

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Тернопільський національний економічний університет
Факультет комп'ютерних інформаційних технологій
Кафедра комп'ютерної інженерії

Баган Ярослав Андрійович

**Програмний модуль пошуку мікрооб'єктів на
зображенні за шаблоном у телемедичній системі
"HIAMS" / The program module for template matching on
image in the telemedical system "HIAMS".**

спеціальність: 6.050102 - Комп'ютерна інженерія
освітньо-професійна програма - Комп'ютерні системи та мережі

Випускна кваліфікаційна робота

Виконав: студент групи КСМ-41
Баган Ярослав Андрійович

Науковий керівник:
Піцун О. Й.

Випускну кваліфікаційну роботу
допущено до захисту:

" ____ " _____ 20__ р.

Завідувач кафедри
О. М. Березький

ТЕРНОПІЛЬ - 2019

РЕЗЮМЕ

Дипломний проект містить 71 сторінку пояснюючої записки, 25 рисунків, 17 таблиць, 3 додатки. Обсяг графічного матеріалу 2 аркуші формату А3.

Метою бакалаврської роботи є розробка модулю для пошуку мікрооб'єктів на зображеннях за шаблоном в телемедичній системі «НАІMS».

В бакалаврській роботі, аналізуючи поставлену задачу та досліджуючи існуючі розробки, визначено основні функції та вимоги до розроблюваного програмного модуля, також виділено та розглянуто напрямки розвитку інформаційних технологій а саме телемедицини і комп'ютерного аналізу мікробіологічних зображень.

Зважаючи на стрімкий розвиток телемедицини, для виконання поставлених задач був складений алгоритм та розроблений програмний модуль на мові програмування Java із використанням бібліотеки комп'ютерного зору OpenCV .

Розроблений програмний модуль виконує функції завантаження, попередньої обробки та безпосереднього пошуку об'єктів на зображеннях, його функціонал надає очевидні переваги в сфері телемедицини а особливо при роботі з мікробіологічними зображеннями.

Ключові слова: ТЕЛЕМЕДИЦИНА, ПОШУК ОБ'ЄКТІВ НА ЗОБРАЖЕННІ, OPENCV, КОМП'ЮТЕРНИЙ ЗІР, АНАЛІЗ БІОМЕДИЧНИХ ЗОБРАЖЕНЬ.

RESUME

The diploma project contains 71 pages of explanatory records, 25 drawings, 17 tables, 3 appendixes. The volume of graphic material is 2 sheets of A3 format.

The purpose of the bachelor's work is to develop a module for detecting microscopic objects in images using a template in the telemedicine system "HAIMS".

In the bachelor's work, analyzing the task and studying the existing developments, the main functions and requirements for the developed software module are determined, directions of the development of information technologies, namely telemedicine and computer analysis of microbiological images, are also highlighted and considered.

Due to the rapid development of telemedicine, an algorithm was developed for the accomplished tasks and a program module was developed in the Java programming language using the OpenCV computer vision library.

The developed program module performs functions of loading, pre-processing and direct search of objects in images, its function provides obvious advantages in the field of telemedicine and especially when working with microbiological images.

Key words: TELEMEDICINE, DETECTING OBJECT ON IMAGE, OPENCV, COMPUTER VISION, ANALYSIS OF BIOMEDICAL IMAGES.

ЗМІСТ

Вступ.....	9
1 Аналіз програмних засобів опрацювання зображень.....	10
1.1 Технології опрацювання зображень	10
1.2 Аналіз біомедичних зображень	18
1.3 Аналіз підходів до пошуків об’єктів на зображенні	21
1.4 Постановка задач дипломного проекту	24
2 Алгоритми роботи програмного модулю	25
2.1 Структура модулю аналізу зображень.....	25
2.2 Алгоритм пошуку об’єкту за шаблоном.....	30
2.3 Структура бази даних	31
3 Розробка та тестування модулю	38
3.1 Структура розробленого програмного модулю.....	38
3.2 Модуль роботи з БД.....	41
3.3 Тестування розробленого модулю	44
4 Техніко-економічне обґрунтування розробки програмного засобу	49
4.1 Розрахунок витрат на розробку програмного забезпечення	49
4.2 Визначення експлуатаційних витрат.....	56
4.3 Розрахунок ціни споживання програмного продукту.....	59
4.4 Визначення показників економічної ефективності	60
Висновки	63
Список використаних джерел	64
Додаток А Лістинг класу «Matching.java».....	67
Додаток Б Світокопія публікації	68
Додаток В Довідка про використання.....	71

					БР.КСМ.07224/17.00.00.000 ПЗ			
Змн.	Лист	№ докум.	Підпис	Дата	ПРОГРАМНИЙ МОДУЛЬ ПОШУКУ МІКРООБ’ЄКТІВ НА ЗОБРАЖЕННІ ЗА ШАБЛОНОМ У ТЕЛЕМЕДИЧНІЙ СИСТЕМІ «HIAMS»	Літ.	Арк.	Акрушів
Розробив	Баган Я.А.						8	71
Перевір.	Піцун О.Й.							
Консульт.	Паздрій І.Р.							
Н. Контр.	Гураль І.В.							
Затвердив	Березький О.М.							ТНЕУ.ФКІТ. КСМ-41/1

ВСТУП

Із розвитком інформаційно-комунікаційних технологій у галузі охорони здоров'я виник цілий величезний, магістральний напрямок досліджень, пов'язаний із дистанційним лікуванням, який назвали телемедициною. У цьому напрямку широко застосовується різного роду програмне забезпечення. Одною з найважливіших областей цієї галузі є аналіз мікробіологічних зображень які повинні бути проаналізовані з метою збору інформації та, або для прийняття рішень. Як правило, інформація представлена наявністю цілого ряду об'єктів, а саме клітин та патологій, що представляють інтерес. Метою аналізу є виявлення та ідентифікація цих об'єктів.

Залежно від задачі та конкретного поля застосування витягнуті об'єкти можуть бути використані для різних цілей. Наприклад, у цитології лікарі опрацьовують оцифровані мікроскопічні слайди, що містять людські тканини, з метою діагностування конкретних захворювань, про які йдеться, присутність клітин, що мають певні характеристики та для визначення концентрації певних мікроорганізмів. Ці зображення зазвичай аналізуються фахівцями вручну. Однак через величину проблеми аналіз не завжди виконується вичерпно. Це має очевидний недолік, що збільшує ризик не знаходження об'єктів що можуть представляти небезпеку.

Через ризики, і тому, що ручний аналіз повних зображень є довгим і виснажливим, комп'ютерні програми можуть бути використані для надання допомоги експертам.

Метою бакалаврської роботи є розробка модулю для пошуку мікрооб'єктів на зображеннях за шаблоном.

Актуальність роботи полягає у вдосконаленні засобів пошуку досліджуваних мікрооб'єктів на гістологічних та цитологічних зображеннях.

					БР.КСМ.07224/17.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		9

1 АНАЛІЗ ПРОГРАМНИХ ЗАСОБІВ ОПРАЦЮВАННЯ ЗОБРАЖЕНЬ

1.1 Технології опрацювання зображень

Зображення є одним із найпоширеніших типів мультимедійної інформації, що циркулює в сучасних обчислювальних системах. Статичні зображення являють собою заміну звичайних паперових фотографій (або їхніх негативів), які переважна більшість користувачів зберігає для довготривалої фіксації певних визначних моментів свого життя, тобто «на пам'ять». Однак, звичайно, існує і широке коло практичних задач, у яких використовуються статичні графічні дані.

Серед таких задач можна назвати наступні:

- збереження та обробка наукової інформації (такої, як знімки зоряного неба, мікроскопічних об'єктів, фіксація над-повільних фізичних процесів, тощо);
- збирання та використання інформації практичного (промислового) характеру, що може бути використана у наступній виробничій (професійній) діяльності в т.ч. біологічного та медичного характеру;
- передача, транспортування інформації каналами цифрового зв'язку (аналог факсимільної передачі).

Для цілей даного дослідження найбільший інтерес представляє робота з зображеннями медичного характеру. Автоматизація цього процесу відкриває нові широкі можливості щодо:

- суттєвого підвищення продуктивності праці суб'єкта, що здійснює первинний аналіз фотографій;
- можливість організації масованого контролю потоків медичних зображень на предмет наявності окремих важливих факторів;
- неупередженість автоматизованої інформаційної системи при винесенні вердикту, в т.ч. її свобода від навмисних зловживань.

Очевидно, у медичній практиці можуть використовуватися зображення із дуже і дуже різними характеристиками. Це, в першу чергу, різні розміри кінцевого зображення, а також і його поля зору. В подальшому основну увагу

					БР.КСМ.07224/17.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		10

будемо приділяти процесу аналізу та обробки зображень мікро-масштабів, а саме, клітинного рівня.

Такий вибір характеристик цільових фотозображень обумовлений надзвичайною актуальністю процесів розпізнавання змінених клітин, адже рання діагностика онкологічних захворювань є дуже важливим фактором успішного лікування пацієнтів, хворих на рак. [1]

Отже, дуже актуальним завданням галузі інформаційних технологій у медичній сфері став розвиток систем автоматизованої мікроскопії (далі – САМ).

При обробці зображень медичного характеру за допомогою САМ можна виділити три рівні «комп'ютерного зору» – рисунок 1.1: низький, середній та високий, які відрізняються ступенем інтелектуального внеску, що його виконує комп'ютер у відповідний процес[9, 10]. Класифікація методів обробки зображень наведена на рисунку 1.1.

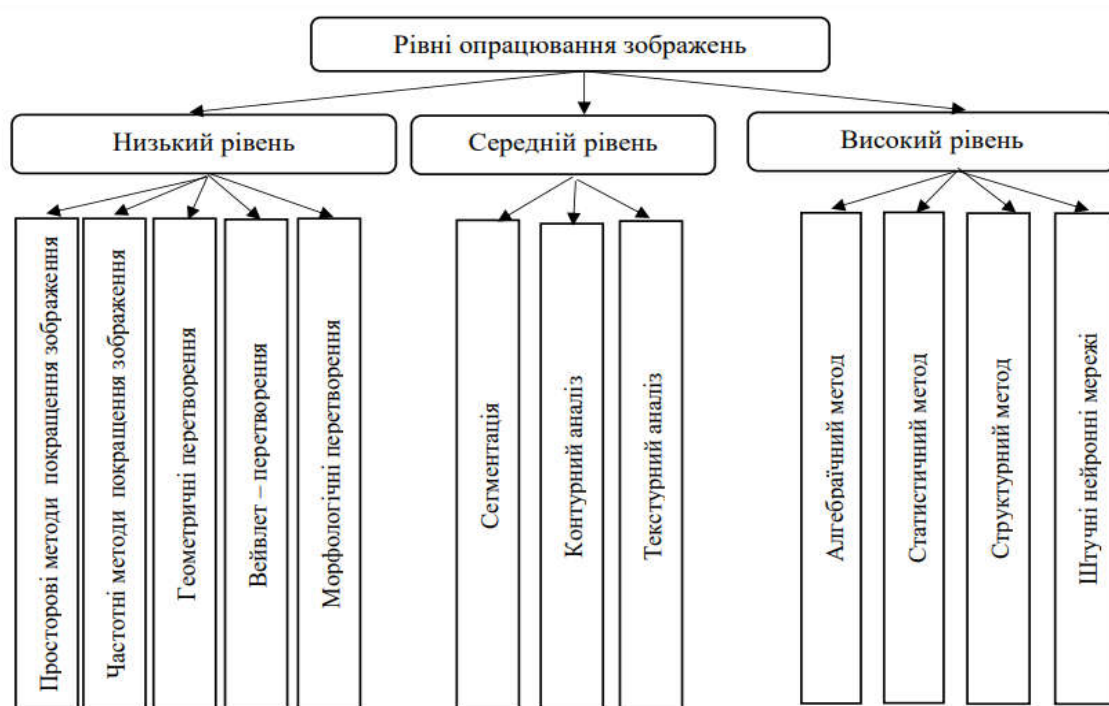


Рисунок 1.1 – Класифікація методів обробки зображень для медичної галузі

Методи, згруповані на низькому рівні обробки, призначені переважно для забезпечення кращої якості зображення, причому під «якістю» тут мається на

увазі відповідність вимогам конкретної прикладної задачі. Це, зокрема, здійснюється за рахунок:

– просторових методів, коли безпосередні числові перетворення застосовуються до пікселів висхідного зображення;

– частотних методів покращення зображень, коли графічний сигнал спочатку підлягає перетворенню Фур'є (відображення у частотну область), а потім обробляється, після чого знову відновлюється до суцільного графічного зображення;

– геометричних методів, коли зображення перебудовується на основі введення нових систем координат;

– вейвлет-перетворення – аналогічне частотним методам, вказаним вище, але розкладання сигналу виконується по оригінальній системі ортогональних функцій (не гармонічних) – вейвлетів, які грубо можна охарактеризувати, як гармонічні сигнали, амплітуда яких спочатку плавно нарощується до максимуму, а потім також плавно спадає;

– морфологічні перетворення, що включають такі елементи, як розтяг та звуження усіх елементів зображення, розмикання та замикання (це комбінації звуження та розширення, а також розширення та звуження відповідно).

Усі перелічені способи низькорівневої обробки зображень можуть бути реалізовані будь-якою мовою програмування загального призначення. Звичайно, оскільки кожна така дія є стандартною і добре описаною, логічним є пошук вже існуючих реалізацій вказаних алгоритмів, наприклад, у певних бібліотеках, що входять до стандартних наборів поставки тієї, чи іншої мови програмування, або можуть долучатися до них як окреме рішення.

Відповідно постає задача вибору мови програмування, що може бути застосована як для реалізації методів низького рівня обробки зображень, так і двох інших рівнів: середнього та високого.

Серед сучасних мов програмування відсутні спеціалізовані рішення, призначені для обробки саме графічної інформації, тому все ж таки тут є зміст

					БР.КСМ.07224/17.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		12

говорити лише про мови загального призначення. Популярні на сьогоднішній день мови програмування наведено на рисунку 1.2.

