

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

БАРАН Євген Романович

«Алгоритми розпізнавання елементів текстових документів на основі стоншення контурів / Algorithms for text documents elements recognizing based on contour thinning»

Студент групи КІм – 21
БАРАН Євген Романович

Науковий керівник
к.т.н., Ю.М. Батько

Тернопіль – 2020

РЕЗЮМЕ

Кваліфікаційна робота на тему “Алгоритми розпізнавання елементів текстових документів на основі стоншення контурів” зі спеціальності 123 «Комп’ютерна інженерія» освітнього ступеня «магістр» написана обсягом 90 сторінок і містить 28 ілюстрації, 4 таблиць, 2 додатків та 50 джерел за переліком посилань.

Метою роботи є розробка алгоритму розпізнавання текстової інформації на основі порівняння скелетів зображень з еталонними моделями символів.

Методи досліджень. Для розв’язання поставлених задач у кваліфікаційній роботі використано: методи: лінійної алгебри та аналітичної геометрії (для створення моделей опису зображень); афінних перетворень (для перетворення елементів зображень); контурного аналізу (для отримання контурів об’єктів та їх скелетів); об’єктно-орієнтованого програмування (для проектування програмних засобів аналізу та розпізнавання зображень).

Результати дослідження: алгоритм опису та розпізнавання текстової інформації на цифрових зображеннях на основі порівняння їх скелетів з еталонними моделями, програмна система аналізу та розпізнавання зображень.

Результати роботи можуть бути використані в створенні нових зображень, цифровому документообігу, бібліотечній справі, індустрії створення комп’ютерних ігор, для наукових досліджень та в навчальному процесі.

Орієнтовні напрямки розвитку досліджень: розроблення програмних систем для підвищення якості цифрових зображень на етапі оцифрування та попередньої обробки, створення нових моделей для представлення символів довільної складності для шаблонів для навчання нейронних мереж.

КЛЮЧОВІ СЛОВА: ПЕРЕТВОРЕННЯ ЗОБРАЖЕННЯ, РОЗПІЗНАВАННЯ ТЕКСТУ, ПРОГРАМНА СИСТЕМА.

RESUME

Graduate qualification work on “ Algorithms for text documents elements recognizing based on contour thinning ” specialty 123 - Computer Engineering is 90 pages long and contains 28 illustrations, 4 tables, 2 appendices and 50 references.

The aim of the work is to develop an algorithm for recognizing text information based on the comparison of image skeletons with reference models of symbols.

Research methods. To solve the tasks in the qualification work used: methods: linear algebra and analytical geometry (to create models of image description); affine transformations (for the transformation of image elements); contour analysis (to obtain the contours of objects and their skeletons); object-oriented programming (for designing software for image analysis and recognition).

Research results: algorithm for description and recognition of textual information on digital images based on comparison of their skeletons with reference models, software system for analysis and image recognition.

The results of the work can be used in the creation of new images, digital document management, librarianship, the computer game industry, for research and in the educational process.

Indicative directions of research development: development of software systems to improve the quality of digital images at the stage of digitization and pre-processing, creation of new models for the representation of symbols of arbitrary complexity for templates for learning neural networks.

KEYWORDS: IMAGE CONVERSION, TEXT RECOGNITION, SOFTWARE SYSTEM.

ЗМІСТ

Вступ.....	7
1 Програмні, апаратні та поєднані системи обробки цифрових зображень	10
1.1 Контурний аналіз, основні алгоритми, поняття та функції	10
1.2 Аналіз методів та алгоритмів стоншення об'єктів	22
1.3 Програмні системи для аналізу, синтезу та обробки зображень	24
1.4 Постановка задач дослідження.....	28
1.5 Висновки до розділу.....	29
2 Методи та алгоритми розпізнавання текстової інформації на зображеннях ...	30
2.1 Методи та алгоритми розпізнавання текстової інформації	30
2.2 Алгоритми виділення контурів на основі їх проходження	39
2.3 Алгоритм розпізнавання текстової інформації на зображеннях	45
2.5 Висновки до розділу.....	50
3 Програмна система аналізу та розпізнавання цифрових зображень	51
3.1 Структура програмної аналізу та розпізнавання зображень.....	51
3.2 Програмні модулі системи перетворення елементів зображення.....	60
3.3 Тестування та аналіз реалізованої системи.....	66
3.4 Аналіз системи захисту доступу до системи	70
3.5 Висновки до розділу.....	72
Висновки.....	73
Список використаної літератури.....	74
Додаток А Лістинг коду програми	78
Додаток Б Світокопії виданих публікацій	85

ВСТУП

Актуальність роботи. У сучасному світі кожен день перекладається з паперу в електронну форму величезна кількість різних документів друковані тексти, платіжні доручення, митні або податкові декларації, бюлетені для голосування, різні анкети і безліч інших. Активно використовуються тисячі різних систем електронного документообігу практично у всіх сферах діяльності. При сучасних обсягах потоків документів подібні операції немислимі без автоматизованої обробки.

Завдяки загальної комп'ютеризації та поширенню електронного документообігу в різних областях людської діяльності накопичені величезні архіви текстової та візуальної інформації. Безперервно розширюється електронним архівом є глобальна мережа Internet. Аналіз сучасних інформаційних систем дозволив зробити висновок про обмежені можливості семантичного аналізу та пошуку зображень. Під семантичним аналізом зображень розуміється автоматичне отримання їх семантичних описів (анотацій) і пошук в просторі цих описів (пошук за змістом). З реалізованих інформаційними системами видів пошуку пошук зображень по ключовими словами найбільш близький до змістовного пошуку, але володіє одним істотним недоліком - ключові слова до зображень формує експерт.

У всіх системах електронного документообігу та системах введення друкованих текстів одним з ключових етапів є розпізнавання текстових символів - переклад інформації з графічної форми - результату сканування - в текстову форму. Незважаючи на багаторічну історію розвитку алгоритмів розпізнавання і існування великої кількості алгоритмів, добре розпізнають чітко надруковані тексти, задача розпізнавання в більш складних випадках далека від вирішення. Виникає запитання подальшого збільшення точності розпізнавання документів поганої якості зокрема, існуючі алгоритми забезпечують відносно невисоку в порівнянні з людиною точність

розпізнавання текстів з графічних зображень, отриманих скануванням з малими дозволами Варто відзначити клас задач, в яких наявне графічне зображення неможливо поліпшити шляхом збільшення дозволу сканування або зміною параметрів сканування До таких завдань відносяться вже створені раніше електронні архіви документів у вигляді растрових зображень, електронні бібліотеки, факсимільні повідомлення тощо

Таким чином, розробка нових високоточних алгоритмів розпізнавання текстів, так само як поліпшення вже існуючих, представляється потенційно корисним завданням. Тому задача створення алгоритму розпізнавання зображень на основі контурного аналізу є актуальною.

Метою роботи є розробка алгоритму розпізнавання текстової інформації на основі порівняння скелетів зображень з еталонними моделями символів.

Для досягнення даної мети ставились наступні завдання:

- проаналізувати методи та алгоритми контурного аналізу;
- провести аналіз підходів до стоншення об'єктів на цифрових зображеннях;
- провести дослідження існуючих програмних систем синтезу, аналізу та обробки цифрових зображень;
- проаналізувати існуючі алгоритми перетворення елементів цифрових зображень;
- розробити алгоритм розпізнавання текстової інформації цифрових зображень на основі аналізу контурів;
- реалізувати програмну систему розпізнавання текстової інформації на цифрових зображень, провести її тестування та порівняти з програмами аналогами.

Об'єкт дослідження – процес обробки цифрових зображень.

Предмет дослідження – методи і алгоритми розпізнавання текстової інформації на цифрових зображеннях.

Наукова новизна одержаних результатів визначається наступним чином:

– проведено комплексний аналіз та класифікацію алгоритмів розпізнавання текстової інформації на зображеннях, що дозволило виділити їх переваги та недоліки та розробити власний алгоритм розпізнавання текстової інформації на основі підходів контурного аналізу;

– розроблено алгоритм розпізнавання текстової інформації на основі підходів контурного аналізу, що дозволило зменшити обчислювальну складність процесу обробки зображень.

Практична цінність одержаних результатів полягає в тому, що:

– розроблено та проведено моделювання програмної системи розпізнавання текстової інформації на основі підходів контурного аналізу, що дозволило в подальшому програмно реалізувати та провести дослідження запропонованих алгоритмів;

– реалізовано програмне забезпечення для аналізу та розпізнавання текстової інформації на цифрових зображеннях на основі мови програмування Delphi та з використанням алгоритмів контурного аналізу.

Публікації та апробація до випускної кваліфікаційної роботи. За результатами наукових досліджень, проведених у випускній кваліфікаційній роботі, підготовлено тези доповіді [1,2]. «Аналіз алгоритмів проходження контурів в системах обробки та розпізнавання цифрових зображень» обсягом 1 сторінка на III Науково-практичній конференції молодих вчених і студентів «Інтелектуальні комп'ютерні системи та мережі», а також «Аналіз алгоритмів виділення та розпізнавання об'єктів на цифрових зображеннях» обсягом 1 сторінка на III Науково-практичній конференції молодих вчених і студентів «Інтелектуальні комп'ютерні системи та мережі».

1 ПРОГРАМНІ, АПАРАТНІ ТА ПОЄДНАНІ СИСТЕМИ ОБРОБКИ ЦИФРОВИХ ЗОБРАЖЕНЬ

1.1 Контурний аналіз, основні алгоритми, поняття та функції

Контури повинні бути пов'язані для відображення межі регіону. Це подання називається контуром. Контур може бути відкритим або закритим. Відзначимо, що контури відповідають межах області, а пікселі в області можуть бути знайдені за алгоритмом заповнення. Відкритий контур може бути частиною регіону її кордон. Проміжки можуть виникати на межі регіону через те що контраст між яскравістю регіонів може бути недостатньо для того, щоб дозволяти бути ребрам по межі знайденим крайовим детектором. Можливо, поріг при виявленні краю був встановлено занадто високо, або може бути контрастність на якійсь ділянці межі настільки слабка щодо інших областей зображення, що жоден поріг не працює скрізь на зображенні. Розкриті контури також трапляються, коли є фрагменти лінії, що пов'язані між собою - наприклад, коли фрагменти лінії пов'язані уздовж штриху на кресленні або зразку почерку.

Контур може бути представлений як упорядкований перелік ребер або кривою. Крива - це математична модель для контуру. Приклади кривих включають відрізки ліній та кубічні сплайни. Існує кілька критеріїв хорошого контуру, а саме:

Ефективність: Контур повинен бути простим, компактним зображенням.

Точність: контур повинен точно відповідати особливостям зображення.

Ефективність: Контур повинен відповідати операціям, які будуть виконуватися на пізніх стадіях застосування.

Точність подання контуру визначається формою кривої, що використовується для моделювання контуру, за показником кривої алгоритму, а також за точністю оцінок розташування краю.

Найпростішим представленням контуру є впорядкований список його ребер. Це подання є не настільки точним, для оцінки розташування країв, але є найменш компактним представленням і може не забезпечити ефективного представлення для подальшого аналізу зображень. Підгонка відповідної моделі кривої до границі збільшує точність, оскільки помилки в розташуванні краю зменшуються завдяки усередненню, і це підвищує ефективність за рахунок надання більш доцільних і більш компактних представлень для подальших операцій. Наприклад, набір ребер, які лежать уздовж лінії, можна найефективніше зобразити, підігнавши лінію по формі до форми країв. Це подання спрощує подальші обчислення, такі як визначення орієнтації або довжини лінії та підвищення точності. Також, оскільки середня квадратична помилка між оцінюваною лінією та справжньою лінією буде меншою за похибку між справжньою лінією та будь-яким із ребром.

