

Міністерство освіти і науки, молоді та спорту України
Тернопільський національний економічний університет
Факультет комп'ютерних інформаційних технологій
Кафедра комп'ютерної інженерії

"До захисту допущено"
Завідувач кафедри
комп'ютерної інженерії
к.т.н., доц. О.М.Березький

“ _____ ” _____ 20__ р.

ДИПЛОМНИЙ ПРОЕКТ
освітньо-кваліфікаційного рівня "Спеціаліст"
зі спеціальність 7.05010201 "Комп'ютерні системи та мережі"
на тему:

ГІБРИДНА ІНТЕЛЕКТУАЛЬНА СИСТЕМА АНАЛІЗУ БІОМЕДИЧНИХ ЗОБРАЖЕНЬ

Студент групи КСМзскп-51

_____ Яблонський Д. І.
(підпис)

Керівник:

к.т.н., доцент _____ Березький О.М.
(підпис)

Нормоконтроль

к.т.н., доцент _____ Васильків Н.М.
(підпис)

Консультант

з охорони праці

доцент _____ Сапожник Г.В.
(підпис)

Міністерство освіти і науки, молоді та спорту України
Тернопільський національний економічний університет
Факультет комп'ютерних інформаційних технологій
Кафедра комп'ютерної інженерії
спеціальність 7.05010201 – “Комп'ютерні системи та мережі”

“Затверджую”
завідувач кафедри
комп'ютерної інженерії
к.т.н., доц. О.М.Березький

“ ____ ” _____ 20__ р.

ЗАВДАННЯ НА ДИПЛОМНИЙ ПРОЕКТ СТУДЕНТА Яблонського Дениса Ігоровича

1. **Тема проекту:** "Гібридна інтелектуальна система аналізу біомедичних зображень" затверджена наказом університету № ____ від “ ____ ” _____ 20__ р.
2. **Термін здачі студентом закінченого проекту** “ ____ ” _____ 20__ р.
3. **Вихідні дані для проекту:** Технічне завдання.
4. **Перелік задач, які мають бути вирішені:**
 - провести дослідження функцій і структури інтелектуальних систем аналізу зображень;
 - визначити оптимальну структурну організацію гібридних інтелектуальних систем;
 - провести дослідження класів біомедичних зображень та алгоритмів їх аналізу;
 - розробити алгоритм статистичного аналізу біомедичних зображень;
 - розробити алгоритм побудови нечітких правил;
 - розробити структуру гібридної інтелектуальної системи;
 - виконати програмну реалізацію розроблених алгоритмів;
 - виконати тестування розробленої системи.
5. **Перелік графічного матеріалу** (з точним вказанням обов'язкових креслень)
 - Типи архітектур ГІС;
 - Структура підсистеми виділення досліджуваних параметрів;
 - Схема структурна гібридної інтелектуальної системи аналізу біомедичних зображень;
 - Блок-схема алгоритму обчислення матриць імовірності розподілу рівнів яскравості;
 - Блок-схема алгоритму текстурної сегментації;
 - Блок-схема алгоритму побудови бази правил.

АНОТАЦІЯ

Дипломний проект на тему “Гібридна інтелектуальна система аналізу біомедичних зображень” на здобуття освітньо-кваліфікаційного рівня “Спеціаліст” зі спеціальності 7.05010201 “Комп’ютерні системи та мережі” містить 84 сторінок основного тексту, 19 рисунків, 1 таблицю, 6 додатків, з яких 4 – графічні схеми.

Метою дипломного проекту є розробка та реалізація гібридної інтелектуальної системи аналізу біомедичних зображень. Система дозволяє опрацьовувати біомедичні зображення та будувати базу знань нечітких правил для визначення типу мікрооб’єктів та будувати опис зображення.

Методи дослідження базуються на використанні методів: комп’ютерного зору, штучного інтелекту, статистичного аналізу, побудови нечітких продукційних правил, створення клієнт–серверних застосувань, теорії і технологіях створення Java-застосувань.

У результаті виконання дипломного проекту розроблені алгоритми текстурного аналізу та сегментації біомедичних зображень, алгоритм побудови нечітких правил виводу. Розроблено структуру гібридної інтелектуальної системи опрацювання біомедичних зображень яка поєднує кількісні та якісні ознаки мікрооб’єктів та використовує ці ознаки для виводу висновку про тип мікрооб’єктів. Проведено тестування розробленої системи на базі реальних біомедичних зображень, що підтвердило доцільність поєднання алгоритмів комп’ютерного зору та нечіткого виводу висновку..

Розроблений програмний продукт є ефективним засобом з простим інтерфейсом, що дозволяє вирішувати проблему нечіткого опису біомедичних зображень.

ANNOTATION

Diploma project on a theme “Hybrid intelligent system for biomedical images analysis” on the receipt of graduating level “Specialist” from specialty 7.05010201 “Computer system and network” contains 84 pages of main text, 19 diagrams, 1 table and 6 appendixes, from which 4 are graphic charts.

The purpose of diploma project is development and implementation of hybrid intelligent system of biomedical images analysis. The system allows to process biomedical images and build the knowledge base of fuzzy rules to determine the type of objects and build description.

Methods of work is based on: computer vision, artificial intelligence, statistical analysis, the construction of fuzzy production rules, a customer server applications, theory and technology of creating Java-applications.

As a result of diploma algorithms texture analysis and segmentation of biomedical images, the algorithm of construction of fuzzy rules of inference. The structure of hybrid intelligent system processing of biomedical imaging that combines quantitative and qualitative features of objects and uses those attributes to output a conclusion about the type of microscope. Testing of the developed system based on real biomedical images confirmed the feasibility of a combination of computer vision algorithms and fuzzy logic conclusion .

The developed software is an effective tool with a simple interface that allows to solve the problem of fuzzy description of biomedical images.

ТЕХНІЧНЕ ЗАВДАННЯ

1. НАЙМЕНУВАННЯ ТА ОБЛАСТЬ ЗАСТОСУВАННЯ

1.1 Гібридна інтелектуальна система аналізу біомедичних зображень.

1.2 Область застосування – системи комп'ютерної мікроскопії, медичні лабораторії, діагностичні центри.

2. ОСНОВА ДЛЯ РОЗРОБКИ

Основою для розробки є індивідуальне завдання на дипломний проект, затверджене кафедрою комп'ютерної інженерії факультету комп'ютерних інформаційних технологій Тернопільського національного економічного університету.

3. ПРИЗНАЧЕННЯ РОЗРОБКИ

Метою дипломного проекту є розробка гібридної інтелектуальної системи аналізу біомедичних зображень. Інтелектуальна система дозволяє визначити тип мікрооб'єктів, побудувати опис біомедичного зображення.

4. ДЖЕРЕЛА РОЗРОБКИ

Джерелами даної розробки є матеріали навчальної і реферативної літератури, технічна документація, науково-дослідні роботи, журнали.

5. ТЕХНІЧНІ ВИМОГИ

5.1 Вимоги до апаратних засобів

5.1.1 Функціональні вимоги до апаратних засобів.

5.1.1.1 Система повинна працювати на ІВМ-сумісних робочих станціях.

5.1.1.2 Мінімальні вимоги до робочих станцій: процесор від 2 ГГц, оперативна пам'ять від 1024 Мб, клавіатура, маніпулятор "миша".

5.2. Вимоги до програмної системи

5.2.1 Функціональні вимоги до програмної системи

5.2.1.1 Гібридна інтелектуальна система аналізу біомедичних зображень повинна містити базу нечітких правил, машину логічного виводу висновку, блок інтерфейсу користувача, блок пояснення висновку, блок попереднього оброблення зображення, блок сегментації зображення.

5.2.2 Вимоги до надійності.

5.2.2.1 Передбачити контроль введеної інформації.

5.2.3 Вимоги до програмного забезпечення:

5.2.3.1 Операційна система сімейства Windows;

5.2.3.2 Графічна бібліотека ImageJ.

5.2.3.4 Сумісність з сучасними форматами даних: вхідна інформація подається у форматах «*.bmp» та «*.jpg»;

5.2.4 Вимоги до програмної документації

5.2.4.1 Код програмних модулів повинен містити необхідні для його розуміння коментарі;

5.2.5. Кліматичні вимоги до експлуатації, при яких забезпечується робота програми повинні відповідати кліматичним умовам експлуатації наявних технічних засобів.

6. ВИМОГИ ОХОРОНИ ПРАЦІ

В розділі “Охорона праці” дипломного проекту повинен бути даний аналіз умов праці розробника програмного забезпечення.

7. ПОРЯДОК КОНТРОЛЮ ТА ПРИЙОМКИ

7.1. Представлення дипломного проекту на попередній захист.

7.2. Представлення дипломного проекту на захист.

ЗМІСТ

Перелік умовних скорочень	9
Вступ.....	10
1. Інтелектуальні системи аналізу зображень	13
1.1 Структура інтелектуальної системи.....	13
1.2 Гібридні інтелектуальні системи	19
1.3 Біомедичні зображення.....	26
2. Проектування інтелектуальної системи.....	38
2.1 Мікрооб'єкти та їх характерні ознаки.....	38
2.2 Алгоритми аналізу біомедичних зображень	49
2.3 Алгоритм побудови нечітких правил.....	60
3. Програмна реалізація гібридної інтелектуальної системи.....	65
3.1 Розроблення архітектури гібридної інтелектуальної системи	65
3.2 Програмна реалізація системи	70
3.3 Тестування гібридної інтелектуальної системи	79
4. Охорона праці	85
Висновки	97
Список використаних джерел	98
Додаток А Вихідний текст головного модуля системи.....	101
Додаток Б Схема структурна гібридної інтелектуальної системи аналізу біомедичних зображень	128
Додаток В Блок-схема алгоритму побудови бази правил на основі чисельних даних	129
Додаток Г Блок-схема алгоритму обчислення матриць імовірності розподілу рівнів яскравості	130
Додаток Д Блок-схема алгоритму текстурної сегментації.....	131
Додаток Е Довідка про використання рекомендацій дипломного проекту	132

					ДП.КСМ.19113/11.00.00.000 ПЗ			
Змн.	Лист	№ докум.	Підпис	Дата				
Розробив	Яблонський Д.І.				Гібридна інтелектуальна система аналізу біомедичних зображень	Літ.	Арк.	Акрушів
Перевір.	Березький О.М.						8	131
Консульт.	Сапожник Г.В					ТНЕУ, ФКІТ, КСМзскп-51		
Н. Контр.	Васильків Н.М							
Затвердив	Березький О.М.							

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

ІС	–	Інтелектуальна система
САМ		систем автоматизованої мікроскопії
ААПР	–	Адаптивний алгоритм планування руху
КТ	–	Комп'ютерна томографія
РЕТ	–	Positive Emission Tomography
ЦГЗ	–	Цифрове гістологічне зображення
ЕС	–	Експертна система
БЗ	–	База знань

					ДП.КСМ.19113/11.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		9

ВСТУП

Історія розвитку штучного інтелекту (ШІ) нерозривно пов'язана з історією розвитку обчислювальної техніки. З появою комп'ютерів з'явилася можливість моделювати процеси діяльності людського мозку, зокрема, процес знаходження людиною розв'язку поставленої задачі. Історію розвитку систем штучного інтелекту можна, досить приблизно, відраховувати з кінця 60-х років ХХ століття, коли Л. Заде ввів поняття нечітких множин [1].

За минулий час були створені різні види систем штучного інтелекту (або інтелектуальних систем), такі як експертні системи, нечіткі системи, системи підтримки прийняття рішень, штучні нейронні мережі, системи планування руху роботів, генетичні алгоритми. Інтенсивного розвитку набули гібридні інтелектуальні системи, що дозволяють використовувати переваги традиційних засобів і методів штучного інтелекту, і, у той же час долаючи деякі їх недоліки, здатні вирішувати задачі, нерозв'язані окремими методами штучного інтелекту. Гібридні інтелектуальні системи дозволяють більш ефективно з'єднувати формалізовані і неформалізовані знання за рахунок інтеграції традиційних засобів штучного інтелекту.

В останні роки інтенсивно розвиваються гібридні інтелектуальні системи, що дозволяють використовувати переваги традиційних засобів і методів штучного інтелекту, і, у той же час долаючи деякі їх недоліки, здатні вирішувати задачі, нерозв'язані окремими методами штучного інтелекту. Гібридні інтелектуальні системи дозволяють більш ефективно з'єднувати формалізовані і неформалізовані знання за рахунок інтеграції традиційних засобів штучного інтелекту.

Стрімкий розвиток медичних, біологічних, фармакологічних наук призвів до значного розширення та поглиблення знань про закономірності функціонування людського організму, появу нових методів обстеження і лікування пацієнтів. Тим не менш, практична медицина все ще залишається важко формалізованою областю людської діяльності, в якій фахівці при

					ДП.КСМ.19113/11.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		10

прийнятті рішень часто виходять з попереднього професійного досвіду та власної інтуїції, а не з аналізу об'єктивних даних. У такій ситуації важко уникнути лікарських помилок, соціальне і економічне значення яких виявляється надзвичайно високим. У результаті виникає необхідність збільшення числа врахованих і аналізованих лікарем взаємозалежних показників діяльності організму, що ще більше ускладнює завдання вибору адекватних лікувальних впливів, також взаємодіючих між собою.

Одним з основних етапів постановки діагнозу при онкологічних дослідженнях є аналіз мікроскопічних зображень. Для аналізу використовуються цифрові зображення цитологічних та гістологічних препаратів (зразків), отримані з допомогою систем автоматизованої мікроскопії. Задача розпізнавання зображень передбачає інтелектуальну обробку отриманої інформації, що представляє певні складнощі. Поза сумнівом, в галузі аналізу біомедичних зображень досягнуті величезні успіхи. Але, універсальних методів обробки зображення, порівнянних по продуктивності і якості розпізнавання з людським зором, немає. Наприклад, в задачах, які ставляться перед експертними системами, потрібен глибший інтелектуальний аналіз і висока швидкодія. Тому обробка зображення в задачі розпізнавання є однією з центральних проблем.

Виділяють наступні питання при розв'язанні задачі розпізнавання зображень: сприйняття поля зору, сегментація, нормалізація виділених об'єктів, розпізнавання. При побудові систем аналізу використовуються наступні основні принципи:

– Принцип цілісності – розпізнаваний об'єкт розглядається як єдине ціле, таке, що складається із структурних частин, зв'язаних між собою просторовими відношеннями.

– Принцип двонаправленості - створення моделі ведеться від зображення до моделі і від моделі до зображення.

– Принцип передбачення. Полягає у формуванні гіпотези про вміст зображення.

– Принцип цілеспрямованості, що включає сегментацію зображення і

					ДП.КСМ.19113/11.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		11

спільну інтерпретацію його частин.

Принцип максимального використання моделі проблемного середовища, використання заздалегідь відомих, апріорних параметрів

В області класифікації типу клітин задачу постановки діагнозу, а саме висновку про наявність злякисного новоутворення, вирішено для окремих видів пухлин. Інтерпретація у виконанні спеціаліста патолога повинна привести до збору морфологічних доказів (числових характеристик об'єктів), що дозволяють ідентифікувати патологічні процеси, які можуть виявитися доброякісними або злякисними. Цей крок передбачає досягнення трьох різних завдань: морфологічного аналізу, ідентифікації клітин постановки та інтерпретації діагнозу. В якості умов нечітких правил виступають ознаки окремих мікрооб'єктів (клітин).

Проблемам підтримки прийняття рішень задач діагностики традиційно приділяється багато уваги. Це пов'язано з постійним підвищенням складності об'єктів діагностики та зростаючою роллю вирішення задач діагностики в сучасному суспільстві. В даний час ця задача стала особливо актуальною через стрибкоподібний розвиток потужності обчислювальних машин, за допомогою яких стало можливим переробка великих масивів інформації та перетворення їх в бази знань для інтелектуальних розв'язуючих систем, що дозволяють істотно підвищити якість діагностичних рішень. Значний внесок у дослідження і рішення проблем побудови інтелектуальних систем внесли вітчизняні вчені Д. А. Поспелов, О.І. Е.В. Попов, Г.С. Осипов, В.М. Вагін, В.Б. Л.М. Аверкін та інші, а також ряд зарубіжних вчених Ф. Форсайт, К. Нейлор та ін.

Наведені факти роблять актуальним розроблення інформаційно-аналітичних систем, що поєднували б декілька технологій штучного інтелекту.

					ДП.КСМ.19113/11.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		12

1. ІНТЕЛЕКТУАЛЬНІ СИСТЕМИ АНАЛІЗУ ЗОБРАЖЕНЬ

1.1 Структура інтелектуальної системи

З розвитком комп'ютерних технологій змінювався зміст, вкладений у поняття інформаційної системи. Сучасна інформаційна система — це набір інформаційних технологій, спрямованих на підтримку життєвого циклу інформації, що й включає три основні процеси: обробку даних, керування інформацією й керування знаннями. В умовах різкого збільшення обсягів інформації перехід до роботи зі знаннями на основі штучного інтелекту є єдиною альтернативою інформаційного суспільства [2]

Згідно з визначенням Д.А. Поспелова [3], "Система називається інтелектуальною, якщо в ній реалізовані наступні основні функції:

- накопичувати знання про світ навколо системи, класифікувати й оцінювати їх з погляду прагматичної корисності й несуперечності, ініціювати процеси одержання нових знань, здійснювати співвіднесення нових знань із раніше збереженими;
- поповнювати отримані знання за допомогою логічного виводу, що відбиває закономірності в світі навколо системи накопичені нею раніше, одержувати узагальнені знання на основі більш часткових знань і логічно планувати свою діяльність;
- спілкуватися з людиною мовою, максимально наближеною до природної людської мови;
- одержувати інформацію з каналів, аналогічних тим, які використовують люди при сприйнятті навколишнього світу;
- уміти формувати для себе або на прохання людини (користувача) пояснення власної діяльності;
- надавати користувачеві допомогу за рахунок тих знань, які зберігаються в пам'яті, і тих логічних засобів міркувань, які властиві системі".

					ДП.КСМ.19113/11.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		13

Перераховані функції можна назвати функціями представлення й обробки знань, міркування й спілкування. Поряд з обов'язковими компонентами, залежно від розв'язуваних задач і області застосування в конкретній системі ці функції можуть бути реалізовані в різному ступені, що визначає індивідуальність архітектури. На рисунку 1.1 у найбільш загальному виді представлена структура інтелектуальної системи у вигляді сукупності блоків і зв'язків між ними [2].

База знань являє собою сукупність середовищ, що зберігають знання різних типів. Розглянемо коротко їхнє призначення.

База фактів (даних) зберігає конкретні дані, а база правил — елементарні вирази, називані в теорії штучного інтелекту продукціями.

База процедур містить прикладні програми, за допомогою яких виконуються всі необхідні перетворення й обчислення.

База закономірностей включає різні відомості, що відносяться до особливостей того середовища, у якому діє система.

База метазнань (база знань про себе) містить опис самої системи й способів її функціонування: відомості про те, як усередині системи представляються одиниці інформації різного типу, як взаємодіють різні компоненти системи, як був отриманий розв'язок задачі.

База цілей містить цільові структури сценарії, що дозволяють організувати процеси руху від вихідних фактів, правил, процедур до досягнення тієї мети, яка поступила в систему від користувача або була сформульована самою системою в процесі її діяльності в проблемному середовищі.

Керування всіма базами, що входять у базу знань, і організацію їх взаємодії здійснює система керування базами знань. З її ж допомогою реалізуються зв'язки бази знань із зовнішнім середовищем. Таким чином, машина бази знань здійснює першу функцію інтелектуальної системи.

Виконання другої функції забезпечує частина інтелектуальної системи, що називаються розв'язувачем, що й складається з ряду блоків, які

					ДП.КСМ.19113/11.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		14

управляються системою керування розв'язувача. Частина блоків реалізує логічний вивід.

Блок дедуктивного виводу здійснює в розв'язувачі дедуктивні міркування, за допомогою яких із закономірностей з бази знань, фактів з бази фактів і правил з бази правил виводяться нові факти. Крім цього, даний блок реалізує евристичні процедури пошуку розв'язків задач як пошук шляхів розв'язку задачі по сценаріях при заданій кінцевій меті. Для реалізації міркувань, які не носять дедуктивного характеру, тобто для пошуку за аналогією по прецеденту і т.д., використовуються блоки індуктивного й правдоподібного виводів.

Блок планування застосовується в задачах планування розв'язків разом із блоком дедуктивного виводу.

Призначення блоку функціональних перетворень полягає в розв'язку задач розрахунково-логічного й алгоритмічного типів.

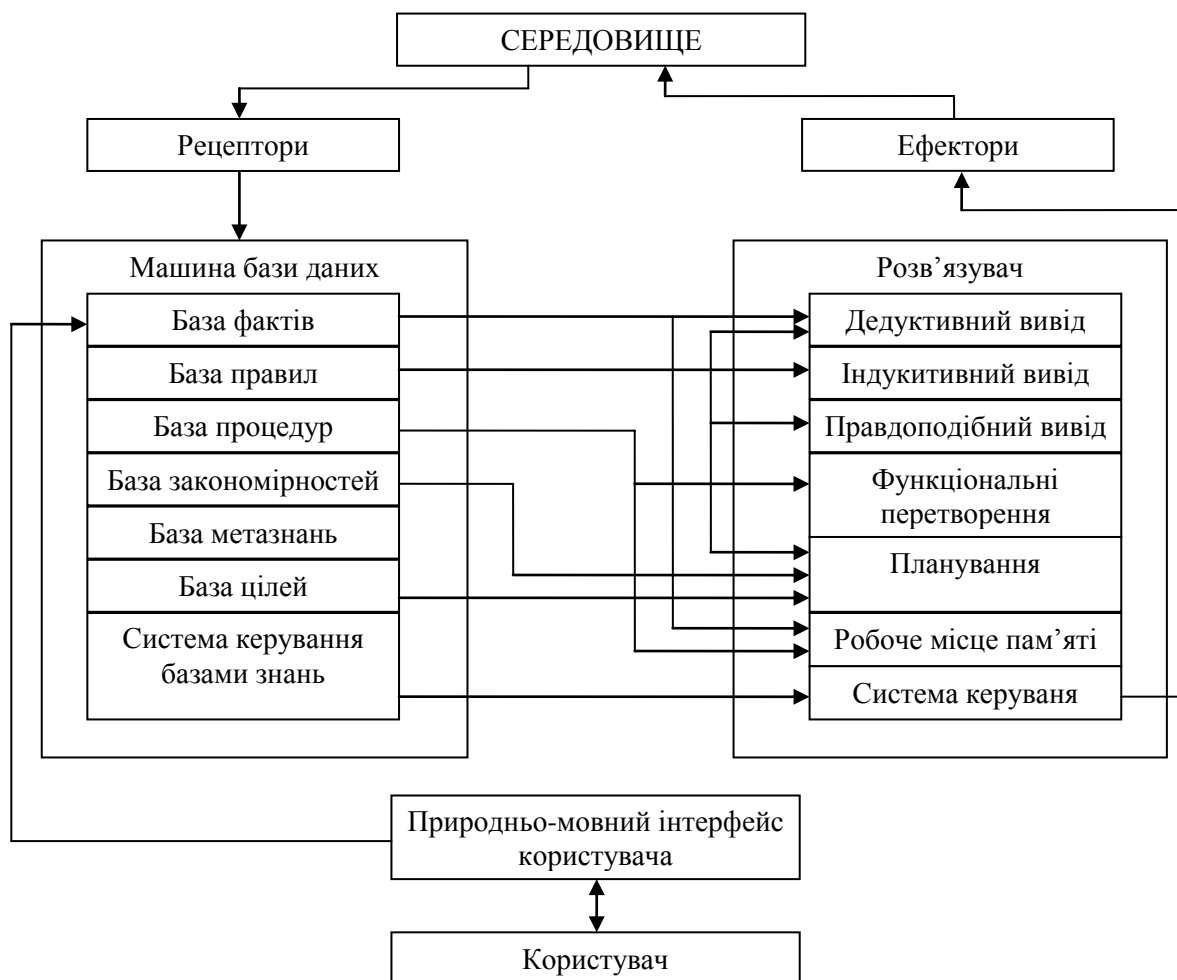


Рисунок 1.1 - Загальна структура інтелектуальної системи

Третя функція — функція спілкування — реалізується як за допомогою компонента природно-мовного інтерфейсу, так і за допомогою рецепторів і ефекторів, які здійснюють так зване невербальне спілкування й використовуються в інтелектуальних роботах.

Різновиди інтелектуальних систем.

Залежно від набору компонентів, що реалізують розглянуті функції, можна виділити наступні основні різновиди інтелектуальних систем:

- інтелектуальні інформаційно-пошукові системи;
- експертні системи (ЕС);
- розрахунково-логічні системи;
- гібридні експертні системи.

Інтелектуальні інформаційно-пошукові системи є системами взаємодії із проблемно-орієнтованими (фактографічними) базами даних на природньому, точніше обмеженому як граматично, так і лексично (професійною лексикою) природній мові (мові ділової прози). Для них характерне використання (крім бази знань, що реалізує семантичну модель представлення знань про проблемну область) лінгвістичного процесора.

Експертні системи є одним із класів, що бурхливо розвиваються, інтелектуальних систем. Дані системи в першу чергу стали створюватися в математично слабоформалізованих галузях науки й техніки, таких як медицина, геологія, біологія й інші. Для них характерна акумуляція в системі знань і правил міркувань досвідчених фахівців у даній предметній області, а також наявність спеціальної системи пояснень.

Розрахунково-логічні системи дозволяють вирішувати управлінські й проектні задачі по їхніх постановках (описах) і вихідним даним незалежно від складності математичних моделей цих задач. При цьому кінцевому користувачеві надається можливість контролювати в режимі діалогу всі стадії обчислювального процесу. У загальному випадку, по опису проблеми мовою предметної області забезпечується автоматична побудова математичної моделі й автоматичний синтез робочих програм при формулюванні функціональних

					ДП.КСМ.19113/11.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		16

задач із даної предметної області. Ці властивості реалізуються завдяки наявності бази знань у вигляді функціональної семантичної мережі й компонентів дедуктивного виводу й планування.

Останнім часом у спеціальний клас виділяються гібридні експертні системи. Зазначені системи повинні увібрати в себе кращі риси як експертних, так і розрахунково-логічних і інформаційно-пошукових систем. Розробки в області гібридних експертних систем перебувають на початковому етапі.

Найбільш значні успіхи в цей час досягнуті в такому класі інтелектуальних систем, як експертні системи.

Важливе місце в теорії штучного інтелекту (ШІ) займає проблема представлення знань. У цей час виділяють наступні основні типи моделей представлення знань:

- семантичні мережі, у тому числі функціональні;
- фрейми й мережі фреймів;
- продукційні моделі.

Семантичні мережі визначають як граф загального виду, у якому можна виділити множину вершин і ребер. Кожна вершина графа представляє деяке поняття, а дуга - відношення між парою понять. Мітка й напрямок дуги конкретизують семантику. Мітки вершин семантичного навантаження не несуть, а використовуються як довідкова інформація.

Різні різновиди семантичних мереж мають різну семантичну потужність, отже, можна описати ту саму предметну область більш компактно або громіздко.

Фреймом називають структуру даних для представлення й опису стереотипних об'єктів, подій або ситуацій. Фреймова модель представлення знань складається із двох частин:

- набору фреймів, що становлять бібліотеку усередині знань, що представляються;
- механізмів їх перетворення, зв'язування і т.д.

Існує два типи фреймів:

- зразок (прототип) - інтенціональний опис деякої множини екземплярів;

					ДП.КСМ.19113/11.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		17

- екземпляр (приклад) - екстенціональне представлення фрейму-зразка.

У загальному виді фрейм може бути представлений наступним кортежем:

$\langle I\Phi, (IC, ZC, PP), \dots, (IC, ZC, PP) \rangle$,

де $I\Phi$ - ім'я фрейму;

IC - ім'я слоту;

ZC - значення слоту;

PP - ім'я приєднаної процедури (необов'язковий параметр).

Слоти - це деякі незаповнені підструктури фрейму, заповнення яких приводить до того, що даний фрейм ставиться у відповідність деякій ситуації, явищу або об'єкту.

У якості даних фрейм може містити звертання до процедур (так звані приєднані процедури). Виділяють два види процедур: процедури-демони й процедури-слуги. Процедури-Демони активізуються при кожній спробі додавання або видалення даних зі слоту. Процедури-Слуги активізуються тільки при виконанні умов, певних користувачем при створенні фрейму.

Продукційні моделі - це набір правил виду " умови-дія", де умовами є твердження про вміст бази даних, а дії являють собою процедури, які можуть змінювати вміст бази даних.

Формально продукція визначається в такий спосіб:

$(i); Q; P; Z; QA B; N$,

де (i) - ім'я продукції (правила);

Q — сфера застосування правила;

P — передумова (наприклад, пріоритетність);

C — предикат (відношення);

$A \rightarrow B$ - ядро;

N — постумови (зміни, внесені в систему правил).

Практично продукції будуються за схемою "ЯКЩО" (причина або, інакше, посилка), "ТО" (наслідок або, інакше, ціль правила).

Отримані в результаті спрацьовування продукцій нові знання можуть використовуватися в наступних цілях:

					ДП.КСМ.19113/11.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		18

- розуміння й інтерпретація фактів і правил із застосуванням продукцій, фреймів, семантичних ланцюгів;
- розв'язок задач за допомогою моделювання;
- ідентифікація джерела даних, причин розбіжностей нових знань зі старими, одержання метазнань;
- складання питань до системи;
- засвоєння нових знань, усунення протиріч, систематизація надлишкових даних.