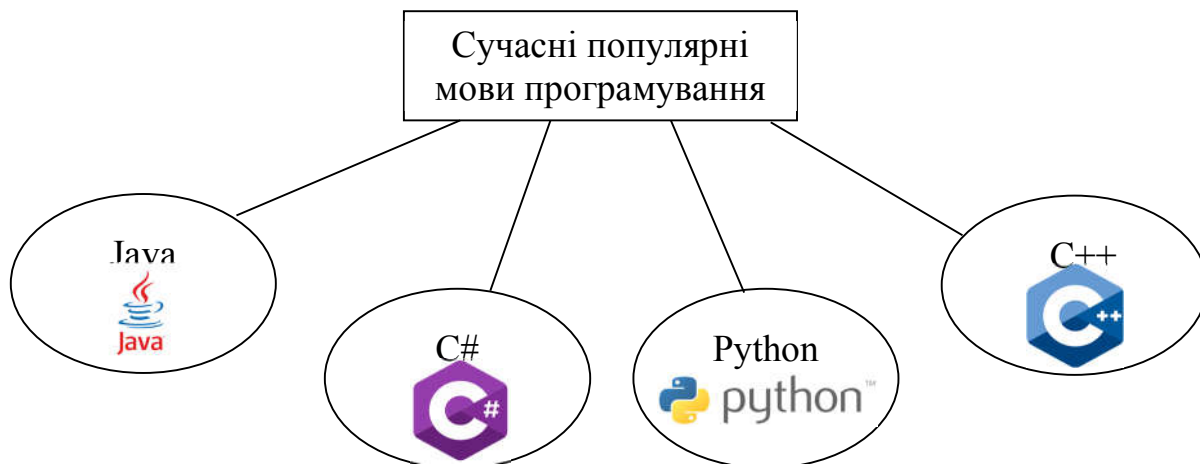


Рисунок 1.2 – Сучасні популярні мови програмування загального призначення

Усі, приведені на рисунку 1.2 мови, є повністю об'єктно-орієнтованими, включають практично усі теоретично існуючі можливості об'єктно-орієнтованого аналізу, тобто відповідають сучасним вимогам до засобу розробки професійного програмного забезпечення.

Однак, у кожного із варіантів рисунку 1.2 є певні особливості, що кінцево спричинюють конкретний вибір мови програмування. На сьогоднішній день існує дві реальних альтернативи для розробки програмного забезпечення для настільних ПК: це мови Java та C#. Слід сказати, що продукт від Microsoft з'явився набагато пізніше, ніж Java, і у великій мірі копіює її властивості та підходи до програмування. Однак, C# настільки динамічно розвивається (і, ймовірно, підтримується фінансово мільйонними інвестиціями гіганту Microsoft), що цілком ймовірно, що у майбутньому він повністю (чи у значній мірі, з окремих ніш) витіснить продукт від Oracle (така ситуація на ринку уже спостерігалася не раз: це і тотальна перемога браузера Microsoft Internet Explorer над початково більш розвинутим Netscape Navigator, і витіснення середовища розробки Borland C++ Builder продуктом Microsoft Visual C++, і т.п.). Особливо ця ситуація видається реальною, якщо взяти до уваги агресивну маркетингову політику Microsoft по просуванню власних продуктів на ринку, що заснована на

					БР.КСМ.07224/17.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		13

монопольному положенні компанії у сфері поставок операційних систем для настільних ПК. Однак, цей сценарій хоча і є можливим, однак не у самому близькому майбутньому, а на сьогодні мова Java має ряд переваг над конкурентами, і, в першу чергу, це наявність дуже значної кількості робочого коду сторонніх виробників, що тим не менше поширюється вільно і може застосовуватися будь-ким у власних програмних продуктах. Отже, наявність численних бібліотек, в т.ч. і для обробки зображень, робить мову Java найкращим вибором для цілей даної дипломної роботи.

Порівняльна характеристика мов програмування з рисунку 1.2 наведена в таблиці 1.1.

Таблиця 1.1 – Порівняльна характеристика популярних мов програмування загального призначення

Характеристика	Java	C#	Python	C/C++
Об'єктно-орієнтована розробка	+	+	+	+
Наявність великої кількості бібліотек	+	+/-	+/-	+/-
Підвищена продуктивність для високонавантажених систем (типу big data, high load, etc)	+	+/-	-	+
Низька вартість комерційних засобів розробки (IDE)	+/-	-	+	+/-
Адаптованість мови для створення великих проектів	+	+	-	+

Аналізуючи таблицю 1.1, бачимо, що по обраним критеріям мова Java не має жодного чистого мінуса (на відміну від інших інструментів), тому саме вона обрана у якості основного засобу для розробки програмного забезпечення, що проектується.

Окрім обрання безпосередньо мови програмування, слід відмітити, що для цілей обробки зображень на даний час створено значну кількість бібліотек, в т.ч. таких, що можуть використовуватися у програмах на мові Java. Розглянемо їх докладніше.

Бібліотека OpenCV (розшифровується Open Source Computer Vision Library) призначена саме для тих цілей, що переслідуються у даній дипломній роботі. Починаючи з версії 2.2 (поточна версія бібліотеки 4.0 випущена в листопаді 2018 р.) відбувся перехід від універсальних великих модулів до компактних, з вузькою спеціалізацією (які, відповідно, швидше завантажуються і працюють):

- `opencv_core` – основна функціональність. Включає в себе базові структури, обчислення (математичні функції, генератори випадкових чисел) і лінійну алгебру, DFT, DCT, введення / виведення для XML і YAML, і т.д.;

- `opencv_imgproc` – обробка зображень (фільтрація, геометричні перетворення, перетворення колірних просторів і т. д.);

- `opencv_highgui` – простий UI, введення / виведення зображень і відео;

- `opencv_ml` – моделі машинного навчання (SVM, дерева рішень, навчання зі стимулюванням і т. д.);

- `opencv_features2d` – розпізнавання і опис плоских примітивів (SURF, FAST і інші, включаючи спеціалізований фреймворк);

- `opencv_video` – аналіз руху і відстеження об'єктів (оптичний потік, шаблони руху, усунення фону);

- `opencv_objdetect` – виявлення об'єктів на зображенні (знаходження облич за допомогою алгоритму Віюлі-Джонса, розпізнавання людей HOG і т.д.);

- `opencv_calib3d` – калібрування камери, пошук стерео-відповідності та елементи обробки тривимірних даних;

					БР.КСМ.07224/17.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		15

– opencv_flann – бібліотека швидкого пошуку найближчих сусідів (FLANN 1.5) і обгортки OpenCV.

Як бачимо, ця бібліотека надає широкий спектр засобів для обробки графіки, в т.ч. для реалізації тих завдань, що стоять перед САМ на усіх рівнях опрацювання зображень. Оскільки ця бібліотека є досить універсальною, в неї наявний інтерфейс і для мови Java, отже вона може бути застосована у даній роботі.

Відповідно до трьох існуючих рівнів обробки медичних зображень, розглянемо порівняльні таблиці характеристик найбільш поширених із цих програмних засобів: таблиця 1.2 – для низького рівня обробки, таблиця 1.3 – для середнього рівня обробки, таблиця 1.4 – для високого рівня обробки зображень.

Таблиця 1.2 – Порівняльна характеристика САМ на низькому рівні обробки зображень

Критерії	Imagej	Axio Visison	BioImage XD	Micro Manager	analySIS FIVE
Контрастування	+	+	+	+	+
Зміна рівня яскравості	+	+	+	+	+
Алгоритми виділення порогів Лапласа	-	+	-	+	+
Фільтри:					
Гаусовий	+	+	+	+	+/-
Медіанний	+	+	+	+/-	+/-
Усереднюючий	+	+	-	+/-	+/-
Швидке перетворення Фур'є	+	+	+	+	+

Таблиця 1.3 – Порівняльна характеристика САМ на середньому рівні обробки зображень

САМ	Метод водорозподілу	Grab Cut	Mean shift	Кодування контурів	RAN SAC	Перетворення Хафа
ImageJ	+	+/-	+	-	+	+/-
AxioVision	+	+	-	+/-	-	+
BioImage	+	-	-	+/-	+	+/-
Micro Manager	+/-	+/-	-	-	+	+

Таблиця 1.4 – Порівняльна характеристика САМ на високому рівні обробки зображень

Критерії	ImageJ	Axio Vision	Bio Image XD	Micro Manager	analySIS FIVE
Автоматична адаптація до зображення	-	-	+	-	-
Детектування об'єктів	+/-	+/-	-	+/-	-
Порівняння зображень	-	+	+	-	-
Класифікатори: нейронні мережі SVM	-	+	-	-	-
	+/-	-	-	+/-	-

Як бачимо, підібрати спеціалізовану САМ, що добре відповідала б різноплановим вимогам, що висуваються до таких систем є непростим питанням, яке буде вирішуватися у наступних розділах [2–8].

1.2 Аналіз біомедичних зображень

Із розвитком інформаційно-комунікаційних технологій з'являлися цілі нові галузі у старих предметних областях. Так, наприклад, інтенсивного розвитку отримало електронне (або дистанційне) навчання. Не стала виключенням і галузь охорони здоров'я: тут виник цілий величезний, магістральний напрямок досліджень, пов'язаний із дистанційним лікуванням, який назвали телемедициною. Основні напрямки цієї, порівняно нової галузі, наведено на рисунку 1.3.

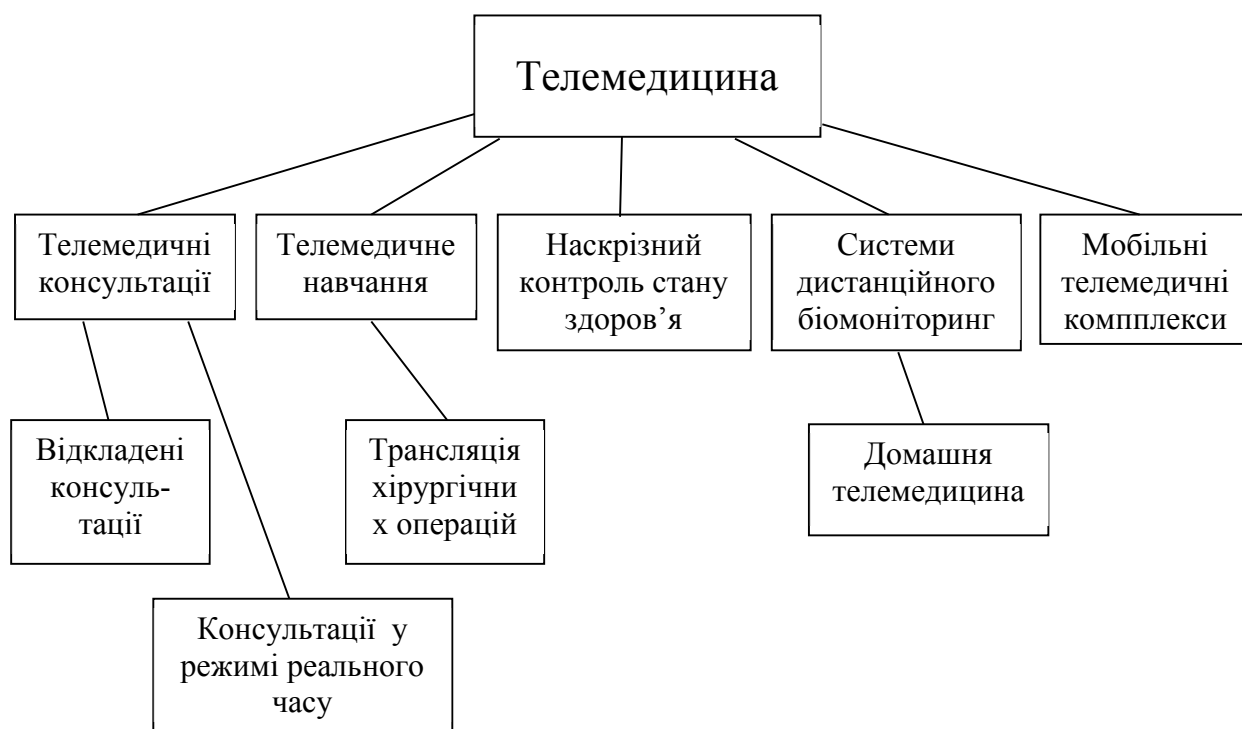


Рисунок 1.3 – Сучасні напрямки розвитку телемедицини

В даний час у багатьох країнах та міжнародних організаціях розвиваються телемедичні проекти. Об'єми світового ринку телемедицини в останні роки оцінюються десятками мільярдів доларів (за даними дослідження P&S Market Research).

					БР.КСМ.07224/17.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		18

Важливим прикладним елементом таких проектів є обробка медичних зображень, а конкретніше, здійснення пошуку об'єктів за відповідними шаблонами. Цей процес вимагає використання глибокої інтелектуальної складової, і втілюється у нових наукових методах обробки зображень. В той же час такі методи не потребують якоїсь дорогої апаратної бази, так як для описаних цілей достатньо сучасних потужних персональних комп'ютерів. Вказаним вимогам у повній мірі відповідають українські науковці, які, зокрема, не маючи доступу до високотехнологічного обладнання, мають високий рівень наукової грамотності, креативності та продуктивності. Так і в галузі телемедицини українські вчені досягли суттєвих успіхів, зокрема, в частині розробки та реалізації алгоритмів обробки медичних зображень. Так, у Тернопільському національному економічному університеті розроблено продукт HIAMS, що дозволяє здійснювати аналіз зображень, виконаних на клітинному рівні з метою виділення передракових станів (та діагностування раку) молочної залози та інших органів і систем.

Цей програмний продукт складається з наступних компонентів:

- база даних;
- модуль обміну повідомленнями – необхідний для забезпечення безперервного зв'язку між лікарями, який використовується для створення можливості оперативних консультацій та швидкого обміну досвідом;
- модуль опрацювання зображень – використовується як для організації нових дослідів із уже існуючими пацієнтами, так і для повторного (чи відкладеного) опрацювання результатів дослідів, що були виконані раніше. Цей модуль містить інтелектуальне ядро розробленого методу та програмного забезпечення в цілому;
- модуль оцінки кількісних та якісних характеристик цитологічних та гістологічних зображень. Сам цей процес є достатньо складним навіть для людини-експерта, тому комп'ютерна обробка відповідних зображень є корисним попереднім етапом, на виході якого може бути сформовано файл із кількісними ознаками зображення. Цей файл може бути як інтерпретований для аналізу

					БР.КСМ.07224/17.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		19

людиною-експертом, так і направлений до підсистеми машинного навчання (штучного інтелекту).