Крива інтерполює список точок, якщо крива проходить через точки. Похибка наближення підходить для кривої, якщо до списку точок з кривою, що проходить близько до точок, але не обов'язково проходження точно через точки. Ребра, забезпечують виявленням країв, що застосовуються до реальних зображень, можуть бути не точним. У передбачуваному розташуванні краю буде помилка.

Список ребер - це впорядкований набір крайових точок або фрагментів.

Контур - це список ребер або кривих, які використовуються для представлення списку граничних точок.

Межа - це замкнутий контур, який оточує область.

Орієнтація границі не використовується більшістю алгоритмів підгонки кривих. У тих небагатьох випадках де алгоритм використовує орієнтацію краю, це буде зрозуміло з контексту, що термін ребра відноситься до фрагментів ребер.

Криві на площині можуть бути представлені трьома різними способами:

– явна форма $y = f(x)$;

- неявна форма $f(x,y) = 0$;
- параметрична форма $(x(u), y(u))$, для деякого параметра u .

Явна форма рідко використовується в машинному зорі оскільки крива в площині x - y може крутитися навколо таким чином, що може бути більше однієї точки на кривій для даного x .

Параметрична форма кривої використовує дві функції, $x(u)$ та $y(u)$, та параметр u , щоб вказати точку вздовж кривої з початкової точки крива при $P_1 = (X(U1), y(U1))$ до кінцевої точки $P_2 = (X(U2), y(U2))$. При цьому довжина кривої задається довжиною дуги:

$$\int_{u1}^{u2} \sqrt{\left(\frac{dx}{du}\right)^2 + \left(\frac{dy}{du}\right)^2} du. \quad (1.1)$$

Одиницею дотичного вектора є

$$t(u) = \frac{p'(u)}{|p'(u)|}, \quad (1.2)$$

де $p(u) = (x(u), y(u))$.

Кривизна кривої є похідною від тангенс: $n(u) = p''(u)$.

Нахил та кривизну важко обчислити саме в цифровому домені, оскільки кут між сусідніми пікселями становить квантовано з кроком 45° . Основна ідея - оцінити орієнтацію дотичної за допомогою точок ребра які не суміжні в списку крайок. Це дозволяє збільшити набір можливих дотичних орієнтації. Нехай $P_i = (X_i, Y_i)$ - координати ребра i в крайовий список. K -нахил - це (кут) вектор напрямку між точками, які є k країв нарізно. Лівий k -схил - це напрямок від P_i до P_{i-k} , правий k -нахил - це напрямок від P_i до P_{i+k} . K -кривизна - це різниця між лівим і правим k -схилами.

Припустимо, що в списку ребер є n точок ребра $(X_1, Y_1), \dots, (X_n, Y_n)$. Тоді довжину цифрової кривої можна наблизити, додавши довжини окремі сегменти між пікселями: $(X_i - X_{i-1})^2 + (Y_i - Y_{i-1})^2$. Хороше наближення виходить шляхом обведення списку країв та додавання 2 вздовж боків і 3 по діагоналях, і ділимо остаточно суму на 2.

Ланцюгові коди - це позначення для запису списку крайових точок уздовж контуру. Ланцюговий код задає напрямок контуру на кожному ребрі в крайовий список. Напрямки квантовані в один із восьми напрямків. Починаючи з першого краю в списку і рухаючись за годинниковою стрілкою по контуру, за напрямком до наступного ребра задається за допомогою одного з восьми ланцюгових кодів. Напрямок - це ланцюговий код для 8-сусіда краю.

Код ланцюжка представляє список ребер за координатами першого ребра і список ланцюгових кодів, що ведуть до наступних ребер. Ланцюговий код має деякі переважливі властивості. Поворот об'єкта на 45° можна легко реалізувати. Якщо об'єкт повернутий на $N \times 45^\circ$, тоді код для обертаного об'єкта отримується додаванням n мода 8 до оригіналу код. Отримано похідну ланцюгового коду, яку також називають різницевим кодом використовуючи першу різницю, є інваріантним описом межі опису. Дещо інші характеристики регіону, такі як площа та кути, можуть бути безпосередньо обчислені за допомогою ланцюгового коду. Обмеженням цього подання є обмежений набір напрямків, що використовуються для представлення дотичної в точці. Це обмеження можна зняти, використовуючи одне із інших представлень зображень кривих. Після того, як крива була підлаштована до списку ребер, можна обчислити будь-яку з геометричних величин, за математичною формулою для кривої.

Відображення нахилу контуру, який також називають W -s сюжетом, схоже на безперервну версію ланцюгового коду. В аному випадку хочемо представити контур за допомогою довільних дотичних напрямків, а не обмеженого набору дотичних напрямків як це дозволяється ланцюговим кодом. Припустимо, що починаємо з початку краю перерахувати та обчислити тангенс та довжину дуги, використовуючи формули, представлені для цифрові криві.

Користувач можемо побудувати тангенс W щодо довжини дуги s , щоб отримати подання для контуру в просторі $W-s$. Графік $W-s$ - це зображення форми контуру. Наприклад, контур, який складається з відрізки ліній та кругові дуги виглядатимуть як послідовність відрізків у $W-s$ сюжет. Горизонтальні відрізки лінії в $w-s$ графіку відповідають прямій сегменти в контурі; відрізки лінії в інших орієнтаціях в $w-s$ сюжеті відповідають круговим дугам. Частини сюжету $w-s$, які не є прямими рядками відповідають іншим примітивам кривих.

Контур можна розділити на прямі лінії та кругові дуги, сегментуючи $w-s$ графік на прямі лінії. Цим методом користувалися багато дослідників, і існує кілька версій цього підходу для розділення контуру на сегменти.

Можна використовувати $w-s$ графік як компактний опис форми оригінального контуру. На рисунку 1.1 наведено контур та його $w-s$ графік. Для замкнутий контур, $w-s$ графік періодичний.

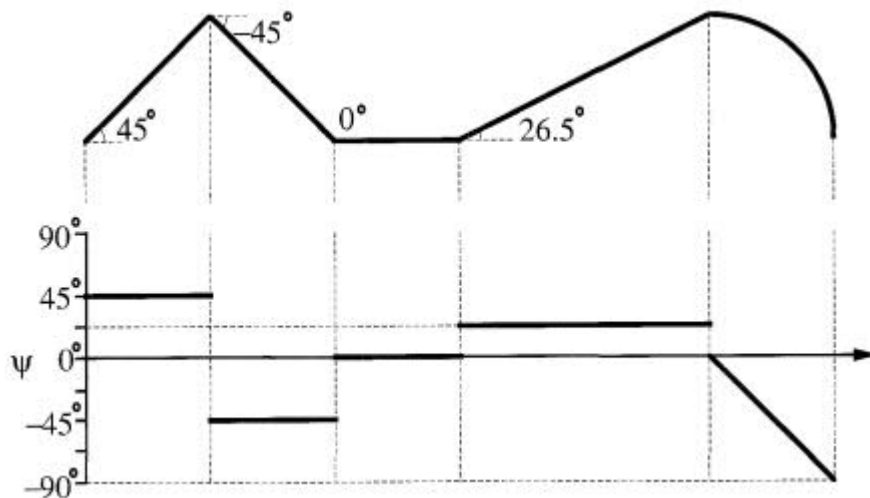


Рисунок 1.1 – Приклад представлення контуру

Для більш точного представлення контуру розглянемо чотири моделі кривих та методи для підгонка моделей до крайових точок. Моделі включають:

- відрізки ліній;
- кругові дуги;

- конічні перерізи;
- кубічні сплайни.

Будь-який алгоритм підгонки повинен вирішувати два питання: який метод застосовується для підгонки кривої до країв та як вимірюється точність передачі контуру.

Нехай d_i - відстань крайової точки i від прямої. Їх декілька мір наближення кривої до крайових точок кандидата. Всі з них залежать від похибки між встановленою кривою та кандидатом точки, що утворюють криву. Нижче наводяться деякі загальноживані методи.

Максимальна абсолютна похибка вимірює (1.3), наскільки відхиляються точки кривої в найгіршому випадку:

$$MAE = \max_i |d_i|. \quad (1.3)$$

Середня квадратична похибка (1.4) дає загальну міру відхилення кривої від крайових точок:

$$MSE = \frac{1}{n} \sum_{i=1}^n d_i^2. \quad (1.4)$$

Нормована максимальна похибка (1.5) - це відношення максимальної абсолютної похибки до довжини кривої:

$$e = \frac{\max |d|}{S}. \quad (1.5)$$

Кількість знакових змін помилки є хорошим показником доцільності кривої як моделі для країв у контурі.

Співвідношення довжини кривої до відстані до кінцевої точки є хорошою мірою складності кривої.

Нормована максимальна похибка забезпечує одиничну міру похибки, незалежно від довжини кривої. Іншими словами, дана сума відхилення від кривої може бути однаково значним у деяких додатках, як вдвічі більше відхилення від кривої, яка вдвічі довша. Якщо крива модель - це відрізок лінії, тоді не потрібно обчислювати довжину дуги, а можна використовувати відстань D між кінцевими точками:

$$D = \sqrt{(y_n - y_1)^2 + (x_n - x_1)^2}. \quad (1.6)$$

Зміни знаків є дуже корисним вказівкою на придатність. Помістіть список крайові точки прямою лінією та вивчіть кількість змін знака. Один змін знака вказує на те, що список ребер може бути змодельований відрізком лінії, дві зміни знака вказують на те, що ребра повинні бути змодельовані квадратично кривою, три зміни знака вказують на кубічну криву тощо. Чисельно знаки зміни вказують на те, що невелике збільшення складності кривої буде істотно не покращують придатність. Хороша підгонка має випадковий шаблон до зміни знака. Помилки одного і того ж знака вказують на систематичну помилку в примірці; можливо через неправильну модель кривої.

Вибір моделі відображення кривої повинен ґрунтуватись на конкретному завданні.

– використання відрізків прямих ліній (поліліній) доцільно, якщо складається сцена з прямих ліній і є відповідною точкою для підгонки інших моделей;

– дуги є корисним поданням для оцінки кривизни, оскільки крива є сегментовані на ділянки з кусково постійною кривизною;

– конічні перерізи забезпечують зручний спосіб представити послідовності відрізків лінії та кругові дуги, а також еліптичні та гіперболічні дуги, і явно представляють перегин балів;

– кубічні сплайни гарні для моделювання плавних кривих і не зусилля оцінки дотичних векторів та кривизни мають бути кусково постійними.

Полілінія - це послідовність відрізків лінії, з'єднаних кінцем до кінця. Полілінія подання для контуру, що відповідає списку ребер із послідовністю відрізків рядка. Полілінія інтерполірує вибрану підмножину точок ребра в список ребер. Кінці кожного відрізка - це точки ребер у вихідному списку ребер. Кожен відрізок лінії моделює пробіг суміжних ребер між його кінцевими точками. Точки з'єднання відрізків прямих називаються вершинами. Зауважимо, що полілінії - це двовимірні криві в площині зображення, як і всі криві, а вершини - точки на зображенні площині.

Алгоритм полілінії бере як вхід впорядкований список точок ребра $\{(X_1, Y_1), (X_2, Y_2), \dots, (X_n, Y_n)\}$. Координати точки краю можуть бути обчислені з роздільною здатністю субпікселів (рисунок 1.2). Оскільки відрізки лінії підходять між двома краями, то ребра, вибрані як вершини, повинні бути точно обчислені.

