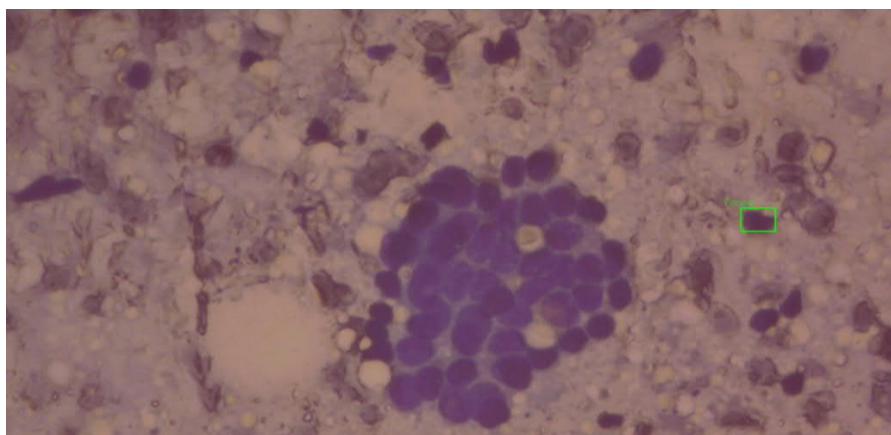
В результаті проведених досліджень можна зробити висновок, що точність пошуку об’єктів на зображенні за двома шаблонами надає кращі результати у порівнянні із пошуком за одним шаблоном приблизно на 20%, при цьому час роботи алгоритму за двома шаблонами близько вдвічі більше часу ніж за одним шаблоном.



а) вхідне зображення



б) шаблон



в) результат

Рисунок 3.12 – Пошук об'єкту на зображенні за шаблоном

3.4 Висновки до розділу 3

У третьому розділі отримано такі результати:

- розроблено структуру програмного модуля;
- розроблено модуль пошуку зображень за шаблонами;
- проведені комп'ютерні експерименти.

ВИСНОВКИ

У кваліфікаційній роботі отримано такі результати:

1. Проаналізовано методи опрацювання зображень, здійснено аналіз методів пошуку зображень, що дало можливість вибрати метод пошуку зображень за шаблонами.
2. Проаналізовано гістологічні зображення, виділено шаблони на зображеннях, що дозволило провести комп'ютерні експерименти.
3. Розроблено алгоритм порівняння зображень на основі скелетонів в метриці Фреше, що дало можливість порівнювати зображення.
4. Розроблено паралельний алгоритм пошуку зображень за шаблонами, які дозволяє проводити пошук за заданими шаблонами одночасно.
5. Розроблено модуль пошуку зображень за шаблонами на основі чого проведені комп'ютерні експерименти.
6. Дослідження показали, що точність пошуку об'єктів на зображенні за двома шаблонами надає кращі результати у порівнянні із пошуком за одним шаблоном приблизно на 20%, при цьому час роботи алгоритму за двома шаблонами близько вдвічі більше часу ніж за одним шаблоном.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Медична інформатика: підручник / за ред. В.Г. Книгавко. Харків: ХНМУ, 2015. 288 с.
2. Салук Р.В., Сидорчук Р.М., Калачик Л.М. Алгоритми пошуку зображень за шаблоном: III Наук.-практ. конф. молодих вчених і студентів «Інтелектуальні комп'ютерні системи та мережі». 26 листопада 2020 р. Тернопіль, 2020. С. 51.
3. Салук Р.В., Сидорчук Р.М., Калачик Л.М. Програмні засоби аналізу біомедичних зображень: III Наук.-практ. конф. молодих вчених і студентів «Інтелектуальні комп'ютерні системи та мережі». 26 листопада 2020 р. Тернопіль, 2020. С. 53.
4. Методичні рекомендації до виконання дипломної роботи з освітньо-кваліфікаційного рівня «Магістр». Спеціальність «Комп'ютерні системи та мережі» / О.М. Березький, Л.О. Дубчак, Г.М. Мельник / Під ред. О.М. Березького. Тернопіль: ТНЕУ, 2016. 47 с.
5. Методичні вказівки до оформлення курсових проектів, звітів про проходження практики, випускних кваліфікаційних робіт для студентів спеціальності «Комп'ютерна інженерія» / І.В. Гураль, Л.О. Дубчак / Під ред. О.М. Березького. Тернопіль: ТНЕУ, 2019. 33 с.
6. Методи, алгоритми і програмні засоби опрацювання біомедичних зображень / Березький О. М., Батько Ю.М., Березька К.М. і ін. Тернопіль: Економічна думка, ТНЕУ, 2017. 330 с.
7. Грицик В.В., Березька К.М., Березький О. М. Моделювання та синтез складних зображень симетричної структури. Львів: УАД – ДНДІІІ, 2005. 140 с.
8. Berezsky O. Regions matching algorithms analysis to quantify the image segmentation results / O. Berezsky, G. Melnyk, Yu. Batko, O. Pitsun // Sensors & Transducers. 2017. Vol. 208, Issue 1. P. 44-50.