В цілому, розроблений програмний продукт має чимало переваг у порівнянні з програмами-аналогами. Результати відповідного порівняльного аналізу наведено в таблиці 1.5.

Таблиця 1.5 – Порівняльні характеристики програмного продукту HIAMS та найближчих програм-аналогів

Критерій	ImageJ	Image Pro Plus	Axio Vision	Bio Image XD	Micro Manager	AMS Diagnosis	HIAMS
Наявність рівнів доступу користувачів	-	-	-	-	-	-	+
Наявність БД	-	-	+	+/-	-	+	+
Наявність БЗ	-	-	-	-	-	-	+/-
Модуль обміну повідомленнями між користувачами	-	-	-	-	-	-	+
Модуль опису кількісних ознак	+	+	+	+	+	+	+
Модуль опису якісних ознак	-	-	-	-	-	-/+	+
Пошук елемента за шаблоном	-	-	-	+	-	-	+
Облік пацієнтів	-	-	-	-/+	-	-/+	+

Як видно з таблиці 1.5 програмне забезпечення української розробки HIAMS є не тільки конкурентоспроможним до програм-аналогів, присутніх на світовому ринку (чи у відкритому світовому доступі), а й переважає їх по цілому ряду показників. Відповідно, науково-практичний інтерес представляє алгоритм організації пошуку об'єктів, заданих своїми шаблонами, який розглянемо докладніше у наступному підрозділі.

1.3 Аналіз підходів до пошуків об'єктів на зображенні

Для цілей даної роботи розглянемо принципи пошуку зображень, заданих своїми шаблонами (англ. *template matching* – «відповідність шаблону»).

Template matching – це метод обробки цифрового зображення для пошуку невеликих елементів, які відповідають певному заданому шаблону зображення. Метод може бути використаний у виробництві як частина контролю якості, спосіб навігації мобільного роботу, або як спосіб виявлення країв у зображеннях.

Даний метод зустрічається з набором певних проблем серед яких можна назвати наступні: часткове закриття, виявлення нежорстких (із деформаціями) перетворень, зміни освітленості та фону, фоновий зсув та зміни масштабу.

Розглянемо основні підходи до пошуку об'єктів, що відповідають шаблону, відповідно до концепцій *Template matching*.

Підхід на основі особливостей (ознак) – при цьому підході прихований шар нейронної мережі, яка є основою для здійснення алгоритму, виводить вектор, який містить інформацію про класифікацію зображення, і використовується у алгоритмі відповідності шаблонів як особливість зображення[14].

Підхід, заснований на особливостях, спирається на вилучення особливостей зображення, наприклад, форм, текстур, кольорів, щоб відповідати цільовому зображенню або кадру. Цей підхід в даний час досягається за допомогою нейронних мереж і класифікаторів *Deep Learning*, таких як *VGG*,

					БР.КСМ.07224/17.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		21

AlexNet, ResNet. Нейронні мережі з глибокою згорткою обробляють зображення, передаючи його через різні приховані шари, і на кожному шарі виробляють вектор з інформацією про класифікацію зображення. Ці вектори витягуються з мережі і використовуються як особливості зображення. Вилучення функцій за допомогою багаторівневих нейронних мереж є надзвичайно ефективним і, таким чином, є стандартом у сучасних алгоритмах відповідності шаблонів[15, 16].

Цей метод вважається досить надійним і сучасним, оскільки він спрацьовує і при наявності трансформацій шаблону нежорсткого типу, або таких, що виконані за межами площини рисунку; також добре обробляються зображення з інтенсивними фонами та змінами освітленості.

Шаблонно-оснований підхід – для шаблонів без ярко виражених ознак або для тих випадків, коли основна частина картинки шаблону відповідає зображенню, яке перевіряється, може бути ефективним шаблонно-оснований підхід. Як зазначено вище, оскільки шаблонно-заснований Template matching може потенційно вимагати вибірки з великої кількості точок, можна зменшити кількість пробних точок шляхом зменшення роздільної здатності зображень: як того, де ведеться пошук, так і зображення-шаблону. При цьому коефіцієнт зменшення має бути однаковим для обох зображень, і виконання операції на результуючі зменшені зображення (багаторезолюція або піраміда), що забезпечують вікно пошуку точок даних у пошуковому зображенні, так що шаблон не повинен шукати кожну життєздатну точку даних, або комбінацію обох.

Відстеження руху та обробка закриття – можливо також, що шуканий елемент зображення було затьмарене або закрите іншим об'єктом у цих випадках необґрунтовано надавати безліч шаблонів для покриття кожного можливого закриття. Наприклад, пошукове зображення може бути ігровою картою, а в деяких пошукових зображеннях картка закривається пальцями когось, хто тримає картку, або іншою картою на ній, або будь-яким об'єктом перед камерою. У тих випадках, коли об'єкт пластичний або позиціонуємий, рух також стає проблемою, і проблеми, що включають як рух, так і закриття, стають

					БР.КСМ.07224/17.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		22

неоднозначними. У цих випадках одне з можливих рішень полягає в тому, щоб розділити зображення шаблону на декілька суб-зображень і шукати відповідність для кожного підрозділу.

Деформовані шаблони в обчислювальній анатомії – відповідність шаблону є центральним інструментом у обчислювальній анатомії. Модель деформівного шаблону моделює простір анатомій людини на орбіті під груповою дією диффеоморфізмів. Відповідність шаблону виникає як проблема у відповідності невідомого диффеоморфізму, який діє на шаблон, щоб відповідати цільовому зображенню.

Алгоритми узгодження шаблонів в обчислювальній анатомії стали називати диффеоморфним метричним відображенням великих деформацій (LDDMM); тепер існують алгоритми узгодження шаблонів LDDMM для узгодження точок, кривих, поверхонь, об'ємів, орієнтирів.

З числової точки зору відповідність на основі шаблонів встановлюється за допомогою перехресної кореляції або суми абсолютних відмінностей. Базовий метод *Template matching* використовує «патч» зображення (шаблон), пристосований до певної особливості шуканого зображення, яку ми хочемо виявити. Ця методика може бути легко виконана на сірих зображеннях або зображеннях країв. Величина перехресної кореляції буде найвищою в місцях, де структура зображення відповідає структурі маски, де великі значення зображення множаться на великі значення маски.

Таким чином, у підрозділі розглянуто основні підходи до організації *Template matching* – встановлення відповідності зображення (частини зображення) шаблону. Для цього використовуються як детерміновані формули на зразок перехресної кореляції або суми абсолютних відмінностей аналогічних елементів зображень, або принципово інший підхід, що полягає у використанні нейронних мереж. Нейронні мережі видаються більш гнучким засобом, що, в першу чергу, доводиться мільйонами років їх експлуатації у біологічних системах (як око людини та відповідні ділянки кортексу, що обробляють зорові сигнали). Причому підхід на базі нейронних мереж не чутливий до деформацій

					БР.КСМ.07224/17.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		23

шаблону різних типів, впливу фону та рівнів освітленості, наявності рухів чи закриттів, тобто являє собою самий універсальний спосіб Template matching.

Звичайно, при використанні нейронних мереж виникає певна кількість нових проблем, у першу чергу, пов'язаних з необхідністю адекватного навчання мережі. Саме тому, мають право на існування і інші способи Template matching, які є більш простими з організаційної точки зору і являють собою набір готових детермінованих формул.

1.4 Постановка задач дипломного проекту

У розділі після аналізу задач розпізнавання образів, машинного зору, або пошуку зображень за шаблоном розглянуто особливості мов програмування, що можуть бути застосовані для вирішення такого типу задач. Встановлено, що найкращими характеристиками для таких задач володіє мова програмування Java. Поставлена задача буде полягати у створенні програмного модуля для реалізації пошуку об'єктів на зображеннях за шаблоном.

Поставленою метою є розробка програмного модулю для пошуку об'єктів на біомедичних зображеннях, з можливістю вибирати шаблон для пошуку.

Для досягнення поставленої мети потрібно виконати наступні задачі:

- здійснити аналіз категорій біомедичних зображень;
- провести огляд підходів до пошуку об'єктів на зображенні;
- розробити структуру модулю, та визначити його основні функції;
- розробити алгоритм пошуку мікрооб'єктів за вказаним шаблоном;
- здійснити порівняльний аналіз роботи модулю з різними класами зображень;

					БР.КСМ.07224/17.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		24

2 АЛГОРИТМИ РОБОТИ ПРОГРАМНОГО МОДУЛЮ

2.1 Структура модулю аналізу зображень

Модуль аналізу зображень є основною частиною програмного рішення, він виконує такі функції як завантаження вхідних зображень у пам'ять, їхню попередню обробку та фільтрацію, завантаження шаблонів, пошук об'єктів за ними і вивід кінцевих результатів. Структура модуля зображена на рисунку 2.1.

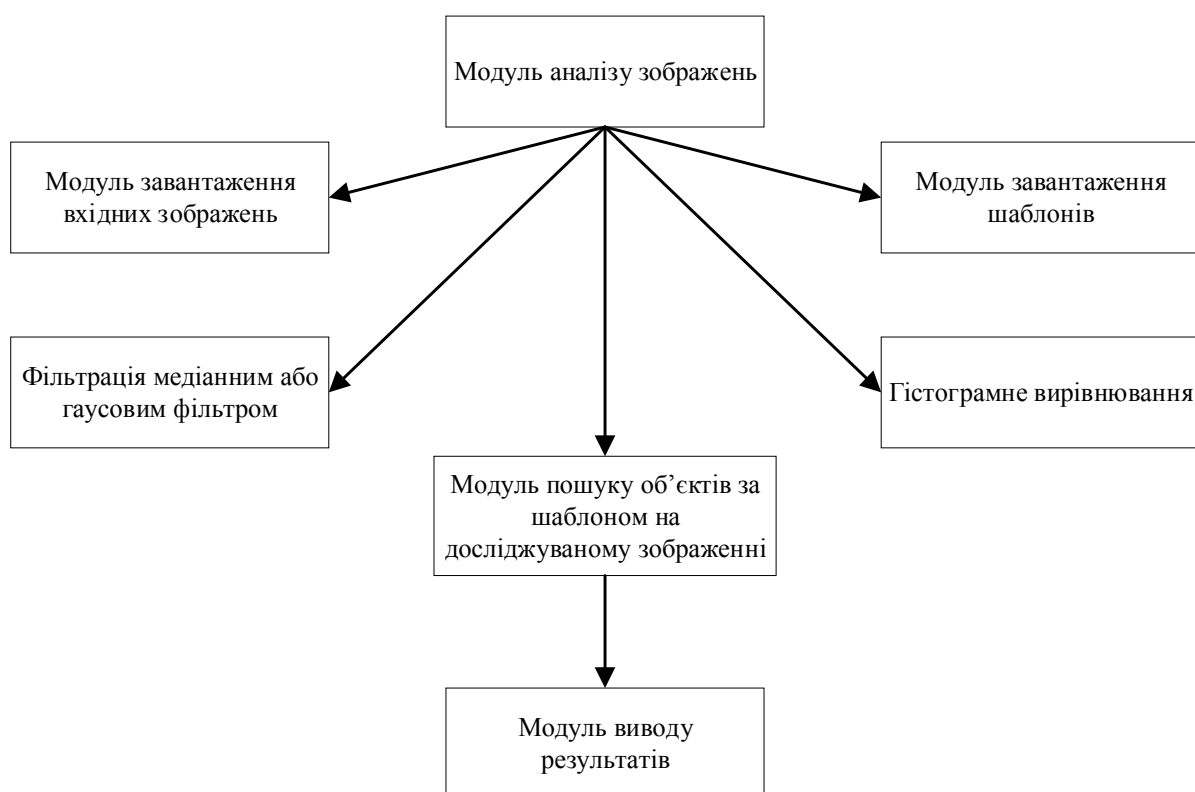


Рисунок 2.1 – Структура модулю аналізу зображень

Робота модулю аналізу зображень відбувається в декілька етапів:

- завантаження зображень у пам'ять;
- завантаження шаблонів;
- фільтрація вхідних зображень медіанним або гаусовим фільтром;
- гістограмне вирівнювання;
- пошук об'єктів за шаблоном;

– вивід результатів пошуку;

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. Алгоритм роботи модуля зображений на рисунку 2.2.

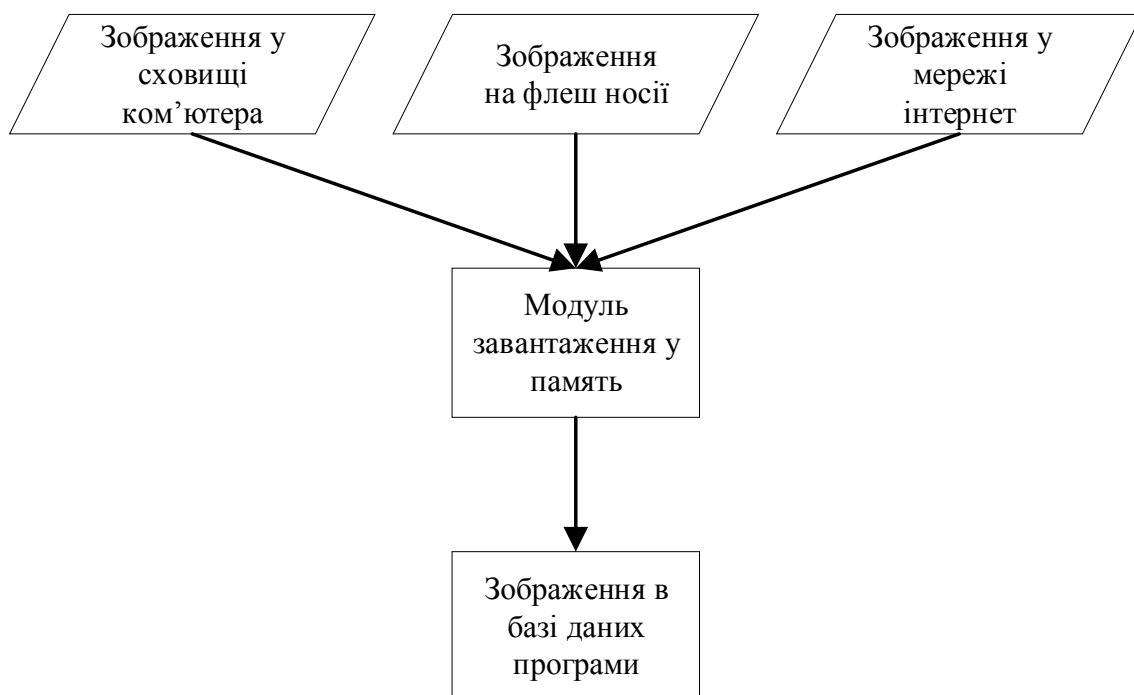


Рисунок 2.2 – Алгоритм завантаження у пам'ять

Після завантаження вхідних зображень відбувається їхня попередня обробка а саме фільтрація медіанним або гаусовим фільтром. Небажаний ефект від завад усувається фільтрацією. Використовуючи фільтрацію для отримання вихідного зображення в якому яскравість (сигнал) окремих пікселів змінюється для усунення завад і спотворень, пікселі які були не були спотворені змінюють яскравість меншим значенням, в результаті отримується більш чисте та чітке зображення. Зображення фільтруються в просторовій і частотній областях. Просторова фільтрація зображення виконує перетворення безпосередньо над значеннями пікселів зображення.