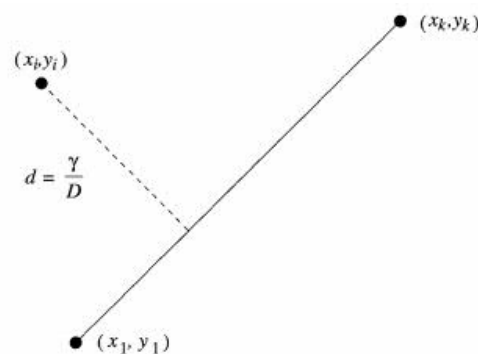


Рисунок 1.2 – Вимірювання відстані від точки до лінії

Формула відрізка лінії, який апроксимує список точок ребра і поєднує першу і останню крайові точки (X_l, Y_l) і (X_k, Y_k) можна отримати зазначивши,

що нахил лінії між кінцевими точками такий самий, як нахил прямої між першою точкою і довільною точкою вздовж рядок:

$$\frac{y - y_1}{x - x_1} = \frac{y_k - y_1}{x_k - x_1}. \quad (1.7)$$

Множення та перестановка термінів дає неявну форму для рядка сегмент, параметризований координатами кінцевих точок. Відстань будь-якої точки (X_l, Y_l) від прямої $d = r / D$, де r і D - відстань між кінцевими точками. Нормована максимальна похибка часто використовується як міра для придатності прилягання відрізка до набору ребер. Всі з ці формули передбачають, що перпендикулярна проекція точки на лінію знаходиться в межах відрізка лінії; тобто як на лінії, так і між кінцем точки відрізка. Це стосується ситуацій, пов'язаних із цим, але в інших випадках для обчислення може знадобитися модифікація формули відстань точки від найближчої кінцевої точки відрізка.

Існує два підходи до підгонки поліліній: розбиття зверху вниз та знизу вгору злиття.

Алгоритм розбиття зверху вниз рекурсивно додає вершини, починаючи з початкова кривої. Початковою кривою є відрізок лінії між першою та останньою крайовими точками, позначеними A та B . Знайдено крапку в списку ребер, що знаходиться найдалше від прямої. Якщо нормалізована максимальна похибка перевищує поріг, тоді вершина вставляється в крайова точка, найдалша від відрізка лінії, позначена як точка C (рисунок 1.3).

Алгоритм розбиття рекурсивно застосовується до двох нових сегментів рядків і крайовий список. Список країв розділений на два відповідні списки до двох відрізків. Крайові точки у списку, які є найдалшими кожен відрізок знайдено, і нові вершини вводяться, якщо точки є занадто далеко від відрізків лінії. Алгоритм розбиття полілінії завершується коли нормована максимальна похибка, для всіх точок ребра вздовж полілінії, знаходиться нижче порога. Ця

рекурсивна процедура є дуже ефективною. Сегмент розщеплення також називають рекурсивним підрозділом.

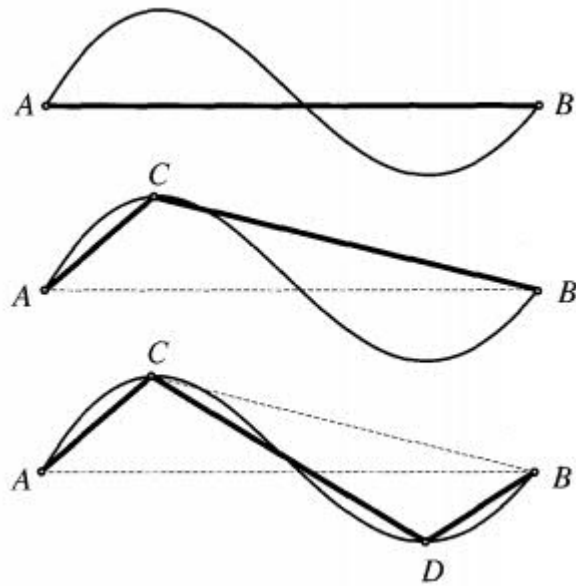


Рисунок 1.3 – Приклад наближення масву кривих до контурної лінії

Нові відрізки запускаються, коли крайові точки також відхиляються далеко від відрізка лінії. Підхід злиття також називається «знизу вгору» для підгонки полілінії. Існує кілька заходів, за допомогою яких можна визначити, чи є край точка знаходиться занадто далеко від відрізка лінії, що формується. Одним із методів є використовувати послідовні найменші квадрати, які виконують відповідність найменших квадратів лінії відрізком до точок ребра та оновлює параметри відрізка лінії поступово по мірі обробки кожної нової точки ребра. Алгоритм підгонки обчислює квадратний залишок між відрізком лінії та крайовими точками. Коли помилка перевищує порогове значення, вводиться вершина та новий сегмент починається з кінцевої точки останнього сегмента.

Алгоритм смуги допусків використовує інший метод визначення розміщення вершин. Два відрізки, паралельні прямій відрізок, наближений до крайових точок на відстані E від центральної лінії сегмент обчислюються. Значення E являє собою абсолютна величина відхилення від встановленої лінії, яка допускається (рисунок 1.4).

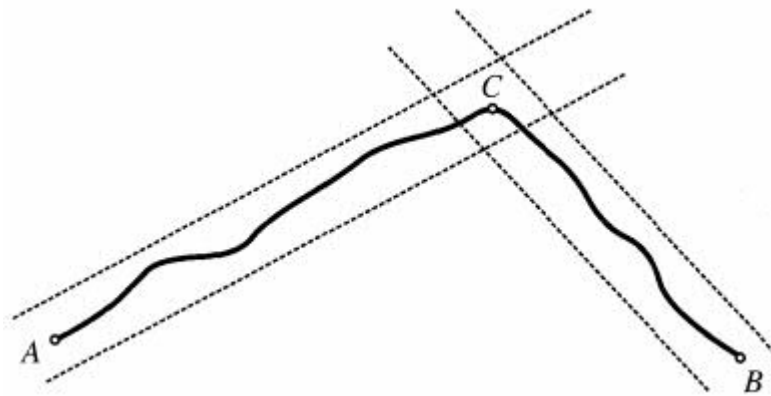


Рисунок 1.4 – Приклад алгоритму смуг допуску

Краї додаються до поточного відрізка, поки нові ребра знаходяться всередині смуга допуску. Параметри сегмента лінії можуть бути перераховані як нові ребра додаються до сегмента. Апроксимуючий відрізок лінії не повинен залишатися паралельно сторонам смуги допуску. Вершина в кінці відрізка - початкова точка для наступного відрізка. Цей підхід, як правило, призводить до великої кількості сегментів. Розташування кутів та кути не точно оцінюються, оскільки вершина не створюється до алгоритму обробив ребра до межі смуги допуску.

Метод рекурсивного поділу зверху вниз і метод злиття знизу вгору може поєднуватися як алгоритм розбиття та злиття. Розщеплення і методи злиття лише частково успішні, якщо їх використовувати самостійно, але точність наближення відрізка лінії до списку ребер може бути покращено за допомогою чергування операцій злиття та розбиття. На рисунку 1.5 показано приклад, коли розбиття з наступним злиттям може відновити неправильно розміщену вершину. Основна ідея полягає в чергуванні проходів розділення та об'єднання. Після рекурсивного підрозділ, дозволити замінити сусідні сегменти на один сегмент між першою та останньою кінцевими точками, якщо новий сегмент відповідає краям з менш нормована помилка. Зверніть увагу, що необхідно використовувати нормалізовану помилку, оскільки кілька сегментів рядків

завжди вміщуватимуть список ребер з меншою похибкою, ніж для один відрізок рядка.

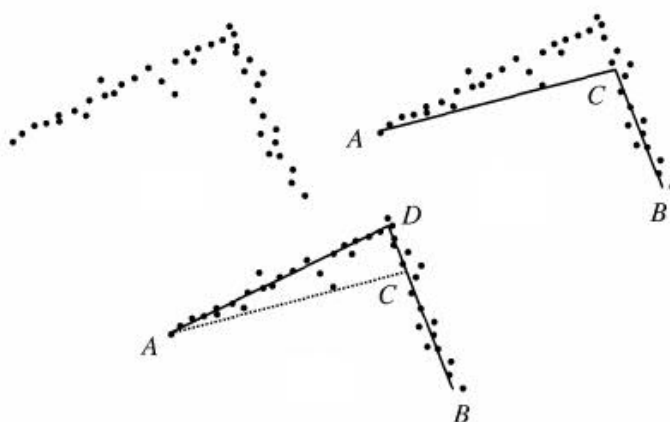


Рисунок 1.5 – Приклад поєднання двох підходів «зверху-вниз» та «знизу - вгору»

Після об'єднання сегментів новий сегмент може бути розділений в інший момент.

Справжнє місце кута можна відремонтувати шляхом розділення та об'єднання проходів що розділяють перший відрізок у точці, що знаходиться ближче до справжнього кута, а потім об'єднати два сегменти в один сегмент рядка.

На сьогоднішній день контурний аналіз широко використовується в різних сферах, серед яких наука, інженерні роботи, інформаційні технології, системи з штучним інтелектом. Основною перевагою використання даних підходів є їх простота та швидкість роботи, а також мінімальний вплив зовнішніх факторів на результати роботи програмних систем з елементами контурного аналізу. Тому саме алгоритми даної групи доцільно використовувати для проектування та програмної реалізації системи аналізу та опису текстової інформації на цифрових зображеннях. А саме доцільно використати алгоритми стоншення моделей шляхом отримання їх скелетону.

1.2 Аналіз методів та алгоритмів стоншення об'єктів

В даний час існує багато алгоритмів побудови скелета [1], але в абсолютній більшості з них вхідне зображення повинно бути оброблено повністю. Поряд з цим, сучасні апаратні прилади дозволяють отримати зображення високої точності, що веде до збільшення оброблюваних даних. Наприклад, аналізовані зображення структур можуть займати кілька гігабайт. Такі зображення не завжди можуть бути завантажені в пам'ять комп'ютера, і для таких зображень повинні бути запропоновані спеціальні методи обробки і аналізу. Сучасні завдання стоншення напівтонових об'єктів полягає у швидкій і якісній обробці тривимірних зображень. Результатом рішення задачі повинні бути бінарні скелети об'єктів, що зберігають їх топологічні особливості. Топологічний опис об'єктів на зображенні відповідає опису форми. Основними особливостями є терміни з цих областей. Так, для області аналізу зображень це піки, плато, хребет, локальні мінімуми і максимуми, які відповідають западині і пагорба. На основі цих понять розроблено декілька алгоритмів розпізнавання об'єктів [1-3]. Наприклад, алгоритм вододілу визначає хребти та впадини. В даному випадку найбільш важливим поняттям є хребти, так як відповідають середньої лінії об'єктів на зображенні і зберігають всі топологічні властивості. В результаті для складання топологічного опису об'єктів, як правило, використовуються оцінки, обчислювані на основі властивостей середньої лінії об'єкта та його кордонів [4]. Найбільш ефективні алгоритми виділення середньої лінії (скелета) і кордонів засновані на операціях математичної морфології [5]. Скелет грає важливу роль при описі топологічних характеристик, але операція утоншення бінарного зображення призводить до неповного скелету і втрати багатьох топологічних особливостей. У той же час алгоритмів для стоншення напівтонових і кольорових зображень трохи і якість більшості з них залежить від топологічних особливостей об'єктів. Основною складністю в побудові скелета є збереження глобальних характеристик скелета

(наприклад, центральність) під час локальних перетворень. На рисунку 1.6 наведено послідовність кроків для отримання скелету об'єкта на цифровому зображенні.

