

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

**Лящинський Павло Борисович**

**Модуль розпаралелення алгоритмів попереднього  
оброблення зображень на основі графічних  
процесорів / The software module of pre-processing  
image algorithms parallelizing based on GPU**

напрямок підготовки: 6.050102 - Комп'ютерна інженерія  
фахове спрямування - Комп'ютерні системи та мережі  
Бакалаврська робота

Виконав студент групи КСМ 42/1  
Павло Борисович Лящинський

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

Тернопіль - 2018

## РЕЗЮМЕ

Дипломний проект містить 67 сторінок пояснюючої записки, 19 рисунків, 9 таблиць, 11 додатків. Обсяг графічного матеріалу 2 аркуші формату А3.

Метою дипломного проекту є розробка програмного модуля для паралельного попереднього оброблення зображень з використанням графічного процесора.

Здійснено аналіз алгоритмів і методів попереднього оброблення зображень, технологій розпаралелення графічної інформації, що показало актуальність розпаралелення алгоритмів попереднього оброблення зображень.

Розроблено адаптивний алгоритм попереднього оброблення зображень і здійснено його розпаралелення.

Програмно реалізовано модуль попереднього оброблення зображень з використанням графічних процесорів, що дозволило ефективно використати поєднання надійності кросплатформеної мови програмування Java та швидкість технології CUDA.

Програмний модуль реалізовано в межах держбюджетної науково-дослідної роботи «Гібридна інтелектуальна інформаційна технологія діагностування передракових станів молочної залози на основі зображень». Номер держреєстрації 1016U002500

Ключові слова: МІКРООБ'ЄКТИ, ОБРОБЛЕННЯ ЗОБРАЖЕНЬ, ГРАФІЧНИЙ ПРОЦЕСОР, РОЗПАРАЛЕЛЕННЯ, JAVA, CUDA

## RESUME

This thesis contains 67 pages of explanatory note, 19 figures, 9 tables, 11 appendices. Volume of graphic material 2 sheets of A3 format.

The aim of the diploma project is to develop a software module for parallel image pre-processing using a graphics processor.

The analysis of algorithms and methods of image pre-processing, technologies of parallelization of graphic information is carried out that showed urgency of parallelization of algorithms of pre-processing of images.

An adaptive algorithm for image pre-processing has been developed and parallelized.

The software implemented a pre-processing module using graphics processors, which allowed to effectively use the combination of reliability of cross-platform Java programming language and the speed of CUDA technology.

The program module was implemented within the state budget research work "Hybrid intelligent information technology for diagnosing precancerous conditions of the breast based on images." Registration number 1016U002500

Keywords: MICROOBJECTS, IMAGE PROCESSING, GRAPHIC PROCESSOR UNIT, PARALLELING, JAVA, CUDA

## ЗМІСТ

|  |    |
|--|----|
| Перелік умовних скорочень .....  | 12 |
| Вступ.....   | 13 |
| 1 Аналіз алгоритмів попереднього оброблення зображень і технологій розпаралелення інформації ..... | 14 |
| 1.1 Технології і технічні засоби розпаралелення графічної інформації .....                         | 14 |
| 1.2 Аналіз алгоритмів попереднього оброблення зображень .....                                      | 20 |
| 1.3 Аналіз сучасних бібліотек комп'ютерного зору орієнтованих для розробки на GPU .....            | 25 |
| 1.4 Аналіз технічного завдання та постановка завдання дипломного проекту                           | 29 |
| 2 Алгоритми попереднього оброблення зображень .....  | 32 |
| 2.1 Лінійні згладжувальні фільтри .....  | 32 |
| 2.2 Нелінійні фільтри .....  | 35 |
| 2.3 Еквалізація гістограми.....  | 37 |
| 3 Програмна реалізація модуля .....  | 40 |
| 3.1 Системні вимоги .....  | 40 |
| 3.2 Опис програмної реалізації.....  | 41 |
| 3.3 Опис дій користувача при роботі з програмним модулем .....                                     | 42 |
| 3.4 Тестування розробленого модуля .....   | 46 |
| 4 Техніко-економічне обґрунтування розробки програмного засобу .....                               | 51 |
| 4.1 Розрахунок витрат на розробку програмного забезпечення.....                                    | 51 |
| 4.2 Визначення експлуатаційних витрат .....  | 57 |
| 4.3 Розрахунок ціни споживання проектного рішення .....  | 59 |
| 4.4 Визначення показників економічної ефективності.....  | 61 |
| Висновки .....   | 63 |

|   |      |                |        |         |
|---|------|----------------|--------|---------|
| ДП.КСМ.07134/14.00.00.000 ПЗ  |      |                |        |         |
| Змн.  | Арк. | № докум.       | Підпис | Дата    |
|   |      | Лящинський П.Б |        |         |
|   |      | Березький О.М. |        |         |
|   |      | Паздрій І.Р.   |        |         |
|   |      | Гураль І.В.    |        |         |
|   |      | Березький О.М. |        |         |
| МОДУЛЬ РОЗПАРАЛЕЛЕННЯ<br>АЛГОРИТМІВ ПОПЕРЕДНЬОГО<br>ОБРОБЛЕННЯ ЗОБРАЖЕНЬ НА<br>ОСНОВІ ГРАФІЧНИХ<br>ПРОЦЕСОРІВ |      |                |        |         |
|   |      | Літ.           | Арк.   | Акрушів |
|   |      | 10             | 92     |         |
| ТНЕУ. ФКІТ. КСМ-42/1  |      |                |        |         |

|   |    |
|---|----|
| Список використаних джерел .....                            | 64 |
| Додаток А Лістинг класу LikDoctorPageView .....             | 68 |
| Додаток Б Лістинг класу LikDoctorController .....           | 69 |
| Додаток В Лістинг класу LikDoctorModel .....                | 79 |
| Додаток Г Лістинг класу FiltersOperations .....             | 83 |
| Додаток Д Лістинг класу SegmentationOperations .....        | 84 |
| Додаток Е Лістинг класу ImageManagerModule .....            | 85 |
| Додаток Ж Лістинг filters.cpp .....                         | 87 |
| Додаток И Довідка про використання .....                    | 89 |
| Додаток К Нагорода за студентську наукову роботу 2017 ..... | 91 |
| Додаток Л Нагорода за студентську наукову роботу 2018 ..... | 92 |
| Додаток М Нагорода за успіхи в науковій діяльності .....    | 93 |

|      |      |          |        |      |                              |      |
|------|------|----------|--------|------|------------------------------|------|
|      |      |          |        |      | ДП.КСМ.07134/14.00.00.000 ПЗ | Арк. |
|      |      |          |        |      |                              | 11   |
| Змн. | Арк. | № докум. | Підпис | Дата |                              |      |

## ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

CUDA – програмно-апаратна архітектура паралельних обчислень;

GPU – графічний процесор;

API – інтерфейс прикладного програмування;

OpenCV – бібліотека комп'ютерного зору;

OpenVX – специфікація для розробки бібліотек комп'ютерного зору;

DDR – подвійна швидкість передачі даних;

GDDR – графічна подвійна швидкість передачі даних;

JavaFX – платформа для створення застосунків;

MVC – Model View Controller, шаблон проектування;

FTP – протокол передачі файлів;

MSE – середньоквадратичне відхилення між двома зображеннями;

PSNR – пікове відношення сигналу до шуму.

|      |      |          |        |      |                              |      |
|------|------|----------|--------|------|------------------------------|------|
|      |      |          |        |      | ДП.КСМ.07134/14.00.00.000 ПЗ | Арк. |
|      |      |          |        |      |                              | 12   |
| Змн. | Арк. | № докум. | Підпис | Дата |                              |      |

## ВСТУП

Зростаюча складність обчислювальних задач вимагає різкого збільшення ресурсів і швидкодії комп'ютерів. Найбільш перспективним напрямком збільшення швидкості розв'язку прикладних завдань є широке впровадження ідей паралелізму в роботу обчислювальних систем. До теперішнього часу спроектовані і випробувані сотні різних комп'ютерів, що використовують у своїй архітектурі той чи інший вид паралельної обробки даних. Основна складність при проектуванні паралельних програм – це забезпечення правильної послідовності взаємодій між різними обчислювальними процесами, а також координація ресурсів, що розділяються між ними.

В наш час особливої актуальності набули задачі класифікації різного роду інформації. Зокрема, класифікації біомедичних зображень і їх попереднього оброблення, що в свою чергу, є підготовчим етапом перед класифікацією.

Як правило, зображення спотворені шумами апаратури сканування, дискретизації або каналів передачі даних, мають нерівномірні яскравість і контрастність. Це приводить до розривів лінійних об'єктів, руйнування символів, маскуванню складних об'єктів аналізу, і, в подальшому, до неточної класифікації зображень.

У дипломному проекті розглядаються алгоритми попереднього оброблення зображень та методи розпаралелення інформації.

|      |      |          |        |      |                              |      |
|------|------|----------|--------|------|------------------------------|------|
|      |      |          |        |      | ДП.КСМ.07134/14.00.00.000 ПЗ | Арк. |
|      |      |          |        |      |                              | 13   |
| Змн. | Арк. | № докум. | Підпис | Дата |                              |      |

# 1 АНАЛІЗ АЛГОРИТМІВ ПОПЕРЕДНЬОГО ОБРОБЛЕННЯ ЗОБРАЖЕНЬ І ТЕХНОЛОГІЙ РОЗПАРАЛЕЛЕННЯ ІНФОРМАЦІЇ

## 1.1 Технології і технічні засоби розпаралелення графічної інформації

Більшість сучасних мікропроцесорних архітектур використовує різні методи підвищення продуктивності виконуваних програм за рахунок їх розпаралелення, а саме:

- конвеєризація виконання операцій – розбиття процесу виконання на стадії (такти) і одночасне виконання операцій, які перебувають на різних стадіях конвеєра;

- паралельне (одночасне) виконання декількох операцій, які перебувають на одній стадії конвеєрного виконання;

- застосування однієї операції до кількох даних одночасно;

- підтримка багатопоточного виконання всередині одного процесорного ядра, на кількох процесорних ядрах або в багатопроцесорній системі, що працює на загальній пам'яті;

- підтримка неоднорідної багатоядерної або багатопроцесорної системи, в якій ядра можуть працювати одночасно, використовуючи при цьому локальну або загальну пам'ять [5].

Оптимальним прискоренням від розпаралелення могло б бути лінійне – збільшення кількості процесорів вдвічі має вдвічі скорочувати час виконання. Наступне збільшення кількості процесорів вдвічі мало б знову прискорювати програму. Тим не менш, лиш кілька паралельних алгоритмів досягають такого прискорення. Більшість з них мають майже лінійне прискорення для малого числа процесорів, яке сповільнюється до константи при великій кількості обчислювальних елементів.

Потенційне прискорення алгоритму при збільшенні числа процесорів задається законом Амдала, що вперше був сформульований Джином Амдалем у

|      |      |          |        |      |                              |      |
|------|------|----------|--------|------|------------------------------|------|
|      |      |          |        |      | ДП.КСМ.07134/14.00.00.000 ПЗ | Арк. |
|      |      |          |        |      |                              | 14   |
| Змн. | Арк. | № докум. | Підпис | Дата |                              |      |



1960-тих. Він стверджує, що невелика частина програми, яка не піддається паралелізації обмежить загальне прискорення від розпаралелення. Будь яка велика математична чи інженерна задача зазвичай буде складатись з кількох частин, що можуть виконуватись паралельно, та кількох частин, що виконуються тільки послідовно. Цей зв'язок задається рівнянням:

$$S = \frac{1}{1 - P}, \quad (1.1)$$

де  $S$  – прискорення програми (відношення до її початкового часу роботи);

$P$  – частка, яку можна виконувати паралельно.

Якщо послідовна частина програми виконується 10% всього часу роботи, ми не можемо прискоритись більш ніж в 10 разів, незалежно від того скільки процесорів застосуємо. Це ставить верхню межу корисності збільшення кількості обчислювальних елементів [6].

«Коли задача не може розпаралелетись через обмеження послідовної частини, прикладання додаткових зусиль не має ніякого ефекту для розкладу. Щоб виносити дитину потрібно дев'ять місяців, незалежно від того, скільки жінок цим займаються.» [2].

З розгляду даних прикладів впливає важливий висновок: ефективність паралелізації в першу чергу залежить від організації обчислювального алгоритму і співвідношення частки послідовних і паралельних програмних блоків.

Два різних типи архітектури комп'ютерів надають два різні способи організації паралельних обчислень:

- векторизація;
- розпаралелення.

Векторизація є типом розпаралелення за даними, тобто одна паралельна інструкція впливає на різні потоки даних. Розпаралелення в загальному випадку надає більш широкі можливості: поряд з розпаралеленням за даними, можливе

|      |      |          |        |      |                              |      |
|------|------|----------|--------|------|------------------------------|------|
|      |      |          |        |      | ДП.КСМ.07134/14.00.00.000 ПЗ | Арк. |
|      |      |          |        |      |                              | 15   |
| Змн. | Арк. | № докум. | Підпис | Дата |                              |      |

розпаралелення по процесах – різні потоки даних беруть участь в обчислювальному процесі під управлінням різних потоків команд.

Як зазначалося вище даний термін має досить широке тлумачення, ми будемо мати на увазі під ним оптимізацію програмного коду призначеного для експлуатації на системах з масовим паралелізмом. Раніше зазначалося, що розпаралелення можливо як за даними, так і по процесах. Причому в обох випадках може застосовуватися один і той же програмний інструментарій для розпаралелення. Далі ми будемо мати на увазі цю обставину, і вважати, що описувані далі кошти, стандарти, пакети застосовні як для розпаралелення за даними, так і по процесах, в іншому випадку ситуації будуть обговорюватися окремо.

Методи розпаралелення можна умовно розділити на наступні:

- директивами оптимізуючого компілятора;
- спеціальними директивами, які розширюють можливості мови до паралелізації;
- використання паралельних мов програмування;
- комунікаційні засоби, або засоби міжпроцесорного інтерфейсу.

Список можливих засобів не вичерпується перерахованими позиціями, однак, мабуть, вищезгадані методи мають найширше розповсюдження.