Процес розгляду комп'ютером набору правил (виконання програми) називають консультацією. Її найбільш зручна для користувача форма — дружній діалог з комп'ютером. Інтерфейс може бути у формі меню, мовою команд і природньою мовою.

Діалог може бути побудований на системі питань, що задаються користувачем, комп'ютером, або фактів — даних, що зберігаються в базі даних. Можливий змішаний варіант, коли в базі даних недостатньо фактів.

При прямому пошуку користувач може задавати дві групи питань, на які комп'ютер дає пояснення:

- як отриманий розв'язок. При цьому комп'ютер повинен видати на екран трасу у вигляді посилань на використані правила;
- чому комп'ютер поставив якесь запитання. При цьому на екран видається своєрідна траса, яку комп'ютер хотів би використовувати для виводу після одержання відповіді на питання, що задається. Питання чому може бути заданий як у процесі консультації, так і після виконання програми.

Специфічний алгоритм пошуку, реалізований логічними мовами: він є фактично послідовним перебором по дереву зверху вниз — ліворуч праворуч.

1.2 Гібридні інтелектуальні системи

Інтелектуальні системи (ІС), створені за останні роки, залежно від

					ДП.КСМ.19113/11.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		19

архітектури можна класифікувати на однокомпонентні (single component) і багатокомпонентні (multi component) ІС [2].

Однокомпонентні ІС засновані на використанні єдиного засоби штучного інтелекту, такого, наприклад, як нечітка логіка або штучна нейронна мережа.

Багатокомпонентні ІС поєднують у собі різні засоби штучного інтелекту в єдину обчислювальну модель. Багатокомпонентні ІС являють собою архітектурно більш складні системи, з компонентами, що володіють власною функціональністю й об'єднаними в ієрархічну багат шарову модель. Коректна робота такої системи, таким чином, залежить від правильної роботи всіх шарів, помилка в роботі одного із шарів може поширюватися на інші шари й відбиватися на роботі всієї системи. Багатокомпонентні ІС сполучають різні засоби ШІ, які взаємодіють між собою для одержання розв'язку поставленої задачі.

Така різноманітність інтелектуальних систем визначається необхідністю формалізації різноманітних даних і знань, причому не завжди процес формалізації можна успішно виконати для даних і знань будь-якого виду. От чому в останні роки стали інтенсивно розвиватися гібридні інтелектуальні системи (один з видів багатокомпонентних інтелектуальних систем), що дозволяють використовувати переваги традиційних засобів штучного інтелекту і в той же час долаючи деякі їх недоліки, здатні вирішувати задачі, нерозв'язані окремими методами штучного інтелекту. Гібридні ІС дозволили більш ефективно з'єднувати формалізовані і неформалізовані знання за рахунок інтеграції традиційних засобів штучного інтелекту [3, 4].

Використання гібридних ІС особливо ефективно в медицині, банківській справі, при розпізнаванні образів, при виявленні несправностей у технічних системах і т.д., тобто в тих галузях, де доводиться мати справа як чіткою, так і з нечіткими знаннями [5–10].

У свою чергу, гібридні ІС можна класифікувати, залежно від архітектури, на наступні типи: 1) комбіновані (combination), 2) інтегровані (integration), 3) об'єднані (fusion) і 4) асоціативні (association) гібридні інтелектуальні системи [11] (рисунок 1.2).

					ДП.КСМ.19113/11.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		20

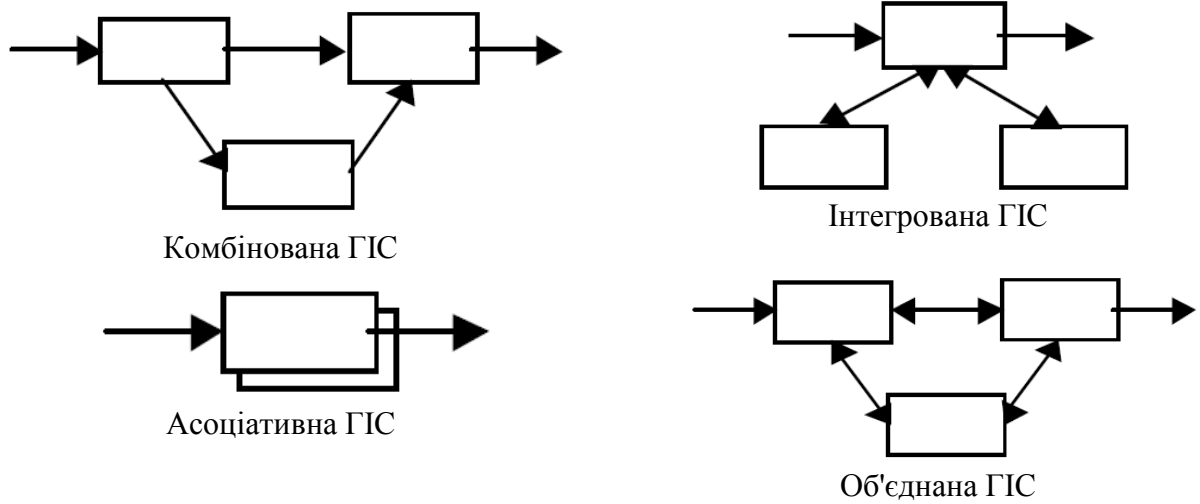


Рисунок 1.2 - Типи архітектур ГІС

Крім того, можна розглядати й ще один тип гібридних ІС, появу якого обумовлено стрімким зростанням обсягів знань і даних, які можуть зберігатися в розподілених базах даних, доступних через глобальну мережу Internet. Архітектуру гібридної ІС такого типу можна назвати розподіленою (distributed) (рисунок 1.3).

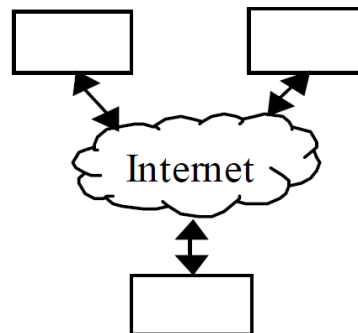


Рисунок 1.3 - Архітектура розподіленої гібридної ІС

Комбіновані (combination) гібридні інтелектуальні системи.

Прикладом комбінованих гібридних ІС служать гібридні експертні системи, що представляють собою інтеграцію експертних систем і нейронних мереж і з'єднуючі як формалізовані знання (в експертних системах), так і неформалізовані знання (у нейронних мережах).

Прикладами комбінованих систем можуть служити: гібридна експертна система для анестезіології важких пацієнтів [12], експертна система адаптивного навчання [13], гібридна експертна система для медичної діагностики [14].

До складу гібридної експертної системи для медичної діагностики входять три основні компоненти: 1) нечітка нейронна мережа, 2) нечітка експертна система й 3) модуль прийняття й пояснення рішень. Крім того, до складу гібридної експертної системи входять ще два додаткові інтерфейсні модулі, що з'єднують нейронну мережу й експертну систему, що й дозволяють взаємно перетворювати дані нечіткої нейронної мережі й нечіткої експертної системи.

Гібридна експертна система для медичної діагностики поєднує у собі чисельне й лінгвістичне представлення знань, у системі використовується ієрархічна інтеграція самоорганізованої нечіткої нейронної мережі (з можливістю online-навчання) і нечіткої експертної системи, оптимізованої з допомогою генетичного алгоритму.

Нейронна мережа сприяє швидкому навчанню, у той час як експертна система дозволяє виконати інтерпретацію нечітких даних і пояснити отриманий розв'язок. Штучна нейронна мережа може навчатися як із учителем, так і без учителя, може навчатися без перепідготовки старої інформації.

Нечітка експертна система – це експертна система, об'єднана з нечіткими множинами. Експертна система побудована на основі знань, впроваджених у навчену нейронну мережу. Експертна система прозора для користувача, розв'язки, одержувані експертною системою, легкі для розуміння, оскільки правила в базі знань в «if then» форматі використовують природню мову.

Зовнішні вхідні дані надходять у гібридну систему як через експертну систему, так і через нейронну мережу. Вихідні дані експертної системи й нейронної мережі надходять на вхід модуля, що знаходить і пояснює розв'язок.

Інтегровані (integration) гібридні інтелектуальні системи

В архітектурі інтегрованих гібридних ІС головним є основний модуль-інтегратор, який, залежно від поставленої мети й поточних умов знаходження розв'язку, вибирає для функціонування ті або інші інтелектуальні модулі, що входять у систему, і поєднує відгуки задіяних модулів.

Прикладами інтегрованих гібридних ІС можуть служити: гібридна експертна система RAISON [15], гібридна інтелектуальна система для

					ДП.КСМ.19113/11.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		22

створення планів руху роботів [16].

Гібридна експертна система RAISON – це інтегрована, заснована на знаннях, експертна система, що включає нейронну мережу, базу даних, геоінформаційну систему, картографічний аналізатор, графічну й програмно-язикову компоненти.

Експертна система є інтегруючим модулем і пов'язана з іншими компонентами гібридної системи. Допоміжні підсистеми, такі як база даних і геоінформаційна система, служать сполучними ланками експертної системи з повнофункціональними зовнішніми базами даних і геоінформаційними системами.

Функціонування експертної системи засноване на використанні алгоритму із застосуванням індукційного дерева (алгоритм дозволяє генерувати правила на основі зразків наборів даних), відомі й існуючі правила пропонуються експертом. Правила можуть використовуватися як для прямої, так і для зворотної послідовності запитів. Тому що структура дерева є майже оптимальною, число запитів, що вимагаються для одержання висновку, є мінімальним.

База даних і карти геоінформаційної системи також доступні для нейронної мережі. Нейронна мережа використовується в системі RAISON як інструмент навчання. Нейронна мережа може приймати вхідні дані й використовувати їх як зразки для навчання (навчатися із учителем). Вихідні дані системи RAISON можуть бути збережені в базі даних, відображені, наприклад, у графічному виді (у вигляді геоінформаційних карт).

Гібридна інтелектуальна система для створення планів руху роботів містить у собі об'єктно-орієнтовану базу знань для моделювання навколишнього середовища й адаптивний алгоритм планування руху (ААПР). На основі отриманих даних (попередньо сформованих карт навколишнього оточення й наступної динамічної інформації про оточення робота) ААПР визначає шлях і генерує послідовність команд для навігації робота.

Гібридна система складається із чотирьох модулів: 1) планувальника, 2) помічників, 3) виконавців рухів і 4) керуючого центру. У системі реалізована

					ДП.КСМ.19113/11.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		23

архітектура з використанням керуючого центру, який пов'язаний з перерахованими модулями й координує їхнє функціонування.

Для керування досить складною концепцією спільно використаної інформації у системі застосована ієрархічна база знань, яка містить у собі три інформаційні області: 1) модель навколишнього середовища, 2) перцептрон і 3) супервізор, що здійснює настроювання керуючого режиму.

У системі використовується спрощена схема нечіткого виводу для визначення дистанції до перешкоди й ширини шляху. Для зменшення обсягу обчислень застосовуються нечіткі трикутні функції приналежності. Машина нечіткого виводу працює з нечіткими правилами й нечіткими трикутними функціями приналежності. Для пошуку підходящого шляху в системі використовується мережа Хопфілда. Для генерації команд поведінки використовуються правила нечіткого виводу.

Об'єднані (fusion) гібридні інтелектуальні системи.

Характерною рисою нейронних мереж і генетичних алгоритмів є їхня здатність до навчання й адаптації за допомогою оптимізації. З'єднання цих методів з іншими методами штучного інтелекту дозволяє збільшити ефективність їх здатності до навчання. Таку архітектуру гібридної ІС можна віднести до об'єданого типу.

Прикладами об'єднаних гібридних ІС можуть служити: гібридна експертна система для інвестиційних рекомендацій [17], гібридна експертна система для визначення несправностей в енергетичних системах [18].

Гібридна експертна система для визначення несправностей в енергетичних системах складається з модуля вхідної умови й модуля виводу. Модуль виводу призначений для оцінки місцезнаходження й типу несправності за допомогою штучної нейронної мережі (багатошарового перцептрона). У розглянутій гібридній експертній системі застосовані методи, засновані на використанні штучних нейронних мереж, що й відрізняються від традиційних використанням приймання вхідної умови (передумови), коли вхідний вектор подається не безпосередньо на вхід нейронної мережі, а на вхід передумовного модуля. У свою чергу, вихідний сигнал передумовного модуля подається на

					ДП.КСМ.19113/11.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		24

вхід нейронної мережі, що дозволяє зробити роботу мережі більш ефективною. У системі застосовано два передумовних методи: один з них використовує швидке перетворення Фур'є для виміру відхилень струму, інший – застосовує детерміновану нелінійну кластеризації для результатів, отриманих за допомогою перетворення Фур'є.

Асоціативні (association) гібридні інтелектуальні системи

Архітектура асоціативних гібридних систем припускає, що інтелектуальні модулі, що входять до складу такої системи, можуть працювати як автономно, так і в інтеграції з іншими модулями. У цей час, через недостатній розвиток систем такого типу, системи з асоціативною архітектурою ще недостатньо надійні й не одержали широкого поширення.

Розподілені (distributed) гібридні інтелектуальні системи

Наступним рівнем у розвитку гібридних систем можуть стати розподілені інтелектуальні системи, що представляють мультиагентний підхід в області розподіленого штучного інтелекту [19, 20]. При цьому підході кожний функціональний інтелектуальний модуль працює автономно й взаємодіє з іншими модулями (агентами) шляхом передачі повідомлень через мережу. Існуючі в цей час інтелектуальні системи, такі як експертні системи, нейронні мережі й т.п. можуть бути перетворені в агенти.

У процесі перетворення інтелектуальні модулі можуть бути доповнені керуючими й комунікативними знаннями, необхідними для їхнього об'єднання в мультиагентну інтелектуальну розподілену систему.

Прикладом розподіленої інтелектуальної системи може служити інтелектуальна навчальна система для навчання викладачів вищої школи новим технологіям, заснована на Web [21].

Інтелектуальна навчальна система формує навчальний курс (презентацію освітнього матеріалу, адаптовану до вимог користувачів з використанням технологій штучного інтелекту для точного визначення моделі кожного користувача, а також для створення педагогічних рішень), що відповідає потребам користувачів з різними рівнями знань і різними здатностями в освоєнні нового матеріалу.

					ДП.КСМ.19113/11.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		25

Інтелектуальна навчальна система заснована на експертній системі, спрямованій на керування процесом навчання. Експертна система використовує формалізм гібридного представлення знань, що називається нейроправилами.

Система складається з наступних компонентів: 1) домену знань, 2) моделі користувача, 3) педагогічної моделі, 4) машини логічного виводу й 5) користувацького інтерфейсу. Домен знань містить знання, що ставляться до досліджуваної теми, що й представляють актуальний навчальний матеріал. Домен складається із двох частин: 1) понять знань і 2) курсових блоків. Курсові блоки містять навчальний матеріал, що представляється користувачам системи як Web-Сторінка. Кожен курсовий блок асоціюється з певним числом понять знань. Система підтримує варіанти однієї й тієї ж сторінки (курсного блоку) з різними представленнями знань. Моделі користувача застосовуються для запису інформації, пов'язаної з користувачем. Моделі користувачів обновляються протягом усього процесу навчання. Педагогічна модель формує процес навчання. Вона пропонує інфраструктуру знань для адаптації презентації досліджуваного матеріалу відповідно до даних, що втримуються в моделі користувача.

Формалізм представлення знань в експертній системі заснований на нейроправилах, гібридних правилах, що інтегрують символічні правила з нейрообчисленнями. Нейроправила створюються або на основі емпіричних даних (навчальних шаблонів), або на основі символічних правил. Кожне нейроправило індивідуально навчається за допомогою спеціального алгоритму. Механізм виводу заснований на стратегії зворотного виводу.

1.3 Біомедичні зображення

Біомедичні зображення використовуються при встановленні діагнозу, подальшому лікуванні і плануванні терапії. Існує ряд чинників, які утруднюють лікування пацієнта засноване на медичних зображеннях, Природно, медичні дані про пацієнта збираються в різних лікувальних

					ДП.КСМ.19113/11.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		26

установах. Лікарі часто не мають доступу до всіх історій хвороб всіх своїх пацієнтів. Біомедичні зображення представляються дуже великими об'ємами даних з складною структурою (клінічно і епідеміологічно значимі показники, такі як вік пацієнта, харчування, спосіб життя і історія хвороби, параметри отримання зображень і анатомічні / фізіологічні зміни). Часто зображень одної модальності виявляється недостатньо, оскільки на отримання зображення впливає багато параметрів, і додаткова інформація накопичується різними системами збору фізичних даних.

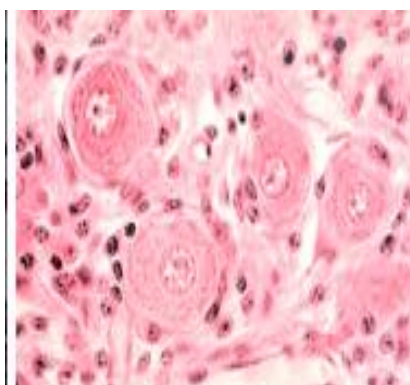
При встановленні діагнозу і проведенні лікування лікарі використовують біомедичні зображення наступних типів: зображення цифрової мікроскопії (цитологічних, гістологічних зразків), рентгенограми (кількість цифрових рентгенограм зростає, але в переважній більшості це рентгенівські знімки), УЗД, магнітно-резонансна томографія (МРТ) (MRI - Magnetic Resonance Imaging), комп'ютерна томографія (КТ, Computed Tomography), томографія на позитивному випромінюванні (PET - Positive Emission Tomography), та ін. Використання медичних зображень буде зростати у міру того, як в багатьох лікарнях (спочатку в базових лікарнях медичних інститутів) будуть встановлені системи з високою пропускною спроможністю для архівації зображень. Біомедичні зображення дають основний об'єм інформації про пацієнта, проте, самі по собі вони недостатні, оскільки потрібний їх аналіз і інтерпретація в контексті історії хвороби пацієнта (тобто потрібні метадані пов'язані із зображеннями). Приклади біомедичних зображень наведені на рисунку 1.4.

Метод комп'ютерної томографії було розроблено на початку 1970-х років і сьогодні застосовується у більш ніж 30000 радіологічних відділень і діагностичних центрів. Головна особливість можливість одночасно відображати м'які тканини, кістки і кров'яні судини. Для підвищення інформативності зображень (виділення найбільш інформативних фрагментів, їх збільшення, здобуття 3-вимірних зображень, кольорокартування, реалізація високої просторової роздільної здатності, підвищення контрастної роздільної здатності).

					ДП.КСМ.19113/11.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		27



КТ зображення грудної частини тіла



Зображення гістологічного препарату



Зображення отримане при УЗД

Рисунок 1.4 – Біомедичні зображення

Об'єкти на медичних зображеннях володіють складністю і варіабельністю параметрів, що обумовлює високі вимоги до надійності, точності і достовірності результатів досліджень. Використання обчислювальної техніки і математичних методів в цій галузі дозволяє не лише прискорити процес обробки матеріалу, але і підвищити точність результатів дослідження.

Джерелом зображень, що містять текстурні елементи, в медицині є цитологія, гістологія, ендоскопія, рентгенологія, ультразвукове дослідження. Із розвитком вимірювальної і цифрової техніки медична діагностика сьогодні неможлива без використання цифрових зображень.

Об'єктами дослідження сучасної гістології і цитології служать живі і мертві (фіксовані) клітини і тканини, і їх зображення, отримані в світлових і електронних мікроскопах.

Наприклад, кожному пухлинному захворюванню відповідають певні особливості зображення текстури, що стосуються його структурних характеристик і характеристик яскравості. Кваліфікований лікар з використанням свого багатолітнього досвіду може по зображенню гістологічного препарату зробити експертний висновок про тип захворювання і, отже, про метод його лікування.

Для цитологічних і гістологічних зображень застосовуються наступні методи досліджень [20]:

– Денситометрія. До задач денситометрії відноситься вимірювання оптичної щільності ядра клітини, вимірювання кольору клітин (колориметрія),

вимірювання яскравості флуоресценції, спектральний аналіз.

– Підрахунок кількості клітин і розподілу клітин по розмірах.

– Вимірювання геометричних розмірів, морфометрія. Найбільш часто вимірюваними параметрами є наступні: площа клітини або її ядра, периметр клітини, лінійні розміри, коефіцієнт витягнутості (відношення двох головних моментів), коефіцієнт форми, відношення площі ядра клітини до площі цитоплазми.

При обробленні біомедичних зображень використовують методи текстурного аналізу для різноманітних задач класифікації, розпізнавання та пошуку і вибірки зображень із баз даних.

Автоматизація аналізу гістологічних структур прискорює діагностику захворювання, дозволяє розширити кордони наукових пошуків в медицині. Автоматичний вимір параметрів гістологічних об'єктів дає можливість уточнити лікування і управління терапевтичними процесами. Так, найбільш перспективним методом ранньої діагностики пухлинних захворювань в даний час є автоматизація цитофотометричного, морфометричного аналізів гістологічних препаратів і розділення їх за принципом норма - патологія.

Однією з головних частин автоматизації вимірювань оптичних і геометричних параметрів є виділення об'єктів на гістологічних препаратах. Основною причиною відсутності автоматизації в гістології є висока варіабельність і слабкий контраст більшості гістологічних структур.

Для аналізу гістологічних препаратів колір складових компонентів тканини грає важливу роль. Фарбування препарату дозволяє виділити ті або інші клітинні структури у зразку [11].

Нижче розглянуто основні результати в області аналізу цифрових гістологічних зображень (ЦГЗ). Вибір методів аналізу ЦГЗ зумовлений, насамперед, варіабельністю ознак мікрооб'єктів – площі, периметру, округлості та ін.

В області сегментації кольорових цитологічних зображень авторами розроблено алгоритм розділення зображення на кольорові складові, що дозволило розділити ядро та цитоплазму. Алгоритм полягає в обчисленні

					ДП.КСМ.19113/11.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		29

внеску кожного із застосовуваних фарбників, базуючись на їх спектрі поглинання світла. Для кожного типу фарбника наперед обчислено вектор кольорів у RGB просторі.

В області класифікації типу клітин задачу постановки діагнозу, а саме висновку про наявність злоякісного новоутворення, вирішено для окремих видів пухлин [4,5]. В роботі [5] детально показано кроки постановки діагнозу. Інтерпретація у виконанні спеціаліста патолога повинна привести до збору морфологічних доказів (числових характеристик об'єктів), що дозволяють ідентифікувати патологічні процеси, які можуть виявитися доброякісними або злоякісними. Цей крок передбачає досягнення трьох різних завдань: морфологічного аналізу, ідентифікації клітин постановки та інтерпретації діагнозу.

Питання побудови нечітких правил для постановки діагнозу на основі якісних ознак описано в роботі. В якості умов правил виступають ознаки окремих мікрооб'єктів (клітин).

Із наведеного вище видно, що задачі діагностування вирішені для частини патологічних процесів, що розглядають як об'єкт клітину. Проте існує багато задач діагностування патологічних процесів, що базуються на аналізі характеристик шарів тканин та інших складних об'єктів.

Основні задачі аналізу зображення - це розпізнавання та класифікація образів. Аналіз можна розділити на певні етапи [29, 30]: сприйняття поля зору, сегментація, нормалізація виділених об'єктів, розпізнавання. Класифікація методів аналізу зображень наведена на рисунку 1.5.

Алгоритми попередньої обробки призначені для покращення вхідного зображення і видалення шумів.

Сегментація зображення – це процес розділення зображення на множини областей, що не перетинаються, об'єднання яких дасть ціле зображення. Сегментація складається з двох етапів: побудова множини ознак (класів, правил), класифікація кожного об'єкта (пікселя, області) відповідно до його признаку. До областей, що отримуються в результаті сегментації, пред'являються наступні вимоги [21]:

					ДП.КСМ.19113/11.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		30

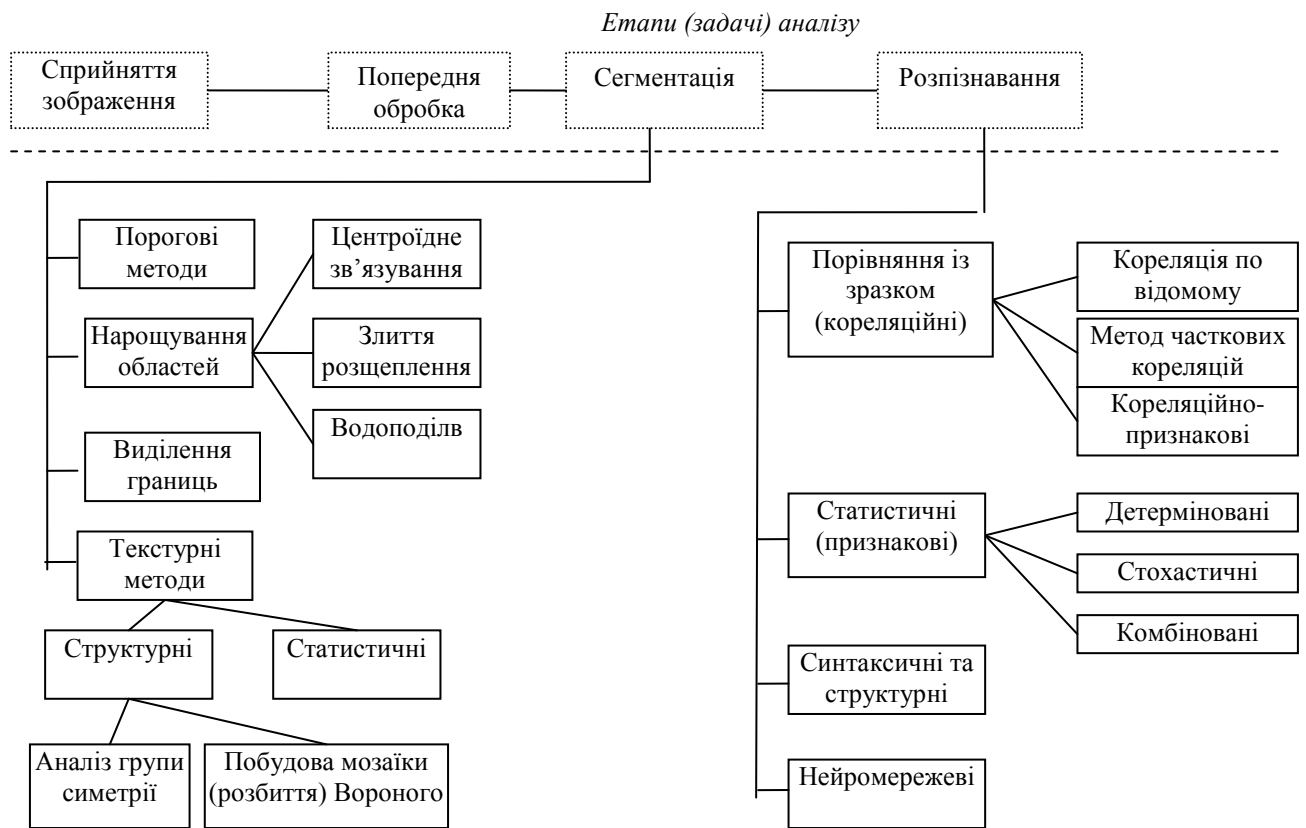


Рисунок 1.5 - Класифікація методів аналізу зображень

1. Области мають бути однорідні відносно певних характеристик.
2. Внутрішні частини областей мають бути простими без великої кількості маленьких отворів.
3. Суміжні області повинні істотно відрізнятися по значеннях вибраних характеристик, відносно яких вони вважаються однорідними.
4. Границі кожного сегменту мають бути простими, просторово точними.

При аналізі конкретного зображення виникає необхідність вибору алгоритму, найбільш відповідного для його сегментації. При цьому доводиться враховувати як властивості зображення, так і особливості конкретного алгоритму.

Порогові методи застосовуються при існуванні стабільних відмінностей в яскравостях окремих областей. Основним недоліком порогових методів є те, що не враховується просторова інформація. Метод нарощування областей ефективний за наявності стійкої зв'язності усередині окремих сегментів. Його основними проблемами є: вибір координат початкового пікселя, вибір критерію

подібності (предиката). Метод виділення границь застосовується, якщо границі досить чіткі і стабільні.

Кластеризація колірному простору схожа до задачі сегментації. Для того, щоб звести задачу сегментації до задачі кластеризації, достатньо задати відображення точок зображення у простір ознак і ввести метрику (міру близькості) у цьому просторі ознак. В якості ознак виступає представлення точки зображення в певному колірному просторі а метрикою, наприклад, Евклідова відстань між векторами в простір ознак. В цьому випадку результатом кластеризації буде квантування кольору у зображенні. Методи кластеризації: k -середніх (узагал. метод Ллойда), EM алгоритм. Недоліки методів кластеризації: не враховується просторове розташування точок (або координати враховуються непрямо, як одна ознак), можуть погано працювати на зашумлених зображеннях. Ще одна проблема – це збіжність алгоритму k -середніх за n ітерацій. Збіжність до локального оптимуму – це такий розподіл точок, при якому перенесення однієї точки в інший кластер збільшує загальну суму відстаней (в кластері) [12].