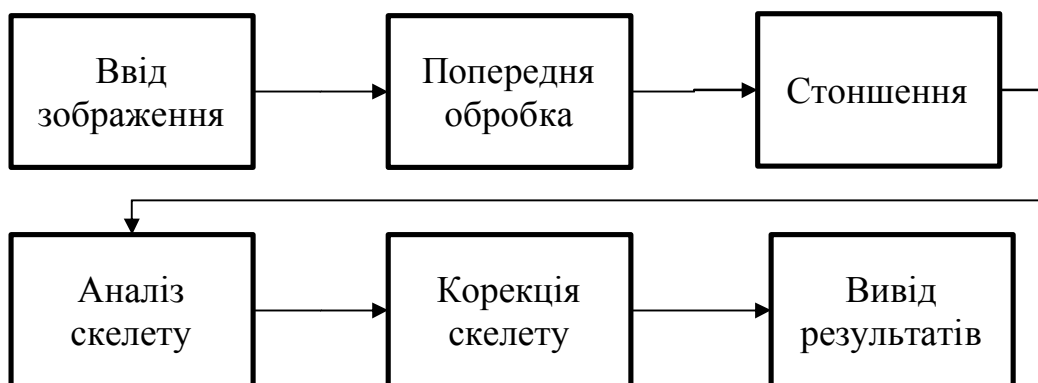


Рисунок 1.5 – Послідовність кроків при скелетизації об'єктів на цифровому зображенні

Для вирішення завдання скелетизації півтонування зі збереженням топологічних особливостей використовують алгоритм, що включає наступні кроки: побудова псевдодистанційної карти відстаней; визначення основних топологічних особливостей; побудова напівтонового скелету; бінаризація скелета. Побудова карти відстаней для напівтонових зображень досить складне завдання. Це пов'язано з тим, що збільшується розмірність. Для побудови скелета не потрібні дистанційні властивості [7]. Скелет відображає топологічні властивості об'єкта, тобто повинен проходити через точки хребта. Це означає, скелет повинен проходити через точки локальних максимумів або пікові точки (які не мають сусідів, що перевищують їх за величиною); точки сідла (через які в їх околиці пролягає єдино можливий шлях між їх максимальними сусідами); вузлові точки сідла (єдиним чином з'єднують більше трьох сусідів, рівних або перевищують їх за величиною) і не повинен проходити через точки локальних мінімумів (що не мають сусідів з меншою величиною). Основні топологічні властивості зображення повинні бути чітко відображені в карті, по якій буде

будуватися скелет, а також включати локальні максимуми і мінімуми, вузлові точки і точки сідла [8]. Розглянемо півтонування як набір шарів, коли кожен нижчий шар включає пікселі попереднього шару. Стандартне півтонування в традиційній обчислювальній техніці має 256 рівнів. Таким чином, якщо кожен бінарний шар збільшити на 256 значень, буде досить простору для побудови дистанційної карти відстаней кожного шару. В результаті маємо набір з 256 шарів зображення з діапазоном значень пікселів від 0 до 255 для кожного шару.

1.3 Програмні системи для аналізу, синтезу та обробки зображень

На сьогоднішній день на ринку присутня велика кількість програмних систем для синтезу, редагування та подільшої обробки цифрових зображень. В даному підрозділі буде проаналізовано наявне програмне забезпечення для розпізнавання документів із графічного формату в текстовий. Безумовно, програма програмного забезпечення для розпізнавання тексту є Abbyy FineReader, але для того, щоб підтвердити це, проведено порівняльний аналіз ще кількох програм для розпізнавання тексту з графічного файлу. У результаті було отримано можливість, по-перше, оцінити ринок доступних програмних систем для розподілу тексту, по-друге, вказати сильні та слабкі сторони кожного з них.

ABBYY FineReader 12 - це професійне програмне забезпечення, за допомогою якого можна розпізнати текст із графічного файлу та перевести його в різні редаговані формати. Програма точно визначає текст і структурно його переводить у новий формат. Вона дозволяє не перепечатати текст у ручній формі, а це, у свою чергу, скорочує затрати на обробку. Дана програмних система використовується у фірмах та організаціях, які постійно працюють з паперовою документацією. Abbyy FineReader конвертує файли у формати pdf, tiff, jpeg у формати pdf, doc, xls, xlsx, txt тощо.

Результати тестування показали що дана система дуже якісно обробляє документи, передає їх із графічного формату в текстовому форматі. У середньому, кількість визначених слів становить 91,3%. Данне програмне забезпечення показало себе, як якісний продукт. Також варто відзначити, що ключові поля, за допомогою яких у подальшому будуть виведені дані з документа, були знайомі добре і без помилок, що означає, основне завдання оптимізації та автоматизації можливо реалізовано.

Переваги: висока якість розподілу текстів, широкий вибір вхідних та вихідних форматів документів, простої та понятний користувацький інтерфейс

Недоліки: дана програмна система є платна, не має відкритого доступу до вихідного коду програми.

Безкоштовна онлайн-послуга ocr - це безкоштовний онлайн сервіс для розпізнавання тексту з різних форматів. Його легко і просто використовувати лютому користувачеві немає необхідності завантажувати та встановлювати програму на комп'ютері. Даний продукт підтримує безліч мов для розподілу (у тому числі та російською мовою).

Присутня можливість перекласти текст в один із трьох форматів: Microsoft Word, Microsoft Excel і Text Plain. В основному, для швидкого розподілу тексту цих форматів досить. Так як іменованій необхідний у цій дипломній роботі формат .txt, для порівняння якості обробленого документа та ймовірності некоректного перевалу тексту.

Виходячи з отриманих результатів, можна зробити висновок, що даний сервіс має високий рівень розподілу тексту. У середньому, кількість визначених слів складає 67,5%. Даний сервіс є хорошим способом швидкої розпізнавання графічного файлу та передає його у текстовому форматі, але для оптимізації реального бізнес-процесу даний продукт не надходить із-за специфічної функціональної можливості.

Переваги: простота у використанні, швидкий доступ до любого пристрою з виходом в Інтернеті, безліч мов для розпізнавання.

Недоліки: на відміну від завантажених програм для оптичного розпізнавання тексту, даний продукт не має таких широких функціональних можливостей.

CuneiForm - це програма, що використовує для розпізнавання текстових документів та подальшого переводу в редактований вид. Даний продукт безкоштовний та доступний будь-якому користувачелю. У результаті роботи програм можна отримати файл у новому форматі з відредагованим текстом.

Виходячи з побаченого вище, робимо висновок, що ці дана програмні продукти не підтримують один з найрозповсюджених форматів файлів, як Microsoft Word (розширення .doc) та Acrobat Reader DC (розширення .pdf). Отже, даний продукт буде програвати за широтою використання того ж ABBYY FineReader 12.

Результати тестування показали, що OCR CuneiForm є низькоякісним програмним продуктів для розпізнавання оптичного тексту. Основну частину слів переводить в незрозумілий набір символів або взагалі не розпізнає. В середньому, кількість розпізнаних слів становить 12,8%. Це дуже низька якість.

Переваги: дана програма безкоштовна, є доступ до вихідного коду програми (використовуючи їх, можна запускати програму для аналізу графічного документа за допомогою написаного коду на C #).

Недоліки: низька якість розпізнавання.

Аналіз проводився без коригування активних полів для розпізнавання, тобто програми переводили текст тільки в тих місцях графічного файлу, де вони його визначили. Суть розпізнавання без коригування в тому, що користувач не витрачає час на вибір того чи іншого поля, відповідно, не витрачаємо час на обробку. У такому вигляді аналізу можливі втрати якості, але так як основна мета дипломної роботи - оптимізувати бізнес-процес, значить, будемо покладатися на коректну обробку документа програмою для оптичного розпізнавання тексту.

Виходячи з результатів аналізу, можна зробити висновок, що програма Abbyu FineReader є найкращим продуктом для оптичного розпізнавання

графічного файлу. Дана програма має широкий спектр можливостей і безліч плюсів. Сервіс free online ocr service також непоганий для швидкого розпізнавання тексту на картинці, але не підходить для автоматизації бізнес-процесу через мінімального функціоналу.

Також були проаналізовані і інші програми для розпізнавання текстів, результати тестування наведено в таблиці 1.1.

Таблиця 1.1 – Порівняння програмних засобів розпізнавання текстової інформації

Функція програми	ABBYY FineReader	OCR CuneiForm	Readiris Pro	Freemore OCR	ABBYY Screenshot Reader	Adobe Acrobat Pro	Scanitto Pro
Сканування і оптичне розпізнавання тексту	+	+	+		+		+
Витяги тексту з зображень, pdf файлів	+	+	+	+	+		
Можливість перекладати текст на кілька мов	+			+			
Обробка у вбудованому текстовому редакторі	+	+	+				
Стиснення відсканованого тексту (зменшення розміру файлу)	+					+	+
Вбудований словник для перевірки орфографії	+	+		+			
Рейтинг	10	8	9	8	7	7	6

За результатами нашого невеликого дослідження, на перше місце рейтингу заслужено претендує Abbyu FineReader. Він коштує недешево, але

вміє розпізнавати тексти краще конкурентів, підтримує більше мов і всі необхідні формати - pdf, doc та поширені розширення графічних файлів.

Якщо сканування, експорт і пакетний режим вам не потрібні, зверніть увагу на ABBYY Screenshot Reader . Він захопить будь-яку область екрану і переведе її в звичайний текст за лічені секунди. Як і «старший брат» продукт забезпечений словником і перекладачем, але коштує в рази дешевше.

CuneiForm, безперечно, працює краще, ніж будь-які інші безкоштовні сервіси розпізнавання тексту. Продукт функціонально наближений до утиліт від ABBYY, але зроблений трохи простіше.

Readiris Pro і Freemore OCR можна назвати альтернативою FineReader і CuneiForm для західного ринку. Проводячи паралелі, ви зрозумієте, наскільки додатки принципово схожі. Правда, у Readiris є великий козир в рукаві - крута система читання рукописного тексту.

Acrobat призначений для інших цілей, але інтегрована функція розпізнавання непогано працює, хоч і менш зручна, ніж у інших рішень з нашої вибірки.

Scanitto Pro і RiDoc зроблять з документа, зображення або аркуша паперу з символами якого читають текст досить швидко. Утиліти легкі, що не вимогливі.

1.4 Постановка задач дослідження

В даному розділі проаналізовано задачі та сфери застосування методів та алгоритмів контурного аналізу. Проведено дослідження алгоритмів обробки цифрових зображень різного виду, а також виділено їх основні характеристики. Проаналізовано програмні комплекси для розпізнавання текстової інформації на цифрових зображеннях.

Для досягнення поставленої мети необхідно розв'язати наступні задачі.

- проаналізувати методи та алгоритми контурного аналізу;
- провести аналіз підходів до стоншення об'єктів на цифрових зображеннях;
- провести дослідження існуючих програмних систем синтезу, аналізу та обробки цифрових зображень;
- проаналізувати існуючі алгоритми перетворення елементів цифрових зображень;
- розробити алгоритм розпізнавання текстової інформації цифрових зображень на основі аналізу контурів;
- реалізувати програмну систему розпізнавання текстової інформації на цифрових зображень, провести її тестування та порівняти з програмами аналогами.

1.5 Висновки до розділу

Проведено дослідження та класифікацію методів та алгоритмів контурного аналізу, що дозволило виділити основні напрямки та алгоритми які використовуються опису цифрових зображень.