9. Berezsky O. Vision-based medical expert system / Berezsky O., Berezska K., BatkoYu., Melnyk G. // 6th International Scientific and Technical Conference «Computer Sciences and Information Technologies» (CSIT'2011), Lviv, November 16-19, 2011. Lviv, 2011. P. 288-289.
10. Berezsky O. Biomedical image search and retrieval algorithms / O. Berezsky, G.Melnyk, Yu Batko // Computing. – 2008. – Vol. 7, Issues 1. – P.108-113.
11. Борисов Е.С. Детектор объектов для неподвижных камер: веб-сайт. URL: <http://mechanoid.kiev.ua/cv-backgr.html>.
12. Борисов Е.С. Обработка видео: детектор объектов на основе цветowych фильтров: веб-сайт. URL: <http://mechanoid.kiev.ua/cv-detector-color.html>.
13. Борисов Е.С. Базовые методы обработки изображений: веб-сайт. URL: <http://mechanoid.kiev.ua/cv-base.html>.
14. Конушин А. Компьютерное зрение (2011). Лекция 3. Простые методы анализа изображений. Сопоставление шаблонов: веб-сайт. URL: <http://www.youtube.com/watch?v=TE99wDbRrUI>.
15. Конушин А. Компьютерное зрение (2011). Лекция 5. Локальные особенности: веб-сайт. URL: <http://www.youtube.com/watch?v=vFseUICis-s>.
16. OpenCV documentation: Harris Corner Detection: веб-сайт. URL: http://docs.opencv.org/3.0-beta/doc/py_tutorials/py_feature2d/py_features_harris/py_features_harris.html.
17. Blob_detection: веб-сайт. URL: http://en.wikipedia.org/wiki/Blob_detection.
18. Histogram of oriented gradients: веб-сайт. URL: http://en.wikipedia.org/wiki/Histogram_of_oriented_gradients.
19. Scale-invariant feature transform: веб-сайт. URL: http://en.wikipedia.org/wiki/Scale-invariant_feature_transform.

20. Конушин А. Компьютерное зрение (2011). Лекция 8. Поиск и локализация объектов: веб-сайт. URL: <http://www.youtube.com/watch?v=I1AiFF6ZkaE>.
21. MathWorks: Image Classification with Bag of Visual Words: веб-сайт. URL: <http://www.mathworks.com/help/vision/ug/image-classification-with-bag-of-visual-words.html>.
22. Борисов Е.С. Автоматизированная обработка текстов на естественном языке: веб-сайт. URL: <http://mechanoid.kiev.ua/ml-text-proc.html>.
23. Борисов Е.С. Кластеризатор на основе алгоритма k-means: веб-сайт. URL: <http://mechanoid.kiev.ua/ml-k-means.html>.
24. Борисов Е.С. Классификатор на основе машины опорных векторов: веб-сайт. URL: <http://mechanoid.kiev.ua/ml-svm.html>.
25. Haar-like features: веб-сайт. URL: http://en.wikipedia.org/wiki/Haar-like_features.
26. Viola M.Jones. Robust Real-Time Face Detection: веб-сайт. URL: <http://www.vision.caltech.edu/html-files/EE148-2005-Spring/pprs/viola04ijcv.pdf>.
27. Борисов Е.С. Бустинг – композиции классификаторов: веб-сайт. URL: <http://mechanoid.kiev.ua/ml-adaboost.html>.
28. Борисов Е.С. Классификатор изображений на основе свёрточной сети: веб-сайт. URL: <http://mechanoid.kiev.ua/ml-lenet.html>.
29. Convolutional Neural Networks for Visual Recognition: веб-сайт. URL: <https://cs231n.github.io/convolutional-networks/>.
30. Howse J., Minichino J. Learning OpenCV 3 Computer Vision with Python – Second Edition, Packt Publishing, September 2015, Packt Publishing, ISBN: 9781785289774.
31. Березький О. М. Інтелектуальна система для діагностування різних форм раку молочної залози на основі аналізу гістологічних і цитологічних зображень / О. М. Березький, Г.М.Мельник, Ю.М. Батько, Т. В. Дацко // Науковий вісник НЛТУ України: зб. наук.-техн. праць. – Львів: РВВ НЛТУ України. – 2013. – Вип. 23.13. – С. 357-367.

32. Березький О. М. Інформаційно-аналітична система дослідження та діагностування пухлинних клітин на основі аналізу їх зображень / О. М. Березький, Ю.М. Батько, Г.М.Мельник // Вісник Хмельницького національного університету. Технічні науки. – 2008. – №4. – С.33-41.