Для сегментації зображень при розгляді однорідності, шорсткості, регулярності сегментів, застосовують текстурні методи які умовно поділяються на дві категорії: статистичні і структурні. Прикладом статистичного підходу є використання матриць суміжності, що формуються з вихідних зображень, з подальшим підрахунком статистичних моментів і ентропії. При структурному підході, наприклад, на основі мозаїки Вороного, будується безліч багатокутників. Багатокутники із загальними властивостями об'єднують в області. Для дослідження загальних властивостей часто використовують ознаки - моменти багатокутників.

Опис зображення базується на апараті математичної лінгвістики, тобто теорії формальних граматик [13].

Розпізнавання образів - це процес, метою якого є класифікація об'єктів або образів, за декількома категоріями або класами. Методи розпізнавання образів застосовуються для класифікації областей (сегментів) та об'єктів.

					ДП.КСМ.19113/11.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		32

Класифікація базується на прецедентах - образах, правильна класифікація яких відома. Вважатимемо, що всі об'єкти або явища розбиті на скінченне число класів. Для кожного класу відомо і вивчено скінченне число об'єктів - прецедентів. Завдання розпізнавання образів полягає в тому, аби віднести новий розпізнаваний об'єкт до якого-небудь класу.

Вимірювання які проводяться для класифікації називаються ознаками. Ознака - це деяка числова міра (характеристика) об'єкта. Сукупність різних ознак одного образу формують його *вектор ознак*. Вектори ознак приймають значення в *просторі ознак*. В рамках задачі розпізнавання вважається, що кожному образу ставиться у відповідність єдине значення вектора ознак і навпаки: кожному значенню вектора ознак відповідає єдиний образ. *Класифікатором* або розв'язуючим правилом називається правило віднесення образу до одного з класів на підставі його вектора ознак.

Наведемо основні етапи розробки системи класифікації : генерація ознак для об'єкта, відбір найбільш інформативних ознак, побудова класифікатора, оцінка побудованої системи з точки зору правильності чи помилковості класифікації.

Використовуватимемо наступну модель задачі класифікації:

Ω - множина об'єктів розпізнавання (простір образів);

$\omega : \varpi \in \Omega$ - об'єкт розпізнавання (образ);

$g(\omega) : \Omega \rightarrow M$, $M = \{1, 2, \dots, m\}$ - індикаторна функція, що розбиває простір образів Ω на m класів, що не перетинаються $\Omega^1, \Omega^2, \dots, \Omega^m$. Індикаторна функція невідома спостерігачеві.

X - простір спостережень, що сприймаються спостерігачем (простір ознак).

$x(\omega) : \Omega \rightarrow X$ - функція, що ставить у відповідність кожному об'єкту ω точку $x(\omega)$ в просторі ознак. Вектор $x(\omega)$ - це образ об'єкту, що сприймається спостерігачем. У просторі ознак визначені непересічні множини $K_i \subset X$, $i=1, 2, \dots, m$ точок, відповідних образам одного класу.

$\hat{g}(x) : X \rightarrow M$ - розв'язуюче правило - оцінка для $g(\omega)$ на основі $x(\omega)$, тобто

					ДП.КСМ.19113/11.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		33

$$\hat{g}(x) = \hat{g}(x(\omega)).$$

Нехай $x_j = x(\omega_j)$, $j=1,2,\dots,N$ - доступна спостерігачеві інформація про функції $g(\omega)$ і $x(\omega)$, але самі ці функції спостерігачеві невідомі. Тоді (g_j, x_j) , $j=1,2,\dots,N$ - є множина прецедентів.

Задача полягає в побудові такого розв'язуючого $\hat{g}(x)$ правила, щоб розпізнавання проводилося з мінімальним числом помилок.

Звичайний випадок - вважати простір ознак Евклідовим, тобто $X=R^l$. Якість розв'язуючого правила вимірюють частотою появи правильних рішень. Зазвичай його оцінюють, наділяючи множину об'єктів Ω деякою імовірнісною мірою. Тоді задачу можна записати у вигляді

$$\min P\{\hat{g}(x(\omega)) \neq g(\omega)\}.$$

Класифікацію методів розпізнавання образів приводять автори [29,30]. Виділяють 4 групи методів розпізнавання:

1) Порівняння із зразком (кореляційні методи). Метод порівняння із зразком заснований на геометричній нормалізації зображення, і обчисленні "відстані" отриманої нормалізації до прототипу. Системи, що побудовані на основі цих методів звичайно призначені для аналізу зображень обмеженого типу і не володіють адаптивністю [1].

2) Статистичні методи. Статистичні методи полягають у виділенні "ознак" зображення, і порівняння їх з відомими статистичними розподілами цих же ознак у зразках.

3) Нейронні мережі. При застосуванні нейромереж для розпізнавання вибирається тип штучної нейронної мережі, і проводиться її навчання по відомих зразках.

4) Структурні і синтаксичні методи. Структурні методи [23] полягають у розбивці зразка на більш прості елементи, і побудові правила в залежності від входження/не входження того або іншого елемента або послідовностей. Потім ті ж елементи шукаються в зображенні, і застосовуються раніше виділені правила.

Методи порівняння зі зразком (кореляційні) засновані на прийнятті рішень по критерію близькості з еталонами [29]. В основному застосовується

					ДП.КСМ.19113/11.00.00.000.ПЗ	Арк.
						34
Змн.	Арк.	№ докум.	Підпис	Дата		

при виявленні і розпізнаванні зображень в системах навігації, стеження, промислової роботизації. Найбільш трудомісткий підхід з точки зору обчислювальної складності. Полягає у багатокроковій кореляції при повністю заданому еталоні, шляхом сканування вхідного поля зору.

Статистичні методи засновані на розбитті простору ознак в процесі навчання на області, що відносяться до розпізнаваних класів або знаходженні в просторі ознак центрів класів. В першому випадку розпізнавання зводиться до визначення області у якій знаходиться розпізнаваний образ, в другому - невідомий образ відноситься до того класу, відстань до центру якого найменша (кластерний аналіз).

Статистичні методи розпізнавання образів базуються на класичній теорії статистичних рішень [34, 30], в якій об'єкти дослідження розглядаються як реалізації багатовимірної випадкової величини, розподіленої в просторі ознак по певному закону. Описи класів являють собою функції умовної густини розподілу імовірності значень ознак для кожного класу.

Елементи кожного класу мають свій розподіл у n -мірному просторі. Якщо функції розподілу відомі рішення про належність об'єкта приймаються за правилом Байеса. Якщо відомий тип розподілу, але не параметри використовують точкові оцінки для параметрів розподілу. Якщо функція розподілу невідома то для її побудови використовують метод гістограм або метод Парзена.

Синтаксичні методи розпізнавання образів засновані на використанні спеціальних граматик, що породжують мови, за допомогою яких може описуватися сукупність властивостей аналізованих об'єктів [23, 25].

Для різних класів об'єктів виділяються непохідні елементи (підобрази, ознаки) і можливі відношення між ними. Граматикою називають правила побудови об'єктів з цих непохідних елементів. Таким чином, кожен об'єкт представляється сукупністю непохідних елементів, сполучених між собою тими або іншими способами або, іншими словами, "реченням" певної "мови". Шляхом синтаксичного аналізу (граматичного розбору) "речення" встановлюється його синтаксична "правильність" або, що еквівалентно - чи

					ДП.КСМ.19113/11.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		35

може деяка фіксована граматика (що описує клас) породити наявний опис об'єкту. Граматичний розбір проводиться синтаксичним аналізатором, який представляє повний синтаксичний опис об'єкту у вигляді дерева граматичного розбору, якщо об'єкт є синтаксично правильним (належить класу, що описується даною граматикою). Інакше, об'єкт не розглядається, або піддається аналізу за допомогою інших граматик, що описують інші класи об'єктів. Відомі безконтекстні, автоматні і інші типи граматик. Проте задача відновлення (визначення) граматик по деякій множині висловлювань, що породжують дану мову важко формалізується.

Синтаксичний підхід найбільш корисний при роботі з образами, які не можуть бути описані числовими ознаками.

Нейромережеві методи базуються на застосування різних типів нейронних мереж (НМ). НМ складається з формальних нейронів, зв'язані з іншими нейронами [31]. Кожен нейрон перетворює сигнал, що поступають на його входи у вихідний сигнал. Перевагами НМ є можливість розпаралелювання процесу роботи, наявність процесу навчання та узагальнююча властивість. Навчання дозволяє уникнути процесу вибору ключових ознак образу та визначення їх інформативності. Тим не менше вибір вихідного представлення даних суттєво впливає на якість рішення.

Геометричні співвідношення, які можуть бути використані як ознаки при розпізнаванні об'єктів по їх зображенням, досить багаточисельні, і їх застосування багато в чому залежить від форми об'єкту і можливих просторових спотворень. Як ознаки можна використовувати [28]:

- площа зображення, тобто число пікселів на зображенні які віднесені до об'єкту на етапі сегментації;
- число пікселів в кожному рядку і стовпці зображення об'єкту;
- довжину контуру зображення в пікселях;
- параметр форми зображення $k = \frac{L^2}{S}$, де L – довжина контуру, S – площа.
- координати екстремальних точок контуру [29], наприклад вершин багатокутника, якщо контур складається з відрізків прямих ліній або точок перегину, якщо лінія контуру криволінійна;

					ДП.КСМ.19113/11.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		36

– максимальну і мінімальну довжини радіусів векторів, що з'єднують центр мас (центр яскравості) з контуром;

– розміри мінімального описаного навколо зображення прямокутника, сторони якого паралельні рядкам і стовпцям зображення;

– координати характерних точок зображення, наприклад центру мас, центру контуру або центру описаного прямокутника;

– параметри орієнтації зображення, тобто напрям діагоналей найменшого описаного прямокутника, напрям осей інерції зображення, напрям мінімальних і максимальних радіусів-векторів, що сполучають центр мас з контуром, напрям осі, що сполучає центр описаного найменшого прямокутника з центром яскравості.

– топологічні характеристики, наприклад число екстремальних точок контуру, число отворів, їх взаємне розташування.

Отже актуальною є задача поєднання методів комп'ютерного зору та виводу висновку на основі нечітких правил для визначення типу клітин на цифрових гістологічних і цитологічних зображеннях.

					ДП.КСМ.19113/11.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		37

2. ПРОЕКТУВАННЯ ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ

2.1 Мікрооб'єкти та їх характерні ознаки

Гістологія – наука, що займається вивченням тканин тварин. Тканиною називають групу клітин, схожих формою, розмірам і функціям і по продуктах своєї життєдіяльності. У всіх рослин і тварин, за винятком найпримітивніших, тіло складається з тканин, причому у вищих рослин і у високоорганізованих тварин тканини відрізняються великою різноманітністю структури і складністю своїх продуктів; поєднуючись один з одним, різні тканини утворюють окремі органи тіла. Гістологію інколи називають мікроскопічною анатомією, оскільки вона вивчає будову (морфологію) організму на мікроскопічному рівні (об'єктом гістологічного дослідження служать дуже тонкі тканинні зрізи і окремі клітини). Останнім етапом приготування зразку для мікроскопії є його розфарбовування спеціальним фарбниками. Розфарбовування є одним з підходів для контрастування зразку при його спостереженні в мікроскопі. Розфарбовування дозволяє виявляти субклітинні структури. Кольорова палітра зображень таких зразків є однаковою для того ж самого типу фарбника [28]. Цю особливість використовують при комп'ютерному аналізі гістологічних зображень.

Задача розпізнавання кластерів зображень по колірній подібності вирішується дослідниками різних наукових сфер, наприклад, при контекстному пошуку в електронних базах даних і при змістовій класифікації зображень (медична діагностика, віддалене спостереження, аналіз документів). Проте, в залежності від сфери застосування і характеру аналізованих зображень кластери, що виділяються, можуть представляти як окремі об'єкти зображень (сегментація зображень), так і значимі кольори зображень, що відображають їх суть (квантування кольорів зображень) [27-29].

Кластеризація – загальна назва множини обчислювальних процедур,

					ДП.КСМ.19113/11.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		38

використовуваних при створенні класифікації об'єктів. В результаті кластеризації утворюються “кластери” (регіони, області) – групи схожі по різних характеристиках об'єктів.

Початкові дані для процедури кластеризації - набір об'єктів, кожен з яких задається вектором своїх характеристик. В ході процедури кластеризації відбувається об'єднання “подібних” об'єктів в окремі класи. Результатом кластеризації є набір класів що містять однорідні об'єкти.

В разі рішення задачі кластеризації зображення заданий набір пікселів, кожен з яких визначається трьома колірними компонентами в одному з колірних просторів. За допомогою процедури кластеризації здійснюється виділення груп пікселів, що мають найбільш “близькі” колірні компоненти. Якщо прийняти два кластери (“фон” і “об'єкти”) то за допомогою кластеризації можна здійснити грубу сегментацію. При визначенні потрібної кількості кластерів можна отримати розфарбовування зображень.

Дослідимо алгоритм k -середніх для сегментації зображень. Алгоритм k -середніх є методом кластеризації, що використовується для поділу набору об'єктів на k груп відповідно до зазначеної міри близькості. Узагальнений алгоритм складається з двох кроків.

1. Обчислення центроїдів кластера. Початкові центроїди встановлюються, використовуючи метод, зазначений користувачем. Існує три методи вибору початкових центрів: вибрати N спостережень для максимізації початкової відстані, випадково вибрати N спостережень, і вибрати перші N спостережень. В даному випадку N означає k . Після присвоєння всіх об'єктів найближчому центроїду, обчислюються нові центроїди, використовуючи всі члени, що їм призначені. Для неперервних змінних значення центроїда є середнє значень всіх членів, призначених цьому кластеру. Для категоріальних змінних значення центроїда є перша модель всіх членів йому привласнених.

2. Присвоєння кожного об'єкта найближчому центроїду. Найближчий центроїд у точності використовує метод заданої міри близькості. Значення всіх неперервних змінних перед обчисленням нормалізуються. Відзначте, що c позначає центроїд і x позначає спостереження. Якщо i -а змінна категоріальна,

					ДП.КСМ.19113/11.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		39

$(x_i - c_i)$ дорівнює 0 якщо їхнього значення збігаються, інакше дорівнює 1. Якщо i -а змінна безперервна, x_i і c_i спершу нормалізуються, використовуючи мінімальне й максимальне значення цих змінних. Для визначення міри близькості застосовуються евклідова відстань, квадрат евклідової відстані, манхетенська відстань, відстань Чебишева.

Якщо всі спостереження належать кластеру, якому належали до поточної ітерації, тоді перервати ітерацію. Також, якщо число ітерацій дорівнює максимальному числу ітерацій, то перервати ітерацію. Оновити центроїди і отримати остаточну кластеризацію.

Міри відстані можуть визначатися в одномірному або багатомірному просторі. Найбільш прямий шлях обчислення відстаней між об'єктами в багатомірному просторі складається в обчисленні евклідових відстаней. Якщо ми маємо двох- або тривимірний простір, то ця міра є реальною геометричною відстанню між об'єктами в просторі.

Евклідова відстань. Це найбільш загальний тип відстані. Вона є геометричною відстанню в багатомірному просторі й обчислюється в такий спосіб:

$$\text{distance}(x, c) = \left\{ \sum_i (x_i - c_i)^2 \right\}^{\frac{1}{2}}.$$

На обчислену відстань можуть сильно впливати розходження між осями, по координатах яких обчислюються відстані. Наприклад, якщо одна з осей вимірюється в сантиметрах, а потім переводиться в міліметри, те остаточна евклідова відстань (або квадрат евклідової відстані), що обчислюється по координатах, сильно зміниться, і, як наслідок, результати кластерного аналізу можуть сильно відрізнятись від попередніх.

Наведемо короткий опис алгоритму кластеризації колірного простору зображення. Фактично число кластерів є кількістю кольорів у палітрі результуючого зображення. Виберемо випадковим чином k значень атрибутів з вихідного зображення й зробимо їх центрами кластерів. Згрупуємо точки кластерах, тобто віднесемо значення до кластера, центр якого перебуває ближче всього до значення. Далі для кожного кластера перерахуємо його центр (тобто

середнє арифметичне всіх значень, що входять у кластер). Останню операцію потрібно повторювати доти, доки або переміщення значень із одного кластера в іншій не припиняться, або після певної (заданої наперед) ітерації відношення переміщених значень до всіх стане менше, ніж задане наперед значення. Таким чином, буде сформовано k кластерів, що відповідають палітрі. Палітру варто заповнити центрами кластерів. Помітимо, що одночасно нам стає відома вся інформація про квантування, тобто не тільки палітра, але й індивідуальна належність значень атрибутів вихідного зображення до конкретного кластера.

На рисунку 2.1 наведено зображення гістологічного зрізу печінки з наявною патологією. Для морфометричного аналізу потрібно з допомогою сегментації виділити ядра клітин та цитоплазму. Ядра при фарбуванні зразку набувають темнофіолетового кольору а цитоплазма рожево-сірого.

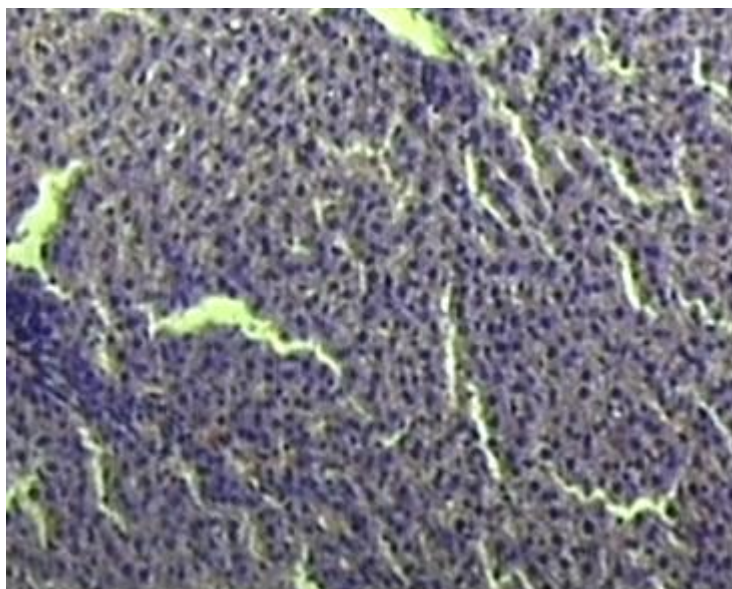


Рисунок 2.1 – Зображення гістологічного зразка

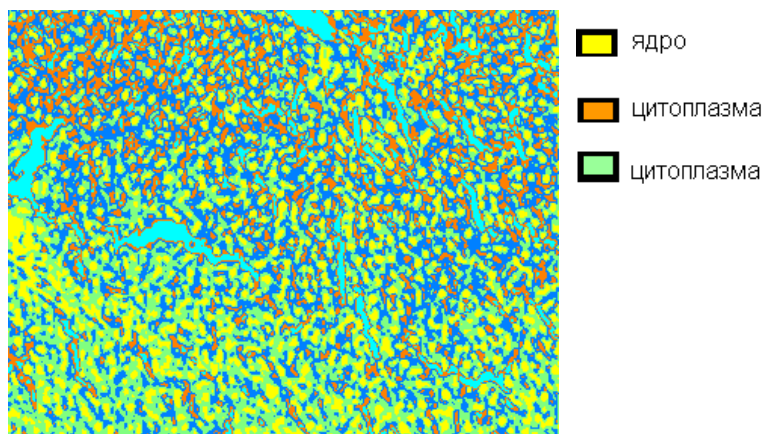
Сегментація цього зображення при різній кількості кластерів наведена на рисунку 2.2. Як видно із результатів моделювання алгоритму сегментації визначення кількості кластерів залежить від якості зображення.

Кожен об'єкт має набір ознак, що якісно відрізняє його серед множини інших об'єктів, підкреслюють його унікальність.

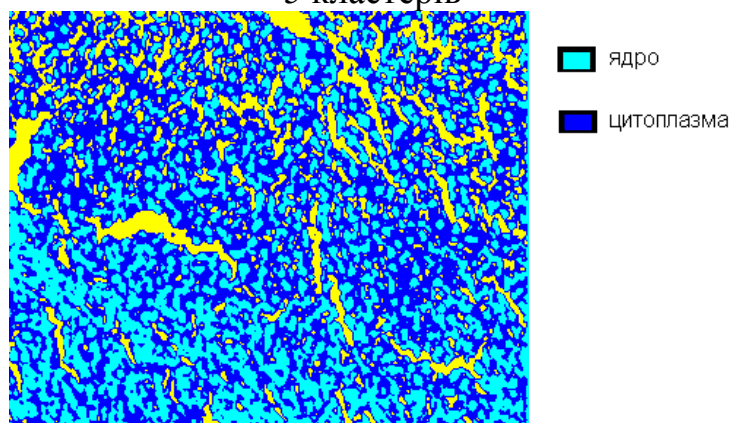
Ознака – це деяка характеристика предмету дослідження [24]. Ознаки об'єктів вводяться в систему за допомогою давачів-рецепторів. В залежності від

					ДП.КСМ.19113/11.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		41

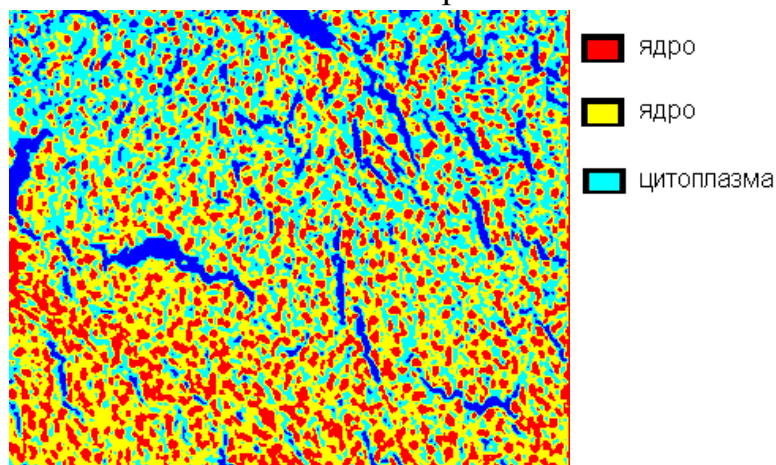
типу датчиків ознаки бувають: акустичні, електромагнітні, хімічні, оптичні, тощо. Як правило ознаки є універсальними, наприклад дані про погоду можна використати в сільському господарстві, рекреаційній та військовій галузях.



5 кластерів



3 кластери



4 кластери

Рисунок 2.2 – Приклад сегментації гістологічного зображення

Ознаки поділяються на об'єктивні та описові [24]. Об'єктивні ознаки можна описати кількісно, як правило в певних числових межах (площа, периметр, центр об'єкта, тощо). Описові ознаки неможна охарактеризувати

					ДП.КСМ.19113/11.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		42

кількісно (враження від об'єкта: гарно, погано, тощо), проте для подальшої машинної обробки їх можна закодувати (гарно=1, погано=2, і т.д.)

Обробка ознак – це виділення, вимірювання та візуалізація тих характеристик сигналу (зображення) які є найбільш інформативними для вирішення конкретної задачі аналізу [15,23,27].

Встановлено, що задача визначення масиву інформативних ознак немає точного об'єктивного рішення, можна лише визначити які із запропонованих людиною ознак є кращими за інші. Таким чином задача визначення масиву інформативних ознак замінюється на задачу відбору підмножини ефективних інформаційних ознак з початкового масиву.

Візуальні інформаційні ознаки розділяють на: метричні, топологічні, аналітичні, статистичні.

Метричні ознаки. Метричні ознаки основані на визначенні відстаней між точками на площині.

– Відстань між точками (найбільш вживані: евклідова, абсолютна, максимальна відстань).

Відстань це кількісна функція $d(x_1, y_1, x_2, y_2)$ двох точок (x_1, y_1) та (x_2, y_2) , що володіє наступними властивостями:

$$\begin{aligned} d(x_1, y_1, x_2, y_2) &\geq 0, \\ d(x_1, y_1, x_2, y_2) &= d(x_2, y_2, x_1, y_1), \\ d(x_1, y_1, x_2, y_2) + d(x_2, y_2, x_3, y_3) &\geq d(x_1, y_1, x_3, y_3). \end{aligned}$$

Найчастіше використовуються метрики:

– евклідова:

$$d_e = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2};$$

– абсолютна:

$$d_M = |x_1 - x_2| + |y_1 - y_2|;$$

– максимальна:

$$d_X = \max(|x_1 - x_2|, |y_1 - y_2|);$$

– Периметр об'єкта P (довжина контуру L), площа об'єкта A та центр об'єкта $O(x_c, y_c)$.

					ДП.КСМ.19113/11.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		43

Периметр об'єкта визначимо як довжину контуру об'єкта, яка обраховується на базі кроку растру при 8-зв'язному кодуванні (зв'язок сусідніх точок контуру можливий по діагоналі):

$$P_k = \sum_{i=1}^N (x_i, y_i).$$

При використанні 8-зв'язного кодування, обрахунок периметру даним методом дає похибку, оскільки відстань між сусідніми точками, що розташовані по діагоналі (тобто контур змінюється на 1 по осі абсцис та ординат одночасно) рівна не 1, а $\sqrt{2}$. Даний підхід підвищує точність обрахунку периметра, при цьому кількість виконаних операцій практично не змінюється.

Якщо контур зображення представлений зо допомогою елементарних векторів, то периметр об'єкта можна обчислити наступним чином:

$$P_k = \sum_{i=1}^N \gamma(n_i).$$

В роботах [16-21] описаний наступний підхід до визначення периметру об'єкта, який базується на евклідовій відстані між двома точками:

$$P_k = \sum_{i=1}^{N-1} \sqrt{(x_i - x_{i+1})^2 + (y_i - y_{i+1})^2}$$

Загальна площа об'єкта визначається як область яку описує контур об'єкта. Площа об'єкта визначається як частина зображення яка належить об'єкту. Під час обрахунку загальної площі враховуються також „дірки” об'єкта. Відповідно якщо об'єкт є суцільним, то загальна площа та просто площа будуть рівні:

$$A_{zag} = \sum_{i=1}^N (x_i, y_i).$$

Центр форми $O(x_c, y_c)$ можна визначити наступним чином:

$$x_c = \frac{M_{10}}{M_{00}}, \quad y_c = \frac{M_{01}}{M_{00}},$$

де M_{00}, M_{01}, M_{10} - моменти нульового та першого порядків.

В деяких випадках центр форми можна визначити наближено:

					ДП.КСМ.19113/11.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		44

$$x_c \approx \frac{(x_{\min} + x_{\max})}{2}, y_c \approx \frac{(y_{\min} + y_{\max})}{2},$$

де $x_{\min}, x_{\max}, y_{\min}, y_{\max}$ відповідно мінімальні та максимальні значення осей абсцис та ординат, що обмежують область зайняту об'єктом.

– Найдовша хорда (велика вісь) D_{\max} . Відрізок, що з'єднує дві найбільш віддалені точки контуру об'єкта (x_i, y_i) та (x_j, y_j) :

$$d[(x_i, y_i), (x_j, y_j)] \rightarrow \max.$$

– Мала вісь D_{\min} . Відрізок що є перпендикуляром до великої осі, а також, прямі, що проведені через кінці великої та малої осі утворюють прямокутник (базовий прямокутник) з найменшою площею, що повністю включає в себе контур.

– Ексцентриситет контуру є відношення великої та малої осей. Перевагою даної характеристики є безрозмірність

$$D_{ek} = \frac{D_{\max}}{D_{\min}}.$$

– Екстремальні точки (рисунок 2.2). Екстремальними точками вважаються крайні точки зображення, тобто значення однієї з координат є максимальним/мінімальним для всього зображення, наприклад $y_2 \rightarrow \max$, $y_4 \rightarrow \min$.

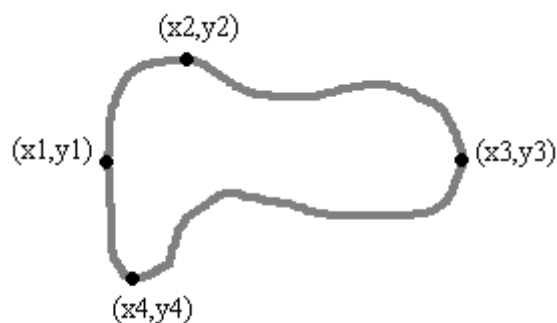


Рисунок 2.2 - Приклад визначення екстремальних (крайніх) точок.

– Кривизна контуру. Контур об'єкта можна охарактеризувати його кривизною $K(s)$ що задається деякою довжиною s , початок відліку – деяка

довільна фіксована точка контуру та швидкістю зміни приросту функції на даному відрізку. $K(s)$ – замкнена періодична функція з періодом, що дорівнює периметру об'єкта.

- Кількість випуклих та ввігнутих сегментів.
- Також важливою є характеристика компактності, що визначається за наступною формулою:

$$T = 4\pi \left(\frac{A}{P^2} \right).$$

Дана характеристика показує відношення площі об'єкта до його периметра. Оскільки великий периметр не завжди гарантує велику площу, то дану характеристику використовують для аналізу форми об'єкта, як показник коло подібності (для кіл вона рівна 1).

Топологічні ознаки. Під топологічними характеристиками розуміють ті характеристики, які залишаються інваріантними відносно топологічних чи гомеоморфних відображень.

- Число зв'язних компонентів об'єкта C . Воно характеризує кількість компонентів, що формують об'єкт, в яких будь-які дві точки можна з'єднати лінією, яка повністю належить компоненті

- Число „дірок” об'єкта H . Дане число характеризує кількість зв'язних компонентів, що не належать об'єкту, але знаходяться всередині об'єкта.