Проаналізовано алгоритми обробки цифрових зображень, що дозволило вигнати групу алгоритмів для отримання компактвого опису об'єктів на цифрових зображень, а також їх переваги та недоліки.

Проведено аналіз програмних систем розпізнавання текстової інформації на цифрових зображень який дозволив виділити основні структурні модулі, а встановити інтерфейси обміну даними між ними.

2 МЕТОДИ ТА АЛГОРИТМИ РОЗПІЗНАВАННЯ ТЕКСТОВОЇ ІНФОРМАЦІЇ НА ЗОБРАЖЕННЯХ

2.1 Методи та алгоритми розпізнавання текстової інформації

Візуально люди сприймають інформацію набагато краща, ніж комп'ютерів, можливо, завдяки чудовому розумінню зображень високого рівня, знанням контексту та масовій паралельній обробці. Але людські можливості різко погіршуються після тривалого періоду спостереження, також певні робочі умови або недоступні, або занадто небезпечні для людей. Отже, з цих причин автоматичні системи розпізнавання розроблені для різних застосувань. Керуючись досягненнями в обчислювальних можливостях та технологією обробки зображень, комп'ютерна імітація людського зору останнім часом знайшла своє місце в ряді практичних застосувань.

Розпізнавання зображень відноситься до технологій, які визначають місця, логотипи, людей, предмети, будівлі та деякі інші змінні в цифрових зображеннях. Людям, може бути дуже легко розпізнати різні зображення, наприклад, зображення тварин. Легко розпізнати образ kota та відрізнити його від образу коня. Але це може бути не так просто для комп'ютера.

Цифрове зображення - це зображення, що складається з елементів зображення, також відомих як пікселі, кожен із кінцевих, дискретних величин числового подання для його інтенсивності або рівня сірого. Отже, комп'ютер розглядає зображення як числові значення цих пікселів, і щоб розпізнати певне зображення, він повинен розпізнати закономірності та закономірності в цих числових даних. Для прикладу розглянемо зображення собаки, представлене розміром 40 x 40 пікселів (рисунок 1.2).

Розпізнавання зображень не слід плутати з виявленням об'єктів. Під час виявлення об'єктів людина аналізує зображення та знаходимо різні об'єкти на зображенні, тоді як розпізнавання зображень займається розпізнаванням зображень та класифікацією їх за різними категоріями.

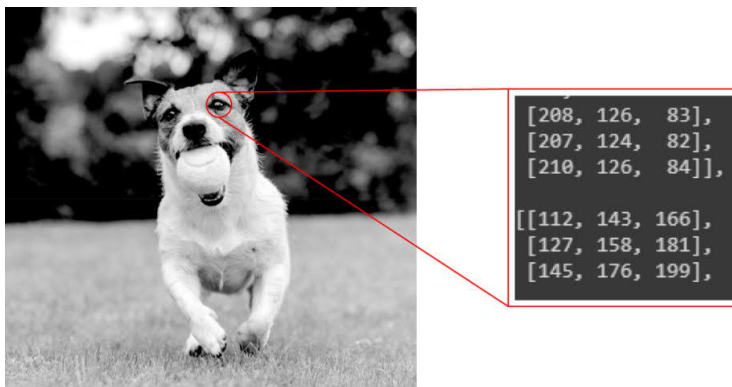


Рисунок 2.1 – Приклад цифрового зображення та його кодування

Для аналізу роботи алгоритмів обробки/розпізнавання зображень розглянемо чотири основні завдання, які сьогодні використовуються в реальних додатках:

- класифікація;
- позначення;
- виділення;
- сегментація.

Класифікація та позначення. Класифікація (рисунок 2.2): при аналізі людина майже впевнена, що є лише собака (97%), а kota (присутність тільки 3%) немає. Позначення в даному випадку аналізатор впевнений що є і собака (97%), і качк (81%).



Рисунок 2.2 – Приклад аналізу цифрового

Першим і найпростішим завданням, яке користувач може виконати, є виявлення того, що є на зображенні, і наскільки впевнений в цьому, тобто відсоток ймовірності на двох зображених вище. Є два основні моменти, на які слід звернути увагу:

Який перелік об'єктів необхідно отримати в кінці. Це називається онтологією. На першому зображенні це коти і собаки. Щоб зробити це (дуже) простим, тут слід повідомити алгоритму, які класи об'єктів він повинен ідентифікувати заздалегідь. І, як і у всіх речах просто ... це насправді складніше, ніж це. Не завжди потрібно перераховувати всі об'єкти. Однак це відкрита область досліджень, яка називається навчання без нагляду.

Інше важливе питання це кількість об'єктів на малюнку. Якщо одночасно є лише один предмет, то це класифікація. В іншому випадку, коли кілька об'єктів знаходяться на одному зображенні, це називається позначенням.

Інша група задач це виділення та сегментація (рисунок 2.3).



Рисунок 2.3 – Приклад виділення(а) та сегментації(б)

Виділення виводить прямокутник або обмежувальну рамку на зображення, де знаходяться об'єкти. Він може бути схильний до невеликих помилок та неточностей щодо позиції, але це дуже надійна технологія.

Сегментації йде на крок далі. Для кожного пікселя, найбільш атомного елемента інформації на зображенні, ми визначаємо, до яких об'єктів, якщо такі є, належить. Результат - дуже точна карта, хоча вона вимагає багато ретельно

анотованих даних. Це виснажливе завдання, коли вам доводиться це робити на кожен піксель, але воно може дати вражаючі результати.

Проблема автоматичного розпізнавання зображень - це складене завдання, яке передбачає виявлення та локалізацію об'єктів на захаращеному тлі, сегментацію, нормалізацію, розпізнавання та перевірку. Залежно від характеру заявки, напр. Розміри бази даних для навчання та тестування, безлад і мінливість фону, шум, оклюзія і, нарешті, вимоги до швидкості, деякі з підзадач можуть бути дуже складними. Припускаючи, що сегментація та нормалізація зроблені, а отже програма повинна зосередитись на підзадачі розпізнавання та перевірки об'єктів та демонструємо ефективність, використовуючи кілька наборів зображень.

Різноманітні парадигми використовувались при розробці алгоритмів розпізнавання зображень, деякими з них є:

- штучні нейронні мережі [7, 8];
- аналіз основних компонентів [9, 10];
- нечіткі моделі [11, 12];
- генетичні алгоритми [13, 14];
- автоматична асоціативна пам'ять [15].

Абрішамбаф та ін. Розробили систему розпізнавання відбитків пальців, засновану на Cellular Neural Networks (CNN). Система включає фазу попередньої обробки, де вдосконалене вхідне зображення відбитків пальців, і фазу розпізнавання, коли розширене зображення відбитків пальців відповідає відбиткам пальців у базі даних. Фази попередньої обробки та розпізнавання реалізуються за допомогою підходів CNN. Нове застосування методу скелетонізації використовується для проріджування гребневих ліній, що покращує якість витягнутих ліній для подальшої обробки, а отже, підвищує загальну продуктивність системи [6].

Ян і Парк розробили систему перевірки відбитків пальців, засновану на наборі незмінних моментних особливостей та нелінійному верифікаторі нейронної мережі зворотного поширення (BPNN). Вони використовували метод

на основі зображення з незмінними моментами для перевірки відбитків пальців, щоб подолати недоліки традиційних методів на основі дрібниць та інших методів, що базуються на зображеннях. Запропонована система містить два етапи: офлайн-етап для обробки шаблону та он-лайн етап для тестування із вхідними відбитками пальців. Система попередньо обробляє відбитки пальців і надійно виявляє унікальну контрольну точку для визначення регіону інтересу (ROI). Всього чотири набори з семи інваріантних моментних ознак витягуються з чотирьох розділених підзображень рентабельності інвестицій. Відповідність між векторами ознак тестового відбитка пальця та шаблоном відбитка пальця в базі даних оцінюється за допомогою нелінійного BPNN, а його ефективність порівнюється з іншими методами з точки зору абсолютної відстані як показник подібності. Експериментальні результати показують, що запропонований метод зі збігом BPNN має вищу точність узгодження, тоді як метод з абсолютною відстанню має вищу швидкість узгодження. Результати порівняння з іншими відомими методами також показують, що запропонований метод перевершує їх у точності перевірки.

У роботі автори представляють класифікатор, заснований на мережі радіальних базових функцій (RBFN) для виявлення фронтальних видів облич. Методика розділена на три основні етапи, а саме: попередня обробка, вилучення ознак, класифікація та розпізнавання. Трансформація кривої, лінійний дискримінантний аналіз (LDA) використовуються для першого вилучення рис із зображень обличчя, а RBFN - для класифікації зображень обличчя на основі ознак. Використання RBFN також зменшує кількість помилкових класифікацій, спричинених нелінійно відокремлюваними класами. З бази даних ORL взято 200 зображень і розраховано такі параметри, як швидкість розпізнавання, коефіцієнт прийнятності та час виконання. Показано, що розпізнавання обличчя на основі нейронної мережі є надійним і має кращі показники коефіцієнта розпізнавання 98,6% та коефіцієнта прийнятності 85%.

Бовмік розробив ефективну техніку злиття для автоматичного розпізнавання обличчя. Злиття візуальних та теплових зображень було

зроблено, щоб скористатися перевагами теплових зображень, а також візуальних зображень. Застосовуючи синтез, можна отримати нове зображення, яке надає найбільш детальну, надійну та виразну інформацію. У цьому методі злиті зображення генеруються з використанням візуальних та теплових зображень обличчя на першому кроці. На другому етапі злиті зображення проєктуються на власний простір і нарешті класифікуються за допомогою нейромережі з радіальною базовою функцією. В експериментах використовувались відстеження та класифікація об'єктів поза видимим спектром (OTCBVS), базовий показник для теплових та візуальних зображень обличчя. Експериментальні результати показують, що запропонований підхід добре розпізнає невідомих з максимальним рівнем успіху 96%.

Зен і Лю описали сучасний стан важливих досягнень нечітких наборів типу 2 для розпізнавання зразків. Успіх нечітких наборів типу 2 багато в чому пояснюється їх тривимірними функціями приналежності для вирішення більшої невизначеності в реальних проблемах. При розпізнаванні зразків як простори ознак, так і гіпотези мають невизначеності, що спонукає нас інтегрувати нечіткі набори типу 2 із звичайними класифікаторами для досягнення кращих показників з точки зору надійності, узагальнення або точності розпізнавання.

Система розпізнавання облич для персональної ідентифікації та перевірки за допомогою генетичного алгоритму (GA) та нейронної мережі зворотного розповсюдження (BPNN) описана в. Система складається з трьох етапів. З самого початку на вхідному зображенні застосовується деяка попередня обробка. По-друге, витягуються риси обличчя, які будуть прийняті як вхідні дані нейронної мережі зворотного розповсюдження та генетичного алгоритму на третьому етапі, а класифікація проводиться за допомогою BPNN та GA. Запропоновані підходи апробовані на ряді зображень обличчя. Експериментальні результати демонструють вищу ефективність цих алгоритмів.

У роботі Vlahuta et al. застосоване розпізнавання образів на ультразвукових зображеннях стовбура мозку з кінцевим набором для створення

нейророзв'язків у медичних проблемах. Для аналізу цих зображень був використаний метод аналізу основних компонентів (PSA). Цей метод - це метод з багатьох методів обробки зображень, саме до розпізнавання зразків, де необхідно вилучення особливостей. Також використовували штучні нейронні мережі (ANN) для цієї проблеми та порівнювали результати. Метод був реалізований в програмному забезпеченні NeuroSolutions, яке є дуже складним симулятором ANN з багатошаровою топологією NN PCA (ML).