Розпаралелення директивами компілятора є найпростішим, "автоматичним" засобом, який може ефективно застосовуватися для вибіркового класу задач, однак, він має наступні недоліки:

- непереносимість з однієї системи на іншу, або навіть при зміні компілятора з одного на інший;
- оскільки обчислювальна ефективність безпосередньо залежить від організації алгоритму, то для хорошого результату все одно потрібна "участь" програміста при написанні паралельного коду;
- досить вузький діапазон можливостей, що надаються компілятором для розпаралелення.

|      |      |          |        |      |                              |      |
|------|------|----------|--------|------|------------------------------|------|
|      |      |          |        |      | ДП.КСМ.07134/14.00.00.000 ПЗ | Арк. |
|      |      |          |        |      |                              | 16   |
| Змн. | Арк. | № докум. | Підпис | Дата |                              |      |

Більш перспективним напрямком в сфері розпаралелення є введення спеціальних директив, що підтримують засоби паралельної обробки, в стандартні мови програмування [4].

Безумовно, найкращою технологією розпаралелення інформації, графічної особливо, є технологія CUDA.

CUDA – програмно-апаратна архітектура паралельних обчислень, яка дозволяє істотно збільшити обчислювальну продуктивність завдяки використанню графічних процесорів (GPUs) фірми Nvidia.

CUDA SDK надає можливість включати в текст програм на C виклик підпрограм, що виконуються на графічних процесорах Nvidia. Це реалізовано шляхом команд, які записуються на особливому діалекті C. Архітектура CUDA дає розробнику можливість на свій розсуд організовувати доступ до набору інструкцій графічного прискорювача й керувати його пам'яттю [1].

Існують, навіть, спеціальні відеокарти сімейства QUADRO та TESLA «заточені» для паралельних обчислень.

У порівнянні з традиційним підходом до організації обчислень загального призначення за допомогою можливостей графічних API, у архітектурі CUDA відзначають такі переваги в цій області:

- Інтерфейс програмування додатків CUDA базується на стандартній мові програмування C з деякими обмеженнями;
- Спільна між потоками пам'ять розміром в 16 Кб може бути використана під організований користувачем кеш з більш широкою смугою пропускання, ніж при вибірці зі звичайних текстур;
- Більш ефективні транзакції між пам'яттю центрального процесора і відеопам'яттю;
- Повна апаратна підтримка цілочисельних і побітових операцій.

Різні відеокарти мають різні обчислювальні можливості. У таблиці 1.1 наведено перелік сучасних відеокарт, що підтримують технологію CUDA.

Таблиця 1.1 – Список підтримуваних відеокарт

| Версія специфікації | GPU  | Відеокарта  |
|---------------------|--|---|
| 1                   | 2  | 3   |
| 1.0                 | G80, G92, G92b, G94, G94b                      | GeForce 8800GTX/Ultra, Tesla C/D/S870, FX4/5600, 360M, GT 420   |
| 1.1                 | G86, G84, G98, G96, G96b, G94, G94b, G92, G92b | GeForce 8400GS/GT, 8600GT/GTS, 8800GT/GTS, 9400GT, 9600 GSO, 9600GT, 9800GTX/GX2, 9800GT, GTS 250, GT 120/30/40, FX 4/570, 3/580, 17/18/3700, 4700x2, 1xxM, 32/370M, 3/5/770M, 16/17/27/28/36/37/3800M, NVS420/50 |
| 1.2                 | GT218, GT216, GT215                            | GeForce 210, GT 220/40, FX380 LP, 1800M, 370/380M, NVS 2/3100M  |
| 1.3                 | GT200, GT200b                                  | GeForce GTX 260, GTX 275, GTX 280, GTX 285, GTX 295, Tesla C/M1060, S1070, Quadro CX, FX 3/4/5800   |
| 2.0                 | GF100, GF110                                   | GeForce (GF100) GTX 465, GTX 470, GTX 480, Tesla C2050, C2070, S/M2050/70, Quadro Plex 7000, Quadro 4000, 5000, 6000, GeForce (GF110) GTX 560 TI 448, GTX570, GTX580, GTX590                                      |
| 2.1                 | GF104, GF114, GF116, GF108, GF106              | GeForce 610M, GT 430, GT 440, GT 640, GTS 450, GTX 460, GTX 550 Ti, GTX 560, GTX 560 Ti, 500M, Quadro 600, 2000   |

|      |      |          |        |      |                              |      |
|------|------|----------|--------|------|------------------------------|------|
|      |      |          |        |      | ДП.КСМ.07134/14.00.00.000 ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата |                              | 18   |

|     |                     |   |
|-----|---------------------|---|
| 3.0 | GK104, GK106, GK107 | GeForce GTX 690, GTX 680, GTX 670, GTX 660 Ti, GTX 660, GTX 650 Ti, GTX 650, GeForce GTX 680MX, 645M, GeForce GT 640M |
|-----|---------------------|---|

Продовження таблиці 1.1

| 1   | 2                   | 3   |
|-----|---------------------|---|
| 3.5 | GK110, GK208        | GeForce GTX TITAN, GeForce GTX TITAN Black, GeForce GTX 780 Ti, GeForce GTX 780   |
| 5.0 | GM107, GM108        | GeForce GTX 750 Ti, GeForce GTX 750, GeForce GTX 860M, GeForce GTX 850M, GeForce 840M, GeForce 830M   |
| 5.2 | GM200, GM204, GM206 | GeForce GTX Titan X, GeForce GTX 980 Ti, GeForce GTX 980, GeForce GTX 970, GeForce GTX 960, GeForce GTX 950, GeForce GTX 970M, GeForce GTX 965M |

Графічні процесори сьогодні використовуються в широкому діапазоні застосувань, головним чином тому, що вони можуть різко прискорити паралельні обчислення, бути доступними та енергоефективними. У галузі медичної візуалізації графічні процесори у деяких випадках є вирішальними для забезпечення практичного використання алгоритмів, що вимагають обчислень.

У [16] описано реалізацію прискореного оброблення зображень на графічному процесорі.

В роботі проведено огляд прискорення на GPU базових операцій оброблення зображень (фільтрація, інтерполяція, оцінка гістограми та дистанційні перетворення), найбільш часто використовуваних алгоритмів медичної візуалізації (реєстрації зображень, сегментації зображень та розпізнавання зображень) та алгоритмів, які є специфічними для окремих

модальностей (СТ, PET, SPECT, MRI, fMRI, DTI, ультразвук, оптична візуалізація та мікроскопія).

У [17] описано вдосконалений метод обробки зображень та класифікації для аналізу даних гіперспектрального зображення для виявлення раку передміхурової залози.

Роботи [26] та [19] описують спектрально-просторову класифікаційну схему для гіперспектральних зображень спеціально адаптовану для обчислення на графічних процесорах.

В роботі [14] описано процес тестування вище згаданої схеми на графічних процесорах.

## 1.2 Аналіз алгоритмів попереднього оброблення зображень

Зображення можна визначити як двовимірну функцію  $f(x, y)$ , де  $x$  і  $y$  – координати в просторі (конкретно на площині) і значення  $f$  якої в будь якій точці, що задається парою координат  $(x, y)$ , називається інтенсивністю або рівнем сірого зображення в цій точці. Якщо величини  $x$ ,  $y$  і  $f$  приймають кінцеве число дискретних значень, то говорять про цифрове зображення.

Цифровим обробленням зображень називається оброблення цифрових зображень за допомогою цифрових обчислювальних машин (комп'ютерів). Треба зауважити, що цифрове зображення складається з кінцевого числа елементів, кожен з яких розташований в конкретному місці і приймає певне значення. Ці елементи називаються елементами зображення або пікселями. Найчастіше для елементів цифрового зображення використовується термін «піксель». [3, с. 22].

У всьому діапазоні від оброблення зображень до машинного зору немає чітких меж, проте можна розрізнити в ньому комп'ютеризовані процеси низького, середнього і високого рівня. Процеси низького рівня стосуються тільки примітивних операцій типу попереднього оброблення з метою зменшення шуму,

|      |      |          |        |      |                              |      |
|------|------|----------|--------|------|------------------------------|------|
|      |      |          |        |      | ДП.КСМ.07134/14.00.00.000 ПЗ | Арк. |
|      |      |          |        |      |                              | 20   |
| Змн. | Арк. | № докум. | Підпис | Дата |                              |      |

підвищення контрасту або поліпшення різкості зображень. Для низькорівневих процесів характерний той факт, що на вході і на виході присутні зображення. Обробка зображень на середньому рівні охоплює такі завдання, як сегментація (розподіл зображення на області або виділення на ньому об'єктів), опис об'єктів і стиснення їх в зручну для комп'ютерної обробки форму, а також класифікація (розпізнавання) окремих об'єктів. Для процесів середнього рівня характерна наявність зображень тільки на вході, на вихід само роблять ознаки і атрибути, які добувають із цих зображень (наприклад кордону областей, лінії контурів, відмінні ознаки конкретних об'єктів). Сюди відноситься сегментація, яка детально описана в [13, 15, 18, 25, 28, 29]. Високорівнева обробка включає в себе «осмислення» набору розпізнаних об'єктів, як це робиться в аналізі зображень, і в межі здійснення пізнавальних функцій, які прийнято пов'язувати із зором.

Таким чином, те, що називається «цифровим обробленням зображень», включає процеси з зображеннями на вході і на виході, а також процеси вилучення ознак із зображень, аж до (або включаючи) розпізнавання індивідуальних об'єктів [3, с. 23-24].

Попереднє оброблення зображень – це набір методів та процедур для покращення якості зображення з метою його подальшого використання. Як правило, зображення спотворені шумами апаратури сканування, дискретизації або каналів передачі даних, мають нерівномірні яскравість і контрастність. Це приводить до розривів лінійних об'єктів, руйнування символів, маскуванню складних об'єктів аналізу. Попереднє оброблення включає такі процедури: вирівнювання загального фону яскравості зображення, усунення на оригінальному документі високочастотних перешкод і різного роду артефактів (засвічення окремих ділянок зображення, провали на ньому, тріщини і т.д.), виконання (у разі потреби), контрастування та інших функціональних перетворень. Опис вихідного інформаційного образу має бути максимально пристосований для зберігання, передачі та аналізу (прийняття рішення на основі цієї інформації). Але при всіх цих перетвореннях одним з найважливіших

|      |      |          |        |      |                              |      |
|------|------|----------|--------|------|------------------------------|------|
|      |      |          |        |      | ДП.КСМ.07134/14.00.00.000 ПЗ | Арк. |
|      |      |          |        |      |                              | 21   |
| Змн. | Арк. | № докум. | Підпис | Дата |                              |      |

показників повинно бути збереження максимального числа особливих елементів вихідної інформації.

У [23] проведено опис аналізу мікроскопічних зображень з допомогою вейвлет-аналізу.

На рисунку 1.1 представлена класифікація алгоритмів попереднього оброблення зображень.



Рисунок 1.1 – Алгоритми попереднього оброблення зображень

Класифікація фільтрів представлена на рисунку 1.2.

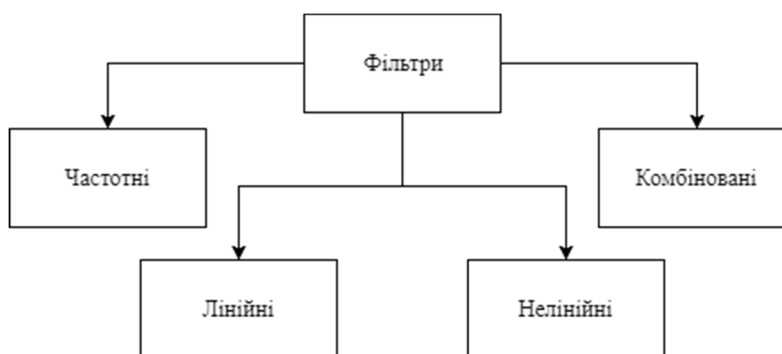


Рисунок 1.2 – Класифікація фільтрів

Широко використовуються в цифровому обробленні зображень методи лінійної фільтрації. Однак, лінійна фільтрація дає хороші результати лише в разі наявності гаусового адитивного шуму. У випадку імпульсного шуму ефективніше працюють методи нелінійної фільтрації, зокрема медіанна та



рангова фільтрація. У випадку комбінованого шуму можна послідовно застосовувати лінійні і нелінійні фільтри або компонувати ці фільтри так, щоб посилити їх сильні сторони і послабити недоліки, це відбувається при використанні гібридних фільтрів. Лінійні фільтри ще називають згладжувальними або усереднюючими тому, що вихід лінійного фільтра усереднює значення пікселів, що містяться в апертурі, і таким чином згладжує зображення [3].

В основі багатьох фільтрів лежить операція згортки. Згортка – це операція обчислення нового значення обраного пікселя, що враховує значення оточуючих його пікселів. Для обчислення значення використовується матриця, що називається ядром згортки або матрицею згортки. Зазвичай ядро згортки є квадратною матрицею  $n * n$ , де  $n$  – непарне число. Під час обчислення нового значення обраного пікселя ядро згортки «прикладається» своїм центром до даного пікселя. Навколишні пікселі так само накриваються ядром. Далі вираховується сума, де складовими є помножені значення пікселів на значення комірки ядра, що накрила даний піксель. Сума ділиться на коефіцієнт нормування (div) або на суму всіх елементів ядра згортки. Отримане значення якраз і є новим значенням обраного пікселя. Якщо застосувати згортку до кожного пікселя зображення, то в результаті вийде певний ефект, що залежить від обраного ядра згортки.

Фільтр Гауса усереднює по закону Гауса пікселі довкола точки і має наступні властивості [11]:

- 1) Згортка з самим собою;
- 2) Розкладання фільтру на 2D та 1D фільтри;
- 3) Згортка по стрічках та стовбцях.