- Число Ейлера E . Характеризує різницю між числом зв'язних компонентів та кількістю „дірок”

$$E = C - H.$$

Топологічні ознаки, не знайшли широкого застосування в задачах обробки зображень. Вони, як правило, використовуються при малому рівні спотворення досліджуваного об'єкта [27].

Аналітичні ознаки. Аналітичні ознаки форми – це математичний опис форми об'єкта для подальшого аналізу. Основними критеріями даних ознак повинна бути простота опису при необхідній достовірності.

- Периметр замкненої області можна представити як сукупність значень кривизни в кожній точці.

					ДП.КСМ.19113/11.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		46

Дану ознаку можна проілюструвати на прикладі неперервної замкнутої кривої представленої в комплексній площині, тоді кожену точку периметру можна представити за допомогою комплексної функції $z(s)$. Комплексну функцію $z(s)$ можна представити за допомогою її дійсної $x(s)$ та уявної $y(s)$ частин:

$$z(s) = x(s) + iy(s).$$

а кривизну функції наступним чином:

$$K(s) = d\Phi(s) / ds,$$

де, $\Phi(s)$ – кут нахилу дотичної, який визначається наступним чином:

$$\Phi(s) = \arctan \left[\frac{dy(s) / ds}{dx(s) / ds} \right].$$

Геометричні моменти. Одновимірні статистичні моменти обраховуються на основі послідовного представлення контуру області або об'єкту. Контур характеризується впорядкованою послідовністю, яка представляє собою Евклідові відстані між центром області і всіма контурними пікселями оцифрованого об'єкту. Моменти безмірні величини, що дозволяє використовувати пряме або комбіноване порівняння.

Момент нульового порядку:

$$M_{00} = \frac{1}{2} \sum_{k=1}^N y_k x_{k-1} + x_k y_{k-1}.$$

Момент першого порядку:

$$M_{10} = \frac{1}{2} \sum_{k=1}^N \left\{ \frac{1}{2} (x_k + x_{k-1}) (y_k x_{k-1} - x_k y_{k-1}) - \frac{1}{6} (y_k - y_{k-1}) (x_k^2 + x_k x_{k-1} x_{k-1}^2) \right\}.$$

Центральний момент:

$$\mu_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} (x - \bar{x})^p (y - \bar{y})^q f(x, y) dx dy,$$

$$\text{де, } \bar{x} = \frac{M_{10}}{M_{00}}, \bar{y} = \frac{M_{01}}{M_{00}};$$

$p, q \in \{0, 1, 2, 3, 4, \dots\}$ - цілі додатні числа.

В загальному випадку двовірні моменти порядку $(p+q)$ зображення представленою неперервною функцією інтенсивності $f(x, y)$, визначаються як:

					ДП.КСМ.19113/11.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		47

$$M_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^p y^q f(x, y) dx dy,$$

де, $p, q \in \{0, 1, 2, 3, 4, \dots\}$ - цілі додатні числа.

В загальному для дискретизованого зображення подвійний інтеграл може бути апроксимований подвійною сумою, звідси момент M_{pq} порядку $p+q$ визначається наступним чином:

$$M_{pq} = \sum_{i=1}^M \sum_{j=1}^N i^p j^q f(i, j),$$

де, M, N – числа, що визначають розмірність матриці інтенсивності $f(i, j)$.

i, j – дискретні позиції пікселів зображення.

А центральні моменти визначаються як:

$$\mu_{pq} = \sum_{i=1}^M \sum_{j=1}^N (i - \bar{i})^p (j - \bar{j})^q f(i, j),$$

$$\text{де, } \bar{i} = \frac{M_{10}}{M_{00}}, \bar{j} = \frac{M_{01}}{M_{00}}$$

M, N – числа, що визначають розмірність матриці інтенсивності $f(i, j)$.

i, j – дискретні позиції пікселів зображення.

Ексцентриситет:

$$elong = \frac{M_{20} + M_{02} + \sqrt{(M_{20} - M_{02})^2 + 4M_{11}^2}}{M_{20} + M_{02} - \sqrt{(M_{20} - M_{02})^2 + 4M_{11}^2}}$$

Дана ознака є інваріантною до просторових перетворень (поворот, масштабування, переносу), що дозволяє успішно використовувати її в процесі розпізнавання.

До недоліків використання моментів, можна віднести їхню неоднозначність та велику кількість арифметичних операцій, для їх обрахунку. На практиці часто зустрічаються ситуації коли два різні об'єкти мають ідентичні моменти низьких порядків, що не унеможливорює проведення процедури розпізнавання. Теоретично збільшуючи порядок моментів можна досягти відмінностей, проте даний підхід є дуже трудомістким, що може негативно позначитись на швидкодії системи в цілому.

						ДП.КСМ.19113/11.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата			48

2.2 Алгоритми аналізу біомедичних зображень

2.2.1 Алгоритми статистичного аналізу

Виходячи з вимог вказаних в п. 1.3, та аналізу існуючих аналогів, розроблено структуру системи текстурного аналізу зображень наведену на рисунку 2.3. Блок попереднього оброблення зображення застосовується для зміни характеристик зображення так, щоб при кількісному його аналізі виділення потрібних об'єктів або фаз, надалі могло б бути проведене найкращим чином.

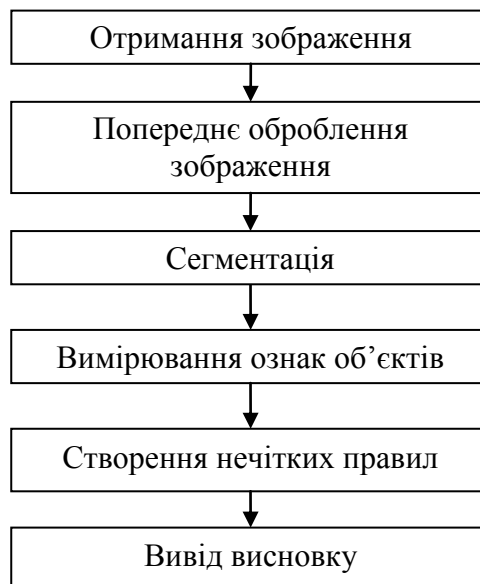


Рисунок 2.3 – Послідовність аналізу зображень

Загальна структурна схема гібридної інтелектуальної системи аналізу біомедичних зображень наведена в додатку Б. Для попереднього оброблення зображень використовуються різні методи фільтрації, які можна умовно розділити на 4 групи:

1. Зміна яскравості і контрастності зображень. При використанні цих методів перетворення змінюються характеристики яскравості зображення, що відбивається на гистограмі яскравості по зображенню.

2. Фільтри вирівнювання, що забезпечують вирівнювання зображень з точки зору яскравості і кольору, що необхідне в тих випадках, коли є дефекти освітлювальної системи або оптики мікроскопа, що виражаються в тому, що фон зображень стає нерівномірним.
3. Фільтри згладжування, що забезпечують очищення зображень від шумів.
4. Фільтри деталізації, що забезпечують перетворення границь, що може виражатися в підсиленні різкості, виділенні границь об'єктів (фаз), деталізації об'єктів і т. п.

Для того, щоб отримувати кількісну інформацію про об'єкти зображення необхідною умовою є їх виділення щодо фону або сторонніх елементів. Виділення, розпізнавання об'єктів зображення - найбільш критичний етап обробки зображення.

Стандартні методи виділення об'єктів включають: виділення по яскравості,

виділення за кольором, виділення по текстурних ознаках, ручне виділення. Найбільш достовірною і точною методикою визначення параметрів служить метод побудови узагальненої текстурної моделі об'єктів на зображенні. Структуру підсистеми отримання досліджуваних параметрів зображено на рисунку 2.4.

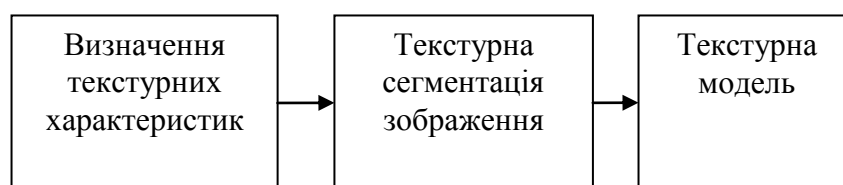


Рисунок 2.4 – Структура підсистеми виділення досліджуваних параметрів

Блок текстурної сегментації призначено для визначення окремих областей. У блоці для визначення текстурних характеристик обчислюють текстурні признаки сукупність яких утворює текстурну модель. На основі текстурної моделі будується поле текстурного признаку. З допомогою

розділення по порозу проводиться сегментація вхідного зображення. Тільки після того, як потрібні об'єкти виділені, стають можливими їх вимірювання.

2.1.2 Визначення текстурних ознак на основі матриці імовірнісного розподілу рівнів яскравості.

Текстурні ознаки обчислюються на основі матриць імовірнісного розподілу яскравості [12], котрі будуються по локальному фрагменту зображення. При послідовній обробці зображення ковзаючим вікном проводиться рекурсивний перерахунок матриці і деяких ознак, і формується поле текстурних ознак.

По матриці імовірнісного розподілу $P(i, j)$ (див. п. 1.2), що описує розподіл яскравості всередині області з центром в точці (x, y) , обчислюються ознаки текстур ($F1...$, $F9$). Після оброблення всього зображення для кожної ознаки формується матриця, що зберігає його значення у всіх оброблюваних точках – поле текстурних ознак. Допоміжні величини: $m_j = \sum_j jP(i, j)$;

$p_j = \sum_j P(i, j)$; математичне сподівання $Mx = \sum_i ip_i$; Нижче наведені

обчислювані текстурні ознаки:

1. Сумарне середнє значення:

$$F_1 = \sum_i m_i p_i$$

2. Інерція:

$$F_2 = \sum_i \sum_j (i - j)^2 P(i, j)$$

3. Другий кутовий момент:

$$F_3 = \sum_i \sum_j P^2(i, j)$$

4. Сумарна кореляція:

$$F_4 = \sum_i \sum_j m_i m_j$$

5. Ентропія:

					ДП.КСМ.19113/11.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		51

$$F_5 = -\sum_i \sum_j \ln(P(i, j))P(i, j)$$

6. Кореляція:

$$F_6 = \sum_i \sum_j (i - Mx)(j - My)P(i, j)$$

7. Затінення:

$$F_7 = \sum_i \sum_j (i + j - 2Mx)^3 P(i, j)$$

8. Контраст:

$$F_8 = \sum_i \sum_j |i - j|P(i, j)$$

9. Сумарна ентропія:

$$F_9 = \sum_i \ln(p_i) p_i$$

Очевидно, що підрахунок матриці імовірнісного розподілу і текстурних ознак, по даному вище визначенню, для кожної точки зображення займе значний час, тому розумно матрицю і деякі ознаки обчислювати рекурсивно. Також відмітимо, що краще не нормувати матрицю; нормування (ділення на загальне число оброблених пар точок всередині вікна) необхідне при обчисленні лише деяких ознак (F_5, F_6, F_7, F_9). Для того, щоб здійснити рекурсивний перерахунок, спочатку необхідно повністю обчислити матрицю P_y^x і текстурні ознаки для першої точки ($x = L, y = L$), де $L = \frac{W-1}{2}$. Вікно по якому обчислюється матриця для наступної точки ($x+1, y$), зсувається вправо, що рівносильно наступному перетворенню:

$$P_y^{x+1}(i, j) = P_y^x(i, j) - \sum_{(m,n) \in D1} f_{i,y}(x_{m,n}; x_{m+d,n+d}) + \sum_{(m,n) \in D2} f_{i,y}(x_{m,n}; x_{m+d,n+d}),$$

де $D1 = \{(n, n) | m \in [-W, x+W], n = y - W\}$,

$D2 = \{(n, n) | m \in [-W, x+W], n = y + W\}$.

Перехід на наступну стрічку зображення здійснюється аналогічно: при цьому зсувається вниз вікно для першої точки попереднього рядка і перераховується матриця для нового положення вікна. Перерахунок ознак

					ДП.КСМ.19113/11.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		52

здійснюється наступним чином: кожного разу, коли до елементу $P(i, j)$ додається (віднімається) одиниця, ознаки текстури перетворюються згідно наведених нижче формул:

$$m_i = m_j \pm j$$

$$p_i = p_i \pm 1$$

$$F_2 = F_2 \pm (i - j)^2$$

$$F_5 = F_5 + \ln\left(\frac{P(i, j)}{N}\right)P(i, j) + \ln\left(\frac{P(i, j) \pm 1}{N}\right)P(i, j) \pm 1$$

$$F_8 = F_8 \pm |i - j|, \quad Mx = Mx \pm \frac{i}{N}, \quad N = 2S^2$$

Якщо $i = j$, то $F_3 = F_3 \pm 4P(i, j) + 4$, у протилежному випадку $F_3 = F_3 \pm 2P(i, j) + 1$. Потім в кожній матриці значень текстурних ознак шукаються максимум і мінімум, і здійснюється приведення до стандартного діапазону градацій сірого кольору 0..255 по формулі:

$$g = 255 \frac{f - \min}{\max - \min},$$

де f елемент матриці, g - значення яскравості, яке йому відповідає, \max і \min - максимальний і мінімальні елементи матриці відповідно.

Після цього матриці текстурних ознак відображаються візуально - формується зображення текстурного поля. Розмір вікна, всередині якого обчислюються текстурні ознаки, впливає на чіткість зображення об'єктів: при збільшенні межі S стають більш розмитими, тому цю величину необхідно вибирати з урахуванням розміру оброблюваного зображення і об'єктів, що виділяються. Параметр d вказує на відстань між точками, яскравості яких враховуються при складанні матриці імовірнісного розподілу. Блок-схема алгоритму обчислення матриць імовірності розподілу рівнів яскравості наведена в додатку Г.

2.2.2 Алгоритми аналізу на основі піраміди зображення

Піраміда зображення – це метод оброблення зображення при якому проводиться його декомпозиція із застосуванням зміни масштабу і фільтрації. При застосуванні орієнтованих фільтрів [23] ми можемо говорити про аналіз на

					ДП.КСМ.19113/11.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		53

основі орієнтованої піраміди зображення. Орієнтована піраміда зображення використовується для зменшення зашумленості, детекції переходів та текстурного синтезу.

Метод аналізу/синтезу текстури на основі піраміди зображення починає роботу з вхідного зображення текстури і зображення шуму (як правило однорідного білого шуму). Розглянемо застосування двох типів пірамід: піраміди Лапласа (радіально симетричне перетворення), і орієнтаційної піраміди (орієнтоване перетворення).

Лінійне перетворення зображення, з допомогою якого утворюється піраміда, представляє зображення як зважену суму базисних функцій. Таким чином, зображення, $I(x; y)$, представлено як суму набору індексованих функцій, $g_i(x, y)$:

$$I(x, y) = \sum_i y_i g_i(x, y)$$

де y_i коефіцієнти перетворення. Ці коефіцієнти обчислюються на основі сигналу, з допомогою проектування на набір *проектуючих* функцій $h_i(x; y)$:

$$y_i = \sum_{x,y} h_i(x, y) I(x, y).$$

Базисні функції перетворення Фур'є синусоїди і косинусоїди різних просторових частот. Проекційні функції для перетворення Фур'є також синусоїди.

У багатьох алгоритмах оброблення зображення, здійснюється його декомпозиція на ряд рівнів, і інформація всередині кожного рівня обробляється більш-менш незалежно від інших. Зображення кожного рівня утворюються з допомогою згортки зображення з банком лінійних фільтрів. Кожна з функцій проектування - перенесена (або зсунута) копія одного з ядер згортки.

Піраміда зображення - специфічний тип перетворення у рівні. Характерна властивість піраміди зображення - це те, що базисні функції і функції проектування це зсунуті і збільшені в масштабі копії одна одної (перенесені і масштабовані у 2^j раз для деякого цілого j). Рівні обчислюються, з допомогою

					ДП.КСМ.19113/11.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		54

згортки і збільшення масштабу. Для кожного наступного значення j , коефіцієнт збільшення масштабу зростає у 2 рази. Це створює множину зображень певного рівня різних розмірів, які відповідають різним смугам частот.

Піраміда Лапласа [24, 25] обчислюється, використовуючи дві основні процедури: зменшення і розширення. Процедура зменшення спочатку застосовує низькочастотний фільтр а потім зменшує зображення у 2 рази в кожному вимірі. Процедура розширення спочатку збільшує зображення у 2 рази (доповнює нулями проміжки між пікселями) і потім застосовує той же низькочастотний фільтр. Звичайно використовується ядро фільтра (окремо до рядків і стовпців) такого вигляду: $\frac{1}{16}$ (1; 4; 6; 4; 1).

Один повний рівень піраміди складається із двох зображень, l_0 і b_0 , які обчислюються в такий спосіб:

$l_0 = \text{зменшення}(\text{im})$

$b_0 = \text{im} - \text{розширення}(l_0)$,

де im - оригінальне вхідне зображення. Воно може бути легко відновлене з l_0 і b_0 :

$\text{відновлене}(\text{im}) = b_0 + \text{розширення}(l_0)$.

Наступний рівень піраміди створюється, застосовуючи той же самий набір операцій до зображення l_0 , що приводить до появи двох нових зображень, l_1 і b_1 . Повна піраміда утворюється, послідовно розділяючи зображення l^i на два нових зображення, l_{i+1} і b_{i+1} . Поєднання ефектів рекурсивної низькочастотної фільтрації і операцій зменшення і збільшення зображення утворює, перетворення піддіапазонів з базисними функціями Гаусіан. Інакше кажучи, перетворення представляє зображення як суму зсунутих, масштабованих функцій Гауса. Проекційні функції цього перетворення функції Лапласа від Гауса. В кінці ми одержуємо множину рівнів піраміди, зображення мають різні розміри через операції збільшення масштабу; менші зображення відповідають більш низьким смугам просторових частот. Оригінальне зображення може

					ДП.КСМ.19113/11.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		55

завжди можна відновити з допомогою представлення піраміди зображення, виконавши операції в зворотному порядку.

Орієнтована піраміда. Текстури, які мають орієнтовані або подовжені структури (узори), не фіксується при аналізі піраміди Лапласа, оскільки базисні функції радіально симетричні. Для того щоб синтезувати анізотропні текстури, застосовуємо орієнтовану піраміду. Як в піраміда Лапласа, під час цього перетворення зображення розкладається на кілька смуг просторових частот. Крім того, кожна смуга частоти ділиться на ряд смуг орієнтації. На рисунку 2.3 показано схему перетворення зображення при аналізі на основі орієнтованої піраміди. Ліва частина схеми представляє аналіз, а права – синтез. Кола посередині позначають аналізовані рівні зображення. Перетворення починається із застосування низькочастотного фільтру L з радіально симетричною частотною характеристикою; смуга високої частоти H відповідає чотирьом кутам просторової частоти. Кожен наступний рівень піраміди утворюються при розкладанні попереднього застосуванням банку смугових фільтрів B_j і фільтра нижніх частот L_{i+1} .

Декомпозиція орієнтації на кожному рівні піраміди "керована" [16], тобто, реакція фільтра, настроєного до будь-яку орієнтацію може бути отримана через лінійну комбінацію реакцій чотирьох базисних фільтрів B , обчислених у тій же області. Властивість орієнтації структур важлива, оскільки це зумовлює подання локальну інваріантність до повороту.

Орієнтована піраміда, на відміну від дискретного вейвлет перетворення яке використовується в алгоритмах стиснення зображення, не ортогональна і надлишкова; число пікселів у піраміді набагато більше ніж, число пікселів у вхідному зображенні. Орієнтована піраміда двонаправлена: фільтри для синтезу - ті ж самі як і для аналізу діаграми. Це дозволяє ефективно виконувати реконструкцію зображення незважаючи на неортогональність. Хоча ядра фільтрів орієнтованої піраміди неподільні, будь-який неподільний фільтр може бути апроксимований сумою декількох подільних ядер фільтра [22].

					ДП.КСМ.19113/11.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		56

Використовуючи ці апроксимовані подільні фільтри можна далі збільшити обчислювальну ефективність.

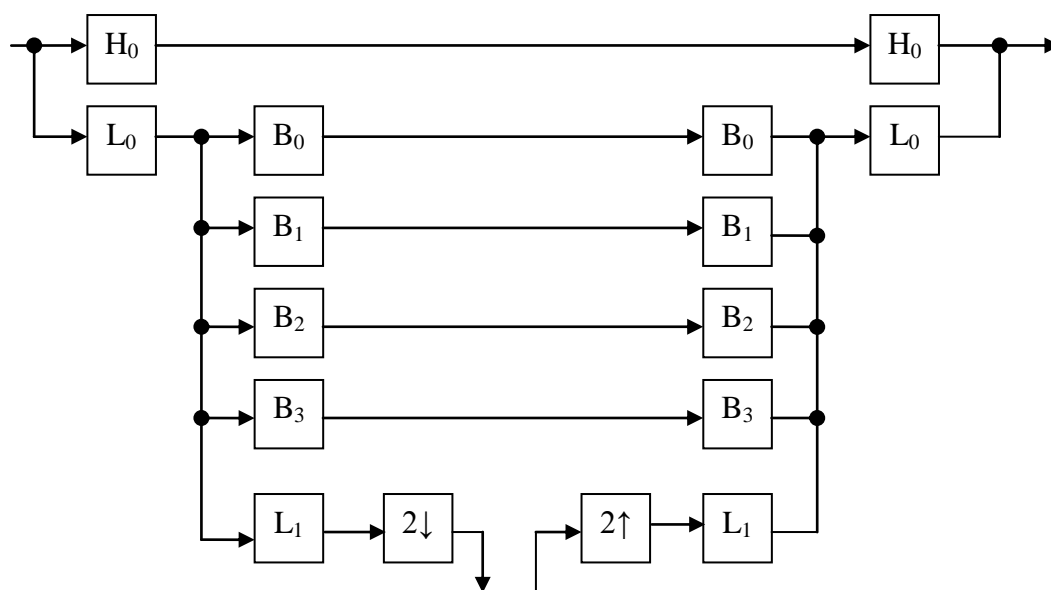


Рисунок 2.5 - Схема перетворень зображення для орієнтованої піраміди

2.2.3 Алгоритми текстурної сегментації

Задачі автоматичної сегментації зображення діляться на два класи:

- виділення областей зображення з відомими властивостями;
- розбиття зображення на однорідні області.

У першому випадку задача сегментації полягає в пошуку певних областей, про які є апіорна інформація (наприклад, ми знаємо колір, форму областей, або області, що нас цікавлять, є зображеннями відомого об'єкту). Сегментація в такій постановці використовується в основному в задачах машинного зору (аналіз сцен, пошук об'єктів на зображенні). У другому випадку ніяка апіорна інформація про властивості областей не використовується, зате на саме розбиття зображення накладаються деякі умови (наприклад, всі області повинні бути однорідні за кольором і текстурою). Такі алгоритми застосовується на початковому етапі рішення задачі, для того, щоб отримати представлення зображення в зручнішому вигляді для подальшої роботи.

Спочатку розглянемо постановку завдання сегментації, як розбиття зображення на однорідні області. Така постановка виникла раніше, ніж завдання виділення областей зображення з відомими властивостями, і методи цієї групи на даний момент добре розроблені. Потрібно зазначити, що далеко не завжди для зображення існує єдина "правильна" сегментація, і далеко не завжди завдання сегментації має єдине рішення. З тієї ж причини немає і об'єктивного критерію оцінки якості розбиття зображення.

Оскільки сегментація зазвичай використовується не самостійно, а як частина деякої системи (наприклад, системи машинного зору), то з практичної точки зору, якість роботи методу оцінюється виходячи з роботи системи в цілому. Тому один і той же метод сегментації може виявитися хорошим для одного завдання і поганим для іншого.

Для грубої оцінки якості методу в конкретному завданні зазвичай фіксують декілька властивостей, якими повинна володіти хороша сегментація. Якість роботи методу оцінюється залежно від того, наскільки отримана сегментація володіє цими властивостями. Найчастіше використовуються наступні властивості [18]:

- однорідність регіонів (однорідність кольору або текстури);
- несхожість сусідніх регіонів;
- гладкість межі регіону;
- маленька кількість дрібних «дірок» усередині регіону;
- і так далі.

Різні методи сегментації орієнтовані на різні властивості розбиття. Тому при виборі методу сегментації для вирішення конкретного завдання, слід визначитися, які властивості розбиття дійсно важливі. У деяких прикладних завданнях достатньо того, щоб розбиття володіло лише першим двома з перерахованих властивостей. Для сегментації зображення існує безліч алгоритмів: кластеризація колірному простору, вирощування областей, дроблення-злиття областей, методи засновані на операторах виділення країв,

					ДП.КСМ.19113/11.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		58

методи теорії графів та ін. Для сегментації з використанням матриць імовірнісного розподілу яскравості поле текстурних ознак розділяється по порогу на окремі області (об'єкти). Пропонований в даній роботі метод кластеризації ознак складається з двох етапів:

грубого розбиття характеристик зображення на свідомо надмірне (заздалегідь невідоме) число класів однорідності та злиття близьких в сенсі вибраної метрики малих класів, поки не буде перевищений заданий поріг злиття.

На етапі 1 використовується алгоритм, подібний методу К-середніх з невідомим числом класів:

- 1) створюється перший кластер, який ініціалізується гістограмою, відповідною деякому вузлу зображення;
- 2) для кожного вузла зображення:
 - а) знаходимо відстані між гістограмою характеристик даного вузла і всіма наявними кластерами;
 - б) вибираємо мінімальну відстань і порівнюємо її із заданим порогом близькості: якщо вона менше, об'єднуємо гістограми, інакше створюємо новий кластер і ініціалізуємо його гістограмою даної точки.

Результат роботи етапу 1 - розбиття характеристик зображення на деяку кількість однорідних в сенсі вибраної метрики і порогу близькості класів. Вибір порогу обумовлений двома чинниками. Дуже низька величина приводить до утворення надмірно великого числа близьких класів, що уповільнює як роботу самого алгоритму, так і наступний етапу злиття. Дуже велика величина приводить до об'єднання в один кластер істотно різних областей. Оскільки така помилка не може бути виправлена на етапі злиття, треба схилитися до вибору меншого порогу. Його чисельне значення вибирається емпірично.

Крім заниженої величини порогу близькості причиною надмірної кластеризації можуть стати локальні неоднорідності зображення, що породжують близькі (в порівнянні з областями різної текстури) класи. Для

					ДП.КСМ.19113/11.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		59

боротьби з цим явищем застосовується процедура злиття класів, в результаті якої можна оцінити кількість різних текстур в початковому зображенні:

- 1) для кожної пари класів обчислюємо значення коефіцієнта грубості злиття $MI = pD$ де p – кількість точок в меншому класі, D відстань між класами;
- 2) вибираємо пару класів з мінімальним значенням MI обчислюваний відношення $MIR = \frac{MI_{cur}}{MI_{max}}$, де MI_{cur} – значення коефіцієнта грубості на поточному кроці, MI_{max} – максимальне значення MI на попередніх етапах;
- 3) об'єднуємо цю пару, якщо значення $MIR < Y$, інакше вважаємо, що отримане оптимальне розбиття, і припиняємо роботу. Сегментація зображення отримується шляхом класифікації всіх його вузлів згідно отриманого набору класів однорідності.

Блок-схема алгоритму текстурної сегментації наведена у додатку Д.

2.3 Алгоритм побудови нечітких правил

Створено базу правил для нечіткої системи із двома входами й одним виходом. Навчальні дані представлено у вигляді множини пар

$$(x_1(i), x_2(i); d(i)), i = 1, 2, \dots, n$$

де $x_1(i), x_2(i)$ - сигнали, що подається на вхід модуля нечіткого керування, $d(i)$ - очікуване (еталонне) значення вихідного сигналу.

Задача полягає у формуванні таких нечітких правил, щоб сконструйований на їхній основі модуль керування при одержанні вхідних сигналів генерував коректні (ті які мають найменшу погрішність) вихідні сигнали.

Крок 1. Поділ просторів вхідних і вихідних сигналів на області.

					ДП.КСМ.19113/11.00.00.000.ПЗ	Арк.
						60
Змн.	Арк.	№ докум.	Підпис	Дата		

Уявімо, що нам відомо мінімальне й максимальне значення кожного сигналу. По них можна визначити інтервали, у яких перебувають припустимі значення. Наприклад, для вхідного сигналу x_1 такий інтервал позначимо $[x_1^-, x_1^+]$. Якщо значення x_1^- й x_1^+ невідомі, то можна скористатися навчальними даними й вибрати з них відповідно мінімальне й максимальне значення

$$x_1^- = \min(x_1), \quad x_1^+ = \max(x_1).$$

Аналогічно для сигналу x_2 визначимо інтервал $[x_2^-, x_2^+]$, а для еталонного сигналу d - інтервал $[d^-, d^+]$.

Кожен визначений у такий спосіб інтервал розділимо на $(2N+1)$ областей (відрізків), причому значення N для кожного сигналу підбирається індивідуально, а відрізки можуть мати однакову або різну довжину. Окремі області позначимо в такий спосіб: M_N (Малий N), ..., M_1 (Малий 1), S (Середній), D_1 (Великий 1), ..., D_N (Великий N) і для кожного з них визначимо одну функцію належності. На рисунку 2.6 представлений приклад такого поділу, де область визначення сигналу x_1 розбита на п'ять підобластей ($N=2$), сигналу x_2 - на сім підобластей ($N=3$), тоді як область визначення вихідного сигналу y - на п'ять підобластей ($N=2$).