Пандіт і Гупта запропонували модель нейронної мережі, яка була використана для навчання системи розпізнавання зображень. Модель NN використовує автоматичну асоціативну пам'ять для навчання. Модель зчитує зображення у вигляді матриці, оцінює вагову матрицю, пов'язану із зображенням. Після завершення тренувального процесу, коли зображення передається системі, модель розпізнає його належним чином. Оцінена матриця ваги використовується для узгодження шаблону зображення. Помічено, що розроблена модель є досить точною, щоб розпізнати зображення, навіть якщо зображення спотворене або якась частина / дані відсутні на зображенні. Ця модель виключає тривалий процес розпізнавання зображень.

У роботі автори представляють дизайн трьох типів нейронних мереж з різними функціями для розпізнавання зображень, включаючи традиційні мережі зворотного розповсюдження, мережі радіальних базових функцій та мережі зустрічного розповсюдження. Складність проектування та можливості узагальнення трьох типів архітектур нейронних мереж перевіряються та порівнюються на основі застосованої задачі розпізнавання цифрових зображень. Традиційні мережі зворотного розповсюдження вимагають дуже складного навчального процесу перед тим, як застосовуватись для класифікації або наближення. Мережі з радіальними базовими функціями спрощують навчальний процес завдяки спеціально організованій 3-шаровій архітектурі. Мережі протидії розповсюдженню взагалі не потребують навчального процесу і можуть бути спроектовані безпосередньо шляхом вилучення всіх параметрів із вхідних даних. Експериментальні результати показують хорошу толерантність

до шуму як RBF-мереж, так і мережі протипоширення щодо проблеми розпізнавання зображень, і якимось чином вказують на погану здатність узагальнення традиційних мереж зворотного поширення. Відмінна здатність шумопоглинання робить мережі RBF дуже придатними для попередньої обробки даних зображень перед поданням заявки на розпізнавання.

Однією з основних груп алгоритмів, що найбільш широко застосовується в області розпізнавання зображень, це група алгоритмів, що базується на використанні класичних моделей-класифікаторів, які в даному випадку проходять попереднє навчання з учителем. Для проведення навчання таких моделей застосовується наперед промаркована вибірка даних, що формуються з масиву зображень та масиву міток які відповідають відповідним зображенням, при цьому визначається категорія (клас), до якої належить відповідне зображення. Під час навчання масив даних ділиться на дві нерівні між собою частини – які називаються: навчальна та тестова вибірки. На наступному кроці на основі специфічного для відповідного алгоритму правила навчання, відбувається налаштування параметрів моделі на основі навчальної вибірки таким чином, щоб отримавши на вхід цифрове зображення, модель на виході встановлювала для зображення відповідну мітку класу. Такий підхід репрезентований множиною моделей, серед яких слід відзначити регресивну модель, метод опорних векторів, штучна нейронна мережа, дерева прийняття рішень, а також деякі моделі, що поєднують в собі вище перераховані моделі.

Метод опорних векторів представляє кожен екземпляр даних (об'єкт на зображенні) як точку в деякому вимірному просторі, який відповідає за розмірність даних та враховує кількість пікселів на зображенні. Кожна з точок зображення належить до певного класу іншими словами категорії.

Проблеми які можуть виникнути під час розпізнавання зображень та ускладнити сам процес розпізнавання (рисунок 2.5).

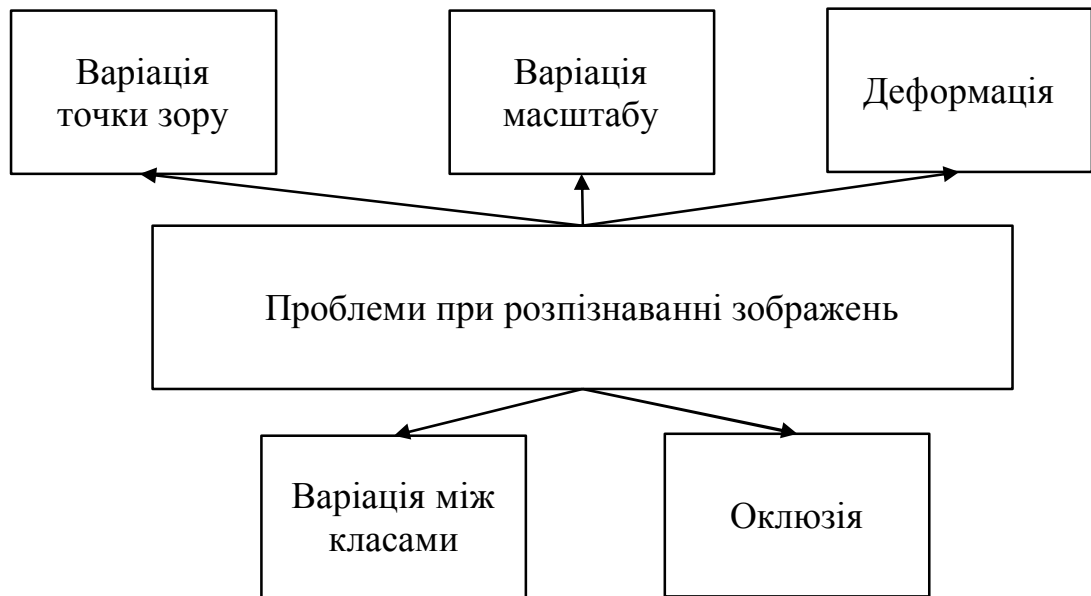


Рисунок 2.5 – Класифікація основних проблем при розпізнаванні зображень

Варіація точки зору: У реальному світі сутності на зображенні вирівнюються в різних напрямках, і коли такі зображення подаються в систему, система передбачає неточні значення. Коротше кажучи, система не розуміє, що зміна вирівнювання зображення (ліворуч, праворуч, знизу, зверху) не зробить його іншим, і саме тому створює проблеми з розпізнаванням зображення.

Варіація масштабу: варіації розміру впливають на класифікацію об'єкта. Чим ближче ви розглядаєте об'єкт, тим більший він виглядає за розміром і навпаки

Деформація: Предмети не змінюються, навіть якщо вони деформовані. Система дізнається на ідеальному зображенні та формує уявлення про те, що конкретний об'єкт може бути лише у певній формі. Ми знаємо, що в реальному світі форма змінюється, і, як наслідок, трапляються неточності, коли система зустрічає деформоване зображення предмета.

Варіація між класами: певний об'єкт змінюється в межах класу. Вони можуть бути різної форми, розміру, але все одно представляють один і той же

клас. Наприклад, гудзики, стільці, пляшки, сумки бувають різних розмірів та зовнішнього вигляду.

Оклюдія: певні об'єкти перешкоджають повному огляду зображення і спричиняють подачу неповної інформації до системи. Необхідно розробити алгоритм, чутливий до цих варіацій і складатися з широкого діапазону вибірок даних.

Для навчання моделей алгоритмів розпізнавання навчальний набір повинен мати різновиди, що належать до одного класу та декількох класів. Сорти, доступні в навчальному наборі, гарантують, що модель точно прогнозує при тестуванні за даними тестів. Однак, оскільки більшість зразків розташовані в довільному порядку, переконання, чи достатньо даних, вимагає ручної роботи, що є нудною справою.

Обмеження які впливають на результативність роботи алгоритмів розпізнавання на основі стоншення об'єктів.

Величезна доступність даних ускладнює їх обробку через обмежену доступність обладнання. Труднощі в інтерпретації моделі, оскільки нечіткий характер моделей забороняє застосовувати їх у ряді областей. Розробка займає більше часу, а отже, гнучкість порушується з часом розробки.

2.2 Алгоритми виділення контурів на основі їх проходження

Границя об'єкта – це частина об'єкта, яка містить велику кількість інформації про форму об'єкта та слабо залежить від кольору та текстури зображення .

Проходження контуру (contour tracing) – це етап отримання дискретного сигналу, що описує границі оцифрованого об'єкта.

Вимоги до алгоритмів проходження контуру такі:

- зменшення обсягів пам'яті для зберігання;

- зменшення часу та складності подальшої обробки;
- отримання інформативних ознак об'єкта.

Виділення контуру може відбуватись двома шляхами: підкресленням границь мікрооб'єкта шляхом фільтрації вхідного зображення або шляхом проходження внутрішнім контуром однорідної області.

Основні алгоритми виділення границь мікрооб'єкта такі: алгоритм змії, алгоритм Кенні, фільтрації на основі операторів Собеля, Лапласа, Превіта тощо. Вони базуються на підкресленні різких перепадів яскравості, які характерні границям об'єктів. Результатом їх роботи є набір незв'язних областей. Для отримання зв'язного контуру необхідно провести додаткову обробку.

Відомі такі алгоритми виділення областей: порогова сегментація, кластеризація, нарощування областей, алгоритм водоподілу, блочна сегментація тощо. Вони базуються на об'єднанні пікселів в однорідні області на основі деякого критерію однорідності. Результатом їх роботи є набір однорідних областей. Для отримання опису контуру об'єкта необхідно використати алгоритми проходження контуром.

На даний час відомі такі алгоритми проходження контуром:

1) Алгоритм „жука” (Square Tracing Algorithm) [54], основною перевагою якого є простота. Проходження контуром здійснюється на основі двох простих правил: якщо значення активного пікселя дорівнює одиниці (активний піксель знаходиться в точці яка належить об'єкту), то проводиться поворот ліворуч, в іншому випадку, коли значення активного пікселя дорівнює нулю (активний піксель знаходиться в точці, яка не належить об'єкту), то проводиться поворот праворуч). Алгоритм зупиняє свою роботу, якщо він повернувся в стартову точку.

2) „Moore-Neighbor Tracing” – алгоритм базується на покроковій перевірці всіх сусідніх точок з метою знаходження наступної контурної точки. Зупинка роботи алгоритму відбувається при його поверненні в стартову точку.

3) „Radial Sweep” – даний алгоритм є модифікацією попереднього. Основна його відмінність полягає в точці початку обходу активного пікселя. В

даному алгоритмі це точка, що була визнана контурною на попередньому кроці алгоритму, а не точка з якої відбувся перехід на активний піксель.

4) „Theo Pavlidi’s Algorithm” – основною ідеєю алгоритму є використання групи з трьох пікселів для визначення наступного контурного пікселя. Перевірку здійснюють шляхом послідовної перевірки сусідніх точок з строго визначеною послідовністю.

Ці алгоритми широко використовуються для проходження та опису контуру [50]. Практично для кожної мови програмування існують бібліотеки з реалізованими алгоритмами, наприклад, для Matlab.

Основним їх недоліком є залежність від складності контуру та критерію зупинки. Алгоритми чутливі до мікрооб’єктів, контур яких містить відгалуження товщиною в один піксель. Це може привести як до помилкового завершення роботи алгоритмів, так і до некоректного виділення контуру. Аналогічна проблема може виникнути, якщо мікрооб’єкт складається з двох і більше частин, що з’єднані між собою тільки одиночними пікселями. Інший недолік алгоритмів полягає у недосконалих критеріях зупинки (повернення в стартову точку, проходження певної точки декілька разів), що призводять до некоректних результатів роботи.

Окрім форми, контур має ряд інших характеристик, зокрема тип зв’язності, що задається під час кодування контуру. Контур може бути 4-зв’язним або 8-зв’язним. Якщо контур 4-зв’язний, то координати двох сусідніх точок, що належать контуру об’єкта, можуть відрізнитись тільки по осі абсцис або тільки по осі ординат. Якщо контур 8-зв’язний, то координати двох сусідніх точок, що належать контуру об’єкта, можуть відрізнитись по осі абсцис і/або по осі ординат.