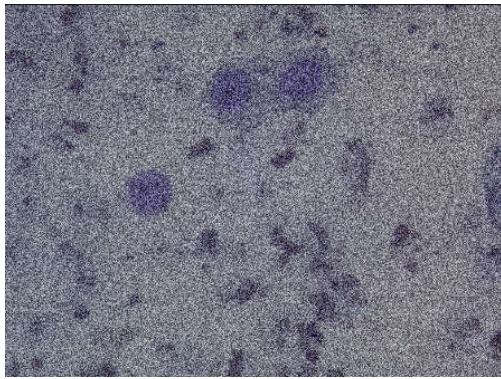
Медіанний фільтр – один з видів цифрових фільтрів, широко використовуваний в цифровій обробці сигналів та зображень для зменшення рівня шуму. Медіанний фільтр є нелінійним FIR-фільтром. Значення відліків усередині вікна фільтра сортуються в порядку зростання (спадання); і значення, що знаходиться в середині упорядкованого списку, надходить на вихід фільтра

|      |      |          |        |      |                              |      |
|------|------|----------|--------|------|------------------------------|------|
| [3]. |      |          |        |      | ДП.КСМ.07134/14.00.00.000 ПЗ | Арк. |
|      |      |          |        |      |                              | 23   |
| Змн. | Арк. | № докум. | Підпис | Дата |                              |      |

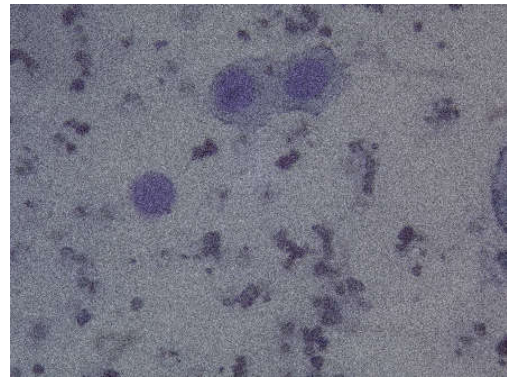
Результати роботи медіанного фільтра зображені на рисунку 1.3 [3].

Морфологічні операції – це нелінійний спосіб оброблення зображень на основі операцій виділення мінімумів та максимумів: erode, dilate.

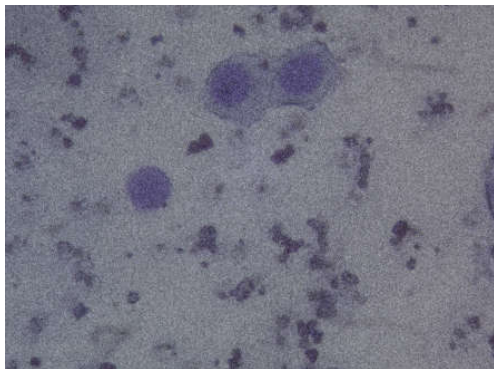
Ці операції застосовуються для видалення шумів на зображенні, виділення окремих елементів і з'єднання розрізнених елементів [7].



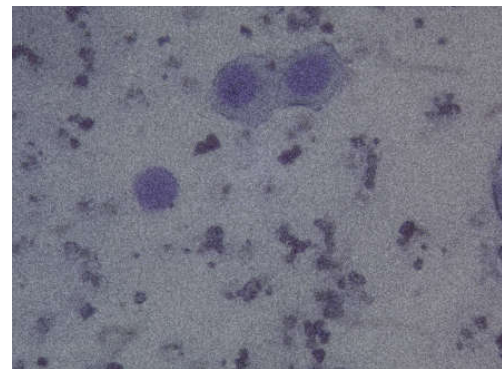
а)



б)



в)



г)

а) зашумлене зображення (комбінований шум); б) медіанний фільтр 5\*5;

в) медіанний фільтр 9\*9; г) медіанний фільтр 21\*21.

Рисунок 1.3 – Результати роботи медіанного фільтра

Широко в попередньому обробленні зображень застосовується зміна яскравості та контрастності зображення, для його кращого сприйняття та підсилення певних деталей.

Яскравість – це світлова характеристика тіл, відношення сили світла, випромінюваного поверхнею, до площі її проекції на площині,

перпендикулярній осі спостереження. Якщо сказати простіше – це кількість білого кольору на зображенні. Чим вище яскравість, тим світліше воно стає.

Контрастність – різниця між відтінками кольору предмета спостереження і навколишнього тла. Якщо сформулювати простіше – це різниця між різними розташованими поруч елементами. Чим вище контрастність, тим різкіше ми спостерігаємо перехід від одного кольору до іншого.

Ще одним дієвим способом попереднього оброблення зображень є робота з гистограмами. Гистограми є основою для чисельних методів просторової обробки. Видозміна гистограм може успішно використовуватися для покращення зображень. Зображення, розподіл значень елементів якого близький до рівномірного і займає весь діапазон можливих значень яскравостей, буде виглядати досить контрастним і міститиме велику кількість напівтонів [10].

Попереднє оброблення зображень, для тестування, буде проводитись на зображеннях приведених на рисунку 1.4.

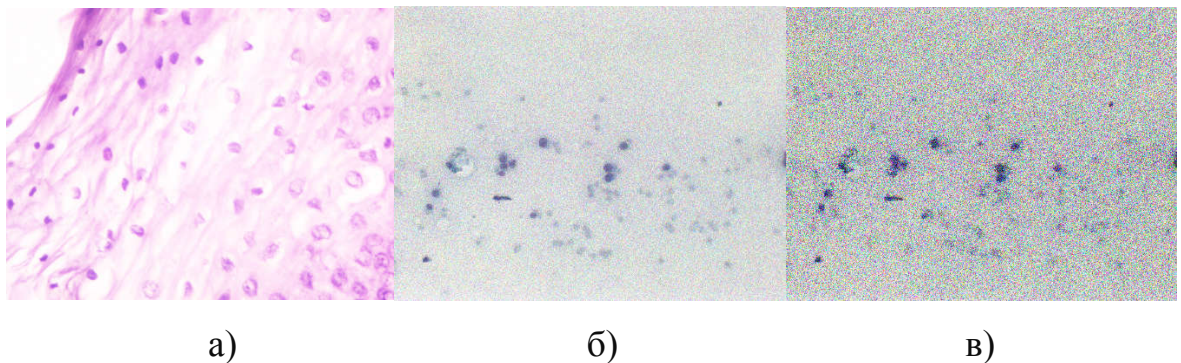


Рисунок 1.4 – Тестові зображення

Зображення а) освітлене, а зображення б) та в) – спотворені шумами. Крім того, деякі області на зображеннях а) та б) штучно затемнені або висвітлені.

### 1.3 Аналіз сучасних бібліотек комп'ютерного зору орієнтованих для розробки на GPU

Комп'ютерний зір – теорія та технологія створення машин, які можуть проводити виявлення, стеження та класифікацію об'єктів.

Як наукова дисципліна, комп'ютерний зір належить до теорії та технології створення штучних систем, які отримують інформацію у вигляді зображень. Відеодані можуть бути представлені у вигляді багатьох форм, таких як відеопослідовність, зображення з різних камер або тривимірними даними з медичного сканера.

Найпопулярнішими бібліотеками комп'ютерного зору для застосування на графічних процесорах є бібліотека OpenCV та специфікація OpenVX.

OpenCV – бібліотека функцій та алгоритмів комп'ютерного зору, обробки зображень і чисельних алгоритмів загального призначення з відкритим кодом. Вона 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 та Itseez. Серцевий 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 [1].

OpenCV містить понад 2500 оптимізованих алгоритмів, серед яких повний набір як класичних так і практичних алгоритмів машинного навчання і комп'ютерного зору.

Алгоритми OpenCV застосовують у таких сферах:

- Аналіз та обробка зображень;
- Системи з розпізнавання обличчя;
- Ідентифікації об'єктів;
- Розпізнавання жестів на відео;
- Відстежування переміщення камери;
- Побудова 3D моделей об'єктів;
- Система взаємодії людини з комп'ютером;

- Пошуку схожих зображень із бази даних;
- Усування ефекту червоних очей при фотозйомці зі спалахом;
- Стеження за рухом очей;
- Ідентифікація об'єктів;
  
- Сегментація зображення;
- Трекінг відео.

OpenVX – специфікація, котра визначає прикладний програмний інтерфейс (API) для розробки переносних, високопродуктивних і енергоефективних застосунків і бібліотек для вирішення задач комп'ютерного зору. Специфікація OpenVX розроблена групою Khronos, що відповідає за розробку стандартів сімейства OpenGL і OpenCL.

OpenVX надає можливість використання серії алгоритмів для розпізнавання і відстеження переміщення особи, тіла і жестів, автоматизації відеоспостереження, автоматичних систем допомоги водієві, реконструкції об'єктів та сцен, доповненої реальності, візуального огляду, робототехніки та багатьох інших застосувань. Обробка даних в OpenVX-застосунках проводиться через маніпуляцію з графом функціональних вузлів, робота з яким може бути прискорена за рахунок залучення CPU, GPU, DSP і додаткових апаратних пристроїв, а також роздільної обробки фрагментів (tiling) [1].

OpenVX являє собою C API двох рівнів: immediate mode і graph mode. Перший – це окремі функції, за структурою дуже схожі на примітиви з OpenCV. У всіх функцій є еквівалент в OpenCV, здебільшого в модулі imgproc. Верхній рівень дозволяє описати алгоритм комп'ютерного зору у вигляді орієнтованого графа, де кожен вузол відповідає функції. Для кожної функції immediate mode існує еквівалентний вузол графа. Граф повністю задається перед виконанням. Користувач може визначити свою функцію через C API callback і включити її в граф. Обидва рівні API працюють з контейнерами для зображень, внутрішня структура яких не специфікована.

|      |      |          |        |      |                              |      |
|------|------|----------|--------|------|------------------------------|------|
|      |      |          |        |      | ДП.КСМ.07134/14.00.00.000 ПЗ | Арк. |
|      |      |          |        |      |                              | 27   |
| Змн. | Арк. | № докум. | Підпис | Дата |                              |      |

Таким чином, дані контейнери є непрозорими (opaque), що дає велику свободу реалізації. Розподіл зображень і вузлів графа по потокам і прискорювачів майже ніяк не керується стандартом, і, відповідно, залишається на розсуд розробника, що реалізує стандарт. В рамках цієї архітектури можна зберігати зображення і виконувати функції OpenVX на прискорювачах, що може сильно прискорити алгоритм або зробити його більш енергоефективним. Взагалі, Graph API відкриває великий простір для оптимізацій.

Наприклад, можна виконувати кілька операцій паралельно на різних ядрах або різних прискорювачах. При певних умовах можна виконувати два послідовних вузла графа одночасно в одному потоці, або обробляти зображення по шматках (tiling), для того, щоб підвищити кеш-ефективність. Підтримка tiling реалізована за допомогою розширення стандарту OpenVX 1.0 Tiling Extension. Надалі, можливо, Tiling API стане обов'язковою частиною наступних версій стандарту [8].

Порівняння OpenCV та OpenVX приведені у таблиці 1.2.

Таблиця 1.2 – Порівняння OpenCV та OpenVX

|                    | OpenCV   | OpenVX   |
|--------------------|--|--|
| Реалізація         | Бібліотека з відкритим вихідним кодом  | Відкритий стандартний API, призначений для реалізації постачальниками обладнання   |
| Сфера застосування | Тисячі функцій оброблення зображень та комп'ютерного зору. Багатофункціональні інтерфейси камер. | Тісна спрямованість на основні апаратні прискорені функції для мобільного зору. Використання зовнішніх / власних камер API |
| Ефективність       | Архітектура на основі пам'яті. Кожна операція читає та записує в пам'ять. Підтримка GPU.         | Graph-based виконання. Оптимізуються обчислення та передача даних. Підтримка GPU.  |

|               |  |   |
|---------------|--|---|
| Послідовність | Доступні функції можуть відрізнятися залежно від реалізації / платформи. | Всі основні функції повинні бути доступними у всіх відповідних реалізаціях. |
|---------------|--|---|

#### 1.4 Аналіз технічного завдання та постановка завдання дипломного проекту

Розроблюваний модуль програмно-апаратної системи, в кінцевому вигляді, має забезпечувати виконання таких завдань:

- завантаження зображення;
- паралельне попереднє оброблення завантаженого зображення;
- отримання вихідного зображення з можливістю його зберігання.

Для вирішення цих завдань існує графічний інтерфейс (Рисунок 1.5), розроблений раніше.

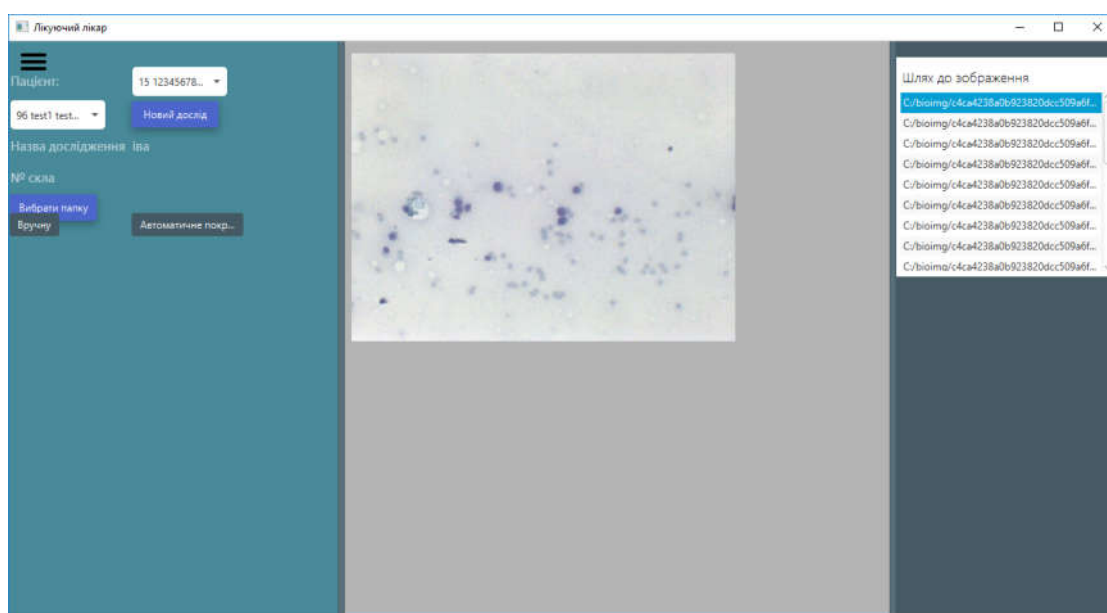


Рисунок 1.5 – Модуль "Цитологія"

Після вибору пацієнта, створення або вибору дослідів, вибору папки з зображеннями, вибору зображення та активації кнопки «Автоматичне покращення», попереднє оброблення зображень повинно відбуватися на графічному процесорі, тобто – паралельно. Коли попереднє оброблення буде завершено, вихідне зображення повинно відобразитись одразу під оригіналом. Таким чином, одразу можна буде порівняти оригінальне та оброблене зображення.

Вихідним параметром розроблюваного програмного модуля є кількісна оцінка якості оброблення зображень.

Для вирішення завдання розпаралелення алгоритмів попереднього оброблення, програмний код буде реалізовуватися на мовах програмування Java/C із використанням бібліотеки комп'ютерного зору OpenCV та технології CUDA.