Кожна функція належності має трикутну форму; одна з вершин розташовується в центрі області і їй відповідає значення функції, рівне 1. Дві інших вершини лежать у центрах сусідніх областей, їм відповідають значення функції, рівні 0. Очевидно, що такий поділ обраний для прикладу. Можна запропонувати багато інших способів поділу вхідного й вихідного простору на окремі області й використовувати інші форми функцій належності.

Крок 2. Побудова нечітких правил на основі навчальних даних.

Спочатку визначимо ступені належності навчальних даних $(x_1(i), x_2(i), d(i))$ до кожної області, виділеної на кроці 1. Ці ступені будуть виражатися значеннями функцій належності відповідних нечітких множин для кожної групи даних. Наприклад, для випадку, показаного на рисунку 3.37, ступінь належності даного $x_1(1)$ до області D_1 становить 0,8, до області D_2 - 0,2, а до інших областей - 0. Аналогічно для даного $x_2(2)$ ступінь належності до області

					ДП.КСМ.19113/11.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		61

S становить 1, а до інших областей - 0. Тепер зіставимо навчальні дані $x_1(i)$, $x_2(i)$ і $d(i)$ областям, у яких вони мають максимальні ступені належності. Помітимо, що $x_1(1)$ має найбільший ступінь належності до області D_1 , а $x_2(2)$ - до області S .

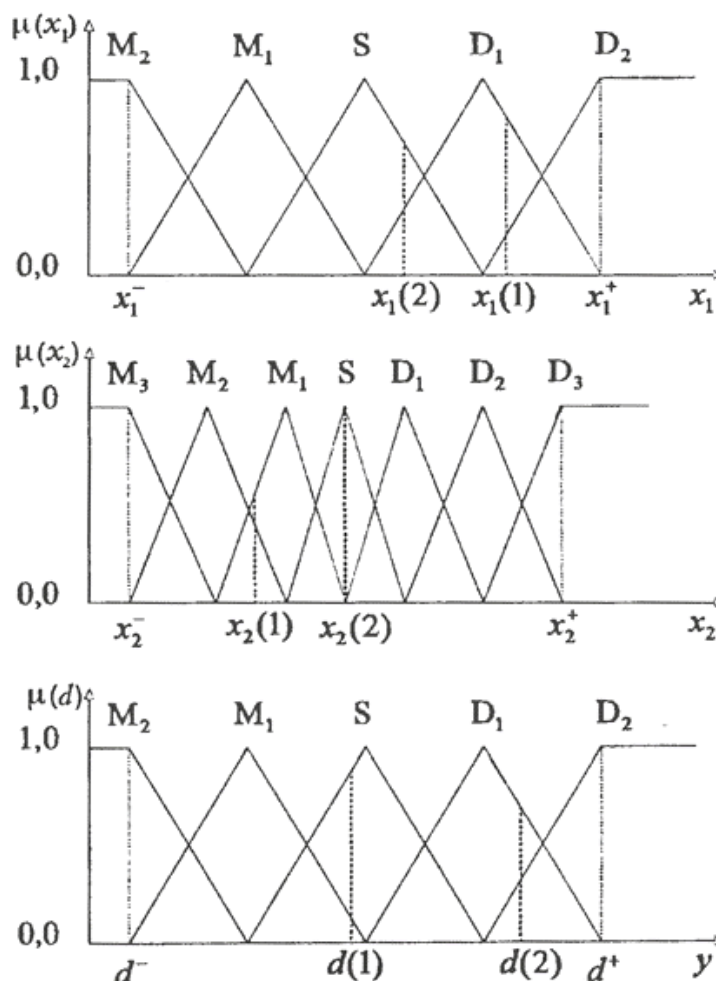


Рисунок 2.6 - Поділ просторів вхідних і вихідних сигналів на області й відповідні їм функції належності.

Остаточно для кожної пари навчальних даних можна записати одне правило, тобто

$$\begin{aligned}
 & (x_1(1), x_2(2); d(1)) \Rightarrow \\
 & \{x_1(1)[\text{max} : 0,8 \text{ в } D_1], x_2(1)[\text{max} : 0,6 \text{ в } M_1]; d(1)[\text{max} : 0,9 \text{ в } S]\} \Rightarrow \\
 & R^{(1)}: \mathbf{IF}(x_1 \text{ це } D_1 \mathbf{AND} x_2 \text{ це } M_1) \mathbf{THEN} y \text{ це } S; \\
 & (x_1(2), x_2(2); d(2)) \Rightarrow \\
 & \{x_1(2)[\text{max} : 0,7 \text{ в } S], x_2(2)[\text{max} : 1,0 \text{ в } S]; d(1)[\text{max} : 0,7 \text{ в } D_1]\} \Rightarrow
 \end{aligned}$$

$$R^{(2)}: \text{IF}(x_1 \text{ це } S \text{ AND } x_2 \text{ це } S) \text{ THEN } y \text{ це } D_1.$$

Крок 3. Приписування кожному правилу ступені істинності.

Як правило, у наявності є велика кількість пар навчальних даних, по кожній з них може бути сформульовано одне правило, тому існує висока ймовірність того, що деякі із цих правил виявляться суперечливими. Це відноситься до правил з однієї й тою же посилкою (умовою), але з різними наслідками (виводами). Один з методів розв'язку цієї проблеми полягає в приписуванні кожному правилу так званої ступеня істинності з наступним вибором із суперечних один одному правил того, у якого цей ступінь виявиться найбільшою. Таким чином, не тільки вирішується проблема суперечливих правил, але й значно зменшується їхня загальна кількість. Для правила виду

$$R: \text{IF}(x_1 \text{ це } A_1 \text{ AND } x_2 \text{ це } A_2) \text{ THEN } (y \text{ це } B)$$

ступінь істинності, позначувана як $SP(R)$, визначається як

$$SP(R) = \mu_{A_1}(x_1) \cdot \mu_{A_2}(x_2) \cdot \mu_B(y).$$

Таким чином, перше правило ($R^{(1)}$) з нашого прикладу має ступінь істинності

$$SP(R^{(1)}) = \mu_{D_1}(x_1) \cdot \mu_{M_1}(x_2) \cdot \mu_S(y) = 0,8 * 0,6 * 0,9 = 0,432,$$

а друге правило -

$$SP(R^{(2)}) = \mu_S(x_1) \cdot \mu_S(x_2) \cdot \mu_{D_1}(y) = 0,7 * 1,0 * 0,7 = 0,49.$$

Крок 4. Створення бази нечітких правил.

Спосіб побудови бази нечітких правил представлений на рисунку 2.8. Ця база представляється таблицею, яка заповнюється нечіткими правилами в такий спосіб: якщо правило має вигляд

$$R^{(1)}: \text{IF}(x_1 \text{ це } D_1 \text{ AND } x_2 \text{ це } M_1) \text{ THEN } y \text{ це } S,$$

то на перетині рядка D_1 (відповідної до сигналу x_1) і стовпця M_1 (сигнал x_2) вписуємо назву нечіткої множини, присутню в наслідку, тобто S (відповідного до вихідного сигналу y). Якщо є кілька нечітких правил з однією і тією ж посилкою, то з них вибирається те, яке має найвищий ступінь істинності.

Крок 5. Дефузіфікація.

					ДП.КСМ.19113/11.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		63

Задача полягає у визначенні за допомогою бази правил відображення $f: (x_1, x_2) \rightarrow \bar{y}$, де \bar{y} - вихідна величина нечіткої системи.

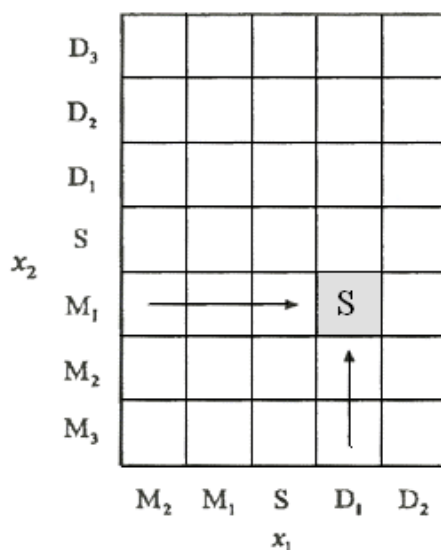


Рисунок 2.8 - Форма бази нечітких правил

При визначенні кількісного значення керуючого впливу \bar{y} для даних, вхідних сигналів (x_1, x_2) необхідно виконувати операцію дефузіфікації. Спочатку для вхідних сигналів (x_1, x_2) з використанням операції добутку об'єднаємо послідовно умови k -го нечіткого правила. Таким чином, визначається так звана ступінь активності k -го правила. Її значення розраховується по формулі $\tau^{(k)} = \mu_{A_1^{(k)}}(x_1) \mu_{A_2^{(k)}}(x_2)$. Наприклад, для першого правила $R^{(1)}$ ступінь активності визначається виразом $\tau^{(1)} = \mu_{D_1}(x_1) \mu_{M_1}(x_2)$.

Для розрахунків вихідного значення \bar{y} скористаємося способом дефузіфікації по середньому центру

$$\bar{y} = \frac{\sum_{k=1}^N \tau^{(k)} \bar{y}^{(k)}}{\sum_{k=1}^N \tau^{(k)}}$$

Розглянутий метод легко можна узагальнити на випадок нечіткої системи з будь-яким числом входів і виходів. Алгоритм побудови бази правил у вигляді блок-схеми представлений в додатку В.

3. ПРОГРАМНА РЕАЛІЗАЦІЯ ГІБРИДНОЇ ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ

3.1 Розроблення архітектури гібридної інтелектуальної системи

Розроблено структуру гібридної інтелектуальної системи (див. додаток Б), що складається з 10 компонентів: блоку інтерфейсу користувача, блоку попереднього оброблення зображення, блоку визначення текстурних ознак, блоку текстурної сегментації, блоку виділення мікрооб'єктів, блоку формування таблиць числових характеристик, блоку побудови бази правил на сонові чисельних даних, блоку виводу висновку на основі правил та блоку зберігання опису зображення у БД.

Система користувацького інтерфейсу забезпечує взаємодію між експертною системою і користувачем. Ця взаємодія зазвичай включає декілька функцій:

1. Обробка даних, отриманих з клавіатури, і вивід вхідних і вихідних даних на екран.
2. Підтримка діалогу між користувачем і системою.
3. Розпізнавання ситуації нерозуміння між користувачем і системою.
4. Забезпечення "дружності" по відношенню до користувача.

Для забезпечення найбільш зрозумілої роботи з програмою розроблена ієрархічна структура меню. Система інтерфейсу з користувачем повинна ефективно обробляти введення і вивід. Для цього необхідно обробляти дані, що вводяться і виводяться, швидко, в ясній і виразній формі. Необхідно також включити можливість роботи з додатковими засобами такими, як магнітні диски і додаткові файли даних.

Блок попереднього оброблення зображення призначений для максимального наближення досліджуваного зображення до еталонного або нормованого. Найчастіше для медичних зображень це просторово-інваріантні операції, зсув, зміна яскравості, зміна контрасту, квантування і геометричні перетворення (зміна масштабу, поворот осі). Теорія цих перетворень добре

					ДП.КСМ.19113/11.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		65

розроблена і як правило, не викликає труднощів при використуванні сучасних комп'ютерів.

Блок визначення текстурних ознак та блок текстурної сегментації призначені для сегментації вхідного зображення. Процес текстурної сегментації можна розділити на 3 етапи:

1) формування з вихідного зображення набору особливостей (характерних ознак), що утворюють багатовимірний простір ознак зображення, кожен вектор якого характеризує особливості певної текстури;

2) етап класифікації, на якому кожна точка зображення співвідноситься з певним класом на основі оцінки вектора ознак відповідного цієї точці в просторі ознак;

3) етап сегментації вихідного зображення ґрунтується на інформації отриманої після класифікації.

Ознака визначається як функція від значень, що містяться в одному або більше пікселях, і обчислюється так, що чисельно виражає деяку значиму характеристику об'єкту. Загальні ознаки (колір, текстура, форма) поділяють на ознаки піксельного рівня (колір, координата), локальні ознаки (обчислюються у певній області або вікні), глобальні ознаки (обчислюються для всього зображення, наприклад, гістограма, дисперсія).

Блок виділення мікрооб'єктів призначений для виділення контурів мікрооб'єктів та обчислення їх ознак.

Блок формування таблиць числових характеристик призначений для збереження значень ознак виділених мікрооб'єктів.

Блок побудови бази правил на основі чисельних даних призначений для побудови нечітких правил, що містять ознак в якості фактів та тип злоякісних клітин в якості висновків.

Блок виводу висновку на основі правил складається з машини логічного виводу на основі нечітких правил. У ЕС прийнято представляти процес логічного виводу у вигляді схеми, яка називається деревом виводу. В процесі консультації ЕС будує дерево виводу і зберігає його в пам'яті в деякій внутрішній формі. Успішному застосуванню правила відповідає додавання

					ДП.КСМ.19113/11.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		66

вузла з його ім'ям, нащадками якого є вузли, відповідні деяким з вже виведених фактів, а предком - новий вузол, відповідний факту, що міститься в правій частині правила.

Одним з важливих питань, що виникають при проектуванні керуючої компоненти систем, заснованих на знаннях, є вибір методу пошуку рішення, тобто стратегії виводу. Від вибраного методу пошуку залежатиме порядок застосування і спрацьовування правил. Процедура вибору зводиться до визначення напряму пошуку і способу його здійснення. Процедури, що реалізують пошук, зазвичай вбудовані в механізм виводу, тому в більшості систем інженери знань не мають до них доступу і, отже, не можуть в них нічого змінювати по своєму бажанню.

При розробці стратегії управління виводом необхідно відповісти на два питання:

1. Яку точку в просторі станів прийняти як початкову? Річ у тому, що ще до початку пошуку рішення система, заснована на знаннях, повинна певним чином вибрати початкову точку пошуку - в прямому або зворотному напрямі.

2. Як підвищити ефективність пошуку рішення? Щоб добитися підвищення ефективності пошуку рішення, необхідно знайти евристики рішення конфліктів, пов'язаних з існуванням декількох можливих шляхів для продовження пошуку в просторі станів, оскільки потрібно відкинути ті з них, які свідомо не ведуть до шуканого рішення.

Механізм логічного виводу включає в себе два компоненти - один з них реалізує власне висновок, інший керує цим процесом. Компонент виводу виконує першу задачу, розглядаючи наявні правила та факти з робочого множини і додаючи до нього нові факти при спрацьовуванні якого - небудь правила. Керуючий компонент визначає порядок застосування правил.

Керуючий компонент визначає порядок застосування правил, а також встановлює, чи є ще факти, які можуть бути змінені у разі продовження консультації. Компонент, що управляє, виконує чотири функції:

1. Зіставлення - зразок правила зіставляється з наявними фактами;

					ДП.КСМ.19113/11.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		67

2. Вибір - якщо в конкретній ситуації можуть бути застосовані відразу кілька правил, то з них вибирається одне, яке найбільше підходить до заданого критерію (розв'язування конфлікту).

3. Спрацювання - якщо зразок правила при зіставленні збігся з будь-якими фактами з робочої множини, то правило спрацює.

4. Дія - робоча множина піддається зміні шляхом додавання в нього висновку правила, що спрацювало. Якщо в правій частині правила міститься вказівка на яку-небудь дію, то воно виконується (як, наприклад, в системах забезпечення безпеки інформації).

Інтерпретатор правил працює циклічно. У кожному циклі він проглядає всі правила, щоб виявити серед них ті посилки, які збігаються з відомими на даний момент фактами з робочої множини. Інтерпретатор визначає також порядок застосування правил. Після вибору правило спрацює, його висновок заноситься в робочу множину, і потім цикл повторюється спочатку.

Розроблено продукційні правила діагностування на основі ознак мікрооб'єктів:

ЯКЩО (1) розміщення_клітин == ізольоване I

(2) розмір_клітин == малий або середній I

(3) форма_клітини == подовжена I

(4) текстура_цитоплазми == однорідна I

(5) ЯЦВ == середнє

ТО тип_клітини = фібробласт.

Розроблено продукційні правила що висувають можливий діагноз базуючись зв'язку тип клітини та можливого діагнозу:

ЯКЩО (1) кількість клітин фібробластів > N I

(2) кількість клітин епітеліальних < M

ТО діагноз = фіброаденома.

Блок зберігання опису зображення у БД призначений для зберігання біомедичних зображень та опису їх вмісту.

Діалог з експертною системою.

					ДП.КСМ.19113/11.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		68

Для того, щоб непідготовлений користувач міг взаємодіяти з системою в неї включено засоби спілкування на природній мові. Переважна більшість систем, заснованих на знаннях, володіють достатньо примітивним інтерфейсом на природній мові – допустимі вхідні повідомлення користувача обмежені набором понять, що містяться в базі знань.

В ході консультації ініціатива діалогу належить системі, а сама консультація у ЕС виглядає так само, як і консультація у експерта-людини: задається ряд питань і на підставі їх аналізу видається експертний висновок. Проте на відміну від бесіди з фахівцем, діалог з ЕС має свої психологічні особливості: більшість користувачів (по цілком зрозумілих причинах, таких, як відсутність досвіду роботи на комп'ютерах, лаконічність діалогу з ЕС, відсутність пояснень в ході консультації і іншим) схильна менше довіряти “думці” ЕС, ніж думці “живого” експерта.

Щоб упевнитися в “розумності” і “компетентності” ЕС, користувач може звернутися до її підсистеми пояснення.

Розглянемо тепер роботу підсистеми пояснення. Для отримання пояснення в процесі консультації користувач може “перехопити ініціативу” діалогу в той момент, коли система ставить чергове питання, “перепитавши” у відповідь систему, чому вона його задала. Таким чином, замість відповіді на питання системи, користувач може поставити їй зустрічне питання:

ЧОМУ?

(“Чому система поставила таке питання?”).

Система інтерпретує питання “Чому?” у “зрозумілих” нею термінах дерева виводу, піднімаючись по ньому на один ярус вище і знаходячи правило, для застосування якого система ставить це питання. Користувачеві видається інформація про це правило, про стан обчислення його умовам про висновок даного правила (поточній меті).

Описана модель пояснення використовується в системі MYCIN. Її перевагою є можливість отримання пояснення будь-якого кроку роботи системи, недоліком - жорстка прив'язка до дерева виводу. Для користувача - непрограміста такий спосіб пояснення може здатися надмірно деталізованим і

					ДП.КСМ.19113/11.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		69

дуже формальним, а потрібно було б пояснення на “вищому рівні”- з погляду здорового глузду і без заглиблення в деталі.

3.2 Програмна реалізація системи

Для проведення аналізу вхідного зображення використано бібліотеку функції програмного засобу ImageJ. До базових можливостей програми можна віднести:

Мультиплатформеність: Оскільки ImageJ написана на Java, те її можна запустити на Linux, Windows (32-bit або 64-bit), Mac OS X .

Відкрита програма: ImageJ і її вихідний Java код вільно доступні. Ні яких ліцензійних обмежень.

Співтовариство користувачів: ImageJ підтримується величезним співтовариством користувачів в усьому світі, які допоможуть розв'язати будь-яке питання. Більш 1700 користувачів і розроблювачів підписані в ImageJ mailing list. І я один з них.

Макроси: Для автоматизації часто повторюваних однакових операцій можна використовувати легко записувані макроси. Що б згенерувати код макросу необхідно тільки скористатися command recorder, при необхідності потім його можна підправити й відтестувати в macro debugger. Більше трьох сотень готових макросів доступно на офіційному сайті програми.

Плагіни: Плагіни дозволяють ще більше підвищити функціональність ImageJ, програма забезпечена власним текстовим редактором для редагування вихідного тексту макросу й Java компілятором. Більш п'ятисот уже готових плагінів доступно на офіційному сайті програми й на деякі інших спеціалізованих сайтах. Плагіни дозволяють одержувати доступ до графічних (і відео) файлів самих екзотичних форматів, виконувати морфологічні операції, згортку, обробку різними фільтрами, вейвлет оброблення, перетворювати в різні формати, працювати з 3-, 4- і т.д. вимірах.

					ДП.КСМ.19113/11.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		70

Інструмент розроблювача: Бібліотека ImageJ дозволяє не тільки розширювати функціональність програми за рахунок написання плагінів, але на її основі можна розробляти аплети, сервлети або самостійні застосування.

Швидкість: ImageJ спростує представлення багатьох про неповороткість програм на Java. Деякі бібліотеки на C++ дозволяють обробляти зображення з такою ефективністю. За заявою на офіційному сайті вона здатна відфільтрувати зображення розміром 2048x2048 за 0.1 seconds - 40 мільйонів пікселів за секунду!

Типи даних: 8-bit grayscale або індексований колір, 16-bit unsigned integer, 32-bit floating-point і RGB колір.

Формати файлів: Форматом за замовчуванням для даної програми є TIFF (без стиснення). У її базовий функціонал закладені можливості відкриття й збереження файлів у форматах GIF, JPEG, BMP, PNG, PGM, FITS, Open DICOM. Підтримка відкриття й збереження в множині інших форматів підтримується за допомогою спеціальних плагінів.

Вікно зображення: Кожне зображення відкривається у власному вікні з можливістю зміни масштабу від 1:32 до 32:1. Якщо зображення не міститься повністю у вікні доступна функція скролінга. Усі функції аналізу працюють із зображеннями представленими в будь-якому масштабі.

Виділення: Реалізовано різні способи виділення ділянок зображення: прямокутне, еліпс, полігон або "вільне" виділення, "чарівна паличка", виділення лінії й точки. Область виділення можна надалі редагувати, інвертувати, зберігати й застосовувати до будь-якого іншого зображення. В області виділення можна проробляти будь-які операції по аналізі зображення, заливанню й очищенню, фільтрації й практично будь-якої іншої функції, доступної для всього зображення.

Поліпшення зображення: У базовій версії підтримується розмиття, підвищення різкості, виявлення границь, медіанний і інші лінійні й нелінійні фільтри, граничний фільтр, як для 8-bit, так і для кольорових RGB зображень. Інтерактивне й автоматичне підстроювання яскравості й контрасту. Можливість

					ДП.КСМ.19113/11.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		71

поділу кольорового зображення на канали, обробка кожного каналу окремо й складання обробленого кольорового зображення.

Геометричні операції: Масштабування, кадрування, довільне обертання, дзеркальне відображення (вертикальне й горизонтальне).

Аналіз: Вимірювання площі, середньої яскравості ділянки або цілого зображення, стандартного відхилення, мінімуму й максимуму. Вимірювання довжин і кутів з автоматичним перетворенням з пікселів у реальні одиниці виміру - міліметри або мікрони, наприклад. Калібрування з використанням стандартних зразків щільності. Гістограми й профілі зображень.

Редагування: Копіювання/вирізання/вставка виділеної частини або всього зображення. Вставка з використанням операторів AND, OR, XOR або режиму "Blend". Додавання тексту, стрілок, прямокутних, овальних і полігональних елементів, а також цілих зображень. У свій час я був уражений багатством математичних операцій, які можна без усяких зусиль зробити із зображеннями.

Робота з кольором: Поділ 32-бітного кольорового зображення на RGB або HSV канали. Об'єднання 8-бітних каналів у кольорове зображення. Конвертування між різними типами даних, переклад RGB зображення в 8-бітне індексоване. Застосування псевдо-кольорових палітр (LUT) до півтонових чорно-білим зображенням.

Стеки: Стек можна представити як стопку зображень, що мають однаковий розмір і формат, розташованих в одному вікні. У вигляді стека в ImageJ можна відкрити .avi відео, при цьому кожний кадр буде представлений у вигляді окремих картинці в стеці. Потім зі стеком можна здійснити самі різні операції, які тільки доступні для окремого зображення, у тому числі й з використанням виділення, при цьому виділити можна той самий ділянка на всіх зображеннях стека, а для виконання операції досить один раз викликати необхідну.

Для реалізації гібридної системи використано наступні функції макромови.

`getResult("Column", row)`. Повертає вимірювання з результуючої таблиці Imagej "Results" або значення NaN якщо обраний стовпець "Column" не

					ДП.КСМ.19113/11.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		72

виявлений у таблиці. Перший аргумент визначає стовпець у таблиці. Він повинен бути присутнім у таблиці вікна "Results" як заголовок стовпця, такий як "Area", "Mean" або "Circ.". Другий аргумент визначає рядок, і розташовується в інтервалі $0 \leq \text{row} < n\text{Results}$. $n\text{Results}$ це змінна, що містить поточну кількість вимірів в таблиці.

`getStatistics(area, mean, min, max, std, histogram)` Повертає площу, середнє значення яскравості, мінімальне й максимальне значення яскравості, стандартне відхилення по шкалі яскравості й гістограму активного зображення або виділення. Гістограма представлена як масив, що містить 256 значень яскравості. Для 8-бітних і RGB зображень, розмір одиниці гістограми = 1. Для 16-бітних і 32-бітних зображень розмір одиниці гістограми розраховується як $(\text{max}-\text{min})/256$.

`imageCalculator(operator, img1, img2)` Запускає інструмент `Process>Image Calculator`, параметр `operator` (приймає наступні значення "add", "subtract", "multiply", "divide", "and", "or", "xor", "min", "max", "average", "difference" or "copy") і визначає операцію, параметри `img1` і `img2` визначають операнди. `img1` і `img2` можуть бути або назвами зображень (рядка), або ідентифікаторами ID (цілочисельні значення). Оператор `operator` рядок, який включає три модифікатори: "create" (наприклад, "add create") формує результат, який зберігається в новому вікні, "32-bit" викликає результат, представлений в 32-бітному форматі а плаваючою точкою й "stack" викликає процес для роботи із цілим стеком.

Модуль попереднього оброблення реалізує алгоритми попереднього оброблення вхідного зображення. Серед його функцій є такі:

`RGBtoHSL (InImage, OutImage)` – функція перекодування вхідного зображення з базису RGB в базис HSL. Параметри функції `InImage` - це об'єкт класу `Tbitmap`, в якому зберігається вхідне зображення. Параметр `OutImage` – область пам'яті, куди буде збережене вхідне зображення в базисі RGB після перекодування в базис HSL;

`zoomimage (TopXYzoom, EndXYzoom)` – процедура масштабування вхідного зображення. Дана функція забезпечує можливість користувачу

					ДП.КСМ.19113/11.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		73

обробляти найбільш інформативні області вхідного зображення. $TopXY_{zoom}$ – координати верхньої лівої точки частини зображення, яке необхідно вивести у робоче вікно програми, $EndXY_{zoom}$ – координати нижньої правої точки частини зображення.

Модуль сегментації зображення. В модулі реалізовано алгоритми сегментації кольорових та бінарних зображень. Функції даного модуля такі:

$segment_threshold(InImage, OutImage: Tbitmap; porig: integer)$ – функція порогової сегментації. Дана функція працює в автоматизованому режимі. *InImage* – це область в якій зберігається вхідне зображення, параметр *OutImage* – область пам'яті, куди буде збережене сегментовано зображення, *Porig* – значення порогу для функції сегментації. Дана функція обробляє зображення, які представлені в градаціях сірого. Результатом роботи даної функції є бінарне зображення, де маркером 0 позначаються пікселі, що належать фону зображення, а 1 – пікселі, що відповідають об'єктам. Параметр порогу може задаватись користувачем (по замовчуванню 90).

Функції автоматизованої сегментації на основі попиксельного виділення, виділення об'єктів на основі контуру, виділення об'єктів на основі характерних точок реалізуються за допомогою комбінації функції середовища програмування ImageJ:

$paint_image(InImage, OutImage: Tbitmap; X, Y :integer; color_image, color: integer; var area: integer)$ реалізує алгоритм зафарбування однорідної області з одного кольору в інший. Вхідні параметри: *InImage* – область пам'яті, в якій зберігається початкове зображення, *OutImage* - область пам'яті, в яку необхідно зберегти вихідне зображення, *X, Y* – координати піксела, що належить однорідній області, *color_image* – початковий колір однорідної області, *color* – колір, в який необхідно перефарбувати однорідну області, *Area* – лічильник, що вказує кількість перефарбованих пікселів. Отримавши на вхід координати піксела, що належить однорідній області, процедура покроково перевіряє всі сусідні пікселі. Якщо сусідній піксел належить однорідній області, то значення його кольору змінюється на нове і рекурсивно запускається функція перевірки всіх сусідніх пікселів. Процедура припиняє свою роботи, якщо всі пікселі, що

належать однорідній області отримали нове значення кольору.

Результатом роботи даного модуля є масив об'єктів, що відповідають клітинам на вхідному зображенні. Інформація про виділені об'єкти зберігається у класі TCell. Окрім основної інформації про виділений об'єкт (площа периметр, масив точок що формують об'єкт) даний клас дозволяє зберігати інформацію про деякі статистичні властивості об'єкта (довжина великої осі, характерні точки).

Параметри класу Cell такі:

– Number – унікальний ключ-ідентифікатор для кожного об'єкта. За допомогою даного ключа відбувається робота з об'єктом;

– Perimetr – в даному параметрі зберігаються інформація про довжину контурної лінії. Периметр об'єкта визначається як сума відстаней між сусідніми точками контуру зображення;

– Contour – масив точок, що відповідають контуру об'єкта;

– A,B,C – характерні точки, визначені для функції афінних перетворень;

– Area – в даному параметрі зберігається інформація про площу об'єкта. (на досліджуваних об'єктах відсутні "дірки").

– Matrix – масив для зберігання координат (x,y) контуру об'єкта. Зберігання інформації про контурну лінію дозволяє проводити додаткові операції, без повторення процесу визначенні контуру. Зокрема, до додаткових операцій можна віднести представлення контурної функції за допомогою елементарних векторів, кодів Фрімена, апроксимацію контурної функції, визначення ознак об'єкта (периметр, компактність, визначення осей, тощо).