Іншим критерієм може бути структура контуру: проста (форма мікрооб’єкта немає малоінформативних відгалужень та мікрооб’єкт знаходиться окремо від інших), з однопіксельними відгалуженнями (на контурі мікрооб’єкта присутні малоінформативні відрости, які є наслідком дефекту під час препарациі досліджуваного зразка), складні (мікрооб’єкти дотикаються або

накладаються один на одного, утворюючи при цьому хибний мікрооб'єкт, який при можливості необхідно коректно роз'єднати).

Представлення (кодування) контуру – це етап отримання дискретного сигналу, що описує границі оцифрованого зображення.

Кодування контуру об'єкта за допомогою ланцюгових кодів Фрімена. Для опису форми об'єкта необхідно на аналогове зображення контуру нанести сітку з заданою величиною комірок, та зафіксувати точки в яких контур перетинає лінії сітки (рисунок 2.6). Потім послідовність вузлів, що є найближчими до перетинів, кодується вісімковими числами, позначаючи напрям від одного вузла до іншого у відповідності із кодами.

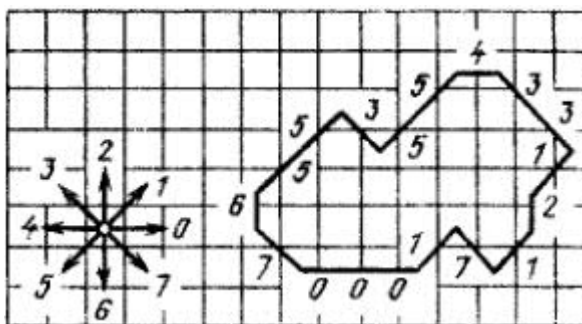


Рисунок 2.6 – Кодування контуру об'єкта за ланцюгових кодів Фрімена.

Різновидами кодування контурної функції ланцюговими кодами Фрімена є також кодування, при якому замість цифр [0-7] використовуються наступна матриці:

$$\begin{array}{ccc}
 -1,-1 & 0,-1 & 1,-1 \\
 -1,0 & \star & 1,0 \\
 -1,1 & 0,1 & 1,1
 \end{array}$$

Перевагою даного кодування є простота реалізації, висока швидкодія та точність представлення контурної функції. До недоліків слід віднести відносно велику кількість пам'яті, що необхідна для зберігання опису, а також чутливість до поворотів.

Функція центрових відстаней $R(s)$ визначається як відстань від центру об'єкта до кожної точки контуру і обчислюється за формулою:

$$R(s) = \sqrt{(x_s - x_c)^2 + (y_s - y_c)^2},$$

де, (x_s, y_s) – координати s -тої точки контуру об'єкта;

(x_c, y_c) – координати центру об'єкта.

Кодування за допомогою елементарних векторів.

Елементарний вектор $y(n)$ - це вектор, що з'єднує центри сусідніх контурних комірок сітки. Де n - порядковий номер вектора ($n=0,1..k$) (рисунок 2.7). Крок сітки вибирається відповідно до необхідної точності представлення.

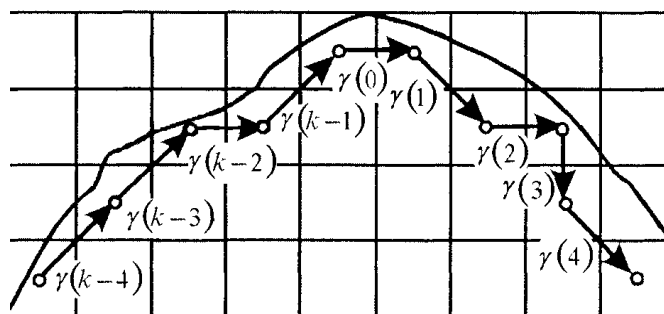


Рисунок 2.7 – Кодування контуру об'єкта за допомогою елементарних векторів.

Кодування елементарними векторами можна розглядати як апроксимацію кривими першого порядку з сталими довжинами. До переваг даного підходу кодування інформації є простота реалізації, низька трудомісткість процесу кодування, малі обсяги пам'яті необхідної для зберігання даних, висока точність представлення, простота порівняння двох контурів.

Кодування за допомогою векторів. Вектори не мають обмеження довжини, що дозволяє суттєво зменшити кількість векторів, в порівнянні з елементарними векторами (рисунок 2.8). Крок сітки залежить від необхідної точності представлення.

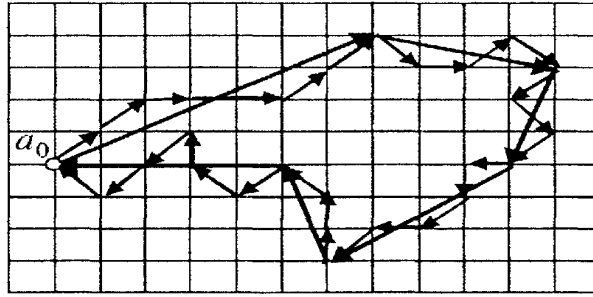


Рисунок 2.8 – Кодування контуру об’єкта за допомогою векторів.

Результатом кодування є наступні масив ознаки: довжина вектора, напрям повороту, куту між сусідніми векторами. В деякому наближенні опис форми об’єкта за допомогою векторів є апроксимацією контуру за допомогою кривих першого порядку.

Перевага – невеликий обсяг пам’ять, необхідної для зберігання векторів, а також висока точність опису контурної функції. До недоліків слід віднести часові затрати на обрахунок векторів які є відносно великі а часові затрати на перевірку точності опису.

Побудова опису на основі перетворення Фур’є є дуже ефективним для представлення форми замкнених контурів, тобто періодична функція, яка отримується при побудові впорядкованої послідовності контурних точок може бути виражена через ряд Фур’є. Контур об’єкта можна розглядати як деяку замкнену періодичну функцію $y(x)$ з періодом P , де P дорівнює периметру об’єкта, а отже, контур можна розкласти за допомогою рядів Фур’є. Тоді дискретне представлення контуру об’єкта з максимальним числом членів N матиме вигляд:

$$x(s) = a_0 + \sum_{n=1}^N [a_n \cos(2\pi ns / P) + b_n \sin(2\pi ns / P)];$$

$$y(s) = c_0 + \sum_{n=1}^N [c_n \cos(2\pi ns / P) + d_n \sin(2\pi ns / P)],$$

де коефіцієнти $a_0, c_0, a_n, b_n, c_n, d_n$ називаються „еліптичними ознаками” та розраховуються наступним чином:

$$a_0 = \frac{1}{N} \sum_{k=1}^N x_k ;$$

$$c_0 = \frac{1}{N} \sum_{k=1}^N y_k ;$$

$$a_n = \frac{2}{N} \sum_{k=1}^N x_k \cos n\delta ;$$

$$b_n = \frac{2}{N} \sum_{k=1}^N x_k \sin n\delta ;$$

$$c_n = \frac{2}{N} \sum_{k=1}^N y_k \cos n\delta ;$$

$$d_n = \frac{2}{N} \sum_{k=1}^N y_k \sin n\delta ,$$

де, $\delta = 2\pi \frac{\Delta s}{P}$, Δs - приріст довжини дуги між двома точками контурної послідовності.

До переваг опису форми за допомогою коефіцієнтів Фур'є можна віднести відносну легкість реалізації та те що даний опис базується на розвинутій теорії перетворення Фур'є. Основний недолік – складність опису локальної інформації про форму, а також проблеми з точністю опису симетричних кривих (вони будуть відрізнятись лише фазою, яку незавжди можна точно обчислити).

Розглянуті алгоритми дозволяють з необхідною точністю та за допустимі часові межі отримати короткий опис контурної функції символів, що в подальшому на етапі проектування алгоритму розпізнавання дозволить з легкістю отримати масив еталонів, на основі яких і буде проходити процедура розпізнавання зображень, що містять текстову інформацію. Подібною процедурою стоншення об'єктів призведе до того, що на вхід алгоритму розпізнавання буде передаватись компактний опис аналізованого зображення.

2.3 Алгоритм розпізнавання текстової інформації на зображеннях

Провівши аналіз відомих алгоритмів та виділивши їх основні переваги та недоліки було спроектовано алгоритм розпізнавання друкованої інформації на основі виділення їх скелету.

Запропонований алгоритм з елементами адаптивного розпізнавання також використовує накопичення характеристик розпізнаваних символів. Алгоритм являє собою двох етапний процес з навчанням на результатах першого проходу.

Схема адаптивного розпізнавання поділяється на кілька етапів: первинне розпізнавання, збір статистики, кластеризація зібраної статистики, формування еталонів і повторне розпізнавання.

Визначимо кожен з названих етапів

- первинна обробка - це аналіз сторінки з виділенням областей інтересу, для подальшого їх опрацювання розпізнавання всієї сторінки за допомогою шрифто-незалежного алгоритму;

- первинне розпізнавання – етап стоншення зображень символів та спроба їх класифікації;

- аналіз розпізнавання - це процес відбору надійно розпізнаних символів, які згодом складуть навчальну вибірку для шрифту-залежного алгоритму

- формування нових еталонів - це створення остаточних наборів даних, за якими буде проводитися дораспознавання

- повторне розпізнавання - це другий прохід розпізнавання по всій сторінці з метою уточнити результати первинного розпізнавання, виставити адекватні оцінки точності, дораспознати те, що було не розпізнано раніше, відзначити ненадійно розпізнані символи.

Запропонований алгоритм ґрунтується на представленні вхідного зображення як можжини окремих областей які містять символи (чорні на білому фоні) з подальшим стоншенням даних відобрежнь з метою отримання компактного представлення того чи іншого символу. Після процесу кодування вхідного зображення відбувається процес порівняння отриманого

представлення з елементами бази даних еталоноів. Якщо кількість невизначених символів буде перевищувти деякий заданий поріг, то відбудеться процедура корекції результатів рообтои програми. Блок-схема алгоритму наведена на рисунку 2.9.