Для виконання завдання буде використовуватися операційна система Windows 10 та комп'ютер з такими комплектуючими:

Таблиця 1.3 – Комплектуючі комп'ютера

|                    |                                     |
|--------------------|-------------------------------------|
| Процесор           | AMD Athlon 64 x2 Dual Core, 2.51ГГц |
| Оперативна пам'ять | 4Гб, DDR2                           |
| Відеокарта         | GTX 950 Asus Strix, 2Гб, GDDR5      |
| PCI Express        | 2.0                                 |

Для правильного функціонування розроблюваного програмного модуля, на комп'ютері користувача має бути встановлено наступне програмне забезпечення:

- операційна система Windows x64 (версія 7 або 10);
- середовище Java 8;
- середовище CUDA 8.0;
- встановлена бібліотека OpenCV 2.4 з підтримкою графічних процесорів;
- Visual Studio 2013.



Також, комп'ютер повинен відповідати таким вимогам:

- процесор з тактовою частотою 2,5ГГц або більше (наприклад, Intel Pentium G4400 або i3-4170);
- оперативна пам'ять обсягом від 4Гб та поколінням пам'яті DDR4 (наприклад, Kingston DDR4-2400 HyperX Fury);
- відеокарта з підтримкою технології CUDA 8.0, обсягом пам'яті не менше ніж 2Гб та поколінням пам'яті GDDR5 (наприклад, GeForce GTX 950 2048MB);
- жорсткий диск розміром від 80 Гб.

Виходячи з аналізу технічного завдання, метою дипломного проекту є розробка програмного модуля для паралельного попереднього оброблення зображень з використанням графічного процесора. Для досягнення мети треба виконати наступні завдання:

- проаналізувати технології розпаралелення графічної інформації;
- проаналізувати алгоритми попереднього оброблення зображень;
- проаналізувати технологію CUDA та бібліотеку OpenCV;
- досягнути високої кількісної оцінки якості оброблення зображень;
- розробити програмний модуль, що має відповідати вищенаведеним вимогам та виконувати описані функції.

У даному розділі було проаналізовано технології та технічні засоби розпаралелення графічної інформації. Також було проаналізовано алгоритми попереднього оброблення зображень, що дало змогу вибрати оптимальний варіант для подальшої розробки. Було проаналізовано графічні процесори та сучасні бібліотеки комп'ютерного зору, проведено їх порівняння. В результаті якого, для розробки обрано OpenCV. Крім того, був проведений аналіз технічного завдання, в якому було прописано чіткі вимоги розроблюваної програмно-апаратної системи.

|      |      |          |        |      |                              |      |
|------|------|----------|--------|------|------------------------------|------|
|      |      |          |        |      | ДП.КСМ.07134/14.00.00.000 ПЗ | Арк. |
|      |      |          |        |      |                              | 31   |
| Змн. | Арк. | № докум. | Підпис | Дата |                              |      |

## 2 АЛГОРИТМИ ПОПЕРЕДНЬОГО ОБРОБЛЕННЯ ЗОБРАЖЕНЬ

### 2.1 Лінійні згладжувальні фільтри

Згладжувальні фільтри застосовуються для розфокусування зображення та зменшення шуму. Розфокусування може застосовуватися в завданнях попереднього оброблення зображень, наприклад для видалення дрібних (незначних) деталей перед виявленням великих об'єктів або для усунення розривів в лініях чи деталях. Для зменшення шумів може застосовуватися розфокусування з використанням як лінійної, так і нелінійної фільтрації.

Вихідним значенням найпростішого лінійного згладжувального просторового фільтра є середнє значення елементів по околиці, закритої маскою фільтра. Такі фільтри іноді називають об'ємними, згладжувальними або низькочастотними фільтрами.

Ідея застосування згладжувальних фільтрів досить зрозуміла. Заміною вихідних значень елементів зображення на середні значення по масці фільтра досягається зменшення «різких» переходів рівнів яскравості. Оскільки, випадковий шум характеризується різкими стрибками яскравості, найбільш очевидним застосуванням згладжування є зменшення шуму. Але контури, які є цікавими у зображенні, також характеризуються різкими перепадами яскравості, тому негативною стороною застосування таких фільтрів є розфокусування контурів. Іншим застосуванням такої процедури може бути згладжування

|      |      |          |        |      |                              |      |
|------|------|----------|--------|------|------------------------------|------|
|      |      |          |        |      | ДП.КСМ.07134/14.00.00.000 ПЗ | Арк. |
|      |      |          |        |      |                              | 32   |
| Змн. | Арк. | № докум. | Підпис | Дата |                              |      |

помилкових контурів, які виникають при перетвореннях з недостатнім числом рівнів яскравості. Головне використання згладжувальних фільтрів полягає у зменшенні «неіснуючих» деталей на зображенні. Під "неіснуючими" мається на увазі сукупність пікселів, які мають малі розміри в порівнянні з розмірами маски фільтра.

На рис. 2.1 показані два згладжувальних фільтри по околиці  $3 \times 3$ . Перший з них (див. рисунок 2.1 а) дає звичайне середнє значення по масці.

Підстановкою коефіцієнтів маски в рівняння отримаємо результат, приведений у формулі (2.1).

$$R = \frac{1}{9} \sum_{i=1}^9 z_i, \quad (2.1)$$

де  $z$  – дев'ятимірний вектор, сформований з коефіцієнтів маски  $i$ , відповідно, значень яскравості пікселів зображення, що покриваються фільтром.

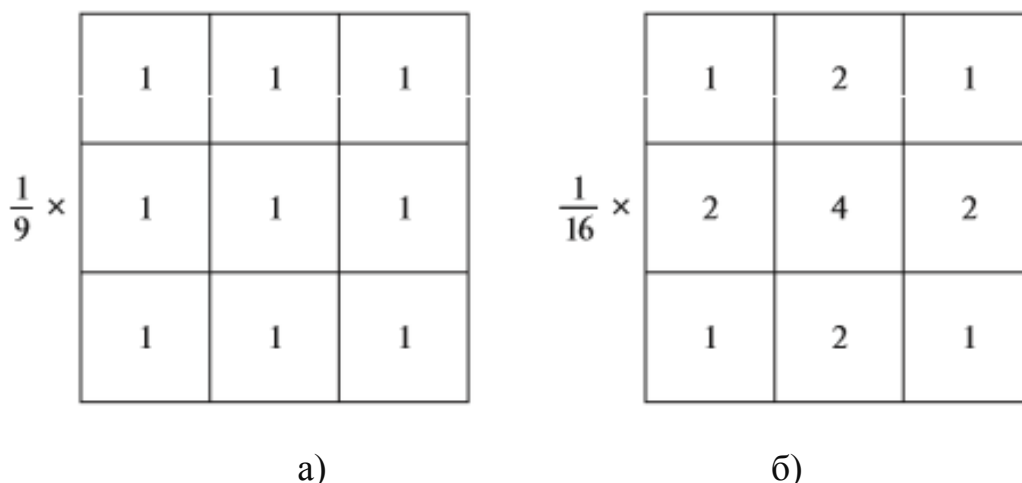


Рисунок 2.1 – Маски згладжувальних фільтрів по околиці  $3 \times 3$

Постійний множник перед кожною з масок дорівнює одиниці, поділений на суму значень коефіцієнтів, так як це необхідно для нормування

Коефіцієнти фільтра показані як 1, замість  $\frac{1}{9}$ . Такий варіант запису є більш ефективним при подальшому комп'ютерному опрацюванні.

По закінченню процесу додавання, отримана величина ділиться на 9. Маска розміром  $m \times n$  матиме нормативний коефіцієнт, рівний  $1 / mn$ . Такий просторовий фільтр, всі коефіцієнти якого однакові, іноді називають однорідним усереднюючим фільтром.

Інша маска на рис. 2.1 (б) більш цікава. Ця маска дає так зване зважене середнє. Цей термін застосовується, щоб показати, що значення елементів додаються з різними коефіцієнтами, що дозволяє присвоїти їм, так би мовити, різні «важливості» (ваги) в порівнянні з іншими.

На рисунку 3.1 (б) коефіцієнт у центрі маски має найбільшу вагу, тим самим даючи відповідному елементу більшу важливість при обчисленні середнього. Значення інших коефіцієнтів у масі зменшуються за ступенем віддалення від центру маски.

Діагональні члени, в порівнянні з ортогональними, розташовані далі від центру і, таким чином, "важать" менше, ніж найближчі сусіди центрального елемента. Основна стратегія присвоєння центральному пікселю найбільшої ваги, а решті — зворотно пропорційно їх відстані, має метою зменшення розфокусування при згладжуванні. Можна було б вибрати і інші значення коефіцієнтів маски для досягнення поставленої цілі, але сума коефіцієнтів, приведених на рис.2.1 (б), рівна 16, що зручно при комп'ютерній реалізації, тому що це степінь двійки. Варто зауважити, що в практиці досить важко помітити різницю між зображеннями, згладженими фільтрами по одній з масок на рис. 2.1 або якимись іншими, аналогічними по конструкції, тому що розміри області, покриті маскою при фільтрації одного елемента, дуже малі.

Загальна формула фільтрації зображення розміром  $M \times N$  фільтром зваженого середнього по околиці  $m \times n$  ( $m$  і  $n$  – непарні) задається рівнянням

|      |      |          |        |      |                              |      |
|------|------|----------|--------|------|------------------------------|------|
|      |      |          |        |      | ДП.КСМ.07134/14.00.00.000 ПЗ | Арк. |
|      |      |          |        |      |                              | 34   |
| Змн. | Арк. | № докум. | Підпис | Дата |                              |      |

$$g(x, y) = \frac{\sum_{s=-a}^a \sum_{t=-b}^b w(s, t) f(x + s, y + t)}{\sum_{s=-a}^a \sum_{t=-b}^b w(s, t)},$$

$$a = (m - 1) / 2,$$

$$b = (n - 1) / 2, \tag{2.2}$$

де  $g(x, y)$  – зображення,

$w(s, t)$  – кореляція фільтра,

$f$  – пікселі.

Повна фільтрація зображення досягається застосуванням формули (2.2) для всіх пар  $x = 0, 1, 2, \dots, M - 1$  і  $y = 0, 1, 2, \dots, N - 1$ . Знаменник в (2.2) – сума всіх коефіцієнтів маски, отже, це константа і вимагає розрахунку тільки один раз [3, с. 196-197].

У [21] наведено вдосконалений обчислювальний підхід, що базується на градієнтному фільтрі зображень та трансформації доменів.

## 2.2 Нелінійні фільтри

Фільтри, засновані на порядкових статистиках, відносяться до класу нелінійних просторових фільтрів.

Відгук такого фільтра визначається попереднім упорядкуванням (ранжуванням) значень пікселів, що покриваються маскою фільтра, та подальшим вибором значення, що знаходиться на певній позиції упорядкованої послідовності (тобто має певний ранг). Фільтрація зводиться до заміни вихідного значення пікселя (в центрі маски) на отримане значення відгуку фільтра.

Найвідомішим є медіанний фільтр, який замінює значення пікселя на значення медіани розподілу яскравостей усіх пікселів в окремій області (включаючи і вихідний). Медіанні фільтри дуже популярні, тому що для певних типів випадкових шумів вони демонструють чудові можливості зменшення

|      |      |          |        |      |                              |      |
|------|------|----------|--------|------|------------------------------|------|
|      |      |          |        |      | ДП.КСМ.07134/14.00.00.000 ПЗ | Арк. |
|      |      |          |        |      |                              | 35   |
| Змн. | Арк. | № докум. | Підпис | Дата |                              |      |

шуму при значно меншому ефекті розфокусування, ніж у лінійних згладжувальних фільтрів з аналогічними розмірами. Медіанні фільтри ефективні при фільтрації імпульсних шумів, які іноді називаються шумами «сіль та перець» і виглядають як накладення на зображення випадкових чорних і білих точок.

Медіана набору чисел – таке число  $\xi$ , що половина чисел з набору менша або рівна  $\xi$ , а друга половина – більша або рівна  $\xi$ . Щоб виконати медіанну фільтрацію для елемента зображення, необхідно спочатку упорядкувати за збільшенням значення пікселів всередині окружості, потім визначити значення медіани і, нарешті, присвоїти отримане значення оброблюваному елементу. Так, для околиці  $3 \times 3$  елементів медіаною буде п'яте значення за величиною, для околиці  $5 \times 5$  – тринадцяте значення і так далі. Якщо декілька елементів в околиці мають однакові значення, ці значення будуть групуватися.

Наприклад, нехай в околиці  $3 \times 3$  елементи мають наступні значення: (10, 20, 20, 20, 15, 20, 20, 25, 100). Після упорядкування вони будуть розташовані наступним чином: (10, 15, 20, 20, 20, 20, 20, 25, 100), а, отже, медіаною буде значення 20. Можна сказати, що основна функція медіанного фільтра полягає в заміні відмінного від фону значення пікселя на інше, ближче до його сусідів.

Насправді, ізольовані темні або світлі (в порівнянні з навколишнім фоном) кластери, що мають площу не більше  $m^2/2$  (половина площі маски фільтра), будуть видалені медіанним фільтром із маскою розміру  $m \times m$ .

У даному випадку «видалені» означає, що значення пікселів у відповідних точках будуть замінені на значення медіан по окружностям. Кластери великих розмірів спотворюються значно менше.

Хоча медіанний фільтр значно більше поширений у обробленні зображень, ніж інші види фільтрів, засновані на порядкових статистиках, тим не менше він не є єдиним. Медіана являє собою квантиль рівня 0,5 порядкового набору чисел, але, як впливає з основ статистики, упорядкування надає багато інших можливостей. Наприклад, використання квантиля рівня 1 призводить до так званого фільтра максимуму, який є корисним при пошуку на зображеннях найяскравіших точок по відношенню до навколишнього фону. Квантиль рівня 0

|      |      |          |        |      |                              |      |
|------|------|----------|--------|------|------------------------------|------|
|      |      |          |        |      | ДП.КСМ.07134/14.00.00.000 ПЗ | Арк. |
|      |      |          |        |      |                              | 36   |
| Змн. | Арк. | № докум. | Підпис | Дата |                              |      |

є фільтром мінімуму, що використовується для пошуку протилежних значень [3, с. 200-201].