– MaxX – точка об'єкта, що максимально віддалена від осі абсцис на площині XOY з початком відліку у верхній лівій точці зображення, тобто $x \rightarrow \max$.

– MinX – точка об'єкта, що мінімально віддалена від осі абсцис на площині XOY з початком відліку у верхній лівій точці зображення, тобто $x \rightarrow \min$.

– MaxY – точка об'єкта, що максимально віддалена від осі ординат на площині XOY з початком відліку у верхній лівій точці зображення, тобто $y \rightarrow \max$.

					ДП.КСМ.19113/11.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		75

– MinY – точка об'єкта, що мінімально віддалена від осі ординат на площині XOY з початком відліку у верхній лівій точці зображення, тобто $y \rightarrow \min$.

– M – інформація про координати точки, в якій міститься середина великої осі. Дана точка є центром описаного кола для даного об'єкта;

– bl – параметр b рівняння $y = kx + b$, що описує велику вісь об'єкта;

– k – параметр k рівняння $y = kx + b$, що описує велику вісь об'єкта.

– LenghtMidLine – довжина великої осі об'єкта.

– Compactness – коефіцієнт, що показує відношення периметра об'єкта до його площі.

– ApproxTop – кількість кривих, за допомогою яких апроксимується верхня частина контуру об'єкта.

– AproxBott – кількість кривих, за допомогою яких апроксимується нижня частина контуру об'єкта.

– PointsTop – масив пікселів верхньої частини контуру об'єкта.

– PointsBott – масив пікселів нижньої частини контуру об'єкта.

– SplinesTop – набір структур, що описують криві, за допомогою яких апроксимується верхня частина контуру об'єкта.

– SplinesBott – набір структур, що описують криві, за допомогою яких апроксимується нижня частина контуру об'єкта.

Методи класу TCell:

– trace_contoure(InImage: Tbitmap; X, Y: integer; Next: integer; color, marker_color, fon_color : integer; var contour: array of TPoint; var perimetr: integer)

- функція проходження контуром на основі запропонованого алгоритму.

Параметр *InImage* – вказує на об'єкт типу *Tbitmap*, в якому знаходиться вхідне зображення, *Next* - вказує напрям згідно розмітаної сітки, який потрібно перевірити на приналежність до контуру об'єкта, *Color* – колір об'єкта,

Marker_color – колір, в який необхідно маркувати контур об'єкта, *Fon_color* –

колір, що відповідає кольору фону зображення, *Contour* – масив типу *TPoint*,

що відповідає масиву точкам, які формують контур об'єкта, *Perimetr* –

довжина периметру об'єкта.

					ДП.КСМ.19113/11.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		76

– CentralLine – метод, який реалізує алгоритм пошуку центральної осі об'єкта.

– Approx (start: integer; secondpoint: integer; tup: integer) – процедура, що реалізує апроксимацію частини контуру кривими n -го порядку. Параметри функції: start – точка початку частини контуру, яку необхідно апроксимувати, secondpoint – точка кінця частини контуру яку необхідно апроксимувати, tup – тип кривої, за допомогою якої необхідно апроксимувати контур. Тип кривої може приймати такі значення – пряма, поліном другого і третього степенів. Функція, отримуючи на вхід масив точок контуру, намагається апроксимувати його кривою заданою в параметрі tup. Після отримання коефіцієнтів апроксимуючої кривої проводиться перевірка на точність сегментації. Поріг точності задається попередньо користувачем. Якщо частина контуру апроксимована із задовільною точністю, то процедура завершує свою роботу, в протилежному випадку проводиться апроксимація контуру за допомогою кривої меншого порядку. Якщо апроксимація кривою найменшого порядку дала негативний результат, то програма відкидає одну крайню точку апроксимованого контуру і знову запускає процедуру апроксимації кривою найбільшого порядку.

– AproxLine(start: integer; finish: integer) – процедура апроксимації частини контуру прямою. Дана процедура є частковим випадком процедури Aprox(), проте реалізує тільки кусково–лінійну апроксимацію.

Модуль виділення контурів та розрахунок їх характеристик. Даний модуль відповідає за визначення контурів об'єктів та визначення їх характеристик (довжина контуру, апроксимація контуру). В даному модулі реалізовано ряд класичних алгоритмів, а також використовуються методи класу TCell. Серед функцій даного модуля є такі:

trace_contour_bug (InImage: Tbitmap; X, Y: integer; Next: integer; color, marker_color, fon_color: integer; var contour: array of TPoint; var perimetr: integer; var step_proxid: integer) – функція, що реалізує алгоритми проходження контуром „жука”. Параметр InImage – вказує на об'єкт типу Tbitmap, в якому знаходиться вхідне зображення, Next - вказує напрям згідно розмітаної сітки,

					ДП.КСМ.19113/11.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		77

який варто перевірити на приналежність до контуру об'єкта, *Color* – колір об'єкта. *Marker_color* – колір, в який потрібно маркувати контур об'єкта, *Fon_color* – колір, що відповідає кольору фону зображення, *Contour* – масив типу *TPoint*, що відповідає масиву точкам, які формують контур об'єкта, *Perimetr* – довжина периметру об'єкта.

trace_contour_neibor (*InImage*: *Tbitmap*; *X*, *Y*: *integer*; *Next*: *integer*; *color*, *marker_color*, *fon_color*: *integer*; *var contour*: array of *TPoint*; *var perimetr*: *integer*; *var step_proxid*: *integer*) – функція, що реалізує алгоритм проходження контуром „Moore-Neighbor Tracing”.

trace_contour_red (*InImage*: *Tbitmap*; *X*, *Y*: *integer*; *Next*: *integer*; *color*, *marker_color*, *fon_color*: *integer*; *var contour*: array of *TPoint*; *var perimetr*: *integer*; *var step_proxid*: *integer*) – функція, що реалізує алгоритм проходження контуром „Redial Sweep”.

trace_contour_teo (*InImage*: *Tbitmap*; *X*, *Y*: *integer*; *Next*: *integer*; *color*, *marker_color*, *fon_color*: *integer*; *var contour*: array of *TPoint*; *var perimetr*: *integer*; *var step_proxid*: *integer*) – функція, що реалізує алгоритм проходження контуром „Theo Pavlidi's Algorithm”.

Даний модуль дозволяє виділити та обчислити характеристики контуру об'єкта. Результатом роботи є масив описів контурів виділених на попередніх етапах роботи системи об'єктів.

Модуль виводу даних на друк. Даний модуль реалізовано за допомогою стандартних класів програмного середовища Borland Delphi 7.0: *PrinterSetupDialog* та *PrintDialog*. Функції класу *PrinterSetupDialog* дозволяють провести попередню діагностику наявності пристрою друку, визначити його тип, перевірити готовність до роботи. *PrintDialog* – даний клас реалізує стандартне вікно Windows для роботи з пристроями друку. Поєднання цих двох класів дозволило реалізувати в системі можливість роботи з усіма пристроями друку, що зареєстровані в системі.

3.3 Тестування гібридної інтелектуальної системи

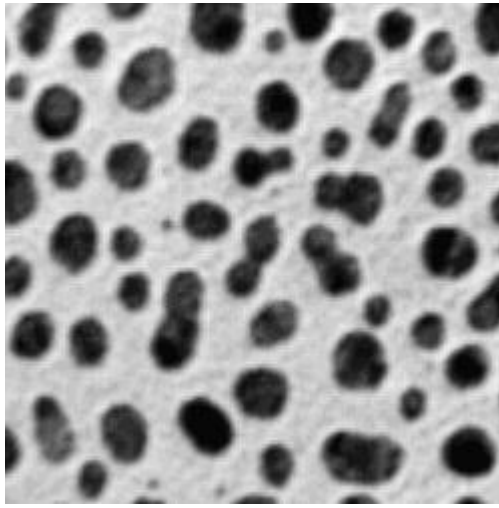
Автоматизувати аналіз зображення за допомогою макросу. Для створення даного плагіна знадобиться встановлена програма ImageJ і доступ в Інтернет. Рекордер – спеціальний редактор, що дозволяє фіксувати всі макрокоманди, які виконуються в цей момент. Для запуску рекордера користувач натискає в Imagej-Plugins-Macros-Record. Порожнє вікно рекордера виглядає в так:



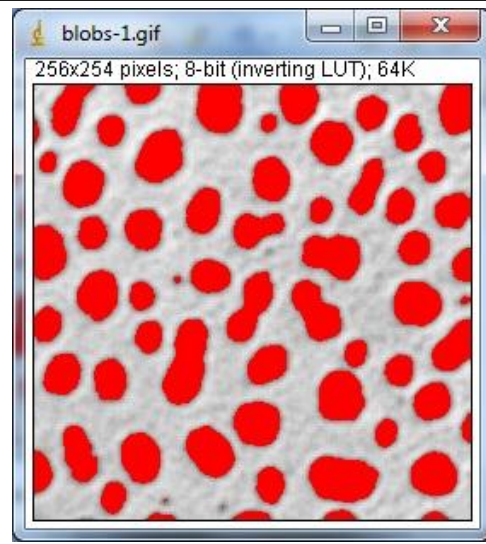
Рисунок 3.1 – Вікно рекордера для запису команд

Для написання плагіна в полі Record користувач змінює Macro на Plugin. Усі команди, які будуть виконуватися в програмі ImageJ, будуть записуватися в рекордері. Наприкінці роботи з макросом користувач збереже його у файл, і при запуску цього файлу наступного разу всі команди будуть виконуватися саме в тій послідовності, яку задали.

Відкриємо цифрове цитологічне зображення в меню File-Open-Open. Користувачеві необхідно автоматизувати обробку зображень як на рисунку 3.2,а. Потрібно порахувати кількість клітин, їх периметр і площу. Для початку виділимо всі клітини, незалежно від розміру й інших параметрів. Це робиться за допомогою порога в меню Image-Adjust-Threshold. Поріг виділив усі клітини (рисунок 3.2,б):



а) Цитологічне зображення



б) порогове виділення клітин

Рисунок 3.2 – Виділення об'єктів на зображенні

Типи порогів бувають різні, для цього плагіна влаштує плагін по “Default”, тобто той, що стоїть за замовчуванням. Користувач встановлює потрібні нам параметри в меню Analyze-Set Measurements зображене на рисунку 3.3.

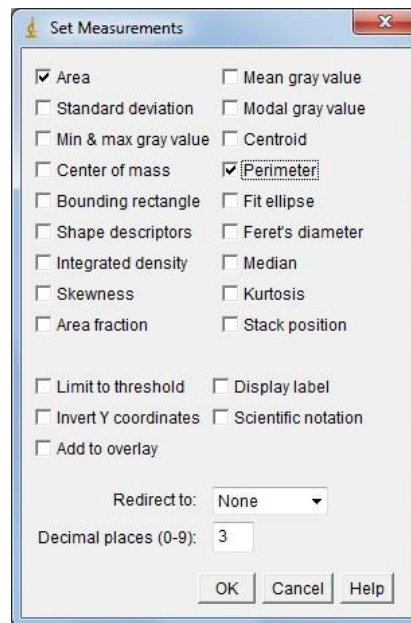


Рисунок 3.3 – Вікно встановлення числових ознак мікрооб'єктів

ImageJ може обчислити множину параметрів, користувачеві потрібно площа і периметр клітин. Кількість порахується автоматично. Вибравши ці параметри, користувач натискає кнопку ОК. У рекордері додається один рядок:

```
run("Set Measurements...", "area perimeter redirect=None decimal=3");
```

					ДП.КСМ.19113/11.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		80

Тепер необхідно вилучити все «сміття», тобто ті дрібні клітини, які є артефактами і їх не потрібно враховувати при аналізі зображень. Не будемо аналізувати клітини, які «склеїлися» один з одним, тому що це буде вносити помилку в аналіз. Також не будемо враховувати клітини, які стикаються із краями картинки, і теж будуть спотворювати загальну картину. Користувач їх також виключить з оброблення. Це робиться в налаштуваннях аналізу часток у меню Analyze-Analyze Particles зображеного на рисунку 3.4.

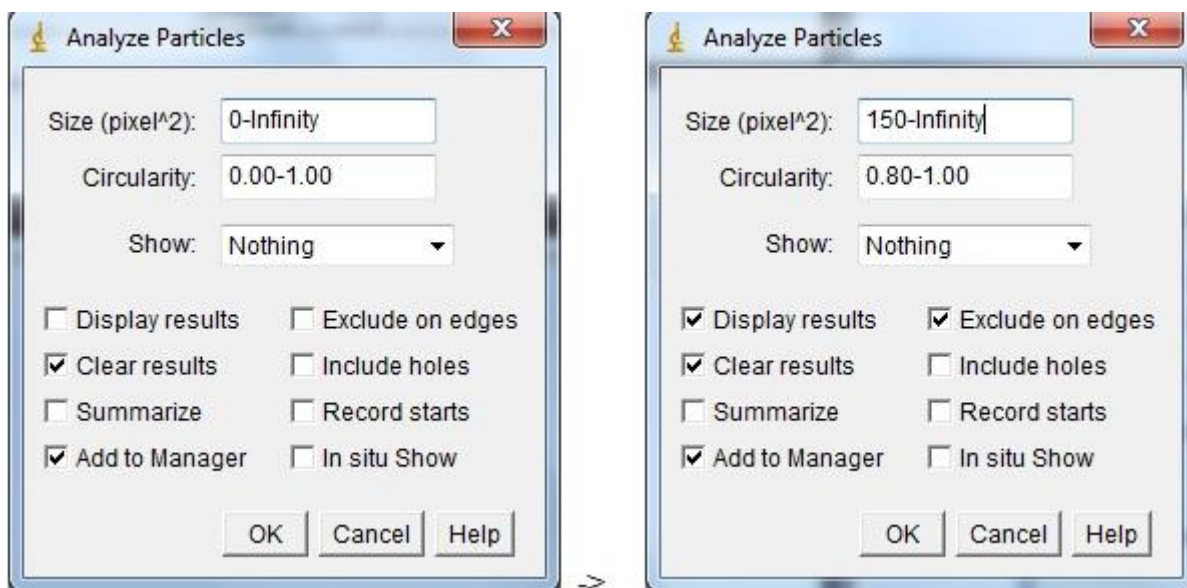


Рисунок 3.4 – Встановлення областей для числових ознак мікрооб’єктів

Параметром Size(pixel) обмежили розмір (дрібні відкинути), параметром Circularity залишили все, що має більш-менш схожу на коло (1.00 – це ідеальне коло), забравши в такий спосіб клітини, що наклались. Параметром Exclude on edges забрали всі клітини на границі поля зору. Параметр Add to Manager залишили для того, щоб візуально побачити, які клітини в нас будуть обраховуватися, які – ні. Галочка в Display results потрібна для відображення вікна результатів. По сегментованих областях на зображенні видно, що порахувалися саме ті клітини, які потрібні. У вікні результатів видно полічені елементи на зображенні, і наведені дані площі й периметра. Командою File-Save As можна зберегти у форматі MS Excel для подальшої статистичної обробки. У рекордері з’явився новий рядок:

					ДП.КСМ.19113/11.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		81

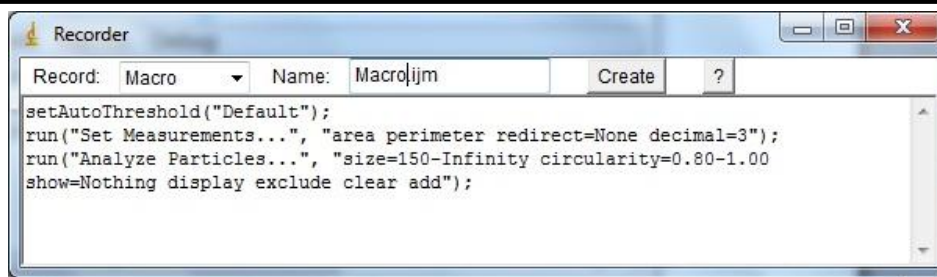


Рисунок 3.5 – Сценарій автоматичного оброблення зображення

Саме ці три рядки й будуть надалі формувати макрос. Якщо користувач натискав ще якісь клавіші або виконував які-небудь команди, то в рекордер запишуться й інші рядки. Щоб з рекордера створити макрос, потрібно дати йому назву, наприклад, analysis.ijm і натиснути клавішу Create. Створений макрос зображений на рисунку 3.6.

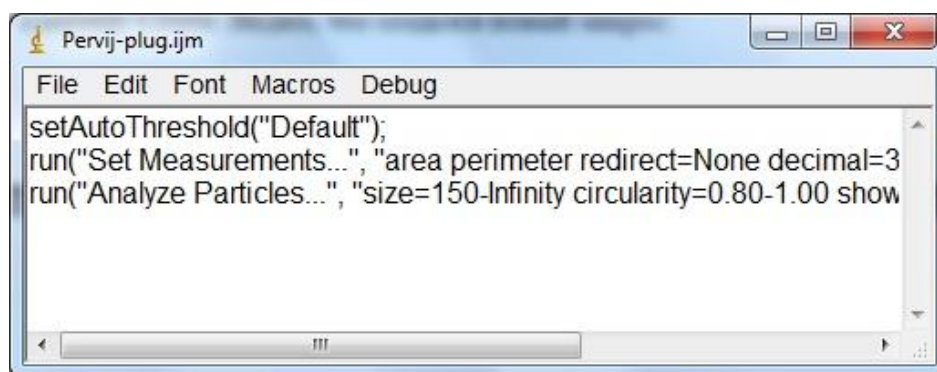


Рисунок 3.6 – Текст плагіну для автоматичного оброблення зображення

Далі користувач зберігає його в будь-якому зручному місці, закриває всі вікна, знову запускає зображення, у головному вікні програми ImageJ натискає Plugins-Macros-Run і вибирає створений макрос. Він виконав обробку точно також, як і вручну, тільки тепер автоматично.

Інтерфейс розробленої системи, зображений на рисунку 3.7, є інтуїтивно зрозумілим будь-якому користувачеві. Головне діалогове вікно програми призначене для відображення та створення сценарію автоматичного оброблення зображення. В правій частині розміщено панель, що відображає наявні в системі алгоритми оброблення. З допомогою кнопки зі стрілкою алгоритми переносять в список, що є відображенням сценарію оброблення.

Кожен алгоритм володіє набором параметрів, що налаштовуються під час навчального режиму роботи системи.

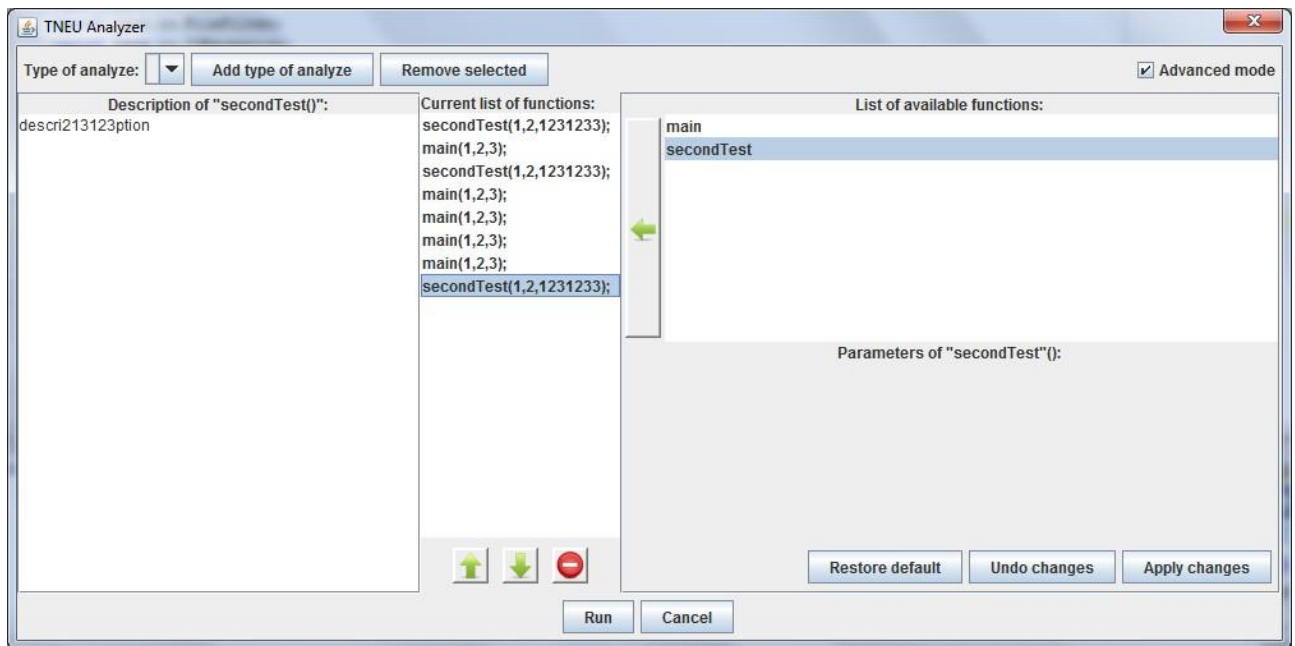


Рисунок 3.7 – Головне вікно програми

В експериментальних цілях було проведено пробні запуски на базі зображень ракових клітин Breast Cancer Dataset. Під час навчання системи створено 45 правил опису типів клітин злочасних процесів. У всіх випадках система коректно визначала тип клітини.

Приклади ознак отриманих під час навчання.

зернистість клітини

до: морфологічні ознаки клітини

вр: морфологічні ознаки клітини

н: зернистість цитоплазми

зернистість ядра

асм: клітини

зернистість цитоплазми

до: морфологічні ознаки клітини

в: зернистість клітини

а: зернистість ядра

асм: цитоплазма

					ДП.КСМ.19113/11.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		83

зернистість ядра

до: морфологічні ознаки клітини

в: зернистість клітини

а: зернистість цитоплазми

асм: ядро клітини

клітини

нв: Nk-Клітини

активовані клітини

базофільні клітини

гістоцити

зрілі клітини

клітини в мітозі

клітини-попередники

лейкоцити

нейтрофіли

периферичні клітини

плазматичні клітини

проліферуючі клітини

клітини, що дозрівають

фібробласти

центробласти

церебріформні клітини

еозинофільні клітини

нч: цитоплазма

ядро клітини

асм: зернистість клітини

форма клітини

колір клітини

Обґрунтування рекомендацій, наведених у роботі, підтверджується практичним використанням результатів у Науково-дослідному інституті інтелектуальних комп'ютерних систем (додаток Е).

					ДП.КСМ.19113/11.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		84

4. ОХОРОНА ПРАЦІ

Основною метою розділу охорони праці є уникнення можливості виробничого травматизму, професійних отруєнь і захворювань, пожеж і вибухів, аварій, забруднення довкілля при будівництві та використанні об'єкта проектування.

Даний дипломний проект передбачає розробку гібридної інтелектуальної системи. В розділі охорона праці проводиться розрахунок безпечних умов праці для приміщення з комп'ютерами.

Обслуговування апаратури виконується в кімнаті розміщення розробленої системи. Обслуговуючий персонал займається обслуговуванням комп'ютерної техніки.

Контроль за роботою здійснюється за допомогою комп'ютерного обладнання, тому ця робота відноситься до категорії легких, тобто робіт, які виконуються в сидячому, стоячому положенні або зв'язані з рухом, але вона не відноситься до систематичної фізичної роботи або до перенесення важких предметів.

Виходячи зі СН 245-71 і ГОСТ 12.1.005-88, а також, беручи до уваги характер робіт, відповідно до яких, площа приміщення на одного працівника в приміщенні дорівнює (6 м^2), приймаємо:

$$S_n = n \cdot S_0, \quad (4.1)$$

де S_0 - площа приміщення, що виводиться на одного працівника;

n - кількість працівників.

Оскільки в приміщенні працює 3 чоловіки, тоді необхідна площа для роботи повинна становити:

$$S_n = 6 \cdot 4 = 24 \text{ м}^2;$$

Реальна площа приміщення становить 48 м^2 , тобто відповідає вимогам санітарних норм.

					ДП.КСМ.19113/11.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		85

Згідно ГОСТ 12.1.005-88 в приміщенні повинні підтримуватися певні метеорологічні умови, що визначаються температурою відносною вологістю повітря, тиском та швидкістю руху повітря. Ці фактори впливають на терморегуляцію, тобто спроможністю організму людини підтримувати нормальну температуру тіла (в межах 36-37 °С).

Тепловіддача від організму може здійснюватись шляхом тепловипромінення, конвекції і випаровування. При підвищеній температурі навколишнього середовища тепловіддача здійснюється лише за рахунок випаровування поту. Перегрівання тіла до 40-41 °С приводить до порушення водно-сольового обміну, виникнення судорожної хвороби і теплового удару з втратою свідомості.

Робота в умовах пониженої температури повітря, особливо при підвищеній вологості і швидкості руху, призводить до переохолодження тіла, що супроводжується виникненням простудних захворювань. Мінусова температура повітря призводить до обморожування, що розглядається, як виробнича травма.

Для робочої зони нашого, приміщення, оптимальні і допустимі значення температури, відносною вологості і швидкості руху повітря, встановлюються з врахуванням трудоемності і складності роботи, яка виконується, а також пори року. Користувачі персональних комп'ютерів належать до групи 1а - легкі роботи.

Відповідно з цим і ГОСТ 12.1.005-88 вибираємо необхідні метеорологічні умови (таблиця 4.1).

Таблиця 4.1 - Оптимальні і допустимі метеоумови

Період року	Категорія робіт	Температура t, °С		Відносна вологість повітря, %		Швидкість повітря	
		Оптимальна	Допустима	Оптимальна	Допустима	Оптимальна	Допустима
Холодний Теплий	Легка 1а Легка 1а	22—24 23-25	21-25 22-28	40—60 40—60	<75 <55	0,1 0,1	<0,1 0,1—0,2

Для підтримання відповідних метеорологічних умов в приміщенні встановлено обладнання системи центрального опалення, але в зимовий період його тепловіддача є недостатньою. Доцільним є проведення ущільнення конструктивів вікон і дверей, щоб припинити втрати тепла.

Решту метеорологічних умов забезпечує обладнання повного кондиціонування повітря. Воно забезпечує постійність температури, вологості, руху і чистоти повітря.

Сприятливі умови роботи забезпечують як високу продуктивність праці, так і позитивно впливають на психологічний стан людини, на її працездатність і здоров'я. Особливо важливе біологічне і гігієнічне значення для людини має природне освітлення, тому при проектуванні виробничих приміщень важливо передбачити наявність природнього освітлення СНиП II-4-79.

Проведемо розрахунок природнього освітлення, згідно зі СНиП II-4-79 «Природне і штучне освітлення. Норми проектування», а при необхідності розрахуємо додаткове штучне освітлення приміщення.

Розрізняють три системи природного освітлення: бокове, верхнє, комбіноване. Для кількісної оцінки виробничого освітлення важливою технічною характеристикою є освітленість робочої поверхні. Густина світлової енергії на площі $E_{(лк)}$ визначається за формулою:

$$E = dF/dS, \quad (4.2)$$

де dF світловий потік, який характеризує потужність світлового випромінювача (лм), рівномірно розподілений по площі dS (m^2).

Коефіцієнт природного освітлення, який являє собою відношення освітленості в даній точці середини приміщення E_v до зовнішнього горизонтального освітлення E_z визначаємо за формулою:

$$I = E_v/E_z. \quad (4.3)$$

Заміри натурного освітлення проводяться люксометром 10116.

Розміри приміщення становлять:

$E_n \cdot B = 6 \cdot 8 = 48 m^2$; висота приміщення $h = 3$ м, S — світловий отвір вікон $1 \cdot 1,9 m^2$. Віконне скло подвійне. Характеристика зорової роботи відноситься

					ДП.КСМ.19113/11.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		87

до високої точності. Це відповідає нормі природного освітлення КПО $I_n = 2\%$ при боковому освітленні.

При боковому освітленні використовується формула:

$$100 \frac{S_0}{S_n} = \frac{I_n \cdot K_3 \cdot \eta_{10}}{\tau_0 \cdot VI} K_6; \quad (4.3)$$

де S_0 — площа світлових отворів, m^2 ;

S_n — площа підлоги, m ;

K_3 — коефіцієнт світлопроникнення;

η_{10} — світлова характеристика вікон;

τ_0 — загальний коефіцієнт світлопроникності;

VI — коефіцієнт, який враховує відбивання світла від поверхні;

K_6 — коефіцієнт, який враховує затемнення будинками, що стоять навпроти.

Для приміщення розмірами $6 * 8 * 3$ площа $48 m^2$;

Для $L_n/V = 8/6 = 1,33$; $V/H = 6/3 = 2$; $\eta_{10} = 16$;

Для середньозваженого коефіцієнта відображення стелі, стін і підлоги, який дорівнює $0,4$, коефіцієнт VI становить $2,4$. K_6 приймаємо — $1,4$.

Для приміщень з повітряним середовищем, в якому концентрація пилу менше $1 mg/m^3$ $K_3 = 1,4$;

Оскільки $I_n = 2\%$, коефіцієнт τ_0 визначаємо за формулою:

$$\tau_0 = \tau_1 \cdot \tau_2 \cdot \tau_3 \cdot \tau_4 \cdot \tau_5; \quad (4.5)$$

де τ_1, τ_2, τ_3 - коефіцієнти світлопропускання матеріалу вікна, виду вікна і його конструкції: для віконного, листового, подвійного скла $\tau_1 = 0,8$; для дерев'яних подвійних роздільних оправ до вікон $\tau_2 = 0,6$; для залізобетонних конструкцій $\tau_3 = 0,8$.