33. Berezsky O. The intelligent system for diagnosing breast cancers based on image analysis / Oleh Berezsky, Tamara Datsko, Sergiy Verbovy // Proceedings of Information Technologies in Innovation Business (ITIB), 7-9 October, 2015, Kharkiv, Ukraine. – P. 27-30.

34. Березький О. М. Нечітка база знань інтелектуальної системи діагностування видів раку молочної залози / О. М. Березький, Г. М. Мельник, К. М. Березька // Вісник Хмельницького національного університету. Технічні науки. 2013. №6. С.284-291.

35. Berezsky O. Fuzzy system diagnosing of precancerous and cancerous conditions of the breas / O. Berezsky, S. Verbovyu, L. Dubchak, T. Datsko // Proceedings of the XIth International Scientific and Technical Conference Computer Sciences and Information Technologies (CSIT'2016), Lviv, 6-10 September, 2016. – Lviv, 2016. – P. 200–203.

36. Berezsky O. An Intelligent System for Cytological and Histological Image Analysis / Oleh Berezsky, Grygoriy Melnyk, Tamara Datsko, Sergiy Verbovy // Proceedings of the 13 th International Conference «The Experience of Designing and Application of CAD Systems in Microelectronics» CADSM 2015, 24-27 February 2015, Polyana-Svalyava (Zakarpattya), Ukraine. – 2015. – P. 28-31.

37. Местецкий Л.М. Непрерывная морфология бинарных изображений. Фигуры, скелеты, циркуляры. М.: Физматлит, 2009. 288 с.

38. Berezsky O. Fréchet Metric for Trees. Proceedings of the 2016 IEEE First International Conference on Data Stream Mining & Processing (DSMP), Lviv, August 23-27, 2016. Lviv, 2016. P. 213-217.

39. Березький О. М. Методи та алгоритми перетворення контурів зображень в афінному просторі // Вісник Національного університету

«Львівська політехніка». Комп'ютерні науки та інформаційні технології. 2008. № 569. С. 123-126.

40. Кормен Т., Лейзерсон Ч., Ривест Р. Алгоритмы: построение и анализ М.: МЦНМО: БИНОМ. Лаборатория знаний, 2004. 960 с.

41. Березький О.М., Піцун О.Й., Долинюк Т.М., Батько Ю.М. Розпаралелення пошуку мікрооб'єктів на цитологічних зображеннях за шаблоном // Штучний інтелект, 2019. №3-4 (85-86). С. 73-81.

42. Березький О., Батько Ю. Аналіз алгоритмів виділення контурів зображень // Вісник Національного університету «Львівська політехніка». Комп'ютерні науки та інформаційні технології. 2006. № 565. С. 212-216.

43. Березький О. М. Аналіз і синтез зображень пухлинних клітин // Інформаційні технології і системи. 2005. Т.8, №1. С.76-85.

44. Глушаков С. В. и др. Компьютерная графика ; Харьковский ин-т информационных технологий. 3. изд., доп. и перераб. Х.: Фолио, 2006. 512 с.

45. JavaScript - What JavaScript can be used for - 2019: веб-сайт. URL: <https://www.theserverside.com/>.

46. Java - Programs created in Java offer - 2019: веб-сайт. URL: <https://www.theserverside.com/>.

47. Java variables - What is variable - 2019: веб-сайт. URL: <https://howtodoinjava.com/>.

48. OpenCV: веб-сайт. URL: <https://uk.wikipedia.org/wiki/OpenCV>.

49. Свідоцтво про реєстрацію авторського права на твір №75359. База даних цифрових гістологічних та цитологічних зображень передракових та ракових станів молочної залози «ВРСІ2100» / О.М. Березький, Г.М. Мельник, С.О. Вербовий, О.Й. Піцун, В.Д. Николук, Т.В. Дацко. Дата реєстрації 14.12.2017 р.

50. Березький О. М. Інформаційно-аналітична система для дослідження і діагностування пухлинних (ракових) клітин людини на основі аналізу їх зображень / О. М. Березький, Т.В. Дацко, Ю.М Батько // Каталог матеріалів Міжнародного Форуму «Регіони знань: Україна в європейському просторі

освіти – науки - інновацій для ревіталізації та процвітання територій», 26-27 березня 2010, м. Тернопіль. – Тернопіль, 2010. С. 37.

51. Березький О. М., Батько Ю. М., Мельник Г. М. Синтез альтернативних рішень при структурному проектуванні систем автоматизованої мікроскопії // Науковий вісник НЛТУ України: зб. наук.-техн. праць. Львів: РВВ НЛТУ України. 2009. Вип. 19.5. С. 258-268.