### 2.3 Еквалізація гистограми

Гистограмою цифрового зображення з рівнями яскравості в діапазоні  $[0, L - 1]$  називається дискретна функція  $h(r_k) = n_k$ , де  $r_k$  –  $k$ -й рівень яскравості, а  $n_k$  – число пікселів на зображенні, що має яскравість  $r_k$ . Загальною практикою є нормування гистограми шляхом ділення кожного з її значень на загальне число пікселів в зображенні, що позначається добутком  $MN$ , де, як правило,  $M$  і  $N$  число рядків і стовпців зображення, тобто його розміри. Значення нормованої гистограми будуть

$$p(r_k) = n_k / MN, k = 0, 1, \dots, L - 1. \quad (2.3)$$

В загальному  $p(r_k)$  – оцінка ймовірності появи пікселя зі значенням яскравості  $r_k$ . Сума всіх значень нормованої гистограми рівна одиниці.

Розглянемо безперервні значення яскравостей, і нехай змінна  $r$  означає яскравості елементів оброблюваного зображення. При цьому вважається, що  $r$  розподілена в діапазоні  $[0, L - 1]$ , причому значення  $r = 0$  відповідає чорному, а  $r = L - 1$  – білому. Для будь якого  $r$ , яке задовільняє вищевказані умови, розглядається перетворення (відображення яскравості) виду

$$s = T(r) \quad 0 \leq r \leq L - 1 \quad (2.4)$$

яке для кожного пікселя, що має значення  $r$ , дає значення  $s$ . Припускається, що функція перетворення  $T(r)$  задовольняє такі умови: (а)  $T(r)$  є монотонно

|      |      |          |        |      |                              |      |
|------|------|----------|--------|------|------------------------------|------|
|      |      |          |        |      | ДП.КСМ.07134/14.00.00.000 ПЗ | Арк. |
|      |      |          |        |      |                              | 37   |
| Змн. | Арк. | № докум. | Підпис | Дата |                              |      |

неспадною функцією на інтервалі  $0 \leq r \leq L - 1$ ; а також (б)  $0 \leq T(r) \leq L - 1$  при  $0 \leq r \leq L - 1$ .

У деяких формулюваннях використовується зворотне перетворення

$$r = T^{-1}(s) \quad 0 \leq s \leq L - 1, \quad (2.5)$$

у випадку якого умова (а) змінюється на (а')  $T(r)$  є монотонно зростаючою функцією на інтервалі  $0 \leq r \leq L - 1$ .

В умові (а) вимога монотонного зростання функції  $T(r)$  гарантує, що зростання яскравості на вході не може привести до зменшення яскравості на виході, запобігаючи тим самим виникненню артефактів, викликаних інвертуванням зміни яскравості. Умова (б) означає, що допустимий діапазон вихідних значень сигналу збігається з діапазоном вхідних значень. Нарешті, умова (а') гарантує, що зворотне відображення з  $s$  в  $r$  буде взаємно однозначним, запобігаючи невизначеності.

В обробленні зображень особливу важливість має наступна функція:

$$s = T(r) = (L - 1) \int_0^r p_r(w) dw, \quad (2.6)$$

де  $w$  – змінна інтегрування.

Права частина даного рівняння є не що інше, як функція розподілу (ФР) випадкової змінної  $r$ . Оскільки ФР завжди додатна, а інтеграл функції дорівнює площі під графіком функції, отже, функція перетворення в рівнянні (2.6) задовольняє умову (а), так як площа під графіком функції не може зменшуватися при збільшенні  $r$ . Коли досягається верхня межа значень  $r = L - 1$ , інтеграл стає рівним 1, а максимальне значення  $s$  стає рівним  $(L - 1)$ . Значить умова (б) також виконується.

|      |      |          |        |      |                              |      |
|------|------|----------|--------|------|------------------------------|------|
|      |      |          |        |      | ДП.КСМ.07134/14.00.00.000 ПЗ | Арк. |
|      |      |          |        |      |                              | 38   |
| Змн. | Арк. | № докум. | Підпис | Дата |                              |      |



У разі дискретних значень замість щільності розподілу ймовірностей і інтегралів ми маємо справу з вірогідністю (значеннями гістограми) і сумами. Як було згадано вище, ймовірність появи на цифровому зображенні пікселя із значенням яскравості  $r_k$  приблизно дорівнює

$$p_r(r_k) = \frac{n_k}{MN}, k = 0, 1, \dots, L - 1, \quad (2.7)$$

де  $MN$  – загальне число пікселів на зображенні,

$n_k$  – число точок яскравості  $r_k$ ,

$L$  – максимально допустима кількість рівнів яскравості на зображенні (256 для 8-бітового зображення).

Залежність  $p_r(r_k)$  зазвичай називають гістограмою.

Дискретним аналогом функції перетворення, що задається рівнянням (2.6), буде

$$s_k = T(r_k) = (L - 1) \sum_{j=0}^k p_r(r_j) = \frac{(L - 1)}{MN} \sum_{j=0}^k n_j, k = 0, 1, \dots, L - 1. \quad (2.8)$$

Таким чином, оброблене (вихідне) зображення отримується відображенням кожного пікселя вхідного зображення, що має яскравість  $r_k$ , в відповідний елемент вихідного зображення зі значенням  $s_k$ , відповідно до рівняння (2.8). Перетворення (відображення), що задається рівнянням (2.8), називається еквалізацією або лінеаризацією гістограми [3, с. 163-167].

У [30] описано процес гістограмного вирівнювання за допомогою конвеєрного підходу.

У даному розділі було проведено аналіз алгоритмів попереднього оброблення зображень, які, надалі, будуть використані у дипломному проєкті.

|      |      |          |        |      |                              |      |
|------|------|----------|--------|------|------------------------------|------|
|      |      |          |        |      | ДП.КСМ.07134/14.00.00.000 ПЗ | Арк. |
|      |      |          |        |      |                              | 39   |
| Змн. | Арк. | № докум. | Підпис | Дата |                              |      |

### 3 ПРОГРАМНА РЕАЛІЗАЦІЯ МОДУЛЯ

#### 3.1 Системні вимоги

Для правильного функціонування розробленого програмного модуля необхідно дотримуватися таких системних вимог:

- процесор з тактовою частотою 2,5ГГц або більше (наприклад, Intel Pentium G4400 або i3-4170);
- оперативна пам'ять обсягом від 4Гб та поколінням пам'яті DDR4 (наприклад, Kingston DDR4-2400 HyperX Fury);
- відеокарта з підтримкою технології CUDA 8.0, обсягом пам'яті не менше ніж 2Гб та поколінням пам'яті GDDR5 (наприклад, Asus PCI-Ex GeForce GTX 950 Strix 2048MB);
- жорсткий диск розміром від 80 Гб.

Також на комп'ютері користувача має бути встановлене наступне програмне забезпечення:

- операційна система Windows x64 (версія 7 або 10);

|      |      |          |        |      |                              |      |
|------|------|----------|--------|------|------------------------------|------|
|      |      |          |        |      | ДП.КСМ.07134/14.00.00.000 ПЗ | Арк. |
|      |      |          |        |      |                              | 40   |
| Змн. | Арк. | № докум. | Підпис | Дата |                              |      |

- середовище Java 8;
- середовище CUDA 8.0;
- встановлена бібліотека OpenCV 2.4 з підтримкою графічних процесорів;
- Visual Studio 2013.

### 3.2 Опис програмної реалізації

Для написання програмного коду використовувалися мови програмування Java 8 та C, фреймворк JavaFX, середовища програмування IntelliJ Idea та Visual Studio 2013.

Програмний код написаний з використанням архітектурного шаблону Model-View-Controller (MVC), що передбачає розподіл функціоналу на 3 окремі класи: для взаємодії з даними, для взаємодії з графічним інтерфейсом, для керування моделлю та графічним інтерфейсом.

Весь функціонал розробленого програмного модуля розділений на наступні класи:

- LikDoctorPageView (Додаток А) – відповідає за завантаження модуля «Цитологія» та відображення графічного інтерфейсу;
- LikDoctorController (Додаток Б) – відповідає за оброблення всіх подій модуля «Цитологія» (кліки по кнопках, збереження зображень і т.п.);
- LikDoctorModel (Додаток В) – відповідає за логічну структуру модуля «Цитологія» (виконання всіх операцій, підключення до бази даних і т.п.);
- FiltersOperations (Додаток Г) – описує всі фільтри, що будуть застосовані до зображення;
- SegmentationOperations (Додаток Д) – описує операції сегментації, що будуть застосовані до зображення;
- ImageManagerModule (Додаток Е) – відповідає за автоматичне покращення зображень.

Всі вищеописані класи реалізовані на мові програмування Java.

|      |      |          |        |      |                              |      |
|------|------|----------|--------|------|------------------------------|------|
|      |      |          |        |      | ДП.КСМ.07134/14.00.00.000 ПЗ | Арк. |
|      |      |          |        |      |                              | 41   |
| Змн. | Арк. | № докум. | Підпис | Дата |                              |      |

Для паралельного оброблення зображень були розпаралелені наступні алгоритми на мові програмування C (Додаток Ж):

- gaussianBlur – фільтр Гауса;
- bilateralFilter – білатеральний фільтр;
- medianFilter – медіанний фільтр;
- thresholding – порогова обробка.

Скомпільована DLL із вищезгаданими алгоритмами підключається до системи автоматизованої мікроскопії з допомогою технології JNA. Після чого, може виконуватися на графічному процесорі.

### 3.3 Опис дій користувача при роботі з програмним модулем

Розроблений модуль в системі автоматизованої мікроскопії зображень на кресленні ДП.КСМ.07134/14.00.00.000 С1.

Після запуску програми з'являється вікно (Рисунок 3.1), що містить дві кнопки, які розміщені у верхньому лівому кутку.

|      |      |          |        |      |                              |      |
|------|------|----------|--------|------|------------------------------|------|
|      |      |          |        |      | ДП.КСМ.07134/14.00.00.000 ПЗ | Арк. |
|      |      |          |        |      |                              | 42   |
| Змн. | Арк. | № докум. | Підпис | Дата |                              |      |

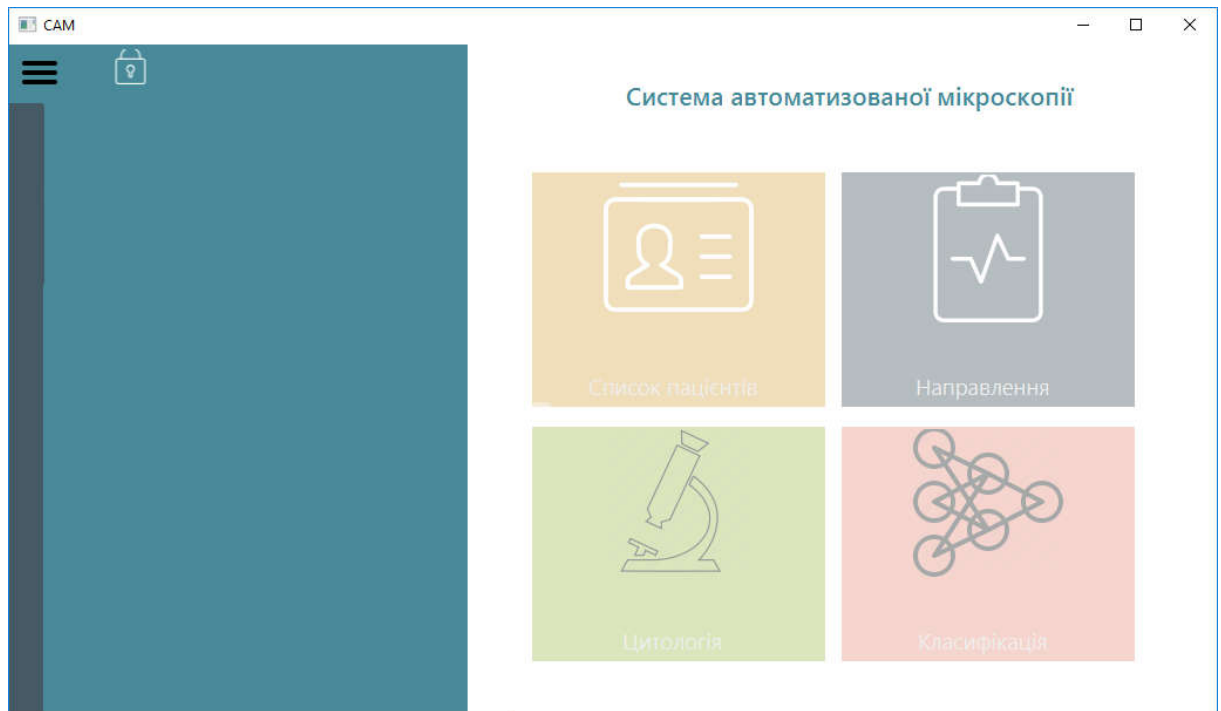


Рисунок 3.1 – Головне вікно програми

Після натиснення на першу з них з'являється спливаюче меню (Рисунок 3.2), з якого можна налаштувати з'єднання з базою даних та сервером FTP.

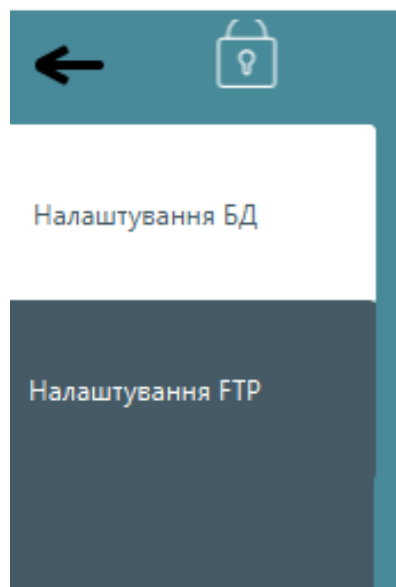


Рисунок 3.2 – Спливаюче меню

Коли всі з'єднання успішно налаштовано – можна авторизуватися в системі. Для цього треба натиснути на кнопку, що має вигляд замка (див.

рисунок 3.2). Після кліку по цій кнопці, з'являється вікно авторизації (Рисунок 3.3), в полях якого треба ввести свої авторизаційні дані.