τ_4 - коефіцієнт, який враховує витрати світла в сонцезахисних конструкціях: для жалюзі і штор, що регулюються, дорівнює 1 .

τ_5 - коефіцієнт, який враховує втрати світла в захисній сітці, що встановлюється під світильником - дорівнює $0,9$.

Отже: $\tau_0 = 0,8 \cdot 0,6 \cdot 0,8 \cdot 1 \cdot 0,9 = 0,35$;

Визначаємо площу світлових отворів S_0 :

					ДП.КСМ.19113/11.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		88

$$S_0 = \frac{I_n \cdot K_z \cdot \eta_{10} \cdot S_n}{100 \cdot \tau_0} = K_6; \quad (4.6)$$

Кількість вікон визначаємо за формулою:

$$S_0 = \frac{2 \cdot 1,4 \cdot 16 \cdot 1,4 \cdot 48}{100 \cdot 0,35 \cdot 24} = 3,47(m^2) \quad (4.7)$$

де S_1 , — стандартна площа вікна.

Відповідно:

$$n = 347/1,9 = 1,83 = 2 \text{ вікна};$$

Таким чином, для забезпечення КПО $I_n = 2\%$ у приміщенні повинно бути два вікна площею $1,9 \text{ м}^2$.

Для освітлення приміщення, коли природного освітлення недостатньо, або взагалі немає, використовується штучне освітлення.

Світловий потік Φ — це потужність світлової енергії, що оцінюється за світловим відчуттям, яке воно справляє на органи зору людини:

$$\Phi = dQ/dt. \quad (4.8)$$

Сила світла I - це відношення світлового потоку до величини тілесного кута, в якому рівномірно розподілено випромінювання:

$$I = dF/d\omega. \quad (4.9)$$

Освітленість E - густина світлового потоку на освітлюваній поверхні.

$$E = d\Phi/dS. \quad (4.10)$$

Яскравість L - поверхнева густина сили світла у заданому напрямку.

$$L = dl/dS \cdot \cos(\alpha). \quad (4.11)$$

Коефіцієнт відбиття β - відношення відбитого світлового потоку до падаючого:

$$\beta = \Phi_{\text{відб.}}/\Phi_{\text{пад.}}$$

Якісні показники:

Фон - поверхня, що прилягає безпосередньо до об'єкта розпізнавання, на який цей об'єкт сприймається. Фон характеризує коефіцієнт відбиття (залежить від кольору поверхні та від її фактури). Фон світлий $\Phi > 0,4$; середній — $\Phi = 0,2—0,4$; темний $\Phi < 0,2$;

Контраст - ступінь розпізнавання яскравості об'єкта і фону.

					ДП.КСМ.19113/11.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		89

$$K=(L_0-L_{\phi}) / L_0. \quad (4.12)$$

Контраст великий $K>0,5$; середній $K = 0,2 - 0,5$; маленький - $K<0,2$;

Коефіцієнт пульсацій K_{Π} - критерій оцінки відносної глибини коливань освітленості в результаті зміни в часі світлового потоку газорозрядних ламп при живленні їх змінним струмом.

$$K_{\Pi} = (E_{\max} - E_{\min}) \cdot 100\% / (2 \cdot E_{\text{сер}}) \quad (4.13)$$

де E - значення освітленості за період.

Розміри приміщення: $A = 8$ м, $B = 6$ м, $H = 3,5$ м. Нормована освітленість 300 лк.

Показник приміщення:

$$i = A \cdot B / (H \cdot (A+B)) = 8 \cdot 6 / (3,5 \cdot (8+6)) = 1,14,$$

де A, B, H - відповідно розміри приміщення.

Вибираємо світильник НОДЛ з коефіцієнтом використання світлового потоку $\eta = 49\%$.

Сумарний світловий потік:

$$\Phi = ((E_n \cdot S \cdot k \cdot Z) / \eta) \cdot 100\%, \quad (4.14)$$

де E_n - нормована освітленість, лк;

S - площа приміщення, м^2 ;

k - коефіцієнт запасу;

Z - коефіцієнт мінімальної освітленості;

η - коефіцієнт використання світлового потоку.

$$\Phi = ((300 \cdot 48 \cdot 1,75 \cdot 1,1) / 49) \cdot 100\% = 56\,572 \text{ лм.}$$

Вибираємо лампи ЛТБ-80 р, $\Phi_{\text{л}}$ - 4300 лм, тоді кількість ламп дорівнює:

$$N = \Phi / \Phi_{\text{л}} = 56572 / 4300 = 14 \text{ шт.}$$

Кількість світильників:

$$N_c = N / 2 = 7 \text{ шт.}$$

Перерахуємо значення E :

$$E = \frac{N \cdot \Phi_{\text{л}} \cdot \eta}{S \cdot k \cdot Z \cdot 100\%} = \frac{14 \cdot 4300 \cdot 49}{48 \cdot 1,75 \cdot 1,1 \cdot 100\%} = 319,3 \quad (4.15)$$

Отже, штучне освітлення забезпечує освітленість $E = 319$ лк, що є більшим за E_n ($E_n = 300$ лк), тобто розрахунок проведений правильно.

					ДП.КСМ.19113/11.00.00.000.ПЗ	Арк.
						90
Змн.	Арк.	№ докум.	Підпис	Дата		

Рівень шуму дорівнює 75 дБ, що відповідає вимогам ГОСТу, тому захисних заходів не передбачається.

Електричний струм при дії на людину може викликати як місцеві, так і загальні пошкодження. Місцеві електротравми — це опіки, нагрівання внутрішніх органів, механічні пошкодження (розрив тканин м'язів), порушення біоелектричних процесів у організмі, електроліз органічних рідин. Зовнішніми проявами електротравм можуть бути термічні опіки, електричні ознаки на шкірі, металізація поверхні шкіри, електрофтальмія (ураження зору під дією ультрафіолетових променів при іскровому розряді). Загальне ураження струмом відбувається при проходженні струму через нервові центри, центри дихання і роботи серця (електричний удар).

Небезпека ураження тим більша, чим більший струм проходить через людину, але крім цього, впливають: тривалість і шлях проходження струму, його вид, частота і виробничі умови.

Умови ураження людини електричним струмом такі:

- двофазне дотикання (двофазне включення людини в мережу);
- однофазне дотикання, наближення на небезпечну віддаль до неізольованих дротів з напругою більше 1000 В;
- дотик до корпусу обладнання, що не проводить струм, але опинилося під напругою;
- перебування в зоні дії атмосферної електрики;
- вхід у зону дії електромагнітного поля.

Згідно класифікації приміщень за ступенем небезпеки ураження електричним струмом (ПУЕ 1.1.6) приміщення роботи системи відноситься до першого (без підвищеної небезпеки).

Електричні установки, до яких відноситься переважна більшість обладнання системи, вимагають дотримання правил електробезпеки, оскільки в процесі експлуатації або проведення профілактичних робіт людина може доторкнутись до частин, що знаходяться під напругою 220 В, тому виникає необхідність у захисті персоналу від ураження електричним струмом. Дуже велике значення для запобігання електротравматизму має правильна

					ДП.КСМ.19113/11.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		91

організація експлуатації; обслуговування системи. Під цим розуміється точне виконання ряду організаційних та технічних заходів, які встановлені діючими «Правилами технічної експлуатації електроустановок споживачів і правил техніки безпеки при експлуатації електроустановок споживачів» (ППЕ і ПТБ споживачів) і «Правилами побудови електропристроїв» (ППЕ). Основними технічними засобами, які забезпечують безпеку робіт в електроустановках, є: захисне заземлення, занулення, вирівнювання потенціалів, захисне включення, електричний розподіл мереж, мала напруга, подвійна ізоляція. Використання цих засобів у різноманітних поєднаннях дозволяє захистити людину від ураження струмом.

Захисне заземлення - це навмисне електричне з'єднання з землею або її еквівалентом металевих неструмопровідних частин, які можуть бути під напругою. У приміщенні розміщення контролера базових станцій заземлено всі шафи з обладнанням, а також вся комп'ютерна техніка. Приміщення, де знаходиться система, обладнується контуром-шиною захисного заземлення, яка з'єднується із заземлювачем. Контур-шина виготовляється з мідного дроту діаметром 6 мм у перерізі і вкладається по периметру приміщення. Місця перетину дротів припадають на застосуванням бікислотного флюсу. Для під'єднання заземлювальних провідників на шину наварюються гвинти М8. У дипломній роботі проведу розрахунок захисного заземлення згідно порядку, встановленого ПУЕ.

Згідно вимог ПУЕ 1.7.65 в електроустановках з напругами до 1 кВ при потужності трансформатора менше 100 кВт опір заземлювача повинен бути не більше 10 Ом.

1. Визначаємо розрахунковий опір землі

$$r_{o.p.z.} = \Phi \cdot r_{o_3}, \quad (4.16)$$

де Φ - коефіцієнт сезонності, який враховує коливання питомого опору при зміні вологості ґрунту протягом року; використовується стержневий заземлювач довжиною $l = 2$ м при глибині закладання від вершини $h = 0,5$ м $\Phi = 1,1$ - четвертої кліматичної зони. Питомий опір ґрунту: $r_{o_3} = 300$ Ом·м - для піску;

					ДП.КСМ.19113/11.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		92

$$r_{0,з.} = 1,1 \cdot 300 = 3300 \text{ м}\cdot\text{м}. \quad (4.17)$$

2. Визначаємо опір R , розтікання струму в землі від одного вертикального заземлювача:

$$R_B = \frac{r_{0,з.}}{2 \cdot 3,14 \cdot l} \left(\ln \frac{2 \cdot l}{d} + \frac{1}{2} \ln \frac{4 \cdot t + l}{4 \cdot t - l} \right), \quad (4.18)$$

де l - довжина заземлювача ($l = 2 \text{ м}$);

$d = 0,05 \text{ м}$ – діаметр заземлювача за таблицею при $U < 1 \text{ кВ}$ та при $S < 100 \text{ кВА}$;

t – відстань від поверхні землі до середини заземлювача,

$$t = h + l/2 = 0,5 + 2/2 = 1,5 \text{ м};$$

$$R_B = \frac{330}{2 \cdot 3,14 \cdot 2} \left(\ln \frac{2 \cdot 2}{0,05} + \frac{1}{2} \ln \frac{4 \cdot 1,5 + 2}{4 \cdot 1,5 - 2} \right) = 133,3 \text{ (Ом)},$$

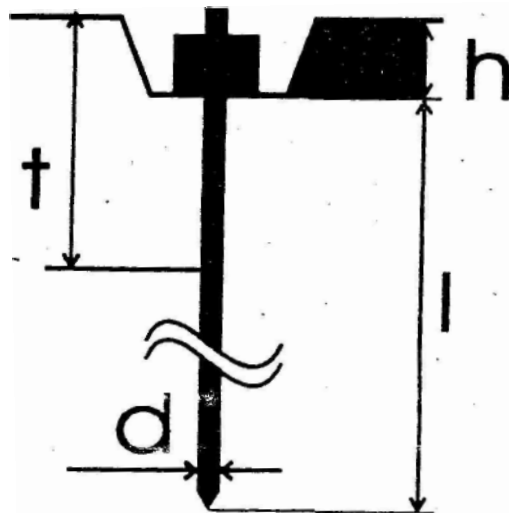


Рисунок 4.1 - Схема розташування одиничного заземлювача в ґрунті.

3. Приблизна кількість заземлювачів

$$n = \frac{R_B}{R_{в.нтмм}} = \frac{133,3}{10} = 13,3 \approx 14. \quad (4.19)$$

4. Знаходимо із таблиць коефіцієнт використання вертикальних заземлювачів, який враховує ефект екранування при вибраному значенні $k = a/l$, де a - віддаль між заземлювачами, м; $k = 1,2$ при $a = 2,4 \text{ м}$; отже коефіцієнт використання вертикального заземлювача за таблицями дорівнює $\eta_e = 0,56$.

5. Кількість вертикальних заземлювачів з урахуванням η_e обчислюємо за

					ДП.КСМ.19113/11.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		93

формулою:

$$n = \frac{R_B}{R_{B, \text{норм}} \cdot \eta_B} = \frac{133,277}{10 \cdot 0,56} = 23,799 = 24. \quad (4.20)$$

6. Довжина горизонтального заземлювача для розміщення по контуру

$$L = a \cdot n = 2,4 \text{ м} \cdot 24 = 57,6 \text{ м}. \quad (4.21)$$

7. Опір горизонтального заземлювача R_r (Ом), прокладеного на глибині $h = 0,5$ м від поверхні землі:

$$R_r = \frac{\rho_{\text{п.к.}}}{2 \cdot 3,14 \cdot L} \ln \frac{2 \cdot L}{b \cdot h} = \frac{330}{2 \cdot 3,14 \cdot 57,6} \ln \frac{2 \cdot 57,6}{0,04 \cdot 0,5} = 7,3 \quad (4.22)$$

де $b = 0,04$ м — ширина штабової сталі, з якої виготовлений заземлювач.

8. Обчислюємо загальний опір:

$$R_k = \frac{R_B \cdot R_o}{n \cdot R_d \cdot \eta_B + R_B \cdot \eta_d} = \frac{133,3 \cdot 7,3}{24 \cdot 7,3 \cdot 0,56 + 133,3 \cdot 0,27} = 7,5 \text{ (Ом)} \quad (4.23)$$

результат менше 10 Ом, тобто виконується нормуюча умова $R_z < R_{z, \text{норм}}$.

Велика увага приділяється дотриманню обслуговуючим персоналом правил роботи в приміщенні, яке призначене для експлуатації системи. У приміщенні не повинно бути сторонніх людей. Працівники повинні використовувати спецодяг. Безпека роботи обслуговуючого персоналу в приміщенні забезпечується:

- наявністю нормальних проходів між обладнанням;
- використанням спеціальних технічних меблів;
- використанням електрозахисних засобів (діелектричних килимків, гумових рукавиць);
- наявністю аварійного освітлення ($E=2$ лк);
- обладнанням розеток з напругою 220 В;
- заземленням корпусів обладнання і апаратури освітлювальних пристроїв.

Одне з основних місць в охороні праці займає пожежна безпека.

Першочергове завдання пожежної профілактики - це запобігання пожеж. Під пожежною профілактикою розуміють комплекс організаційних і технічних заходів, спрямованих на забезпечення безпеки людей, на запобігання пожеж, обмеження їх розповсюдження, а також на створення умов для успішного

					ДП.КСМ.19113/11.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		94

гасіння пожеж. Пожежно-профілактичні заходи розробляються та виконуються разом, в тісному взаємозв'язку з усіма проектними, будівельними та експлуатаційними роботами.

Приміщення чергування технічного персоналу забезпечується проти-пожежним інвентарем (вуглекислотними вогнегасниками типу ВВ-2). Проходи між рядами і вихід не повинні загроможуватись. У випадку виникнення пожежі перш за все потрібно виключити джерело живлення, сповістити про пожежу в пожежну частину. Евакуювати сторонніх людей, які могли опинитися в небезпечній зоні і лише після цього приступити до гасіння пожежі і рятування цінного обладнання.

Один вуглекислотний вогнегасник ВВ-2 розрахований на 40—50 м² приміщення. Для ліквідації невеликих пожеж можна використовувати деякі порошкові матеріали (хлориди лужних металів, соду, пісок і т. д.), що подаються в зону горіння порошковими вогнегасниками.

Будівля, в якій знаходиться наше приміщення обов'язково має резервний вихід на випадок екстреної евакуації працівників і неможливості використання основного виходу.

За вибухопожежною і пожежною безпекою приміщення і будівлі згідно ОНТП-24-86 і СНТП 2.09, СНТП 02-85 діляться на категорії А, Б, В, Г, Д.

Для приміщення чергування персоналу ЛАЦ встановлена категорія пожежної безпеки Д (СНиП 2.09.02-85) при ступені вогнестійкості (СНиП Н-90-81), що означає наявність у приміщенні негорючих речовин та матеріалів у холодному стажі.

Для швидкого сповіщення пожежної сховони при виникненні пожежі приміщенні використовується електрична пожежна сигналізація. Система електричної пожежної сигналізації виявляє пожежу на початковій стадії і сповіщає про місце її виникнення, а також автоматично включає стаціонарні установки гасіння пожеж.

Автоматичні оповіщувачі при ознаках пожежі здійснюють посилення сигналу. Сповіщувачі типу АТИП-1, АТИП-3 і АТИП 3М спрацьовують внаслідок теплової деформації (при 80-100 °С) біметалічних пластинок і мають

					ДП.КСМ.19113/11.00.00.000.ПЗ	Арк.
						95
Змн.	Арк.	№ докум.	Підпис	Дата		

розраховану площу обслуговування в приміщеннях до 15 м². Комбіновані теплові і димові оповіщувачі типу КИ-1 мають чутливий елемент у вигляді іонізуючої камери (реагування на дим) і теоморезистори (реагування на тепло). Температура спрацювання цих оповіщувачів 50—80°С, площа обслуговування 100 м².

Передбачені нами заходи з охорони праці в першу чергу призначені для уникнення нещасних випадків, що можуть виникнути на підприємстві.

В іншому передбачені заходи з охорони праці відповідають вимогам нормативних документів і актів та забезпечують нормальну, ефективну і безпечну для здоров'я людини виробничу діяльність.

					ДП.КСМ.19113/11.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		96

ВИСНОВКИ

Гібридні інтелектуальні системи дозволяють використати переваги традиційних засобів штучного інтелекту, долаючи деякі їх недоліки, і можуть вирішувати задачі, нерозв'язані окремими методами штучного інтелекту. В результаті виконання дипломного проекту отримано наступні висновки.

Проведено класифікацію мікрооб'єктів біомедичних зображень, що дало можливість виділити їх інформативні ознаки. Розроблено алгоритми текстурного аналізу та сегментації біомедичного зображення на основі статистичних ознак, що дало змогу виділити мікрооб'єкти які дотикаються та накладаються.

Розроблено алгоритми аналізу на основі піраміди зображення, що дало змогу зменшити вплив завад на процес автоматичного оброблення зображення.

Розроблено алгоритм побудови нечітких правил виводу висновку на основі числових ознак мікрооб'єктів біомедичних зображень, що дало змогу побудувати базу знань типів мікрооб'єктів.

В якості програмного засобу обрано відкриту бібліотеку ImageJ побудовану мовою програмування Java. Використання даної бібліотеки дало змогу забезпечити вимоги до надійності та швидкодії програмних засобів, розробити інтуїтивно зрозумілий інтерфейс та побудувати ефективні алгоритми оброблення біомедичних зображень.

Розроблено структуру гібридної інтелектуальної системи опрацювання біомедичних зображень яка поєднує кількісні та якісні ознаки мікрооб'єктів та використовує ці ознаки для виводу висновку про тип мікрооб'єктів. Проведено тестування розробленої системи на базі реальних біомедичних зображень, що підтвердило доцільність поєднання алгоритмів комп'ютерного зору та нечіткого виводу висновку.

					ДП.КСМ.19113/11.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		97

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Заде Л. Понятие лингвистической переменной и его применение к принятию приближенных решений. – М., 1976. – 165 С.
2. Советов Б.Я. Информационные технологии / Б.Я. Советов, В.В. Цехановский - М.: Высшая школа, 2005 - 263 с.
3. Поспелов Г.С Искусственный интеллект - основа новой информационной технологии - М.: Высшая школа, 1988 – 280 с.
4. Jacobsen H.A. A generic architecture for hybrid intelligent systems // IEEE Fuzzy Systems. Encourage. – Alaska. 1998. – P. 709-714.
5. Kandel A. Fuzzy intelligent hybrid expert system and their application // IEEE. 1995. – P. 2275-2280.
6. Fenton B., McGinnity M., Maguire L. Fault diagnosis of electronic systems (using artificial intelligence) // IEEE Instrumentation & Measurement Magazine. –2002. –№9. – P. 16-20.
7. Herrmann C. A hybrid fuzzy-neural expert system for diagnosis // Proc. of IJCAI. Montreal. –1995. –P. 1-10.
8. Pacheco R. et al. A hybrid intelligent system applied to financial statement analysis // Proc. of 5th FUZZ IEEE. New Orleans. –1996. P. –1007-1012.
9. Yan H.H. et al. Power system security assessment using a hybrid expert system/neural network architecture // Proc. of IEEE. ISCS. New York. – 1992. – P. 1713-1716.
10. Froning J.N., Olson M.D., Froelicher V.F. Exercise ECG analysis and measurement using an expert system approach // Proc. of 10th IC. IEEE Engineering in medicine & biology society. – 1988. – P. 1-2.
11. Sivathanan S., Cecelja F., Balachandran W. ECG Diagnosis using neural network and fuzzy expert system // Proc. of PREP'99. UMIST. Manchester. – 1999. – P. 340-343.

					ДП.КСМ.19113/11.00.00.000.ПЗ	Арк. 98
Змн.	Арк.	№ докум.	Підпис	Дата		

- 12.El_Fergany A.A., Yousef M.T., El_Alaily A.A. Fault diagnosis in power systems – substation level – through hybrid artificial neural networks and expert system // IEEE. – 2001. – P. 207-211.
- 13.Funabashi M. et al. Fuzzy and neural hybrid expert systems: synergetic AI // AI in Japan. IEEE Expert. – 1995. – P. 32-40.
- 14.Passold F., Ojeda R.G., Barreto J.M. A hybrid expert system in anesthesiology for critical patients // IEEE. – 1996. – P. 1486-1489.
- 15.Wiriyagoonkasem S., Esterline A. Adaptive learning expert system // IEEE. – 2000. – P. 445-448.
- 16.Meesad P., Yen G. A hybrid intelligent system for medical diagnosis // Proc. of IJCNN'01. Washington. – 2001. – P. 2558-2563.
- 17.Lam D., Swayne D. A hybrid expert system and neural network approach for envi-ronmental application // Proc. of ICESD. Bangkok. – 1994. – P. 298-303.
- 18.Luo R.C., Lin M.H., Shen Sh.H. The development of object-oriented knowledge base and adaptive motion planning for autonomous mobile robots // Proc. of the 2001 IEEE/RSJ ICIRS. Maui. – 2001. – P. 108-113.
- 19.Nikolopoulos C. A hybrid expert system for investment advising // IEEE. – 1994. – P. 1818-1820.
- 20.Mori H. et al. A hybrid intelligent system for fault detection in power systems // Proc. of IEEE IJCNN. Honolulu. – 2002. – Vol. 2. – P. 2138-2143.
- 21.Lhotska L., Vlcek T. Efficiency enhancement of rule-based expert systems // Proc. of the 15th IEEE CBMS. – 2002. – P. 1063-1068.
- 22.Averbukh A.B. Hybrid intelligence architecture for real time processing // IEEE. – 1999. – P. 4107-4110.
- 23.Prentzas J., Hatzilygeroudis I., Koutsojannis C. A web-based ITS controlled by a hybrid expert system // Proc. of IEEE ICALT. – 2001. – P. 239-240.
- 24.Автандилов Г.Г. Основы количественной паталлогической анатомии. – М.: Медицина , 2002. – 240 с.
- 25.Егорова О. Компьютерная микроскопия. / В. Пантелеев, О. Егорова, Е. Клыкова – М.: Техносфера, 2005. – 300 с .

					ДП.КСМ.19113/11.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		99

26. Д. Форсайт, Дж. Понс Компьютерное зрение. Современный подход - СПб: Вильямс, 2004 – 316 с..
27. Гуревич И.Б. Тезаурус и Онтология Предметной Области "Анализ Изображений" / И.Б. Гуревич, Ю.О. Трусова // II Всероссийская конференция "ЗНАНИЯ – ОНТОЛОГИИ – ТЕОРИИ". (ЗОНТ-09). 20–22 октября 2009 г. 2009. – 12 С. .
28. Pratt W. K. Digital Image Processing: PIKS Scientific Inside / William K. Pratt - NY, USA: John Wiley & Sons, Inc., 2007. - 782 p.
29. Yanxi Liu, R. C. & Tsin, Y. A Computational Model for Periodic Pattern Perception Based on Frieze and Wallpaper Groups //IEEE Transactions on Pattern Analysis and Machine Intelligence' – 2004 – pp. 354 - 371.
30. Методичні рекомендації до виконання дипломного проекту з освітньо-кваліфікаційного рівня „Спеціаліст”. Спеціальність „Комп’ютерні системи та мережі” // Березький О.М., Н.М.Васильків, І.В.Васильцов, Р.Б.Трембач.- Тернопіль: ТНЕУ, 2008. – 38 с.

					ДП.КСМ.19113/11.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		100

ДОДАТОК А

Вихідний текст головного модуля системи

```
function [iind,jind,numberofpixels]=fastcentroid(X)
% [iind,jind,numberofpixels]=fastcentroid(X)
% повертає координати центроїда для сполучених компонентів в BW
(iind,jind).
% and the number of pixels of each component (numberofpixels)
%
% input:
% X bynary image
%
% output:
% iind [length = number of connected components ]
% jind [length = number of connected components ]
% numberofpixels [length = number of connected components ]
%
XMLABEL=BWLABEL(X);
XMLABELS=sparse(XMLABEL);
maxXM=max(XMLABEL(:));

for i=1:maxXM
    [ii,jj]=find(XMLABELS==i);
    numberofpixels(i)=length(ii);
    iind(i)=mean(ii);
    jind(i)=mean(jj);
end

Текст файлу calculate_min_distance.m
% Calculates the minimum distance between 2 edges and returns the distance
and
```

					ДП.КСМ.19113/11.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		101

```

% the matrix linking the coordinates to both edge coordinates.
% Input :
%     edge1,edge2 - [X,Y] coordinates to be compared and measured
%     'testing' or 'not testing' (optional) - Default 'not testing'
%
% Output :
%     min_distance - Returns the minimum distance between the two
%                   sets of coordinates.
%     matching_coordinates - [edge1X,edge1Y,edge2X,edge2Y]
%                   Pairs of coordinates of edge1 and edge 2 that
%                   are of the minimum distance.

function [min_distance, matching_coordinates] = calculate_min_distance
(edge1, edge2, varargin )
% -----
% Process the input
testing = 'not testing';
if ~isempty(varargin)
    for n = 1:1:length(varargin)
        if strcmp(varargin{n},'not testing') | strcmp(varargin{n},'testing')
            testing = varargin{n};
        end
    end
end
if isempty(edge1) | isempty(edge2)
    error('Input entry is a null matrix');
end
% -----
% Find the closest edge and calculate the distance
% -----
temp1 = [];

```

					ДП.КСМ.19113/11.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		102

```

temp2 = [];
for n = 1:1:length(edge1(:,1))
    temp = edge2;
    temp(:,1) = edge1(n,1);
    temp(:,2) = edge1(n,2);
    temp1 = [temp1;temp];
    temp2 = [temp2;edge2];
    % Distance calculated here to reduce memory usage
    distance = euclidean_distance(temp1,temp2);
    min_distance = min(distance);
    pos = find(distance == min_distance);
    temp1 = temp1(pos,:);
    temp2 = temp2(pos,:);
end
distance = euclidean_distance(temp1,temp2);
pos = find(distance == min_distance);
min_distance = min(distance);

matching_coordinates = [temp1(pos,:),temp2(pos,:)];

% -----
% Displaying the results for testing
% -----

if strcmp(testing,'testing')
    figure;
    plot(edge1(:,1),edge1(:,2),'r+');
    hold on;
    plot(edge2(:,1),edge2(:,2),'b+');
    for n = 1:1:length(matching_coordinates(:,1))
        plot([matching_coordinates(n,1);matching_coordinates(n,3)],...
            [matching_coordinates(n,2);matching_coordinates(n,4)],'g*-');
    end

```

					ДП.КСМ.19113/11.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		103

```

end
title('edge1(red+), edge2(blue+) and selected edges(green*)');
xlabel(strcat('minimum distance = ',num2str(min_distance)));
end

```

Текст файлу segmentation.m

```

function mat = segment(mat,wsiz,thresh,pixNum);
%Синтаксис : mask = segment(image>window_size,threshold,pix_number);
% На вхід подається зображення в градаціях сірого.
% mask = segment(I,100,0.5,10);
% imagesc(I),figure,imagesc(mask)
[xs,ys] = size(mat);
avr = mat2avr(mat,wsiz);
STD = sqrt(mat2avr(mat.^2,wsiz)-avr.^2);
mat = (mat - avr)./STD;
mat = mat > thresh;
mat = bwareaopen(mat,pixNum,4);
function mat = mat2avr(mat,num)
% синтаксис : avr = mat2avr(mat,num)
% знаходить 2*num+1 середнє по квадратному вікну
[xs,ys] = size(mat);
mat = [mat(:,num+1:-1:2,,:); mat mat(:,ys-1:-1:ys-num,,:)];
mat = [mat(num+1:-1:2,,:); mat; mat(xs-1:-1:xs-num,,:)];
mask = zeros(size(mat));
mask(1:2*num+1,1:2*num+1,,:) = 1;
mat = real(ifft2(fft2(mat).*fft2(mask)))./[(2*num+1).^2];
mat = mat(2*num+1:end,2*num+1:end,,:);
im1=rgb2gray(im);
im1=medfilt2(im1,[3 3]); %Median filtering the image to remove noise%
BW = edge(im1,'sobel'); %finding edges
[imx,imy]=size(BW);