У результаті отримуємо оцінку корисного сигналу зображення. Це досягається завдяки тому, що цифрові зображення часто являють собою двовимірну функцію просторових координат, значення зміни яких є повільніше, ніж завади, яка також є двовимірною функцією. Цей принцип дозволяє оцінювати корисний план сигналу в кожній точці зображення беручи до уваги сусідні точки, використовуючи певну подібність сигналу. В деяких випадках, навпаки, різкі перепади яскравості є ознакою корисного сигналу. Всі типи фільтрів поділяються на чотири класи зображені на рисунку 2.3.

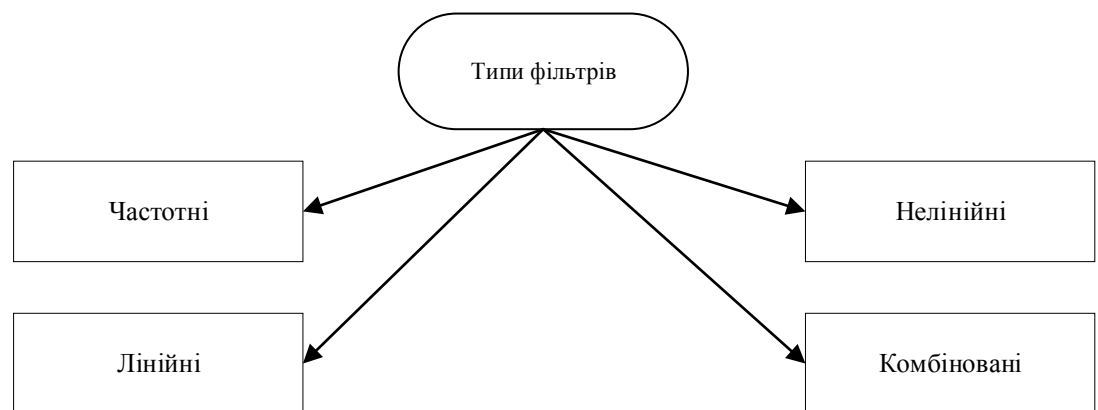


Рисунок 2.3 – Типи фільтрів зображень

При використанні частотних фільтрів обробці підлягають коефіцієнти розкладання зашумленого сигналу по базису Фур'є або інших базисах, зокрема, вейвлет-базису. Перетворення Фур'є грає основну роль для лінійних фільтрів, оскільки множення в Фур'є-області для частотних методів – це операція згортки для вихідного зображення. У обробці цифрових сигналів часто використовуються методи лінійної фільтрації. Проте, хороші результати лінійна фільтрація дає лише в разі наявності гауссово адитивного шуму. Для імпульсного шуму ефективніше використовувати методи нелінійної фільтрації, а саме медіанна та рангова фільтрація. А якщо шум комбінований то можна послідовно застосовувати лінійні і нелінійні фільтри або компонувати ці фільтри щоб посилити їх сильні сторони і мінімізувати недоліки, це відбувається

при використанні гібридних фільтрів. Лінійні фільтри також називають згладжуючими або усереднюючими, тому що відповідь лінійного фільтра усереднює значення пікселів, що містяться в апертурі, і таким чином згладжує зображення [11–13]. Фільтр гауса є один з найбільш поширених методів очистки вхідних зображень від шумів, приклад його роботи зображений на рисунку 2.4.

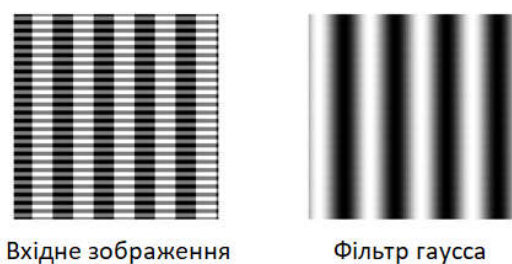


Рисунок 2.4 – Фільтр гауса

Для подальшого покращення якості вхідних зображень використовується вирівнювання гістограми та різного роду її модифікації. Основні параметрами, які використовуються в даних методах покращення якості зображення є середнє значення гістограми. Приклад гістограми зображення наведений на рисунку 2.5.

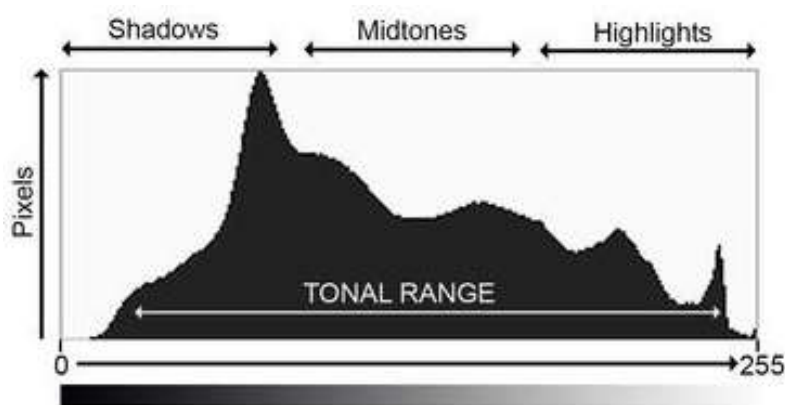


Рисунок 2.5 – Приклад гістограми яскравості зображення

Для звичайного гістограмного вирівнювання спочатку проводиться розрахунок гістограми вхідного зображення, а на наступному етапі відбувається саме нелінійне перетворення [16]. При модифікації гістограми відбувається

зміна рівнів яскравості та контрасту зображення, його затемнення або висвітлення яке дає змогу виділити певні мікрооб'єкти на зображенні. Результатом гістограмного вирівнювання є сильний перепад яскравості на межах та краях об'єктів, і їх відповідне виділення на фоні зображення.

На етапі пошуку об'єкту за шаблоном, шаблонне зображення послідовно накладається на досліджуване зображення і між ними вираховується кореляція. Приклад пошуку об'єкту зображено на рисунку 2.6.

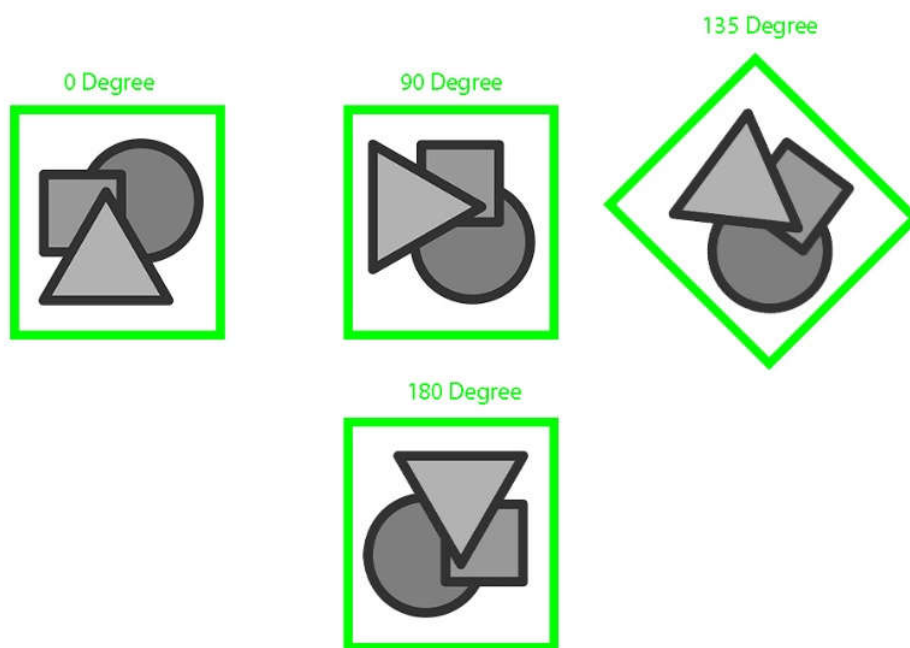


Рисунок 2.6 – Пошук за шаблоном

Кореляція це метод співставлення зображень, який використовує швидкий підхід з переходом у частотну область для визначення взаємного паралельного зсуву двох однакових зображень. Результат кореляції заноситься у вихідне зображення [17]. Після пошуку результат виводиться у вигляді початкового зображення на якому виділені області де знаходиться шуканий нами об'єкт, отримане вихідне зображення можна зберегти на комп'ютері або зовнішньому носії.

2.2 Алгоритм пошуку об'єкту за шаблоном

Для пошуку об'єкта на зображенні був застосований метод кореляції. До переваг даного методу відносяться простота застосування, стійкість до адитивної і мультиплікативної зміни яскравості зображення. Однак даний метод чутливий до неоднорідних змін яскравості і контрастності, що знижує область його застосування.

Алгоритм пошуку складається з п'яти кроків.

Крок 1 – зчитування шаблону та досліджуваного зображення.

Крок 2 – розрахунок кількості пікселів, та створення матриці.

Крок 3 – підрахунок кореляції для кожного сегменту зображення.

Крок 4 – визначення позиції з найбільшою кореляцією.

Крок 5 – вивід результату пошуку об'єкту.

Для опису зображення будується матриця, в якій відображені показники порівнянь областей зображення кожен з кожним. Елементи такої матриці будемо називати ознаками. Приклад матриці кореляцій зображення наведено на рисунку 2.7.

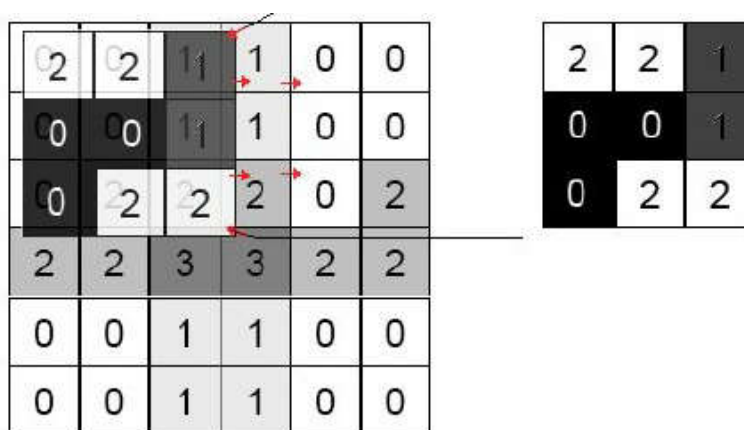


Рисунок 2.7 – Матриця кореляцій

Для пошуку об'єктів на зображенні, розроблений метод використовує ознаки яскравості. Для усунення впливу освітлення і інших особливостей,

пов'язаних з умовами зйомки використовуються відносні показники яскравості: світліше, темніше, приблизно рівні. У БР.КСМ. 07224/17.00.00.000 А2 наведено блок-схему алгоритму пошуку об'єкту за шаблоном.

Для роботи розробленого методу використовується шаблон. Якщо розділити складену матрицю на дві трикутних по головній діагоналі, то можна помітити, що показники мають зворотну залежність, наприклад: область А світліше області В, відповідно область В темніше області А. Розмір областей вибирається шляхом перебору. Критерієм якості є відсоток правильно класифікованих об'єктів. Розмір областей повинен бути не менше цікавлять особливостей об'єкта. Використання всіх пікселів зображення в якості ознак не дає поліпшення якості роботи. Це пов'язано з попаданням шумів, які усереднюються при використанні групи пікселів.

Показник подібності з шаблоном визначається як відношення накладних ознак до загальної кількості ознак. Розмірність матриць шаблону і зображення повинні збігатися. Так само шаблон чутливий до масштабу і повороту об'єкта. Отже, на практиці необхідно використовувати методи попередньої обробки для нормалізації геометрії об'єкта.

2.3 Структура бази даних

База даних – набір впорядкованих та логічно взаємопов'язаних даних, призначених для використання в конкретній галузі чи сфері, і для покращення доступу до відповідної інформації користувачами [23].

Головні цілі розробки БД:

- структуроване зберігання потрібної інформації;
- зручний доступ до всіх даних;
- зменшення надмірності і дублювання даних;
- забезпечення цілісності даних;

					БР.КСМ.07224/17.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		31

Система управління базами даних (СУБД) – набір програмних, та лінгвістичних методів спеціалізованого або загального використання, що забезпечують управління створенням і підтримання роботи з базою даних. СУБД виконують основні завдання по інтеграції та взаємодії з даним із сторони додатків чи корпоративних систем [22].