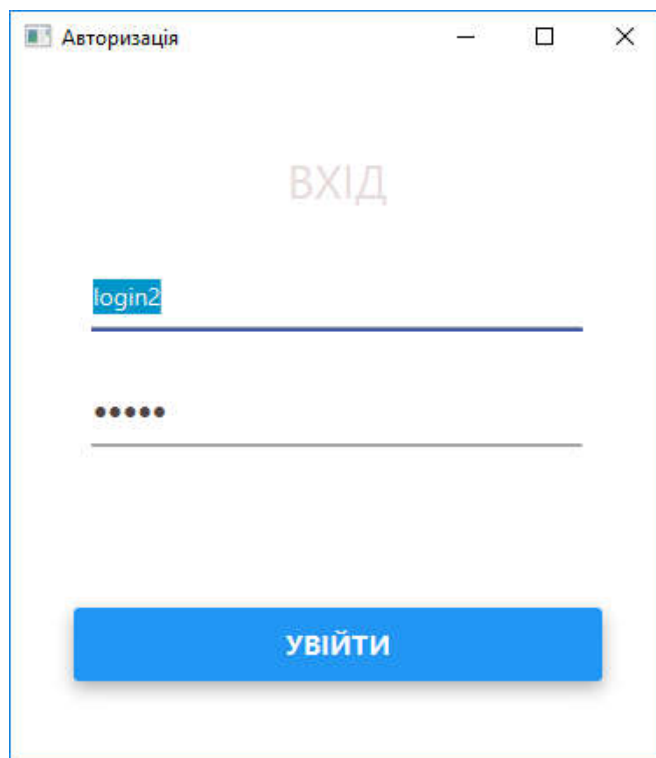


Рисунок 3.3 – Вікно авторизації користувача

Коли авторизацію пройдено, користувач потрапляє у свій робочий інтерфейс (Рисунок 3.4) з всім доступним функціоналом системи, який передбачено для кожного її користувача.

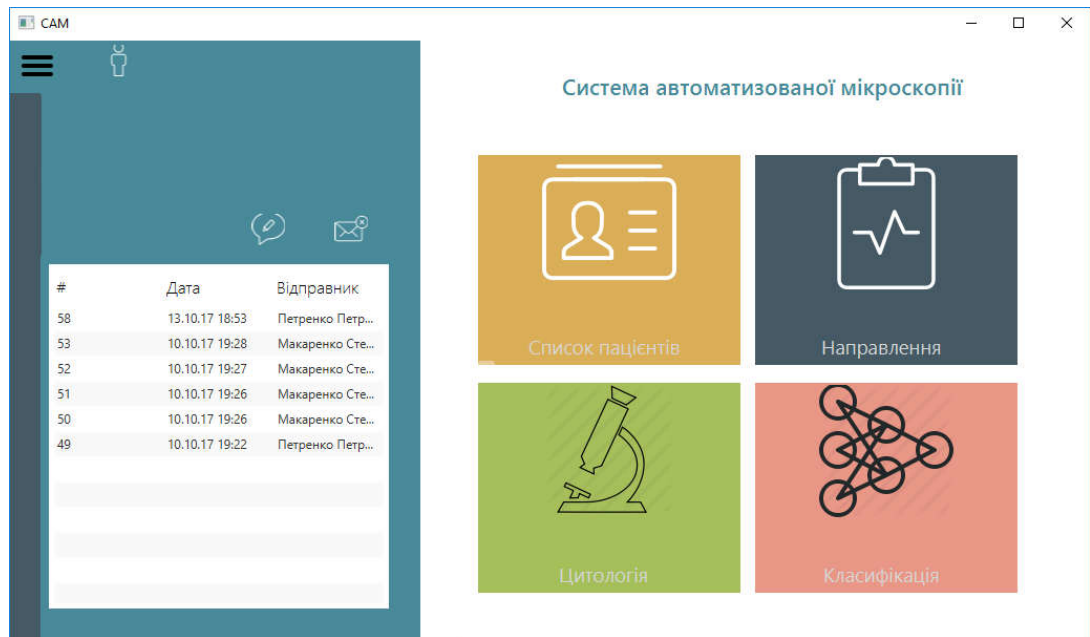


Рисунок 3.4 – Робочий інтерфейс користувача

Щоб потрапити в інтерфейс модуля попереднього оброблення зображень (Рисунок 3.5) – клікаємо на кнопку «Цитологія».

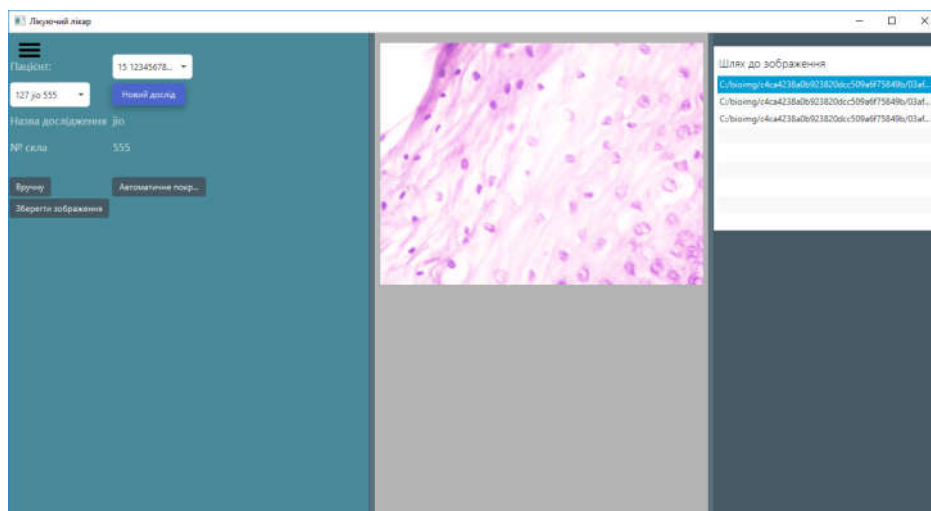


Рисунок 3.5 – Модуль "Цитологія"

Модуль надає можливість вибору пацієнта; створення або вибору вже існуючого дослідження; вибору директорії з зображеннями (тільки при створенні нового дослідження); можливість ручного та автоматичного покращення вибраного

зображення; зберігання обробленого зображення на робочу станцію. Блок-схема алгоритму роботи модуля зображена у ДП.КСМ.07134/14.00.00.001 С1.

Після активації кнопки «Автоматичне покращення», оброблене зображення з'являється одразу під оригіналом (Рисунок 3.6).

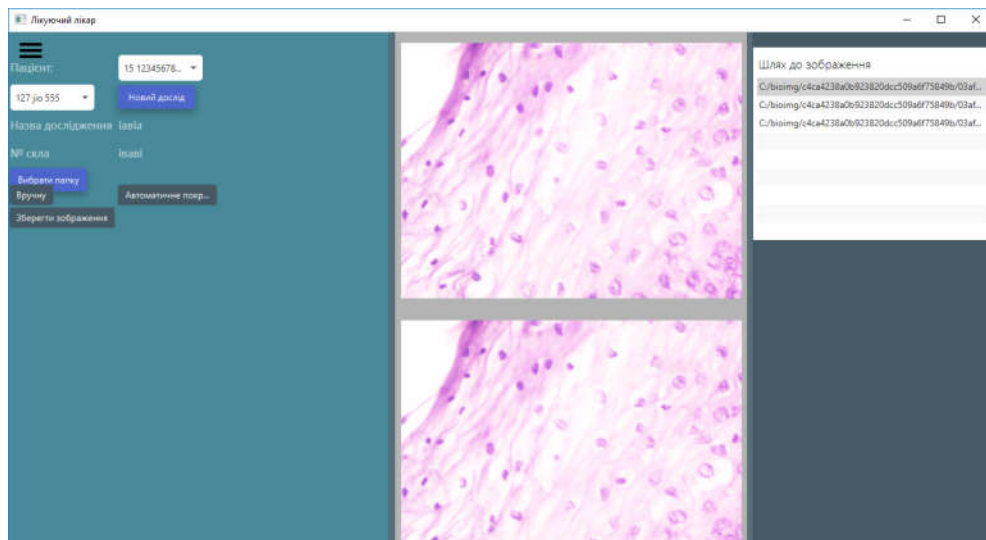


Рисунок 3.6 – Оригінальне та оброблене зображення

Таке розташування зображень має свої переваги. Завдяки такому розташуванню – одразу, візуально, можна порівняти оригінальне та оброблене зображення.

Після автоматичного попереднього оброблення – зображення можна зберегти, натиснувши на кнопку «Зберегти зображення».

### 3.4 Тестування розробленого модуля

Для проведення тестування були підготовлені зображення (див. рисунок 1.4). Зображення готувалися в програмному продукті Adobe Photoshop СС. Зображення а) було освітлено, зображення б) та в) спотворені шумами. Причому, зображення в) зашумоване більше, ніж зображення а).

Деякі області на цих зображеннях штучно освітлені чи затемнені.

|      |      |          |        |      |                              |      |
|------|------|----------|--------|------|------------------------------|------|
|      |      |          |        |      | ДП.КСМ.07134/14.00.00.000 ПЗ | Арк. |
|      |      |          |        |      |                              | 46   |
| Змн. | Арк. | № докум. | Підпис | Дата |                              |      |



Автоматичне попереднє оброблення зображень на низькому рівні відбувається із застосуванням медіанного фільтра з розміром вікна 3\*3.

Подамо на вхід штучно освітлене зображення (див. рисунок 3.6). Після його оброблення, програмно визначимо середньоквадратичне відхилення (MSE) та пікове відношення сигналу до шуму (PSNR) вихідного зображення відносно вхідного (Рисунок 3.7).

```
"C:\Program Files\Java\jdk1.8.0_101\bin\java" ...  
MSE = 0.3268414391900551  
PSNR = 52.98743246695879  
  
Process finished with exit code 0
```

Рисунок 3.7 – MSE та PSNR першого тестового зображення

Як видно з рисунка, PSNR є досить великим.

Тепер подамо на вхід зображення із великим рівнем шуму (Рисунок 3.8).

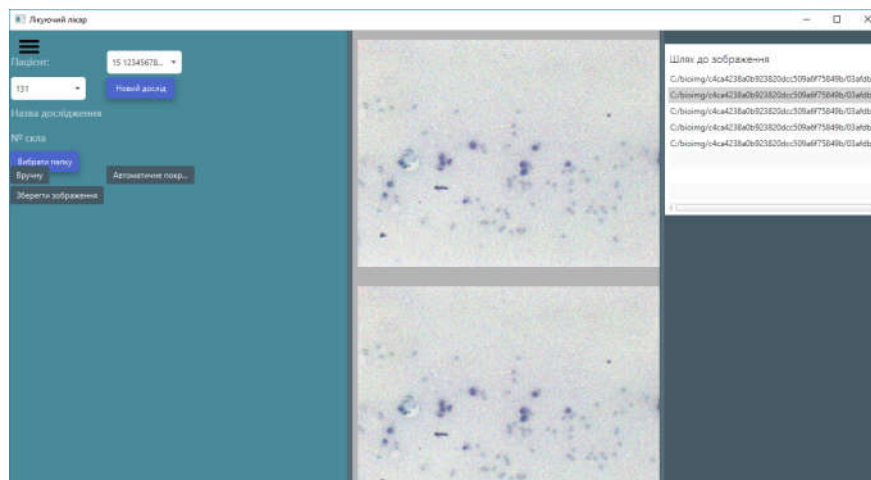


Рисунок 3.8 – Оброблення зображення із великим рівнем шуму

Для цього зображення також визначимо MSE та PSNR (Рисунок 3.9).

```

tools
"C:\Program Files\Java\jdk1.8.0_101\bin\java" ...
MSE = 52.893624764901624
PSNR = 30.89677030917298

Process finished with exit code 0

```

Рисунок 3.9 – MSE та PSNR другого тестового зображення

Подемо на вхід зображення із дуже великим рівнем шуму (Рисунок 3.10).

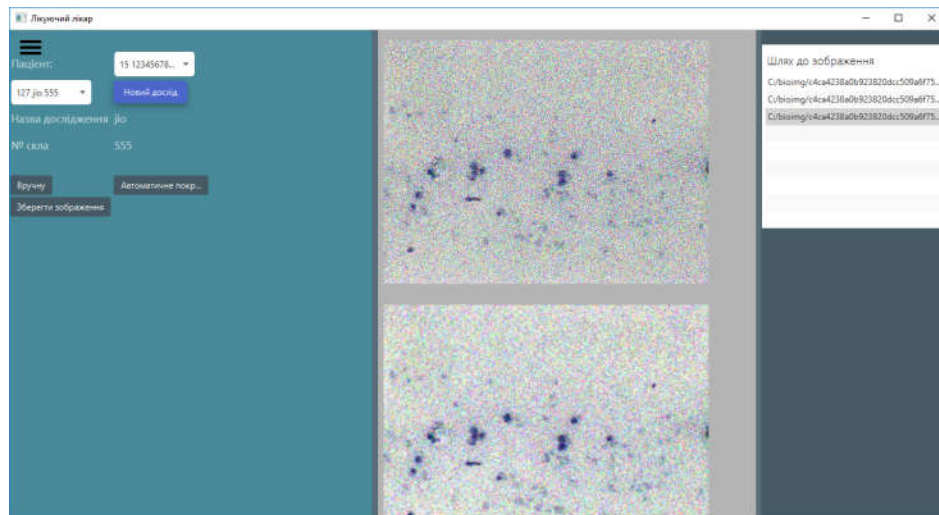


Рисунок 3.10 – Оброблення зображення із дуже великим рівнем шуму

Як і для попередніх зображень, визначимо для третього MSE та PSNR (рисунок 3.11).

```

tools
"C:\Program Files\Java\jdk1.8.0_101\bin\java" ...
MSE = 28.500289916992188
PSNR = 33.58231083009943

Process finished with exit code 0

```

Рисунок 3.11 – MSE та PSNR третього тестового зображення

Як видно з результатів тестування, попереднє оброблення зображень не дає надто хорошого результату для зображень спотворених великими шумами. Хоча, зображення, зашумовані настільки, практично не зустрічаються в реальних випадках.

|      |      |          |        |      |                              |      |
|------|------|----------|--------|------|------------------------------|------|
|      |      |          |        |      | ДП.КСМ.07134/14.00.00.000 ПЗ | Арк. |
|      |      |          |        |      |                              | 48   |
| Змн. | Арк. | № докум. | Підпис | Дата |                              |      |

Проведемо ще один дослід із зображенням зашумованим низьким рівнем шуму – 5% (Рисунок 3.12).