```

					ДП.КСМ.19113/11.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		104

```

msk=[0 0 0 0 0;
      0 1 1 1 0;
      0 1 1 1 0;
      0 1 1 1 0;
      0 0 0 0 0;];

B=conv2(BW,msk); %Розмиття зображення

L = bwlabel(B,8); % Підрахунок з'єднаних компонентів

mx=max(max(L))

[r,c] = find(L==17);

rc = [r c];

[sx sy]=size(rc);

n1=zeros(imx,imy);

for i=1:sx
    x1=rc(i,1);
    y1=rc(i,2);
    n1(x1,y1)=255;
end % Зберігання зображення в масиві

ima=double(ima);

copy=ima; % make a copy
ima=ima(:); % vectorize ima

mi=min(ima); % deal with negative
ima=ima-mi+1; % and zero values

s=length(ima);

% створити гістограму зображення
m=max(ima)+1;
h=zeros(1,m);
hc=zeros(1,m);
for i=1:s
    if(ima(i)>0) h(ima(i))=h(ima(i))+1;end;
end

```

					ДП.КСМ.19113/11.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		105


```

ind=find(h);
hl=length(ind);
mu=(1:k)*m/(k+1);
while(true)
    oldmu=mu;
    % поточна класифікація
    for i=1:hl
        c=abs(ind(i)-mu);
        cc=find(c==min(c));
        hc(ind(i))=cc(1);
    end
    %перерахунок середніх значень
    for i=1:k,
        a=find(hc==i);
        mu(i)=sum(a.*h(a))/sum(h(a));
    end
    if(mu==oldmu) break;end;
end
s=size(copy);      % обчислити маску
mask=zeros(s);
for i=1:s(1),
for j=1:s(2),
    c=abs(copy(i,j)-mu);
    a=find(c==min(c));
    mask(i,j)=a(1);
end
end
mu=mu+mi-1;

Файл main.h
#include "wx/toolbar.h"

```

					ДП.КСМ.19113/11.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		106

```

#include "wx/foldbar/foldpanelbar.h"
#include "wx/propgrid/propgrid.h"
#include "wx/treectrl.h"
#include <vector>

// Визначити нове застосування
class MyApp : public wxApp
{
public:
    bool OnInit();
};

// клас для дочірніх
вікон
class mTexture : public wxScrolledWindow
{
public:
    mTexture(wxWindow *parent, const wxPoint& pos, const wxSize&
size); //constructor

    virtual void OnDraw(wxDC& dc);
    bool IsDirty() const { return m_dirty; }
    void OnEvent(wxMouseEvent& event);
    void SetText(const wxString& text) { m_text = text;
Refresh(); }

private:
    wxString m_text;
    bool m_dirty;

    DECLARE_EVENT_TABLE()
};

```

					ДП.КСМ.19113/11.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		107

```

// клас батьківського фрейму
class mainFrame : public wxMDIParentFrame
{
private:
void CreatePanels(wxSizer *sizer);
void SelectNode(wxTreeEvent& event);    //повідомлення від Дерева
перегляду

public:
    wxSashLayoutWindow* m_leftPanel1;
    wxSashLayoutWindow* m_leftPanel2;
    wxTextCtrl *textWindow;           //text control, single or multi-line
    wxTreeCtrl *treeWindow;
    wxPropertyGrid *propertyWindow;

    CMTree          *Tree;
    typedef std::vector<CTexPalette*> m_PaletteList;

    mainFrame(wxWindow *parent, const wxWindowID id, const
wxString& title,
                const wxPoint& pos, const wxSize& size, const long style);

    void InitToolBar(wxToolBar* toolBar);

    void OnFoldPanelBarDrag(wxSashEvent& event);

    void OnSize(wxSizeEvent& event);
    void OnAbout(wxCommandEvent& event);

    void OnNewFile(wxCommandEvent& event);    //MENU HANDLERS

```

					ДП.КСМ.19113/11.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		108

```

void OnOpenTexture(wxCommandEvent& event);
void OnOpenColormap(wxCommandEvent& event);
void OnSave(wxCommandEvent& event);
void OnCloseFile(wxCommandEvent& event);
void OnQuit(wxCommandEvent& event);

void OnMDIBlank( wxCommandEvent& event); //adds image writer
void OnAdding( wxCommandEvent& event); //adds
image+renderer
//adds noisemap+noisemapbulderpale
void OnAddNoiseMap( wxCommandEvent& event);
void OnAddGenPerlin( wxCommandEvent& event);

void OnClose(wxCloseEvent& event);

DECLARE_EVENT_TABLE()

};

// клас дочірнього фрейму
class mChildWnd: public wxMDIChildFrame
{
public:
    mTexture *mtexture;
    mChildWnd(wxMDIParentFrame *parent, const wxString& title);
    ~mChildWnd();
    void OnActivate(wxActivateEvent& event);
    void OnUpdateRefresh(wxUpdateUIEvent& event);
    void OnSize(wxSizeEvent& event);
    void OnMove(wxMoveEvent& event);
    void OnClose(wxCloseEvent& event);

/*

```

					ДП.КСМ.19113/11.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		109

```

#if wxUSE_CLIPBOARD
    void OnPaste(wxCommandEvent& event);
    void OnUpdatePaste(wxUpdateUIEvent& event);
#endif // wxUSE_CLIPBOARD
*/

    DECLARE_EVENT_TABLE()

};

// menu items ids
enum
{
    NEW_FILE,
    OPEN_TEXTURE,
    OPEN_COLORMAP,
    M_SAVE, M_CLOSE,
    M_QUIT,
    ID_LEFT_PANEL1,      ID_LEFT_PANEL2,
    ID_PROPGRIDCTRL,
    MID_ADDIMAGE, MID_ADDIMGRENDERER,
    MID_ADDBUILDPLANE, MID_ADDBUILDSPHERE,
    MID_ADDGENPERLIN,
    MDI_QUIT = wxID_EXIT,
    MDI_NEW_WINDOW = 101,
    MDI_REFRESH,
    MDI_CHANGE_TITLE,
    MDI_CHANGE_POSITION,
    MDI_CHANGE_SIZE,
    MDI_CHILD_QUIT,
    MDI_ABOUT = wxID_ABOUT,
    ID_IMGWRT_FILENAME,
    ID_NMAPBLD_X,      ID_NMAPBLD_Y,
    ID_NMAPBLD_BX1, ID_NMAPBLD_BX2,

```

					ДП.КСМ.19113/11.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		110

```

ID_NMAPBLD_BY1, ID_NMAPBLD_BY2,
ID_PERLIN_FREQ,
ID_PERLIN_OCTAVES,
ID_PERLIN_PERSISTENCE,
ID_PERLIN_LACUNARITY,
ID_PERLIN_SEED,
ID_TREECTRL = 1000

};

```

Файл main.cpp

```

/////////////////////////////////////////////////////////////////
// Name: main.cpp
// Purpose: TextureWork project
/////////////////////////////////////////////////////////////////
// declarations
// headers
#include "wx/wxprec.h" //Includes the appropriate files for precompiled
headers
#include "wx/propgrid/propgrid.h"

#ifndef WX_PRECOMP
    #include "wx/wx.h" //wxWidgets central header including the
most often used ones
    #include "wx/treectrl.h"
    #include "wx/mdi.h" //wxMDI base header
    #include "wx/artprov.h"
#endif

// verify that the item is ok and insult the user if it is not
#define CHECK_ITEM( item ) if ( !item.IsOk() ) { \

```

					ДП.КСМ.19113/11.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		111

```

        wxMessageBox(wxT("Please select some item first!"), \
        wxT("Tree sample error"), \
        wxOK | wxICON_EXCLAMATION,
\
        this);
\
        return; \
    }

```

```

#include "wx/toolbar.h" //wxToolBar interface declaration

```

```

#include "wx/laywin.h"

```

```

#include "wx/spinctrl.h"

```

```

#include "wx/slider.h"

```

```

#include "bitmaps/m01blank.xpm"

```

```

#include "bitmaps/m02plane.xpm"

```

```

#include "bitmaps/m03sphere.xpm"

```

```

#include "bitmaps/m04perlin.xpm"

```

```

#include "bitmaps/m05voroni.xpm"

```

```

#include "bitmaps/m06genbillow.xpm"

```

```

#include "bitmaps/m07genridget.xpm"

```

```

#include "bitmaps/m08genspheres.xpm"

```

```

#include "myclass.h"

```

```

#include <vector>

```

```

#include "main.h"

```

```

DECLARE_APP(MyApp)

```

```

IMPLEMENT_APP(MyApp) //let wxWidgets know your app entry point

```

```

// global variables #####

```

					ДП.КСМ.19113/11.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		112

```

mainFrame *frame = (mainFrame *) NULL;
wxList my_children;
static int gs_nFrames = 0;

// MyApp -----
bool MyApp::OnInit()
{
// Create the main frame window
frame = new mainFrame((wxFrame *)NULL, wxID_ANY, _T("Texture
work"),
wxDefaultPosition, wxSize(500, 400),
wxDEFAULT_FRAME_STYLE | wxHSCROLL | wxVSCROLL);

// створити меню "File" -----
wxMenu *file_menu = new wxMenu;
file_menu->Append(NEW_FILE, _T("&New ...\tCtrl-N"), _T("Create new
bitmap file"));
file_menu->AppendSeparator();
file_menu->Append(OPEN_TEXTURE, _T("&Open\tCtrl-O"), _T("Load
texture"));
file_menu->Append(OPEN_COLOMAP, _T("Open co&lormap \tCtrl-L"),
_T("Load colormap"));
file_menu->AppendSeparator();
file_menu->Append(M_SAVE, _T("&Save\tCtrl-S"), _T("Save bitmap"));
file_menu->Append(M_CLOSE, _T("&Close\tCtrl-W"), _T("Close bitmap"));
file_menu->AppendSeparator();
file_menu->Append(M_QUIT, _T("&Exit\tAlt-X"), _T("Quit the program"));

wxMenuBar *menu_bar = new wxMenuBar;
menu_bar->Append(file_menu, _T("&File"));

```

					ДП.КСМ.19113/11.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		113


```

frame->SetMenuBar(menu_bar);           // Associate the menu bar
with the frame

//-----
frame->CreateStatusBar();
frame->Show(true);
SetTopWindow(frame);

return true;
}

// mainFrame -----
BEGIN_EVENT_TABLE(mainFrame, wxMDIParentFrame)
    EVT_MENU( NEW_FILE,    mainFrame::OnNewFile)
    EVT_MENU( OPEN_TEXTURE, mainFrame::OnOpenTexture)
    EVT_MENU( OPEN_COLORMAP, mainFrame::OnOpenColormap)
    EVT_MENU( M_SAVE,     mainFrame::OnSave)
    EVT_MENU( M_CLOSE,    mainFrame::OnCloseFile)
    // EVT_MENU( MDI_ABOUT, mainFrame::OnAbout)      // діалог
про програму
    EVT_MENU( M_QUIT,     mainFrame::OnQuit)
    EVT_MENU( MID_ADDIMAGE, mainFrame::OnAddimg)
    EVT_MENU( MID_ADDBUILDPLANE,
mainFrame::OnAddNoiseMap)
    EVT_MENU( MID_ADDGENPERLIN, mainFrame::OnAddGenPerlin)
    EVT_SASH_DRAGGED_RANGE(ID_LEFT_PANEL1,
ID_LEFT_PANEL2, mainFrame::OnFoldPanelBarDrag)
    EVT_CLOSE( mainFrame::OnClose)
    EVT_SIZE( mainFrame::OnSize)
    EVT_TREE_ITEM_ACTIVATED( ID_TREECTRL,
mainFrame::SelectNode )

```

					ДП.КСМ.19113/11.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		114

```

        EVT_TREE_SEL_CHANGED( ID_TREECTRL,
mainFrame::SelectNode )

        END_EVENT_TABLE()

        // Define my frame constructor

        mainFrame::mainFrame(wxWindow *parent, const wxWindowID id, const
wxString& title,          const wxPoint& pos, const wxSize& size, const long style)
            : wxMDIParentFrame(parent, id, title, pos, size,
                style | wxNO_FULL_REPAINT_ON_RESIZE)
        {
            SetSize( wxSize(800, 600) );

            wxSashLayoutWindow* win = new wxSashLayoutWindow(this,
ID_LEFT_PANEL1,
                wxDefaultPosition, wxSize(800, 150),
                wxNO_BORDER | wxSW_3D | wxCLIP_CHILDREN);

            win->SetDefaultSize(wxSize(120, 1000));
            win->SetOrientation(wxLAYOUT_VERTICAL);
            win->SetAlignment(wxLAYOUT_LEFT);
            win->SetSashVisible(wxSASH_RIGHT, true);
            win->SetExtraBorderSize(0);
            m_leftPanel1 = win;

            win = new wxSashLayoutWindow(this, ID_LEFT_PANEL2,
                wxDefaultPosition, wxSize(800, 80),
                wxNO_BORDER | wxSW_3D | wxCLIP_CHILDREN);
            win->SetDefaultSize(wxSize(120, 1000));
            win->SetOrientation(wxLAYOUT_VERTICAL);
            win->SetAlignment(wxLAYOUT_LEFT);
            win->SetSashVisible(wxSASH_RIGHT, true);
            win->SetExtraBorderSize(0);

```

					ДП.КСМ.19113/11.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		115

```

        m_leftPanel2 = win;
        CreatePanels(0);

        /*
        wxImageList* imglist = new wxImageList(16, 16, true, 2);
        imglist->Add(wxArtProvider::GetBitmap(wxART_FOLDER,
wxART_OTHER, wxSize(16,16)));
        imglist->Add(wxArtProvider::GetBitmap(wxART_NORMAL_FILE,
wxART_OTHER, wxSize(16,16)));
        treeWindow->AssignImageList(imglist);
        */

        CreateToolBar(wxNO_BORDER | wxTB_FLAT |
wxTB_HORIZONTAL); //create
        InitToolBar(GetToolBar()); //load xpms,
show
        /*
        #if wxUSE_ACCEL
            // Accelerators
            wxAcceleratorEntry entries[3];
            entries[0].Set(wxACCEL_CTRL, (int) 'N', MDI_NEW_WINDOW);
            entries[1].Set(wxACCEL_CTRL, (int) 'X', MDI_QUIT);
            entries[2].Set(wxACCEL_CTRL, (int) 'A', MDI_ABOUT);
            wxAcceleratorTable accel(3, entries);
            SetAcceleratorTable(accel);
        #endif // wxUSE_ACCEL
        */
    }

    void mainFrame::InitToolBar(wxToolBar* toolBar)
    {
        wxBitmap* bitmaps[8];
        bitmaps[0] = new wxBitmap( m01blank_xpm );
    }

```

					ДП.КСМ.19113/11.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		116

```

    bitmaps[1] = new wxBitmap( m02plane_xpm );
    bitmaps[2] = new wxBitmap( m03sphere_xpm );
    bitmaps[3] = new wxBitmap( m04perlin_xpm );
    bitmaps[4] = new wxBitmap( m05voroni_xpm );
    bitmaps[5] = new wxBitmap( m06genbillow_xpm);
    bitmaps[6] = new wxBitmap( m07genridget_xpm);
    bitmaps[7] = new wxBitmap( m08genspheres_xpm);

    int width = 40;
    int currentX = 5;

    toolBar->SetToolBitmapSize( wxSize(32,32) );

    //      AddTool( toolId, const wxString& label
    toolBar->AddTool( MID_ADDIMAGE, *bitmaps[0], wxNullBitmap, false,
currentX, wxDefaultCoord, (wxObject *) NULL, _T("Додати зображення"));
    currentX += width + 5;
    toolBar->AddTool( MID_ADDBUILDPLANE,*bitmaps[1], wxNullBitmap,
false, currentX, wxDefaultCoord, (wxObject *) NULL, _T("Додати плоску
карту"));
    currentX += width + 5;
    toolBar->AddTool( MID_ADDBUILDSPHERE, *bitmaps[2], wxNullBitmap,
false, currentX, wxDefaultCoord, (wxObject *) NULL, _T("Додати сферичну
карту"));
    currentX += width + 5;
    toolBar->AddSeparator();
    toolBar->AddTool( MID_ADDGENPERLIN, *bitmaps[3], wxNullBitmap,
false, currentX, wxDefaultCoord, (wxObject *) NULL, _T("Perlin generator"));
    currentX += width + 5;
    toolBar->AddTool( 5, *bitmaps[4], wxNullBitmap, false, currentX,
wxDefaultCoord, (wxObject *) NULL, _T("Voroni generator"));

```

					ДП.КСМ.19113/11.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		117

```

currentX += width + 5;
toolBar->AddTool( 6, *bitmaps[5], wxNullBitmap, false, currentX,
wxDefaultCoord, (wxObject *) NULL, _T("Billow generator"));
currentX += width + 5;
toolBar->AddTool( 7, *bitmaps[6], wxNullBitmap, false, currentX,
wxDefaultCoord, (wxObject *) NULL, _T("Ridget generator"));
currentX += width + 5;
toolBar->AddTool( 8, *bitmaps[7], wxNullBitmap, false, currentX,
wxDefaultCoord, (wxObject *) NULL, _T("Spheres generator"));

toolBar->Realize();

int i;
for (i = 0; i < 5; i++)
delete bitmaps[i];
}

void mainFrame::CreatePanels(wxSizer *sizer)
{
// -----
treeWindow = new wxTreeCtrl(m_leftPanel1, ID_TREECTRL,
wxDefaultPosition, wxDefaultSize,
wxTR_HAS_BUTTONS|wxTR_SINGLE,
wxDefaultValidator, wxT("treeCtrl"));

propertyWindow = new wxPropertyGrid(m_leftPanel2, ID_PROPGRIDCTRL,
wxDefaultPosition, wxDefaultSize,
wxPG_SPLITTER_AUTO_CENTER, wxPG_DEFAULT_STYLE);
propertyWindow->SetExtraStyle( wxPG_EX_HELP_AS_TOOLTIPS );
}

```

					ДП.КСМ.19113/11.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		118

```

void mainFrame::OnFoldPanelBarDrag(wxSashEvent& event)
{
if (event.GetDragStatus() == wxSASH_STATUS_OUT_OF_RANGE)
    return;
switch (event.GetId())
{
    case ID_LEFT_PANEL1:
    {
        m_leftPanel1->SetDefaultSize( wxSize(event.GetDragRect().width,
1000) );
        break;
    }
    case ID_LEFT_PANEL2:
    {
        m_leftPanel2->SetDefaultSize( wxSize(event.GetDragRect().width, 1000)
);
        break;
    }
}

wxLayoutAlgorithm layout;
layout.LayoutMDIFrame(this);

// Leaves bits of itself behind sometimes
GetClientWindow()->Refresh();
}

void mainFrame::OnSize(wxSizeEvent& WXUNUSED(event) )
{
    wxLayoutAlgorithm layout;
    layout.LayoutMDIFrame(this);
    // Leaves bits of itself behind sometimes

```

					ДП.КСМ.19113/11.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		119

```

        GetClientWindow()->Refresh();
    }

void mainFrame::OnAbout(wxCommandEvent& WXUNUSED(event) )
{
    (void)wxMessageBox(_T("Texture workwxWidgets 2.0 MDI Demo\n")
        _T("Author: Julian Smart (c) 1997\n")
        _T("Usage: mdi.exe"), _T("About MDI Demo"));
}

void mainFrame::OnNewFile(wxCommandEvent& WXUNUSED(event))
{
    // Make another frame, containing a canvas
    mChildWnd *subframe = new mChildWnd( frame, _T("Texture Frame"));
    wxString title;
    title.Printf(_T("Texture Frame %d"), ++gs_nFrames);
    subframe->SetTitle(title);
    subframe->CreateStatusBar();
    subframe->SetStatusText(title);

    // всановити параметри розміщення -----
    int width, height;
    subframe->GetClientSize(&width, &height);
    mTexture *mtexture = new mTexture(subframe, wxPoint(0, 0), wxSize(width,
height));

    subframe->mtexture = mtexture;           // ???
    mtexture->SetScrollbars(20, 20, 50, 50); // Give it scrollbars
    subframe->Show(true);
    Tree = new CMTree;
    Tree->Init( treeWindow );
}

```

					ДП.КСМ.19113/11.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		120

```

        Tree->createRootNode();
    }
    void mainFrame::OnOpenTexture(wxCommandEvent& event)
    {}
    void mainFrame::OnOpenColormap(wxCommandEvent& event)
    {}
    void mainFrame::OnSave(wxCommandEvent& event)
    {}
    void mainFrame::OnCloseFile(wxCommandEvent& event)
    {}
    void mainFrame::OnQuit(wxCommandEvent& WXUNUSED(event))
    {
        Close();
    }
    void mainFrame::OnMDIBlank(wxCommandEvent& WXUNUSED(event))
    {
        //створити дочірнєвікно де буде кінева текстура
        mChildWnd *subframe = new mChildWnd(frame, _T("Texture"));
        subframe->CreateStatusBar();
        subframe->SetStatusText( wxT("Texture Frame"));
        int width, height;
        subframe->GetClientSize(&width, &height);
        mTexture *mtexwnd = new mTexture(subframe, wxPoint(0, 0),
wxSize(width, height));
        // Give it scrollbars
        mtexwnd->SetScrollbars(20, 20, 50, 50);
        subframe->Show(true);
    }
    void mainFrame::OnAdding(wxCommandEvent& WXUNUSED(event))
    {
        wxTreeItemId *item;

```

					ДП.КСМ.19113/11.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		121


```

        item = Tree->insertNode( treeWindow->GetSelection(), IMAGE);
        if ( item->IsOk() ) Tree->insertNode( *item, RENDERER);

    }

void mainFrame::OnAddNoiseMap(wxCommandEvent& event)
{
    wxTreeItemId *item;
    item = Tree->insertNode( treeWindow->GetSelection(), NOISEMAP);
    if (item->IsOk() ) Tree->insertNode( *item, NMAPBUILDERPLANE);
}

void mainFrame::OnAddGenPerlin(wxCommandEvent& event)
{
    Tree->insertNode( treeWindow->GetSelection(), PERLIN);
}

void mainFrame::OnClose(wxCloseEvent& event)
{
    if ( event.CanVeto() && (gs_nFrames > 0) )
    {
        wxString msg;
        msg.Printf(_T("%d windows still open, close anyhow?"),
gs_nFrames);
        if ( wxMessageBox(msg, _T("Please confirm"),
            wxICON_QUESTION | wxYES_NO) != wxYES )
        {
            event.Veto();
            return;
        }
    }
}

```

					ДП.КСМ.19113/11.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		122

```

        event.Skip();
    }

void mainFrame::SelectNode(wxTreeEvent& event)
{
    // 1) отримати вибраний елемент
    wxTreeItemId selectedItem = treeWindow->GetSelection();
    // 2) передати в дерево і отримати параметри вузла і їх значення
    wxArrayString nodeInfo;
    Tree->selectNode( selectedItem, &nodeInfo);
    // 3) заповини таблицю іменами параметрів і значеннями
    if ( nodeInfo.GetCount() <= 1) return;

    propertyWindow->Clear();
    long nType;
    nodeInfo[0].ToLong( &nType, 10 );
    switch( (int) nType )
    {
        case IMAGE:
        {
            // some new controls

            propertyWindow->Append(
wxStringProperty(wxT("Label"),wxT("Name"),wxT("Initial Value"))) );
            long x=0; nodeInfo[1].ToLong( &x, 10 );
            propertyWindow->Append( wxIntProperty(wxT("ширина"),
wxPG_LABEL, x) );
            nodeInfo[2].ToLong( &x, 10 );
            propertyWindow->Append( wxIntProperty(wxT("висота"), wxPG_LABEL, x)
);

            m_leftPanel2->SetSashVisible(wxSASH_RIGHT, true);

        }break;
    }
}

```

					ДП.КСМ.19113/11.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		123

```

        case RENDERER:
        {
            double y=0;
            propertyWindow->Append(
wxStringProperty(wxT("Label"),wxT("Name"),wxT("Initial Value"))) );
            nodeInfo[1].ToDouble(&y);
            propertyWindow->Append( wxFloatProperty( wxT("Lower X Bound"),
wxPG_LABEL, y) );
            nodeInfo[2].ToDouble(&y);
            propertyWindow->Append( wxFloatProperty( wxT("Upper X Bound"),
wxPG_LABEL, y) );
            nodeInfo[3].ToDouble(&y);
            propertyWindow->Append( wxFloatProperty( wxT("Upper Y Bound"),
wxPG_LABEL, y) );
            nodeInfo[4].ToDouble(&y);
            propertyWindow->Append( wxFloatProperty( wxT("Upper Y Bound"),
wxPG_LABEL, y) );
            }break;
        }

        GetClientWindow()->Refresh();
    }
//
// mTexture -----
BEGIN_EVENT_TABLE(mTexture, wxScrolledWindow)
//     EVT_MOUSE_EVENTS(mTexture::OnEvent)
END_EVENT_TABLE()

mTexture::mTexture( wxWindow *parent, const wxPoint& pos, const
wxSize& size)
: wxScrolledWindow(parent, wxID_ANY, pos, size,
                    wxSUNKEN_BORDER |

```

					ДП.КСМ.19113/11.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		124

```

wxNO_FULL_REPAINT_ON_RESIZE |
wxVSCROLL | wxHSCROLL)

{
    SetBackgroundColour(wxColour(_T("WHITE")));
    m_dirty = false;
}

// REPAINTING
void mTexture::OnDraw(wxDC& dc)
{}

// EVENT:MOUSE
void mTexture::OnEvent(wxMouseEvent& event)
{}

//
// mChildWnd -----
// заміть що MDI_NEW_WINDOW та MDI_ABOUT commands get passed
// to the parent window for processing, so no need to
// duplicate event handlers here.
BEGIN_EVENT_TABLE(mChildWnd, wxMDIChildFrame)
    EVT_SIZE( mChildWnd::OnSize)    // СИСТЕМНИЙ ВИКЛИК
    EVT_MOVE( mChildWnd::OnMove)    // СИСТЕМНИЙ ВИКЛИК
    EVT_CLOSE(mChildWnd::OnClose)   // СИСТЕМНИЙ ВИКЛИК
END_EVENT_TABLE()

// CONSTRUCTOR
mChildWnd::mChildWnd(wxMDIParentFrame *parent, const wxString& title)
    : wxMDIChildFrame(parent, wxID_ANY, title,
wxDefaultPosition, wxDefaultSize,
wxDEFAULT_FRAME_STYLE |
wxNO_FULL_REPAINT_ON_RESIZE)
{

```

					ДП.КСМ.19113/11.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		125

```

mtexture = (mTexture *)NULL;
my_children.Append(this);
// this should work for MDI frames as well as for normal ones
SetSizeHints(100, 100);
}
// DESTRUCTOR
mChildWnd::~mChildWnd()
{
    my_children.DeleteObject(this);
}

// ACTIVATE
void mChildWnd::OnActivate(wxActivateEvent& event)
{
    if ( event.GetActive() && mtexture )
        mtexture->SetFocus();
}

void mChildWnd::OnMove(wxMoveEvent& event)
{
    // to be the width of the MDI canvas border)
    wxPoint pos1 = event.GetPosition(), pos2 = GetPosition();
    wxLogStatus(wxT("position from event: (%d, %d), from frame (%d,
%d)"),
        pos1.x, pos1.y, pos2.x, pos2.y);
    event.Skip();
}

void mChildWnd::OnSize(wxSizeEvent& event)
{
    // VZ: under MSW the size event carries the client size (quite
    // unexpectedly) *except* for the very first one which has the full

```

					ДП.КСМ.19113/11.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		126

```

// size... what should it really be? TODO: check under wxGTK
wxSize    size1 = event.GetSize(),
          size2 = GetSize(),
          size3 = GetClientSize();

wxLogStatus(wxT("size from event: %dx%d, from frame %dx%d, client
%dx%d"),
size1.x, size1.y, size2.x, size2.y, size3.x, size3.y);

event.Skip();
}

// CLOSE
void mChildWnd::OnClose(wxCloseEvent& event)
{
    if ( mtexture && mtexture->IsDirty() )
    {
        if ( wxMessageBox(_T("Really close?"), _T("Please confirm"),
wxICON_QUESTION | wxYES_NO) != wxYES )
        {
            event.Veto();
            return;
        }
    }
    gs_nFrames--;
    event.Skip();
}

```

					ДП.КСМ.19113/11.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		127

ДОДАТОК Б СХЕМА СТРУКТУРНА ГІБРИДНОЇ ІНТЕЛЕКТУАЛЬНОЇ
СИСТЕМИ АНАЛІЗУ БІОМЕДИЧНИХ ЗОБРАЖЕНЬ

					ДП.КСМ.19113/11.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		128

ДОДАТОК В БЛОК-СХЕМА АЛГОРИТМУ ПОБУДОВИ БАЗИ ПРАВИЛ НА
ОСНОВІ ЧИСЕЛЬНИХ ДАНИХ

					ДП.КСМ.19113/11.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		129

ДОДАТОК Г БЛОК-СХЕМА АЛГОРИТМУ ОБЧИСЛЕННЯ МАТРИЦЬ
ІМОВІРНОСТІ РОЗПОДІЛУ РІВНІВ ЯСКРАВОСТІ

					ДП.КСМ.19113/11.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		130

ДОДАТОК Д БЛОК-СХЕМА АЛГОРИТМУ ТЕКСТУРНОЇ СЕГМЕНТАЦІЇ

					ДП.КСМ.19113/11.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		131

ДОДАТОК Е ДОВІДКА ПРО ВИКОРИСТАННЯ РЕКОМЕНДАЦІЙ
ДИПЛОМНОГО ПРОЕКТУ

					ДП.КСМ.19113/11.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		132