Основні функції СУБД такі:

– управління даними в постійній пам'яті – використовується користувачам для виконання найбільш поширених операцій, які здійснюються з даними, а саме – додавання, збереження, видалення та оновлення інформації, а також підтримка структури зовнішньої пам'яті для ефективного зберігання даних, і прискореного до них доступу;

– управління транзакціями – організація та упорядкування операцій над БД у послідовному вигляді з метою доступу або зміни вмісту бази даних;

– відновлення бази даних – забезпечення надійності збережених у БД даних а також реалізація можливості відновлення останнього узгодженого стану БД після виникнення помилок чи збоїв;

– підтримка різних мов – використання спеціальних мов, що дозволяють формувати запити для роботи з БД;

– реалізація паралельного доступу – надання одночасного доступу для багатьох користувачів до оброблюваних даних. СУБД повинна вирішувати конфлікти коли два або більше користувачів одночасно отримують доступ до бази даних, основною задачею в цьому випадку є усунення небажаних конфліктів, наприклад, при одночасному оновленні та зчитуванні даних. СУБД повинна гарантувати, що при одночасному доступі до бази даних багатьох користувачів цих конфліктів не відбудеться;

– керування буферами в оперативній пам'яті - підтримка окремого числа буферів оперативної пам'яті з протоколом заміни буферів для збільшення швидкодії та продуктивності роботи БД;

– контроль та розподілення доступу до даних – механізм, який забезпечує захист бази даних від несанкціонованого доступу з боку сторонніх користувачів

					БР.КСМ.07224/17.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		32

і захищає її від будь-якого зловмисного втручання. В сучасних системах управління базами даних використовуються декілька з широко поширених методів вирішення питання безпеки даних: вибірковий підхід або обов'язковий підхід [21]. Сьогодні у більшості СУБД задіяний вибірковий підхід, при якому різні користувачі володіють різними правами доступу при роботі з різними об'єктами. Доволі нечасто застосовують обов'язковий, де кожен об'єкт даних має певний рівень класифікації до якого він належить, а користувачі розділяються деякими рівнями допуску;

– підтримка цілісності даних – забезпечує коректність і несуперечність збережених у БД даних. СУБД завжди отримує інструкції та правила, які вказують як саме проводити роботу з даними, і володіє відповідними інструментами контролю щоб дані та їх зміни задовольняли задані інструкції.

Сам процес проектування БД складається з таких етапів:

- визначення ролей користувачів;
- визначення станів і переходів;
- визначення сутностей;
- створення схеми сутностей і їх зв'язків;
- створення таблиць та зв'язків у рамках конкретної СУБД;
- створення об'єктів БД;
- розробка скриптів створення БД.

Перші 4 завдань відносяться до логічного проектування БД тобто тут проектувальник вирішує загальні завдання організації БД, не прив'язуючись до конкретної СУБД, хоча найчастіше вже на ранніх стадіях розробки ІС приймається рішення про застосовувану СУБД.

Тому в реальному житті вже всі рішення приймаються з урахуванням особливостей конкретної СУБД. Вибір СУБД майже завжди диктується не технічними, а економічними або кон'юнктурними міркуваннями [25].

Останні три завдання ставляться до технічного проектування, на цьому етапі логічні схеми і правила виливаються в конкретні об'єкти обраної СУБД. Конструкція таблиць, обмежень, уявлень, і інших об'єктів БД оформляється у

					БР.КСМ.07224/17.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		33

вигляді операторів мови SQL або характерного для обраної СУБД розширення мови SQL. Отримана таким чином програма і є результатом проектування БД.

Отже для проектування бази даних вибрано мову SQL, наступними будуть наведені переваги цієї мови:

- стандартизованість – мова запитів SQL є стандартизованою і використовується в багатьох програмах міжнародного рівня;
- наявність широкого вибору СУБД – реляційна база даних може бути перенесена з однієї СУБД на іншу;
- легкість в розумінні та інтуїтивність з роботою із табличними структурами реляційної БД;
- можливість написання багатофункціональних запитів – SQL надає користувачам можливість роботи з базою даних написання програми;
- можливість підключення програм для доступу до БД – мову SQL зручно використовувати в програмних модулях які потребують зв'язку з базою даних;
- надання багатьох методів представлення даних – дані які зберігаються у БД можуть бути згруповані і виведені у вигляді однієї простої таблиці;

Крім цього, мова SQL відповідає всім вимогам для роботи з базами даних та здатна виконувати всі поставлені вище завдання при мінімізації зусиль з сторони користувача, також структура і синтаксис її команд – досить прості і доступні для швидкого вивчення [26].

Основні типи даних які можуть зберігатися в SQL базі даних:

- текстовий – зберігає інформацію у вигляді звичайного тексту;
- числовий – може зберігати числа як цілі так і з дійсні;
- дата/час – для зберігання дати та часу;
- грошовий – зберігає суми у національних валютах;
- логічний – зберігає двійкове значення «так» або «ні»;
- гіперпосилання – використовується для збереження URL адресс.

В даній роботі було створено саме реляційну базу даних вона показує класи включно з їх методами і атрибутами, а також статичні взаємозв'язки між ними,

					БР.КСМ.07224/17.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		34

котрі класи знають, які класи існують і які класи наслідують інші класи. Класифікація баз даних зображена на рисунку 2.8.

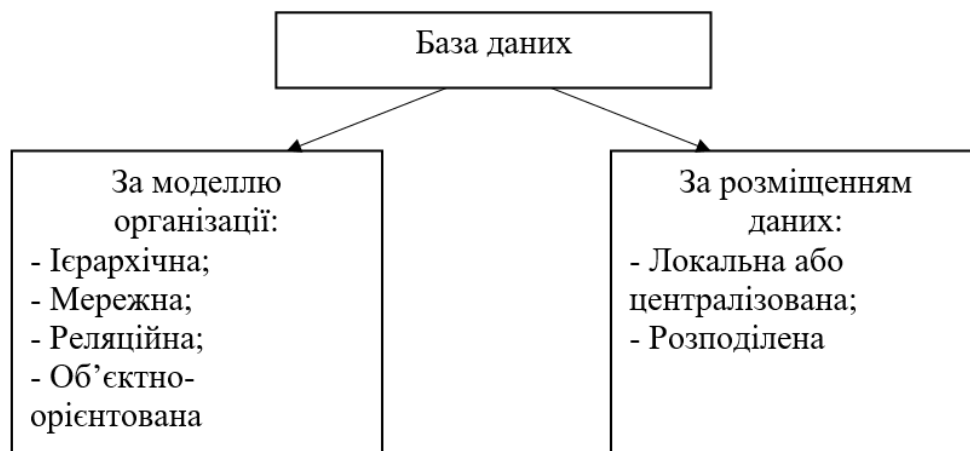


Рисунок 2.8 – Класифікація баз даних за структурою організації

Реляційна модель даних зберігає дані у вигляді певної кількості таблиць. Таблиця це сукупність записів з фіксованим числом полів. Таблиці в реляційних БД об'єднуються між собою за допомогою первинних ключів. Первинний ключ - поле що однозначно визначає запис (екземпляр сутності), використовується для зв'язку з іншими таблицями. Значення цього поля повинні бути унікальні в межах таблиці. Такий ключ може бути тільки один на таблицю [27]. У реальних БД в якості первинних використовуються сурогатні ключі. Структура таблиць реляційної бази даних розроблюваного програмного модулю наведена нижче. Структура таблиці «Patients» наведена у таблиці 2.1. Дана таблиця використовується для зберігання даних про пацієнтів у системі.

Таблиця 2.1 – Структура таблиці «Patients»

Назва поля	Тип даних	Опис
id_patients	Int	Первинний ключ
patient_name	Varchar(100)	Ім'я та прізвище пацієнта
patient_gender	Varchar(1)	Стать пацієнта
patient_dob	Date	Дата народження пацієнта
adress	Tinytext	Місце проживання пацієнта

Наступною важливою сутністю у розроблюваній базі даних є лікар структура та атрибути якого наведені у таблиці 2.2.

Таблиця 2.2 – Структура таблиці «Doctors»

Назва поля	Тип даних	Опис
id_doctors	Int	Первинний ключ
doctors_name	Varchar(100)	Ім'я та прізвище лікаря
doctors_speciality	Varchar(60)	Спеціальність лікаря
clinic	Tinytext	Клініка в якій працює лікар

Кожне нове звернення пацієнта до лікаря створює нову медичну справу, пацієнта в якій заноситься дата звернення, основні скарги пацієнта і всі зображення які будуть отримані в ході аналізів. Структура наведена у таблиці 2.2.

Таблиця 2.2 – Структура таблиці «MedResearch»

Назва поля	Тип даних	Опис
id_med_res	Int	Первинний ключ
date	Date	Дата звернення пацієнта
images	Mediumtext	Посилання на зображення
id_patients	Int	Зовнішній ключ на ID пацієнта
subject	Mediumtext	Записи скарг та симптомів пацієнта

Структуру таблиці «ImageCharacteristics» наведено у таблиці 2.3 Дана таблиця, вміщує розширену інформацію про зображення їхні кількісні та якісні характеристики та результати їх аналізів, кожна сутність цієї таблиці має зв'язок багато до одного із сутністю медичної справи так як важливим є зберігання інформації із всіх забраних в ході аналізів зображень пацієнта.

Таблиця 2.3 – Структура таблиці «ImageCharacteristics»

Назва поля	Тип даних	Опис
idImageCharacteristics	Int	Первинний ключ
summary	Mediumtext	Примітки лікаря
quantitative	Mediumtext	Кількісні характеристики зображень
qualitative	Mediumtext	Якісні характеристики зображень
MedResearch_id_med_res	Int	Зовнішній ключ на ID медичної справи

Після отриманої інформації з аналізів та зображень пацієнта може бути поставлений діагноз, для зберігання інформації про діагноз дату його постановки лікаря який його встановив та назначене лікування створена таблиця «Diagnosis» структуру якої наведено у таблиці 2.4.

Таблиця 2.4 – Структура таблиці «Diagnosis»

Назва поля	Тип даних	Опис
idDiagnosis	Int	Первинний ключ
finaldiagnosis	Mediumtext	Текст кінцевого діагнозу
MedResearch_id_med_res	Int	Зовнішній ключ на ID медичної справи
id_doctor	Int	Зовнішній ключ на ID лікаря
date_of_diagnosis	Date	Дата постановки діагнозу
treatment	Mediumtext	Назначене лікування

Таким чином, у даному підрозділі було розглянуто та проаналізовано основні технології розробки та впровадження баз даних у програмний продукт, а також визначено основні етапи розробки БД, було розроблено її структуру та визначено основні сутності їх властивості та встановлено зв'язки між ними. У БР.КСМ. 07224/17.00.00.000 С1 наведено структуру бази даних.

3 РОЗРОБКА ТА ТЕСТУВАННЯ МОДУЛЮ

3.1 Структура розробленого програмного модулю

Робота програмного модулю поділена на декілька основних функцій, кожна з яких отримує та передає інформацію наступній, і забезпечує коректну роботу та зручний вивід результату для кінцевого користувача. Список функцій зображений на рисунку 3.1.

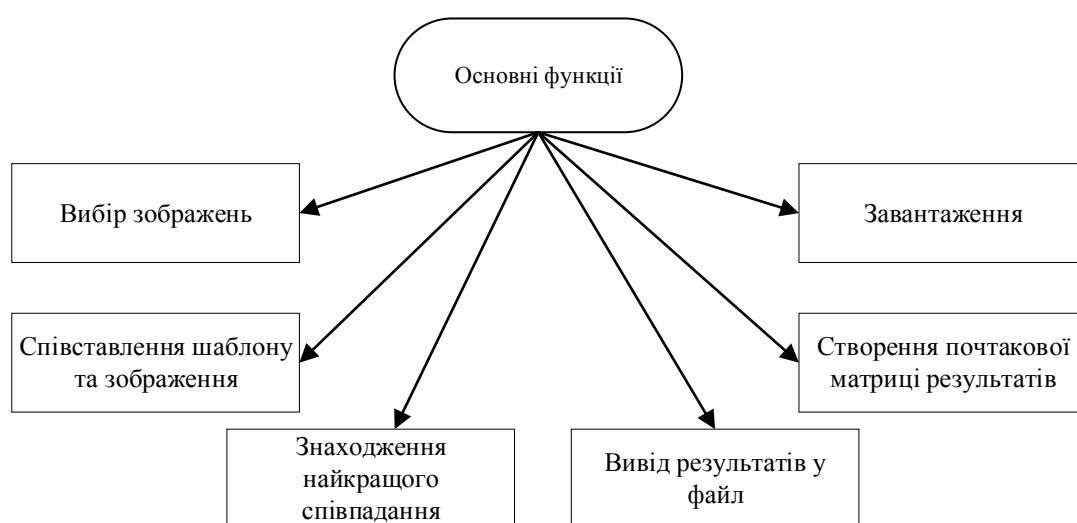


Рисунок 3.1 – Основні функції програмного модулю

Робота програмного модулю починається із вибору вхідних зображень, який здійснюється за допомогою отримання шляху до їхнього розміщення на комп'ютері користувача. Даний функціонал реалізується наступним програмним кодом.

```
String infile = "D:\\concertest1.jpg";  
String templateFile = "D:\\templatel.png";
```

Змінні типу «String» зберігають в собі дані у текстовому форматі які надалі будуть використовуватися для завантаження файлів зображень.

Наступною є функція завантаження вхідних зображень які знаходяться на комп'ютері користувача. На них буде здійснюватися пошук об'єктів відповідно до вибраного шаблону. Нижче наведений програмний код який відповідає за завантаження зображень.

```
Mat img = Imgcodecs.imread(inFile2);  
Mat templ = Imgcodecs.imread(TemplateFile2);
```

У даному коді використовується контейнер «Mat» для завантаження вхідних зображень, він являє собою клас, який зберігає матрицю, інформацію про її розмір та вказівник на її розташування, комірки самої матриці зберігають в собі числові значення відповідних пікселів зображення, яке було завантажене. Метод imread отримує атрибутом шлях до зображення і завантажує його.

Після завантаження створюється початкова матриця, в яку буде занесено результати співставлення зображення та шаблону. Реалізація цієї функції наведена нижче.

```
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);
```

Спочатку проводиться визначення розміру матриці, і в змінні записується потрібна кількість рядків і стовпців, потім створюється новий об'єкт класу «Mat» який отримує значення цих змінних як атрибути.