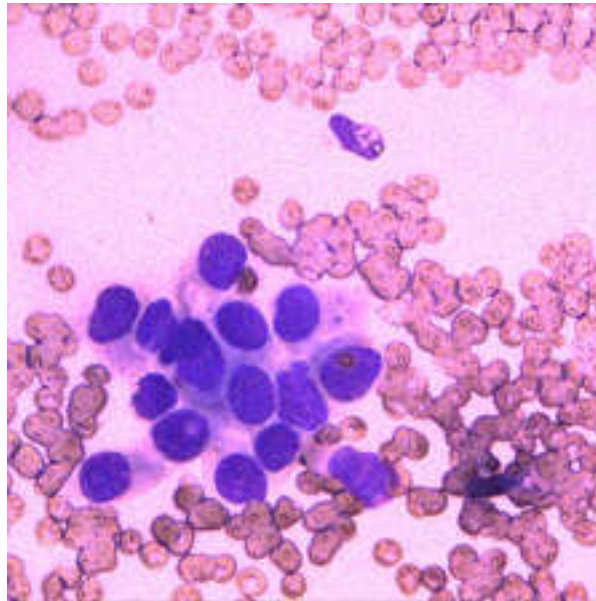


Рисунок 3.12 – Зображення із рівнем шуму 5%

Подамо його на вхід та опрацюємо (Рисунок 3.13).

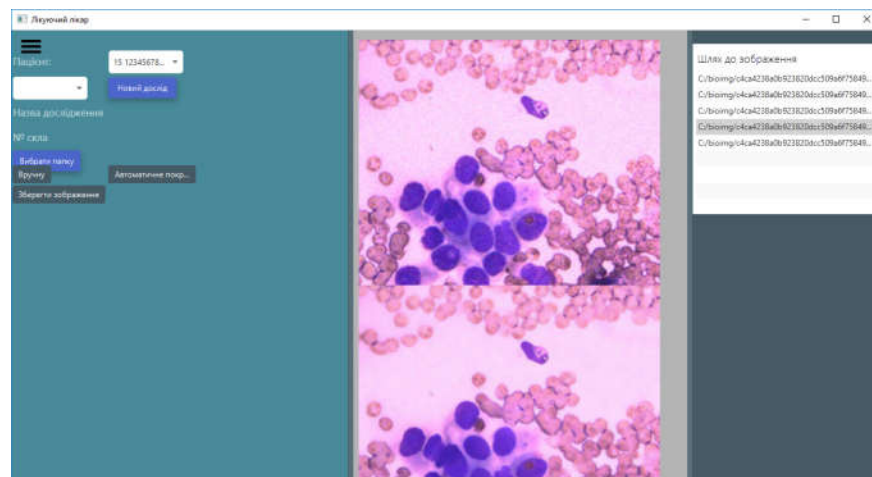


Рисунок 3.13 – Оброблення зображення з рівнем шуму 5%

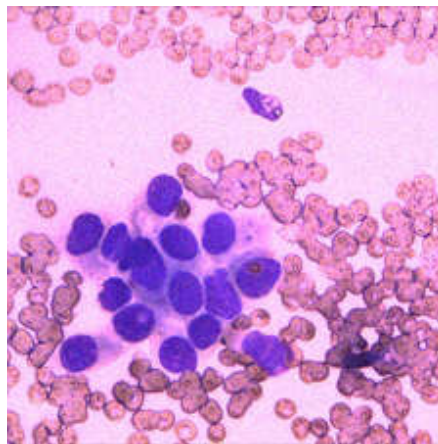
MSE та PSNR для обробленого зображення приведені на рисунку 3.14.

```
013
"C:\Program Files\Java\jdk1.8.0_101\bin\java" ...
MSE = 36.66968271683673
PSNR = 32.48773208609057

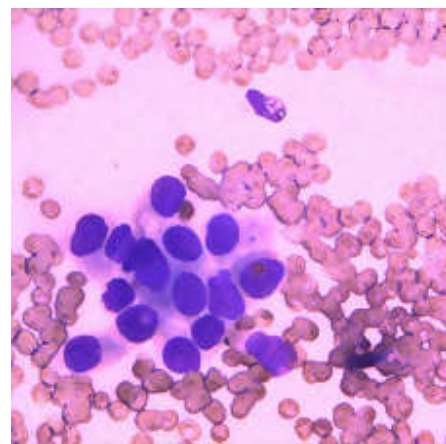
Process finished with exit code 0
```

Рисунок 3.14 – MSE та PSNR четвертого тестового зображення

Як видно з результатів, для зображень із меншим рівнем шуму (Рисунок 3.15 а) попереднє оброблення дає кращий результат (Рисунок 3.15 б).



а)



б)

Рисунок 3.15 – Зображення з низьким рівнем шуму (оригінальне та оброблене)

## 4 ТЕХНІКО-ЕКОНОМІЧНЕ ОБҐРУНТУВАННЯ РОЗРОБКИ ПРОГРАМНОГО ЗАСОБУ

В цьому розділі дипломного проекту проводиться економічне обґрунтування доцільності розробки програмного забезпечення. Зокрема, здійснюється розрахунок витрат на розробку програмного забезпечення, експлуатаційних витрат, ціни споживання проектного рішення. В заключній частині визначаються показники економічної ефективності нового програмного продукту, обґрунтовуються відповідні висновки.

Розроблене програмне забезпечення призначене для візуалізації метричних ультразвукових сигналів і характеризується підвищеною ефективністю виконання алгоритму, що призводить до зменшення часу візуального представлення об'єкту дослідження.

### 4.1 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 ( $B\phi$ );
- витрати на покупні вироби ( $Пв$ );

|      |      |          |        |      |                              |      |
|------|------|----------|--------|------|------------------------------|------|
|      |      |          |        |      | ДП.КСМ.07134/14.00.00.000 ПЗ | Арк. |
|      |      |          |        |      |                              | 51   |
| Змн. | Арк. | № докум. | Підпис | Дата |                              |      |

– витрати на придбання спецобладнання для проведення експериментальних робіт (*Об*);

– накладні витрати (*H*) та інші витрати (*Iв*).

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.

У розробці проектного рішення задіяні наступні спеціалісти - розробники, а саме: керівник проекту; студент-дипломник; консультант техніко-економічного розділу.

Таблиця 4.1 – Вихідні дані для розрахунку витрат на оплату праці

| № п/п | Посада виконавців                                | Місячний оклад, грн. |
|-------|--|----------------------|
| 1     | Керівник ДП, професор                            | 7000                 |
| 2     | Консультант техніко-економічного розділу, доцент | 6026                 |
| 3     | Студент  | 1300                 |

Витрати на оплату праці розробників проекту визначаються за формулою:

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

де  $n_{ij}$  – чисельність розробників  $i$ -ої спеціальності  $j$ -го тарифного розряду;

$t_{ij}$  – 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..

Середньогодинна ставка працівника може бути розрахована за формулою:

|      |      |          |        |      |                              |      |
|------|------|----------|--------|------|------------------------------|------|
|      |      |          |        |      | ДП.КСМ.07134/14.00.00.000 ПЗ | Арк. |
|      |      |          |        |      |                              | 52   |
| Змн. | Арк. | № докум. | Підпис | Дата |                              |      |

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

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

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

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

Результати розрахунку записують до 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, професор                            | 16                | 102,9                               | 1646,4                   |
| 2     | Консультант техніко-економічного розділу, доцент | 2                 | 88,59                               | 177,18                   |
| 3     | Студент  | 140               | 19,11                               | 2675,4                   |
| Разом |  |                   |                                     | 4498,98                  |

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:

$$B_{\phi} = \frac{20,5}{100} \times 4498,98 = 922,29 \text{ грн} \quad (4.4)$$

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             | 95,0      | 1                           | 95,00     | 9,5                                | 104,5              |
| 2     | Ручка кулькова                 | шт             | 5,0       | 1                           | 5,00      | 0,5                                | 5,5                |
| 3     | Олівець простий                | шт             | 3         | 1                           | 3         | 0,3                                | 3,3                |
| 4     | Диски CD-R                     | шт             | 6,0       | 1                           | 6,00      | 0,6                                | 6,6                |
| 5     | Зошит, 96 арк                  | шт             | 10        | 1                           | 10        | 1                                  | 11                 |
| 6     | Тонер для принтера             | уп             | 80        | 1                           | 80        | 8,0                                | 88                 |
| Разом |                                |                |           |                             |           |                                    | 218,9              |

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. За даними обчислювального центру ТНЕУ для комп'ютера типу IBM PC/ATX вартість години роботи становить 5,3 грн. Середній щоденний час роботи на комп'ютері – 2 години. Розрахунок витрат на використання комп'ютерної техніки приведений в таблиці 4.4.



Таблиця 4.4 – Розрахунок витрат на використання комп'ютерної техніки

| № п/п | Назва етапів робіт, при виконанні яких використовується комп'ютер | Час використання комп'ютера, год. | Витрати на використання комп'ютера грн. |
|-------|---|-----------------------------------|---|
| 1     | Проведення досліджень та оформлення їх результатів                | 60                                | 318                                     |
| 2     | Оформлення техніко-економічного розділу                           | 8                                 | 42,4                                    |
| 4     | Оформлення ДП   | 12                                | 63,6                                    |
| Разом |   | 80                                | 424                                     |

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:

$$H = 1,5 \times 4498,98 = 6748,47 \text{ грн} \quad (4.5)$$

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

$$I = 4498,98 \times 0,1 = 449,898 \text{ грн} \quad (4.6)$$

Витрати на розробку програмного забезпечення складають:

$$K_1 = B_{ОП} + B_{\phi} + B_{ПВ} + H + I \quad (4.7)$$

$$K_1 = 4498,98 + 922,29 + 218,9 + 6748,47 + 449,898 = 12838,538 \text{ грн} \quad (4.8)$$

|      |      |          |        |      |                              |      |
|------|------|----------|--------|------|------------------------------|------|
|      |      |          |        |      | ДП.КСМ.07134/14.00.00.000 ПЗ | Арк. |
|      |      |          |        |      |                              | 55   |
| Змн. | Арк. | № докум. | Підпис | Дата |                              |      |

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

$$K_2 = S_{M.Г.} \times t_{від}, \quad (4.9)$$

де  $S_{M.Г.}$  – 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 40 днів. Середній щоденний час роботи на комп'ютері – 2 години. Вартість години роботи комп'ютера дорівнює 5,3 грн.

$$K_2 = 5,3 \times 80 = 424 \text{ грн} \quad (4.10)$$

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

Таблиця 4.6 – Кошторис витрат на розробку програмного забезпечення

| № п/п | Найменування витрат   | Сума витрат, грн. |
|-------|---|-------------------|
| 1     | Витрати на оплату праці   | 4498,98           |
| 2     | Відрахування у спеціальні державні фонди                              | 922,29            |
| 3     | Витрати на куповані вироби  | 218,9             |
| 4     | Накладні витрати  | 6748,47           |
| 5     | Інші витрати  | 449,898           |
| 6     | Витрати на відлагодження і дослідну експлуатацію програмного продукту | 424               |
| Разом |   | 13262,538         |

## 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:

$$E_{\Pi} = E_{1\Pi} + E_{2\Pi}, \quad (4.11)$$

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

$E_{1\Pi}$  – вартість підготовки даних для експлуатації ПЗ (аналог), грн.;

$E_{2\Pi}$  – вартість роботи комп'ютера для виконання проектного рішення (аналог), грн.

Річні експлуатаційні витрати  $B_{\text{еп}}$  визначаються за формулою:

$$B_{\text{еп}} = E_{\Pi} \times N_{\Pi}, \quad (4.12)$$

де  $N_{\Pi}$  – періодичність експлуатації ПЗ (аналог), раз/рік.

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

$$E_{1\Pi} = \sum_{i=1}^n n_i t_i C_i, \quad (4.13)$$

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.;

|      |      |          |        |      |                              |      |
|------|------|----------|--------|------|------------------------------|------|
|      |      |          |        |      | ДП.КСМ.07134/14.00.00.000 ПЗ | Арк. |
|      |      |          |        |      |                              | 57   |
| Змн. | Арк. | № докум. | Підпис | Дата |                              |      |

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

$$C_i = \frac{C_i^0(1+b)}{m}, \quad (4.14)$$

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

$b$  – коефіцієнт, який враховує додаткову заробітну плату (прийємо 0,57);

$m$  – кількість робочих годин у місяці, год.

Для роботи з даними як для проектного рішення так і аналогу потрібен один працівник, основна місячна заробітна плата якого складає:  $C_i^0 = 3700$  грн. Тоді:

$$C_1 = \frac{3700(1+0,57)}{22 \times 8} = 33 \frac{\text{грн}}{\text{год}} \quad (4.15)$$

Трудомісткість підготовки даних для проектного рішення складає 1 год., для аналога – 1,5 год.