Коли матриця результатів є створена можна переходити до співставлення зображення та шаблону. Наступний код виконує дану функцію.

```
Imgproc.matchTemplate(img, templ, result, 4);
```

Функція «matchTemplate» порівнює шаблон із кожним сегментом вхідного зображення і записує результат їх кореляції. Із всіх співставлень проводиться знаходження найкращого збігу, та остаточне визначення позиції об'єкту. Ця функція реалізована наступним кодом.

					БР.КСМ.07224/17.00.00.000 ПЗ	Арк.
						39
Змн.	Арк.	№ докум.	Підпис	Дата		

```

MinMaxLocResult mmr = Core.minMaxLoc(result);
Point matchLoc = mmr.maxLoc;

```

Функція «minMaxLoc» отримує атрибутом матрицю кореляцій, шукає найбільш точні збіги і повертає місце де знаходиться шуканий об'єкт.

Останнім кроком є обведення знайденого об'єкту та збереження вихідного зображення. Нижче наведена реалізація даної функції з використанням бібліотеки OpenCV.

```

Imgproc.rectangle(img, matchLoc,
    new Point(matchLoc.x + (double)templ.cols(), matchLoc.y +
    (double)templ.rows()), new Scalar(0.0D, 255.0D, 0.0D), 4);
    Imgproc.putText(img, "Found", new Point(matchLoc.x - 60.0D,
    matchLoc.y - 10.0D), 3, 1.0D, new Scalar(0.0D, 255.0D, 0.0D));
    Imgcodecs.imwrite(outFile2, img);

```

Метод «rectangle» використовується для обведення знайденого об'єкту прямокутником заданого розміру і кольору, висота та ширина якого визначається розмірами самого шаблону а колір вводиться в параметрах. Метод «putText» використовується для додавання надпису біля знайденого об'єкту, а «imwrite» для збереження остаточного результату у файл, після чого він може бути переглянутий користувачем. Діаграма класів наведена на рисунку 3.2.

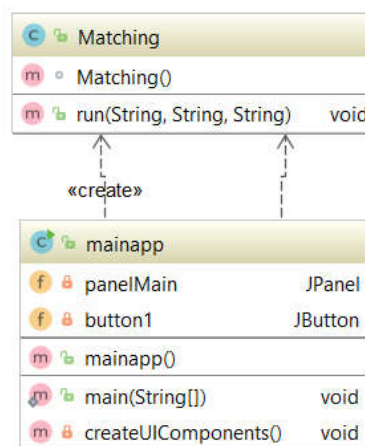


Рисунок 3.2 – UML діаграма класів модулю аналізу зображень

Вона показує класи включно з їх методами і атрибутами, а також статичні

взаємозв'язки між ними: котрі класи знають, які класи існують і які класи наслідують інші класи.

Головний клас «mainapp» вміщує функції роботи елементів інтерфейсу, метод «createUIComponents» використовується для виклику компонентів інтерфейсу при запуску програми, метод «main» є основною функцією в якій використовуючи команду «create» створюється новий об'єкт класу «Matching». Клас «Matching» має основний метод «run» який містить в собі всі функції та змінні які безпосередньо використовуються для пошуку об'єктів на зображенні.

3.2 Модуль роботи з БД

Для покращення зручності та швидкості користування програмним засобом, було розроблено модуль роботи з SQL базою даних, в якій знаходиться вся потрібна інформація для функціонування та додаткові дані про досліди та аналізи зображень які можуть бути використані в сфері телемедицини. Робота модулю здійснюється з використанням багатьох SQL запитів до БД, для вибірки, додавання, та редагування даних і записів у таблиці.

Як і аналіз зображень, модуль роботи з БД складається із декількох функцій, які виконуються при відповідних діях користувача, а саме:

- підключення до бази даних;
- додавання записів у таблиці;
- вибірка записів з таблиці;
- редагування та видалення записів;

Кожній з цих дій відповідає певний SQL запит. Деякі з основних запитів наведені на рисунку 3.3.

					БР.КСМ.07224/17.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		41

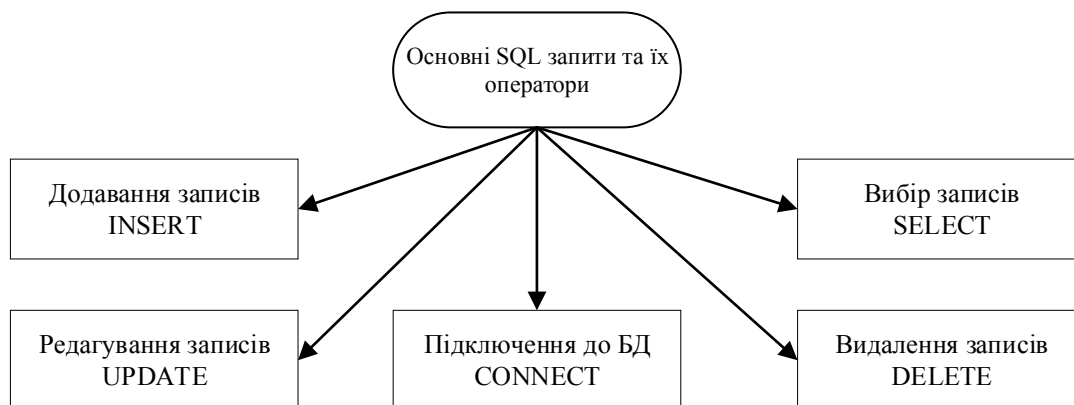


Рисунок 3.3 – SQL запити та оператори які використовувалися

Підключення до БД здійснювалося з використання вбудованої бібліотеки `java.sql`. Код для підключення до бази даних наведений нижче.

```

Connection con=DriverManager.getConnection( "jdbc:mysql://localhost:3306/ haimsdb", "username", "password");
  
```

Тут «localhost:3306» є адреса і порт SQL сервера, «haimsdb» назва нашої бази даних до якої буде здійснюватися підключення, «username» і «password» це імя користувача та його пароль.

Для додавання нових записів у таблиці використовується оператор `insert`. Приклад використання оператора в коді програми

```

insert into patients (id_patients, patient_name, patient_gender, patient_dob, adress) values (4, 'Kate', 'F', 19980211, 'Lviv')
  
```

Синтаксис оператора вимагає вказати назву таблиці, назви стовпців та після значення «values» дані які будуть заноситися. Будь який стовпець, для якого явно не вказано значення, буде встановлений у своє значення за замовчуванням. Наприклад, якщо в заданому списку стовпців не зазначені всі стовпці в даній таблиці, то не згадані стовпці встановлюються у свої значення за замовчуванням.

Для вибору запитів з таблиці використовується запит «select», він застосовується для вилучення рядків, вибраних з однієї або декількох таблиць.

Приклад використання

```
select * from patients where patient_gender = 'M' order by patient_dob
```

У даному виразі задаються стовпці, в яких необхідно проводити вибірку, значення яке повинно бути у записі в конкретному стовпці, та порядок сортування. Вираз «from» задає ім'я таблиці, з якої належить витягувати рядки. У виразі «where» можна використовувати будь-яку з функцій, яка підтримується в MySQL.

Запит «delete» використовується для видалення записів із таблиць, його синтаксис схожий до синтаксису «select» лише замість вибору відповідних записів відбувається їх видалення. Використання зпиту в кодї програми:

```
delete from patients where patient_gender = 'M'
```

Як і в попередньому запиті тут вибираються та видаляються записи із таблиці «patients» які відповідають умові заданій після виразу «where».

Коли потрібно оновити вже існуючий запис чи додати до нього нові дані, використовується оператор «update». Приклад використання запиту в кодї програми.

```
update patients set adress = 'Kiev' where id_patients = 4
```

Оператор update оновлює стовпці відповідно до їх нових значень в рядках існуючої таблиці. У виразі set вказується, які саме стовпці слід модифікувати і які величини повинні бути в них встановлені. У виразі «where», якщо воно присутнє, задається, які рядки підлягають оновленню. В інших випадках оновлюються всі рядки.

					БР.КСМ.07224/17.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		43

3.3 Тестування розробленого модулю

Для тестування роботи модулю була використана вибірка із багатьох мікробіологічних зображень клітин, та тканин із різними патологіями та захворюваннями. Перевірка пошуку із одним шаблоном та вхідним зображенням, на рисунку 3.4 зображено вхідне мікробіологічне зображення.



Рисунок 3.4 – Зображення з раковою клітиною

Задача модулю на цьому етапі знайти на заданому зображенні об'єкт який буде відповідати шаблону. Шаблон для пошуку взято із вхідного зображення і він зображений на рисунку 3.5.



Рисунок 3.5 – Шаблон для пошуку

Параметри пошуку зображень були встановлені для знаходження лише одного об'єкту на зображенні.

					БР.КСМ.07224/17.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		44

Результат роботи можна перевірити на рисунку 3.6.

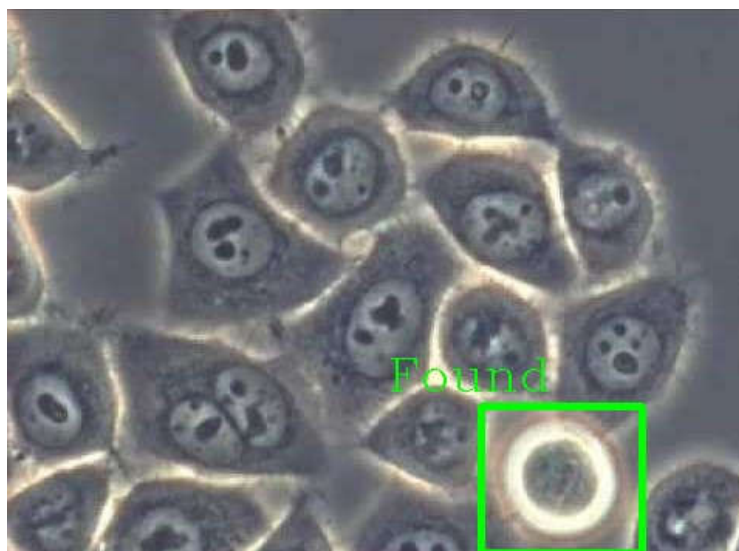


Рисунок 3.6 – Результат пошуку

Подальше тестування роботи модулю проводилося на різних класах цитологічних зображень (Цито-непроліферативна мастопатія – клас А, цито-кістозна мастопатія – клас Б, цито фіброзно-кістозна мастопатія – клас В). На рисунку 3.7 зображено приклад роботи на першому класі зображень.

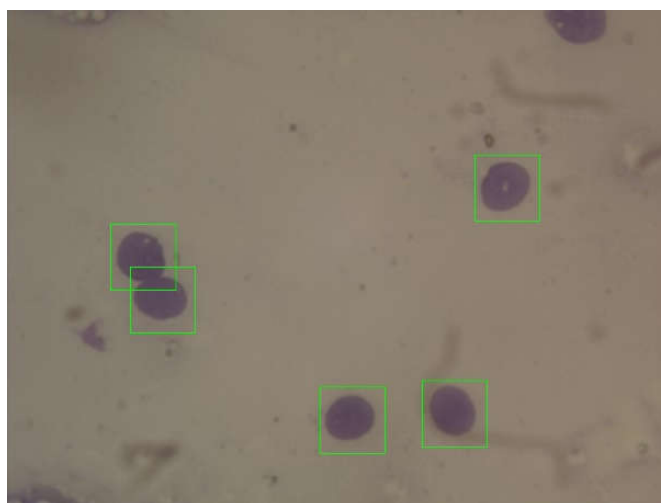


Рисунок 3.7 – Пошук на цитологічному зображенні класу А

Було взято окремі шаблони для кожного класу зображень та перевірено точність знаходження об'єктів по кожному шаблону.

					БР.КСМ.07224/17.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		45

Пошук клітин на зображенні класу Б зображено на рисунку 3.8.

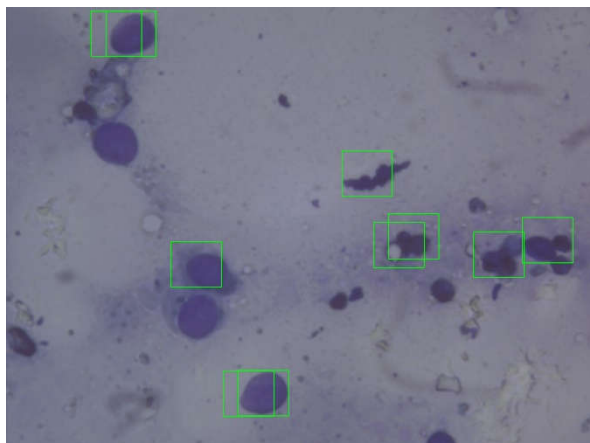


Рисунок 3.8 – Пошук на цитологічному зображенні класу Б

В результатах пошуку проявляється неточність та помилкове визначення клітин особливо при наявності скупчення. Наступним був проведений пошук на зображеннях класу В, результат якого можна побачити на рисунку 3.9.

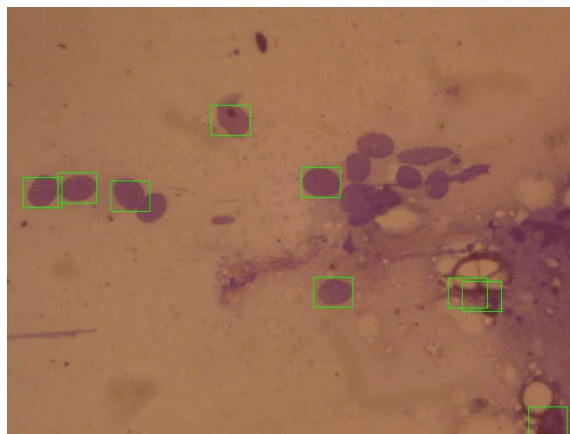


Рисунок 3.9 – Результат пошуку на зображеннях класу В

Як бачимо при аналізі різних класів зображень є розбіжності в точності та якості знаходження об'єктів на кожному з них. Помилкові визначення клітин та виділення не всіх клітин у їх скупченнях є найбільш поширеним типом помилок які допускаються роботою модуля. Дані зібрані після тестування роботи модуля, є зображенні у вигляді діаграми на рисунку 3.10.

					БР.КСМ.07224/17.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		46

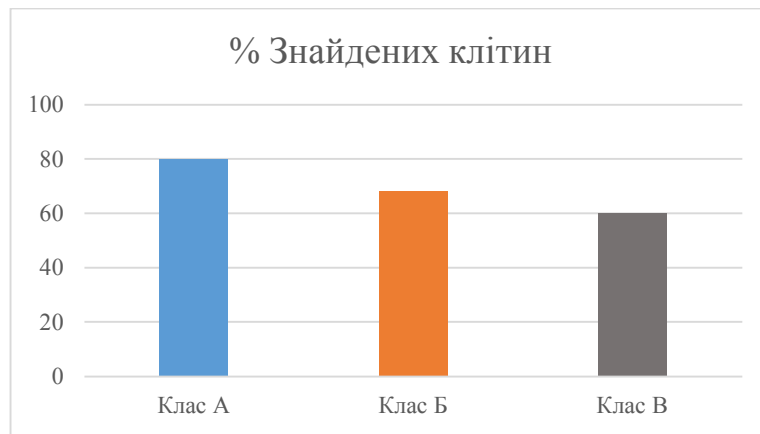


Рисунок 3.10 – Відсоток успішно знайдених клітин

Після аналізу отриманої інформації, можна побачити що розроблений програмний продукт все ж таки має не абсолютну точність в розпізнаванні на зображеннях. На рисунку 3.11 зображено декілька з невдало визначених клітин.

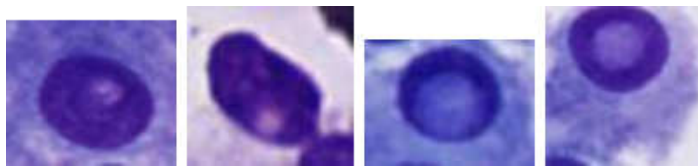


Рисунок 3.11 – Клітини які не були знайдені

Для порівняння на рисунку 3.12 можна побачити успішно знайдені програмним модулем клітини.

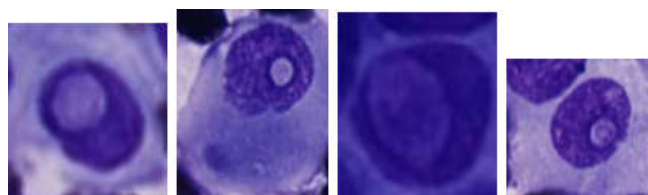


Рисунок 3.12 – Успішно визначені клітини

Після перегляду результатів було визначено що одним з чинників помилкових визначень є низька якість вхідного зображення, а також висока скупченість клітин. Запропонованим вирішенням цієї проблеми було збільшення

розміру вхідного зображення і відповідно підвищення його якості, проте це значно впливає на час роботи програми. На рисунку 3.13 зображена діаграма часу роботи програми в залежності від розміру вхідного зображення та кількості сегментів.

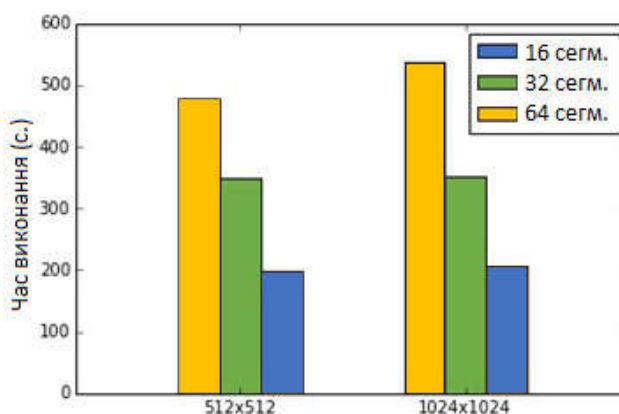


Рисунок 3.13 – Діаграма часу пошуку об’єктів

Аналізуючи час роботи модулю над зображеннями різного розміру та різного рівня сегментації було отримано наступні дані. На рисунку 3.14 наведено діаграму часу виконання кожного етапу роботи модуля.

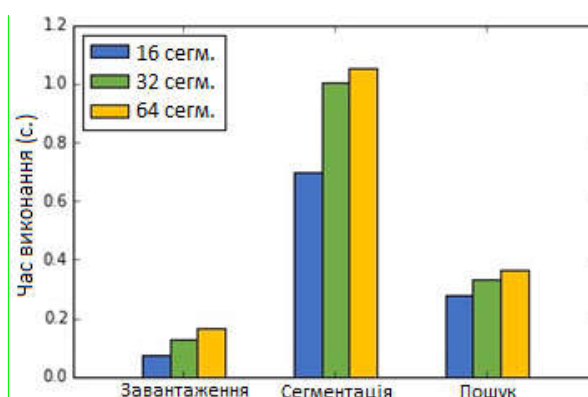


Рисунок 3.14 – Діаграма часу виконання кожного етапу

Із наведених діаграм та таблиць про успішність знаходження об’єктів, а саме клітин на мікробіологічних зображеннях, можна зробити висновок що робота модулю задовільняє поставлені в ТЗ критерії, але його продуктивність та точність може бути збільшена.

4 ТЕХНІКО-ЕКОНОМІЧНЕ ОБҐРУНТУВАННЯ РОЗРОБКИ ПРОГРАМНОГО ЗАСОБУ

4.1 Розрахунок витрат на розробку програмного забезпечення

4.1.1 Розрахунок витрат на розробку програмного забезпечення

Витрати на розробку і впровадження програмних засобів (K) включають:

$$K = K_1 + K_2, \quad (4.1)$$

де K_1 – витрати на розробку програмних засобів, грн.;

K_2 – витрати на відлагодження і дослідну експлуатацію програми рішення задачі на комп'ютері, грн.

Витрати на розробку програмних засобів включають:

– витрати на оплату праці розробників ($B_{оп}$);

– витрати на відрахування у спеціальні державні фонди ($B_{ф}$);

– витрати на покупні вироби ($П_B$);

– витрати на придбання спецобладнання для проведення експериментальних робіт ($O_б$);

– накладні витрати (H);

– інші витрати (I_B).

4.1.2 Розрахунок витрат на оплату праці та соціальні заходи

Витрати на оплату праці включають заробітну плату (ЗП) всіх категорій працівників, безпосередньо зайнятих на всіх етапах проектування. Розмір ЗП обчислюється на основі трудомісткості відповідних робіт у людино-днях та середньої ЗП відповідних категорій працівників.

					БР.КСМ.07224/17.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		49

У розробці програмного забезпечення задіяні наступні спеціалісти-розробники, а саме: керівник проекту; студент-дипломник; консультант техніко-економічного розділу. Вихідні дані для розрахунку витрат на оплату праці наведено у таблиці 4.1.

Таблиця 4.1 – Вихідні дані для розрахунку витрат на оплату праці

№ п/п	Посада виконавців	Місячний оклад, грн.
1	Керівник ДП, викладач	5950
2	Консультант техніко-економічного розділу, доцент	7293
3	Студент	1400

Витрати на оплату праці розробників проекту визначаються за формулою:

$$B_{оп} = \sum_{i=1}^N \sum_{j=1}^M n_{ij} \cdot t_{ij} \cdot C_{ij}, \quad (4.2)$$

де n_{ij} – чисельність розробників i -ої спеціальності j -го тарифного розряду, осіб;
 t_{ij} – затрачений час на розробку проекту співробітником i -ої спеціальності j -го тарифного розряду, год;

C_{ij} – годинна ставка працівника i -ої спеціальності j -го тарифного розряду, грн.

Середньо годинна ставка працівника може бути розрахована за формулою:

$$C_{ij} = \frac{C_{ij}^0 (1+h)}{PЧ_i}, \quad (4.3)$$

де C_{ij}^0 – основна місячна заробітна плата розробника i -ої спеціальності j -го тарифного розряду, грн.;

					БР.КСМ.07224/17.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		50

h – коефіцієнт, що визначає розмір додаткової заробітної плати (при умові наявності доплат);

$РЧ_i$ – місячний фонд робочого часу працівника i -ої спеціальності j -го тарифного розряду, год. (приймаємо 168 год.).

Результати розрахунку записано у таблиці 4.2.

Таблиця 4.2 – Розрахунок витрат на оплату праці

№ п/п	Посада виконавців	Час розробки, год	Погодинна заробітна плата, грн/год.	Витрати на розробку, грн
1	Керівник ДП, викладач	16	44,27	708,32
2	Консультант техніко-економічного розділу, доцент	2	63,81	127,62
3	Студент	122	8,33	1016,66
Разом				1852,6

Величну відрахувань у спеціальні державні фонди визначають у відсотковому співвідношенні від суми основної та додаткової заробітних плат.

Згідно діючого нормативного законодавства сума відрахувань у спеціальні державні фонди складає 20,5 % від суми заробітної плати:

$$B_{\phi} = \frac{20,5}{100} \cdot 1852,6 = 463,15 \text{ грн.}$$

4.1.3 Розрахунок витрат на матеріали та комплектуючі

Загальна сума витрат на матеріальні ресурси (B_M) визначається за формулою:

$$B_M = \sum_{i=1}^n K_i \cdot C_i, \quad (4.4)$$

де K_i – витрата i -го типу матеріалу, натуральні одиниці вимірювання;

C_i – ціна за одиницю i -го типу матеріалу, грн, i -тип матеріального ресурсу;

n – кількість типів матеріальних ресурсів.

Звідси, витрати на матеріальні ресурси дорівнюватимуть:

$$B_M = 488,2 \text{ грн.}$$

У таблиці 4.3 наведений перелік купованих виробів і розраховані витрати на них.

Таблиця 4.3 – Розрахунок витрат на матеріали та комплектуючі

№ п/п	Найменування купованих виробів	Одиниця виміру	Ціна, грн	Кількість купованих виробів	Сума, грн	Транспортні витрати (10% від суми)	Загальна сума, грн
1	Папір (формат А4)	уп	90	1	90	9	99
2	Ручка кулькова	шт	7	2	14	3,4	17,4
3	Олівець простий	шт	4	1	4	0,4	4,4
5	Зошит, 48 арк	шт	14	1	14	1,4	15,4

Продовження таблиці 4.3

6	Тонер для принтера	уп	320	1	320	32	352
Разом							488,2

4.1.4 Витрати на використання комп'ютерної техніки

Для розробки КС використовується електрообладнання, тому необхідно розрахувати витрати на електроенергію.

Витрати на використання комп'ютерної техніки включають витрати на амортизацію комп'ютерної техніки, витрати на користування програмним забезпеченням, витрати на електроенергію, що споживається комп'ютером.

Загальна сума витрат на електроенергію розраховується за формулою:

$$B_E = \sum_{i=1}^n P_i \cdot k_i \cdot T_i \cdot C, \quad (4.5)$$

де P_i – паспортна потужність i -го електрообладнання, кВт;

k_i – коефіцієнт використання потужності i -го електрообладнання;

T_i – час роботи i -го устаткування за весь період розробки, год;

C – ціна електроенергії, грн / кВт год;

i – тип електрообладнання;

n – кількість електрообладнання.

Для розробки проекту даної системи використовується один ноутбук потужністю $P = 0,045$ кВт, який за весь період розробки працює 100 годин та друкуючий пристрій потужністю $P = 0,3$ кВт, який працює 1,5 години.

Розрахунки на витрату електроенергії наведено у таблиці 4.4.

					БР.КСМ.07224/17.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		53

Таблиця 4.4 – Проміжні розрахунки на витрату електроенергії

Найменування устаткування	Паспортна потужність, кВт	Коефіцієнт використання потужності	Час роботи обладнання для розробки, год	Ціна електроенергії,	Сума, грн.
Ноутбук	0,045	0,98	100	0,90	3,96
Принтер	0,3	0,98	1,5	0,90	0,39
Разом					4,35

4.1.5 Накладні витрати

Накладні витрати проектних організацій включають три групи видатків: витрати на управління, загальногосподарські витрати, невиробничі витрати. Вони розраховуються за встановленими відсотками до витрат на оплату праці.

Середньостатистичний відсоток накладних витрат приймемо 150% від заробітної плати:

$$H_B = 1,5 \cdot B_{оп}, \quad (4.6)$$

де H_B – накладні витрати.

Розрахунок накладних витрат для даного проекту:

$$H_B = 1,5 \cdot 1852,6 = 2778,9 \text{ грн.}$$

4.1.6 Обчислення інших витрат

Інші витрати є витратами, які не враховані в попередніх статтях. Вони становлять 10% від заробітної плати:

$$I = 1852,6 \cdot 0,1 = 185,26 \text{ грн.}$$

					БР.КСМ.07224/17.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		54

Витрати на розробку програмного забезпечення складають:

$$K_1 = B_{OP} + B_{\phi} + B_{PB} + H + I . \quad (4.7)$$

$$K_1 = 1852,6 + 463,15 + 448,2 + 2778,9 + 185,26 = 5768,11 \text{ грн.}$$

Витрати на відлагодження і дослідну експлуатацію програмного продукту визначаємо за формулою:

$$K_2 = S_{м.г.} \cdot t_{від} , \quad (4.8)$$

де $S_{м.г.}$ – вартість однієї машино-години роботи ПК, грн./год;

$t_{від}$ – комп'ютерний час, витрачений на відлагодження і дослідну експлуатацію створеного програмного продукту, год.

Загальна кількість днів роботи на комп'ютері дорівнює 30 днів. Середній щоденний час роботи на комп'ютері – 2 години. Вартість години роботи комп'ютера дорівнює 5,2 грн. Тому

$$K_2 = 5,2 \cdot 60 = 312 \text{ грн.}$$

На основі отриманих даних складаємо кошторис витрат на розробку програмного забезпечення.

Загальні витрати (B_{KC}) розрахуємо за формулою:

$$B_{KC} = B_{OP} + B_{\phi} + B_M + B_E + B_{AM} + B_T + H_B . \quad (4.9)$$

					БР.КСМ.07224/17.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		55

Таблиця 4.5 – Кошторис витрат на розробку програмного забезпечення

№ п/п	Найменування витрат	Сума витрат, грн.
1	Витрати на оплату праці	1852,6
2	Відрахування у спеціальні державні фонди	463,15
3	Витрати на куповані вироби	488,2
4	Накладні витрати	2778,9
5	Інші витрати	185,26
6	Витрати на відлагодження і дослідну експлуатацію програмного продукту	312
Разом		6080,11

4.2 Визначення експлуатаційних витрат

Для оцінки економічної ефективності розроблюваного програмного продукту слід порівняти його з аналогом, тобто існуючим програмним забезпеченням ідентичного функціонального призначення.