Таблиця 4.7 – Розрахунок витрат на підготовку даних та реалізацію проектного рішення на комп'ютері

| №                | Час роботи співробітників, год. | Середньогодинна заробітна плата, грн./год. | Витрати, грн. |
|------------------|---------------------------------|--|---------------|
| Проектне рішення |                                 |  |               |
| 1                | 1                               | 33   | 33            |
| Аналог           |                                 |  |               |
| 1                | 1,5                             | 33   | 49,5          |

Витрати на експлуатацію комп'ютера визначається за формулою:

$$E_{2\Pi} = t \times S_{MG}, \quad (4.16)$$

де  $t$  – витрати машинного часу для реалізації проектного рішення (аналогу), год.;

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

$$E_{2\Pi} = 1 \times 5,3 = 5,3 \text{ грн.};$$

$$E_{2a} = 1,5 \times 4,5 = 7,95 \text{ грн};$$

$$E_{\Pi} = 33 + 5,3 = 38,3 \text{ грн.};$$

$$E_a = 49,5 + 7,95 = 57,45 \text{ грн};$$

$$B_{E\Pi} = 38,3 \times 252 = 9651,6 \text{ грн};$$

$$B_{ea} = 57,45 \times 252 = 14477,4 \text{ грн}.$$

#### 4.3 Розрахунок ціни споживання проектного рішення

Ціна споживання - це витрати на придбання і експлуатацію проектного рішення за весь строк його служби:

$$Ц_{C(\Pi)} = Ц_{\Pi} + B_{(E)NPV}, \quad (4.17)$$

де  $Ц_{\Pi}$  – ціна придбання проектного рішення, грн.:

$$Ц_{\Pi} = K \left(1 + \frac{\Pi_p}{100}\right) + K_0 + K_k, \quad (4.18)$$

де  $K$  – кошторисна вартість;

|      |      |          |        |      |                              |      |
|------|------|----------|--------|------|------------------------------|------|
|      |      |          |        |      | ДП.КСМ.07134/14.00.00.000 ПЗ | Арк. |
|      |      |          |        |      |                              | 59   |
| Змн. | Арк. | № докум. | Підпис | Дата |                              |      |

$P_p$  – рентабельність;

$K_0$  – витрати на прив'язку та освоєння проектного рішення на конкретному об'єкті, грн.;

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

$$C_D = 13262,538 \times (1 + 0,3) + 5000 = 22241,29 \text{ грн} \quad (4.19)$$

Вартість витрат на експлуатацію проектного рішення (за весь час його експлуатації), грн.:

$$B_{ВІРВ} = \sum_{t=0}^T \frac{B_{ЕП}}{(1 + R)^t}, \quad (4.20)$$

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

$T$  – строк служби проектного рішення, років;

$R$  – річна ставка проценту банку.

$$B_{ВІРВ} = \sum_{t=1}^5 \frac{9651,6}{(1 + 0,08)^t} = 38535,8 \text{ грн} \quad (4.21)$$

$$B_{ВІРВ} = \sum_{t=1}^5 \frac{14477,4}{(1 + 0,08)^t} = 57804 \text{ грн} \quad (4.22)$$

Тоді ціна споживання проектного рішення дорівнюватиме:

$$C_{СП} = 22241,29 + 38535,8 = 60777,09 \text{ грн} \quad (4.23)$$

Аналогічно визначається ціна споживання для аналогу:

|      |      |          |        |      |                              |      |
|------|------|----------|--------|------|------------------------------|------|
|      |      |          |        |      | ДП.КСМ.07134/14.00.00.000 ПЗ | Арк. |
|      |      |          |        |      |                              | 60   |
| Змн. | Арк. | № докум. | Підпис | Дата |                              |      |

$$Ц_{СА} = 10400 + 57804 = 68204 \text{ грн} \quad (4.24)$$

#### 4.4 Визначення показників економічної ефективності

Економічний ефект в сфері проектування рішення:

$$E_{ПР} = Ц_{П} - Ц_{А} \quad (4.25)$$

$$E_{ПР} = 22241,29 - 10400 = 11841,29 \text{ грн} \quad (4.26)$$

Річний економічний ефект в сфері експлуатації:

$$E_{КС} = B_{ЕА} - B_{ЕП} \quad (4.27)$$

$$E_{КС} = 14477,4 - 9651,6 = 4825,8 \text{ грн} \quad (4.28)$$

Додатковий економічний ефект у сфері експлуатації:

$$\Delta E_{ВКС} = \sum_{t=1}^T E_{ВКС} \times (1 + R)^{T-t} \quad (4.29)$$

$$\Delta E_{ВКС} = \sum_{t=1}^5 4825,8 \times (1 + 0,08)^{5-t} = 28310,9 \text{ грн} \quad (4.30)$$

|      |      |          |        |      |                              |      |
|------|------|----------|--------|------|------------------------------|------|
|      |      |          |        |      | ДП.КСМ.07134/14.00.00.000 ПЗ | Арк. |
|      |      |          |        |      |                              | 61   |
| Змн. | Арк. | № докум. | Підпис | Дата |                              |      |

Сумарний ефект складає:

$$E = E_{IP} + \Delta E_{BKC} = 11841,29 + 28310,9 = 40152,19 \text{ грн} \quad (4.31)$$

Таблиця 4.8 – Показники економічної ефективності проектного рішення

| № | Найменування                           | Одиниці вимірювання | Значення показників |               |
|---|--|---------------------|---------------------|---------------|
|   |  |                     | Базовий варіант     | Новий варіант |
| 1 | Капітальні вкладення                   | грн.                | -                   | 12838,538     |
| 2 | Ціна придбання                         | грн.                | 10400               | 22241,29      |
| 3 | Річні експлуатаційні витрати           | грн.                | 14477,4             | 9651,6        |
| 4 | Ціна споживання                        | грн.                | 68204               | 60777,09      |
| 5 | Економічний ефект в сфері проектування | грн.                | -                   | 11841,29      |
| 6 | Економічний ефект в сфері експлуатації | грн.                | -                   | 4825,8        |
| 7 | Додатковий ефект в сфері експлуатації  | грн.                | -                   | 28310,9       |
| 8 | Сумарний ефект                         | грн.                | -                   | 40152,19      |

В даному розділі проведено розрахунок витрат на розробку проектного рішення. Здійснено порівняння з існуючим аналогом, і цим показано, що дане проектне рішення має переваги в порівнянні з аналогами, зокрема: є більш надійним і простішим у використанні. Згідно проведеного економічного обґрунтування дане проектне рішення є конкурентноздатним. Крім того, отримано економічний ефект у розмірі 40152,19 грн. і тому розробка і впровадження цього проектного рішення є економічно доцільною.



## ВИСНОВКИ

1. Здійснено аналіз алгоритмів і методів попереднього оброблення зображень, технологій розпаралелення графічної інформації, що показало актуальність розпаралелення алгоритмів попереднього оброблення зображень.

2. Розроблено адаптивний алгоритм попереднього оброблення зображень і здійснено його розпаралелення.

3. Програмно реалізовано модуль попереднього оброблення зображень з використанням графічних процесорів, що дозволило ефективно використати поєднання надійності кросплатформеної мови програмування Java та швидкість технології CUDA.

4. Програмний модуль реалізовано в межах держбюджетної науково-дослідної роботи «Гібридна інтелектуальна інформаційна технологія діагностування передракових станів молочної залози на основі зображень». Номер держреєстрації 1016U002500.

|      |      |          |        |      |                              |      |
|------|------|----------|--------|------|------------------------------|------|
|      |      |          |        |      | ДП.КСМ.07134/14.00.00.000 ПЗ | Арк. |
|      |      |          |        |      |                              | 63   |
| Змн. | Арк. | № докум. | Підпис | Дата |                              |      |

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Бібліотека OpenCV [Електронний ресурс] – Режим доступу до ресурсу: <https://uk.wikipedia.org/wiki/OpenCV>
2. Брукс Ф. Мифический человеко-месяц или как создаются программные системы II издание. – М.: Символ-Плюс , 2016. – 304 с.
3. Гонсалес Р.С., Вудс Р.Е. Мир цифровой обработки: цифровая обработка изображений / Р.С. Гонсалес, Р.Е. Вудс. – М.: Техносфера, 2012. – 1104с.
4. Лабусов А. Н. Технологии распараллеливания [Електронний ресурс] / А. Н. Лабусов – Режим доступу до ресурсу: <http://masters.donntu.org/2009/fvti/turashov/library/book5.htm>
5. Методы распараллеливания программ в оптимизирующем компиляторе [Електронний ресурс] – Режим доступу до ресурсу: <http://www.mcst.ru/metody-rasparallelivaniya-programm-v-optimiziruyushhem-kompilyatore>
6. Паралельні обчислення [Електронний ресурс] – Режим доступу до ресурсу: [https://uk.wikipedia.org/wiki/Паралельні\\_обчислення](https://uk.wikipedia.org/wiki/Паралельні_обчислення)
7. Піцун О.Й. Алгоритми попереднього оброблення біомедичних зображень на базі бібліотеки OpenCV / О.Й. Піцун, А.Р. Боднар //Сучасні комп'ютерні інформаційні технології: Матеріали VI Всеукраїнської школи-семінару молодих вчених і студентів 20-21 травня, 2016 р.: – Тернопіль: ТНЕУ, 2016. - С. 81-83
8. Сенин А. OpenVX: стандарт комп'ютерного зрення [Електронний ресурс] / А. Сенин – Режим доступу до ресурсу: <https://habrahabr.ru/company/intel/blog/204236/>
9. Технологія CUDA [Електронний ресурс] – Режим доступу до ресурсу: <https://uk.wikipedia.org/wiki/CUDA>
10. Фисенко В.Т, Фисенко Т.Ю. Компьютерная обработка и распознавание изображений: учеб. Пособие / В.Т. Фисенко, Т.Ю. Фисенко. - СПб: СПбГУ ИТМО, 2008. – 192 с.

|      |      |          |        |      |
|------|------|----------|--------|------|
|      |      |          |        |      |
| Змн. | Арк. | № докум. | Підпис | Дата |

11. Фільтрація зображень [Електронний ресурс] – Режим доступу до ресурсу: <http://ki.tneu.edu.ua/?c=CV&f=show#filtration>

12. Aswathy M. A. Detection of breast cancer on digital histopathology images: Present status and future possibilities / M. A. Aswathy, M. Jagannath // Informatics in Medicine Unlocked. – 2017. – Vol. 8. – P. 74-79.

13. Chen S. Image segmentation with one shape prior – A template-based formulation / S. Chen, D. Cremers, R. J. Radke // Image and Vision Computing. – 2012. – Vol. 30, No. 12. – P. 1032-1042.

14. Delgado J. Fast Spatial Preprocessing for Spectral Unmixing of Hyperspectral Data on Graphics Processing Units / J. Delgado, G. Martin, J. Plaza, L. Jimenez, A. Plaza // IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing. – 2016. – Vol. 9, No. 8. – P. 952-961.

15. Divya S. Analysis of contour evolution methods for segmentation of medical images / S. Divya, K. B. Jayanthi // 2015 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS) – 2015. – P. 1-4.

16. Eklund A. Medical image processing on the GPU - Past, present and future / A. Eklund, P. Dufort, D. Forsberg, S. M. LaConte // Medical Image Analysis. – 2013. – Vol. 17, No. 8. – P. 1073-1094.

17. Fei B. Hyperspectral imaging and spectral-spatial classification for cancer detection / B. Fei, H. Akbari, L. V. Halig // 2012 5th International Conference on BioMedical Engineering and Informatics – 2012. – P. 62-64.

18. Iglesias J. E. Multi-atlas segmentation of biomedical images: A survey / J. E. Iglesias, M. R. Sabuncu // Medical Image Analysis. – 2015. – Vol. 24, No. 1. – P. 205-219.

19. Jiménez L. GPU Implementation of Spatial–Spectral Preprocessing for Hyperspectral Unmixing / L. Jiménez, G. Martín, S. Sánchez, C. García, S. Bernabé, J. Plaza, A. Plaza // IEEE Geoscience and Remote Sensing Letters. – 2016. – Vol. 13, No. 11. – P. 1671-1675.

20. Katsigiannis S. Grow-Cut Based Automatic cDNA Microarray Image Segmentation / S. Katsigiannis, E. Zacharia, D. Maroulis // IEEE Transactions on

|      |      |          |        |      |                              |      |
|------|------|----------|--------|------|------------------------------|------|
|      |      |          |        |      | ДП.КСМ.07134/14.00.00.000 ПЗ | Арк. |
|      |      |          |        |      |                              | 65   |
| Змн. | Арк. | № докум. | Підпис | Дата |                              |      |

NanoBioscience. – 2015. – Vol. 14, No. 1. – P. 138-145.

21. Kaur P. Contour Detection of Gradient Images Using Morphological Operator and Transform Domain Filtering / P. Kaur, A. Gupta // 2015 IEEE International Conference on Computational Intelligence & Communication Technology – 2015. – P. 107-111.

22. Lapini A. Comparison of super-resolution methods for quality enhancement of digital biomedical images / A. Lapini, F. Argenti, A. Piva, L. Bencini // 2014 8th International Symposium on Medical Information and Communication Technology (ISMICT) – 2014. – P. 1-5.

23. Lyashenko V. Contour Detection and Allocation for Cytological Images Using Wavelet Analysis Methodology / V. Lyashenko, O. Kobylin, R. Matarneh, Y. P. Putyatin // International Journal of Advance Research in Computer Science and Management Studies. – 2016. – Vol. 4, No. 1. – P. 85-94.

24. OpenVX [Електронний ресурс] – Режим доступу до ресурсу: <https://uk.wikipedia.org/wiki/OpenVX>

25. Pratondo A. Integrating machine learning with region-based active contour models in medical image segmentation / A. Pratondo, C.-K. Chui, S.-H. Ong // Journal of Visual Communication and Image Representation. – 2017. – Vol. 43. – P. 1-9.

26. Quesada-Barriuso P. Wavelet-Based Classification of Hyperspectral Images Using Extended Morphological Profiles on Graphics Processing Units / P. Quesada-Barriuso, F. Argüello, D. Heras, J. Benediktsson // IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing. – 2015. – Vol. 8, No. 6. – P. 2962-2970.

27. Riaz F. Content-Adaptive Region-Based Color Texture Descriptors for Medical Images / F. Riaz, A. Hassan, R. Nisar, M. Dinis-Ribeiro, M. Tavares Coimbra // IEEE Journal of Biomedical and Health Informatics. – 2017. – Vol. 21, No. 1. – P. 162-171.

28. Wang X.-Y. A pixel-based color image segmentation using support vector machine and fuzzy C-means / X.-Y. Wang, X.-J. Zhang, X.-Y. Yang, J. Bu // Neural Networks. – 2012. – Vol. 33. – P. 148-159.

|      |      |          |        |      |                              |      |
|------|------|----------|--------|------|------------------------------|------|
|      |      |          |        |      | ДП.КСМ.07134/14.00.00.000 ПЗ | Арк. |
|      |      |          |        |      |                              | 66   |
| Змн. | Арк. | № докум. | Підпис | Дата |                              |      |

29. Yeo S. Y. Segmentation of biomedical images using active contour model with robust image feature and shape prior / S. Y. Yeo, X. Xie, I. Sazanov, P. Nithiarasu // International Journal for Numerical Methods in Biomedical Engineering. – 2014. – Vol. 30, No. 2. – P. 232-248.

30. Yeow Wong C. Histogram equalization and optimal profile compression based approach for colour image enhancement / C. Yeow Wong, G. Jiang, M. Arifur Rahman, S. Liu, S. Ching-Feng Lin, N. Kwok, H. Shi, Y.-H. Yu, T. Wu // Journal of Visual Communication and Image Representation. – 2016. – Vol. 38. – P. 802-813.

|      |      |          |        |      |                              |      |
|------|------|----------|--------|------|------------------------------|------|
|      |      |          |        |      | ДП.КСМ.07134/14.00.00.000 ПЗ | Арк. |
|      |      |          |        |      |                              | 67   |
| Змн. | Арк. | № докум. | Підпис | Дата |                              |      |