Експлуатаційні одноразові витрати по програмному забезпеченню і аналогу включають вартість підготовки даних і вартість роботи комп'ютера (за час дії програми):

$$E_{II} = E_{III} + E_{2II} , \quad (4.10)$$

де E_{II} – одноразові експлуатаційні витрати на ПЗ (аналог), грн.;

E_{III} – вартість підготовки даних для експлуатації ПЗ (аналог), грн.;

E_{2II} – вартість роботи комп'ютера для розробки програмного забезпечення (аналог), грн.

					БР.КСМ.07224/17.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		56

Річні експлуатаційні витрати B_{EP} визначаються за формулою:

$$B_{EP} = E_{EP} \cdot N_{EP} , \quad (4.11)$$

де N_{EP} – періодичність експлуатації ПЗ (аналогу), раз/рік.

Вартість підготовки даних для роботи на комп'ютері визначається за формулою:

$$E_{EP} = \sum_{i=1}^n n_i t_i c_i , \quad (4.12)$$

де i – категорії працівників, які приймають участь у підготовці даних ($i=1,2,\dots,n$);

n_i – кількість працівників i -ої категорії, осіб;

t_i – трудомісткість роботи співробітників i -ої категорії по підготовці даних, год.;

c_i – середнього годинна ставка працівника i -ої категорії з врахуванням додаткової заробітної плати, що знаходиться із співвідношення:

$$c_i = \frac{c_i^0 (1 + b)}{m} , \quad (4.13)$$

де c_i^0 – основна місячна заробітна плата працівника i -ої категорії, грн.;

b – коефіцієнт, який враховує додаткову заробітну плату;

m – кількість робочих годин у місяці, год.

Для роботи з даними як для поточного програмного забезпечення так і аналогу потрібен один працівник, основна місячна заробітна плата якого складає: $c^0 = 1400$ грн. Тоді:

$$c_1 = \frac{1400 \cdot (1 + 0)}{168} = 8,33 \text{ грн/год.}$$

					БР.КСМ.07224/17.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		57

Трудомісткість підготовки даних для програмного забезпечення складає 1,5 год., для аналога 2 год.

Таблиця 4.6 – Розрахунок витрат на підготовку даних та реалізацію програмного забезпечення на комп'ютері

№	Час роботи співробітників, год.	Середньогодинна заробітна плата, грн./год.	Витрати, грн.
Проектне рішення			
1	1,5	8,33	12,49
Аналог			
1	2	11,2	22,4

Витрати на експлуатацію комп'ютера визначається за формулою:

$$E_{2П} = t \cdot S_{МГ} , \quad (4.14)$$

де t – витрати машинного часу для реалізації програмного продукту (аналогу), год.;

$S_{МГ}$ – вартість однієї години роботи комп'ютера, грн./год.

$$E_{2П} = 1,5 \cdot 5,2 = 7,8 \text{ грн. } E_{2А} = 2 \cdot 5,2 = 10,4 \text{ грн.}$$

$$E_{П} = 12,49 + 7,8 = 20,29 \text{ грн. } E_{А} = 22,4 + 10,4 = 32,8 \text{ грн.}$$

$$B_{ЕП} = 15,51 \cdot 250 = 5072,5 \text{ грн. } B_{ЕА} = 32,8 \cdot 250 = 8200 \text{ грн.}$$

4.3 Розрахунок ціни споживання програмного продукту

Ціна споживання – це витрати на придбання і експлуатацію програмного продукту за весь строк його служби:

$$Ц_{C(П)} = Ц_{П} + B_{(E)NPV}, \quad (4.15)$$

де $Ц_{П}$ – ціна придбання програмного продукту, грн.:

$$Ц_{П} = K \left(1 + \frac{П_p}{100}\right) + K_0 + K_k, \quad (4.16)$$

де K – кошторисна вартість;

$П_p$ – рентабельність;

K_0 – витрати на прив'язку та освоєння програмного забезпечення на конкретному об'єкті, грн.;

K_k – витрати на доукомплектування технічних засобів на об'єкті, грн.

$$Ц_{П} = 9000 \cdot (1 + 0,3) = 11700 \text{ грн.}$$

$$Ц_{ПА} = 7800 \cdot (1 + 0,3) = 10140 \text{ грн.}$$

Вартість витрат на експлуатацію програмного забезпечення (за весь час його експлуатації), грн.:

$$B_{enpv} = \sum_{t=0}^T \frac{B_{EP}}{(1+R)^t}, \quad (4.17)$$

					БР.КСМ.07224/17.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		59

де B_{EP} – річні експлуатаційні витрати, грн.;

T – термін служби програмного забезпечення, років;

R – річна ставка проценту банку.

$$B_{\text{внр}} = \sum_{t=0}^5 \frac{5072,5}{(1+0,5)^t} = 13881,53 \text{ грн.}$$

$$B_{\text{внр}A} = \sum_{t=0}^5 \frac{8200}{(1+0,5)^t} = 22440 \text{ грн.}$$

Тоді ціна споживання програмного забезпечення дорівнюватиме:

$$Ц_{СП} = 11700 + 13881,53 = 25581,53 \text{ грн.}$$

Аналогічно визначається ціна споживання для аналогу:

$$Ц_{СП} = 10140 + 22440 = 32580 \text{ грн.}$$

4.4 Визначення показників економічної ефективності

Економічний ефект в сфері розробки програмного продукту:

$$E_{PP} = Ц_{П} - Ц_{A} . \quad (4.18)$$

$$E_{PP} = 1170 - 10140 = 1560 \text{ грн.}$$

					БР.КСМ.07224/17.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		60

Річний економічний ефект в сфері експлуатації:

$$E_{KC} = B_{EA} - B_{EP} \cdot \quad (4.19)$$

$$E_{KC} = 8200 - 5072,5 = 3127,5 \text{ грн.}$$

Додатковий економічний ефект у сфері експлуатації:

$$\Delta E_{ekc} = \sum_{t=1}^T E_{ekc} (1 + R)^{T-t} \quad (4.20)$$

$$\Delta B_{BKC} = \sum_{t=1}^5 3127,5 \cdot (1 + 0,5)^{5-t} = 41243,9 \text{ грн.}$$

Сумарний ефект складає:

$$E = E_{IP} + \Delta E_{BKC} = 1560 + 41243,9 = 42803,9 \text{ грн.}$$

Таблиця 4.7 – Показники економічної ефективності програмного забезпечення

№	Найменування	Одиниці вимірювання	Значення показників	
			Базовий варіант	Новий варіант
1	Капітальні вкладення	грн.	-	6080,11
2	Ціна придбання	грн.	10140	11700
3	Річні експлуатаційні витрати	грн.	8200	5072,5
4	Ціна споживання	грн.	32580	25582,53

Продовження таблиці 4.7

6	Економічний ефект в сфері експлуатації	грн.	-	3127,5
7	Додатковий ефект в сфері експлуатації	грн.	-	41243,9
8	Сумарний ефект	грн.	42803,9	

В результаті виконання економічного розділу було проведено розрахунки економічного ефекту розробки програмного продукту, порівняно його з існуючими аналогами, визначено функціональну вартість та рентабельність. На основі функціонально-вартісного аналізу було вибрано найбільш ефективні варіанти реалізації продукту з найбільшим економічним ефектом.

Було встановлено що розроблений програмний продукт має переваги над існуючими аналогами, і крім цього, отриманий економічний ефект у розмірі 42803,9 грн., а отже спроектований програмний продукт та його розробка є економічно вигідними.

ВИСНОВКИ

Отже в результаті виконання бакалаврської роботи можна зробити наступні висновки:

1. Проаналізувавши поставлену задачу було визначено основні перспективи розвитку медичної індустрії в напрямку інформаційних технологій а саме телемедицини і комп'ютерного аналізу мікробіологічних зображень. Розглянуто та проведено порівняння різних методів для вирішення задачі пошуку об'єктів на зображеннях за шаблоном.

2. Розроблено та реалізовано програмний модуль для виявлення об'єктів і класифікації в мікробіологічних зображеннях. Програмний модуль був написаний на мові програмування Java із використанням бібліотеки для комп'ютерного зору «OpenCV». Також було проаналізовано і порівняно інші бібліотеки комп'ютерного зору, та зроблено висновок про перевагу «OpenCV» як вільного проекту з відкритим кодом та широким функціоналом.

3. Розроблено структуру реляційної бази даних для використання в програмному модулі, визначено необхідні сутності та зв'язки, для мінімізації надлишковості даних.

4. Проведено тестування та дослідження розробленого модулю які показують як наукову так і економічну перспективність подальшого розвитку програмного забезпечення для аналізу біомедичних зображень. Визначено напрямки вдосконалення алгоритму для зменшення кількості помилкових спрацьовувань, а також кращої сегментації клітин і ядер.

Результати дипломного проектування доповідалися на інтернет-конференції «Науково-практична конференція інтелектуальні ком'ютерні системи та мережі» (додаток Б).

Розроблений програмний модуль має практичне значення, що підтверджено довідкою про впровадження (додаток В).

					БР.КСМ.07224/17.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		63

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Л.Шапіро, Дж. Стокман. Комп'ютерний зір Computer Vision. М.: Біном. Лабораторія знань, 2006. 752 с.
2. Комп'ютерний зір URL: <https://opencv.org/> (дата звернення: 12.02.19).
3. ImageJ документація URL: <https://imagej.nih.gov/ij/docs/index.html> (дата звернення: 12.02.19).
4. AxioVision технічна інформація та документація URL: <https://www.zeiss.com/microscopy/int/products/microscope-software/axiovision.html> (дата звернення: 12.02.19).
5. Image Pro Plus URL: <http://www.mediacy.com/imageproplus#overview> (дата звернення: 12.02.19).
6. BioImageXD функціонал та технічні вимоги URL: <https://www.bioimagexd.net/index.php/features> (дата звернення: 12.02.19).
7. Micro-Manager технічна інформація та документація URL: <https://micro-manager.org/wiki/Overview%20of%20the%20documentation> (дата звернення: 12.02.19).
8. analySIS FIVE функціонал та технічні вимоги URL: <https://www.olympus-global.com/en/news/2005a/nr050118asfe.html> (дата звернення: 12.02.19).
9. Електронна енциклопедія Вікіпедія: Редагування зображень URL: http://uk.wikipedia.org/wiki/Редагування_зображень. (дата звернення: 12.02.19).
10. Електронна енциклопедія Вікіпедія: Обробка зображень URL: http://uk.wikipedia.org/wiki/Обробка_зображень. (дата звернення: 12.02.19).
11. А.В. Липанов, Системи обробки інформації. Х.: Хупс, 2007. – 52 с.
12. D.G. Lowe, Distinctive Image Features from Scale Invariant Keypoints. D.G. Lowe, International Journal of Computer Vision. 2004. Vol. 60, № 2.110 с.
13. Єрмолаєв В. А. Програмне забезпечення ЕОМ. Структури даних та алгоритми. Запоріжжя: ЗНУ, 2005. 110 с.

					БР.КСМ.07224/17.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		64

14. Template Matching Java URL:
https://docs.opencv.org/3.4/de/da9/tutorial_template_matching.html. (дата звернення 02.03.19)
15. Електронна енциклопедія Вікіпедія: Template matching URL:
<https://en.wikipedia.org/wiki/> (дата звернення: 12.02.19).
16. Електронна енциклопедія Вікіпедія: Сегментація зображення URL:
http://uk.wikipedia.org/wiki/Сегментація_зображення (дата звернення: 12.02.19).
17. Путятин Е.П. Нормализация и распознавание изображений URL:
<http://sumschool.sumdu.edu.ua/is-02/rus/lectures/pytyatin/pytyatin.htm>. (дата звернення: 12.02.19).
18. CBIR: Texture Features URL: <http://www.cs.auckland.ac.nz/compsci708s1c/lectures/Glect-html/topic4c708FSC.htm> (дата звернення: 12.02.19).
19. Бондіна Н. М. Порівняльний аналіз алгоритмів фільтрації медичних зображень. Харків: НТУ “ХПІ”, 2012. № 38. С.14–25
20. Бронников А. В. Комбинированные алгоритмы фильтрации зашумленных сигналов и изображений М.: Автометрия, 1990. 32 с.
21. Аткинсон Л. Mysql. Библиотека профессионала. М Энергоатомиздат, 2002. 496 с.
22. Дейт К. Руководство по реляционной СУБД DB2 М.: Финансы и статистика, 1988. 320 с.
23. Мейер М. Теория реляционных баз данных М.: Мир, 1987. 608 с.
24. Риккарди Грег. Системы баз данных. Теория и практика использования в Internet М.:Вильямс, 2001. 240с.
25. Дейт К. Дж. Введение в системы баз данных СПб: Изд-во "Пітер", 2005 1315с.
26. Девід Форсайт Комп'ютерний зір: Сучасний підхід. М.: "Вильямс", 2004. 928 с.
27. Острей О.Р. Діаграми класів UML як засіб моделювання інформаційної системи моніторингу освіти М.: 2008. № 2. С.85–89.

					БР.КСМ.07224/17.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		65

28. Методичні вказівки до написання техніко-економічного розділу для дипломних проектів на здобуття освітньо-кваліфікаційного рівня «Бакалавр» напряму підготовки 6.050102 «Комп'ютерна інженерія» І.Р. Паздрій. Тернопіль: ТНЕУ, 2015. 36с.

29. Методичні рекомендації до виконання дипломного проекту освітньо-кваліфікаційного рівня «Бакалавр» напряму підготовки 6.050102 «Комп'ютерна інженерія» фахового спрямування «Комп'ютерні системи та мережі» Л.О. Дубчак О.М. Березький, Р.Б Трембач, Г.М. Мельник, Ю.М. Батько, С.В. Івасьєв Під ред. О.М. Березького. Тернопіль: ТНЕУ, 2016. 60с.

30. 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.

					БР.КСМ.07224/17.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		66