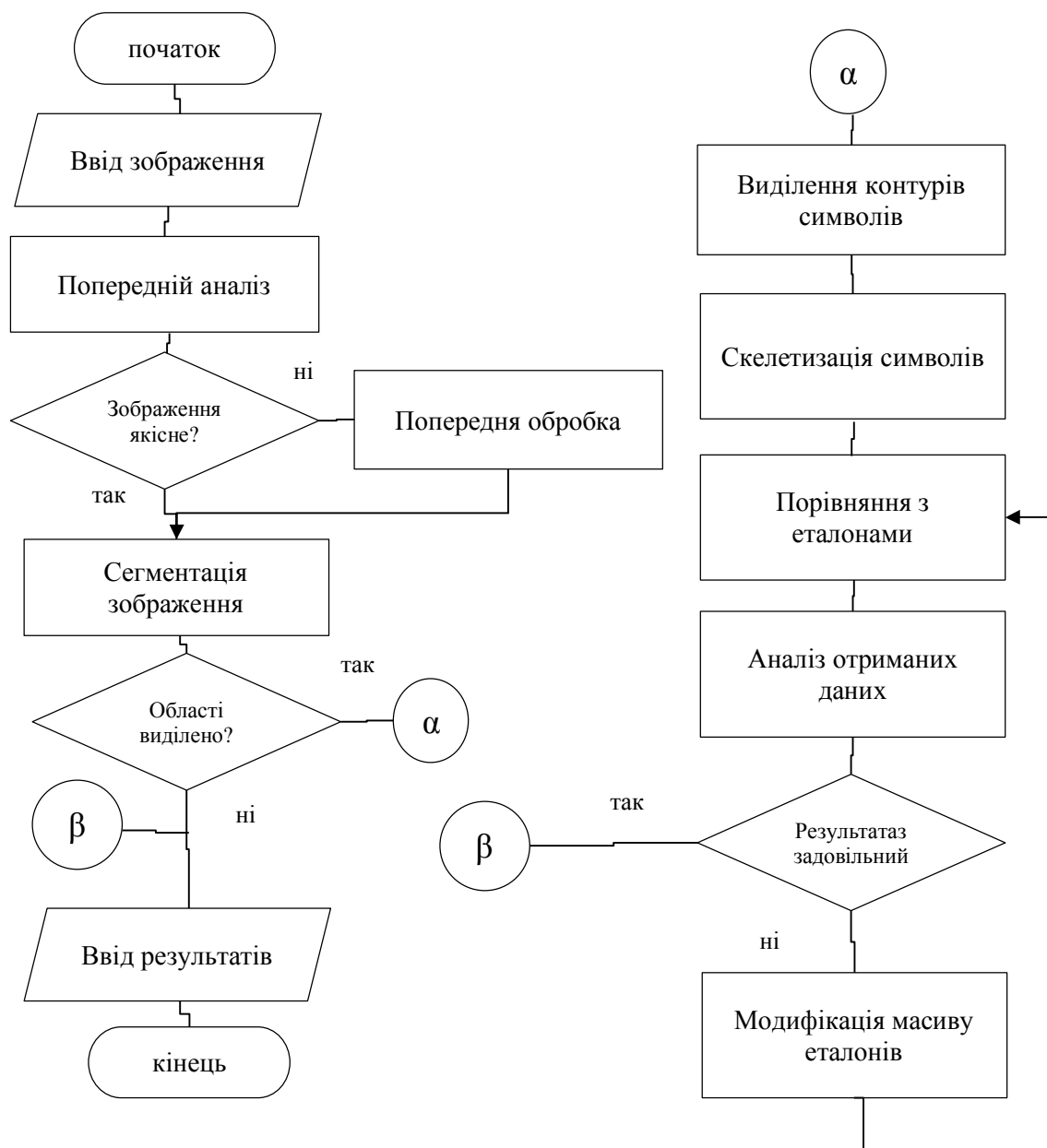


Рисунок 2.9 – Блок-схема алгоритму розпізнавання текстової інформації

Серед особливостей запропонованого алгоритму слід відмітити поєднання алгоритмі попередньої обробки, фільтрації, контурного аналізу та

скелетизації, порівняльного аналізу та алгоритмів розпізнавання на основі масиву еталонів на різних етапах роботи.

Запропонований алгоритм ґрунтується на твердженні, що при розпізнаванні символів, різновид вхідних зображень буде обмежений певною вибіркою, що дозволяє зробити припущення про те, що масив еталонних значень буде мати обмежену злічену кількість елементів. Дана кількість буде обмежуватись символами з таблиці ASCII. Іншим припущенням було те, що якщо деяке зображення не буде класифіковано однозначно, то необхідно провести повторний аналіз і визначити його приналежність до класу з максимальною подібністю, а модель вхідного символу замінити на модель вхідного зображення, чим досягнеться адаптивність алгоритму до додаткових спотворень та дефектів отриманих на етапі оцифрування.

Поріг визначення точності розпізнавання теж буде адаптивним, оскільки його рівень буде визначатись на основі більшості з класифікованих символів, що додатково дозволить підняти точність розпізнавання відсікаючи символи які не вдалось розпізнати під час первинного процесу аналізу зображення.

Даний набір припущень в повній мірі забезпечує можливість роботи запропонованого алгоритму та дозволяє спроектувати програмну систему для розпізнавання текстової інформації. Для уточнення параметрів роботи алгоритму достатньо провести декілька додаткових запусків алгоритму з різними параметрами.

Запропонований алгоритм складається з такої послідовності кроків:

Крок 1. Завантаження вхідного зображення.

Крок 2. Проведення попередньої обробки зображення. На даному етапі проводиться перетворення вхідного зображення з кольорового у градації сірого. Проводиться видалення сторонніх шумів тощо.

Крок 3. Виділення областей інтересу. Встановлення областей зображення, що містять точки чорного кольору та вписання їх в прямокутники розміром $L \times M$. Початкові розміри задаються з розрахунку 1 символ на 10×8 пікселів

зображення, проте дане вікно може бути розширене шляхом аналізу вхідного зображення, та аналізу висоти на ширини областей чорного кольору;

Крок 4. Виділення контурів об'єктів. Відбувається на основі алгоритму проходження контуром. Результатом роботи є масив послідовно зєднаних точок які описують відповідний символ.

Крок 5. Скелетизація. Процес стоншення форми вхідних символів, з метою отримати більш компактний опис.

Крок 6. Порівняння з масивом еталонів. Послідовне порівняння отриманого опису символу з кожним елементом масиву еталонів. Символ відносить до того класу, подібність з яким максимальна.

Крок 7. Визначення основного порогу подібності. Обчислення кількості елементів масиву еталонів, подібність до яких була найбільша. Визначення порогу якості розпізнавання, який отримується шляхом віднімання 10% від значення порогу, що найчастіше зустрічалось.

Крок 8. Адаптація масиву еталонів. Спроба порівняти символи, які не були розпізнані або рівень їх подібності був нижче за погоговий та корекція масиву еталонів.

Крок 9. Повторна процедура порівняння.

Крок 10. Вивід результатів роботи програми.

Дана послідовність кроків забезпечує повноцінний аналіз вхідного зображення та дозволяє виділити відповідні точки та розити зображення на відповідні сегменти, що в кінцевому варіанті дозволить провести процедуру передворення зображення на основі технології морфінгу.

До переваг даного алгоритму слід віднести його простоту та швидкість роботи. До недоліків: необхідність проведення процесу налаштування програми пред її роботою, оскільки для коректного та швидкого функціонування необхідно встановити початкові параметри роботи алгоритму (розміри сітки, відповідні точки).

2.5 Висновки до розділу

Проведно аналіз алгоритмів розпізнавання текстових зображень за допомогою контурного аналізу, що дало можливість обрати метематичне обґрунтування для проведення розпізнавання текстових цифрових зображень.

Розроблено алгоритм розпізнавання текстової інформації на цифрових зображень на основі контурного аналізу та процесу стоншення образів об'єктів, що дозволило спроектувати програмний додаток для обробки цифрових зображень.

3 ПРОГРАМНА СИСТЕМА АНАЛІЗУ ТА РОЗПІЗНАВАННЯ ЦИФРОВИХ ЗОБРАЖЕНЬ

3.1 Структура програмної аналізу та розпізнавання зображень

Завдання розпізнавання тексту залишається актуальною на сьогоднішній день, тому що не існує стовідсоткової універсальної системи з розпізнавання тексту. Система розпізнавання тексту передбачає наявність на вході зображення з текстом (в форматі даних графічного файлу). На виході системи повинен сформувався текст, виділений з цього зображення.

Розпізнавання тексту включає в себе наступні етапи:

попередня обробка зображення, що поступає на вхід системи зображення повинно бути очищено від шуму і приведено до виду, що дозволяє ефективно виділяти символи і розпізнавати їх;

система повинна розбити зображення на блоки тексту, ґрунтуючись на особ ності його вирівнювання та розподілу за кількома колонкам;

зображення з текстом має бути розділене на зображення рядків, а потім на зображення символів для того, щоб в подальшому обробити кожен символ окремо. Після цього кроку різні системи розпізнавання працюють за своїми специфічними алгоритмами;

зображення символу може оброблятися цілком, для цього воно порівнюється з наявними шаблонами, розпізнається за допомогою нейронних мереж або виділяються характеристики зображуваного символу. На виході четвертого шага з'являється можливий варіант літери. Однак зазвичай системи на цьому не зупиняються і продовжують роботу на основі інших методів, уточнюючи получение результат.

На першому етапі у всіх системах оптичного розпізнавання символів виконують поліпшення якості та аналіз зображення, поданого на обробку. У тому числі використовуються спеціальні фільтри відновлення пошкоджених

зображень, наприклад, за допомогою гіпоелліптичної дифузії, фільтри, що усувають змазування.

На другому етапі виконується робота по виявленню текстів на зображеннях. використовують такі підходи: на основі використання контурної інформації (скелетизації, виділення країв і виділення кутів, методи на основі інваріантних моментів); на основі колірної інформації (метод гістограм, аналіз головних компонент і різні алгоритми адаптивної бінаризації).

Спрощену структуру програмної системи наведено на рисунку 3.1.

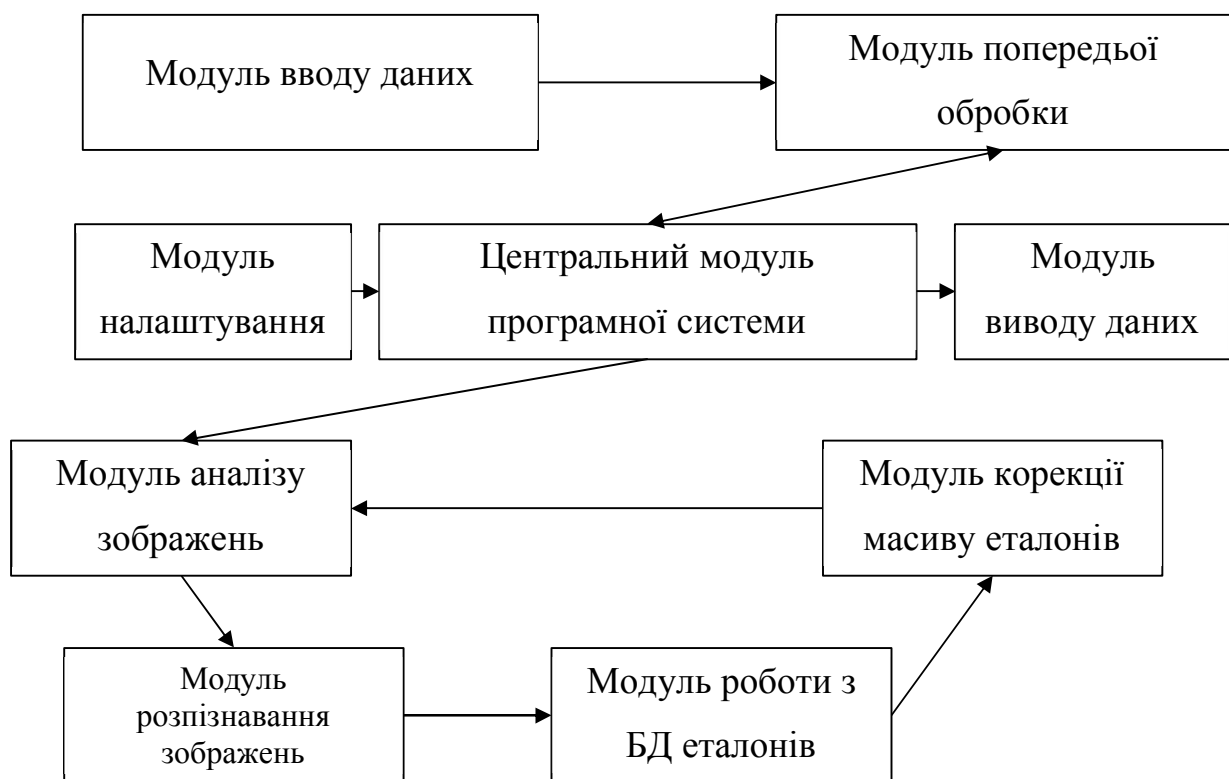


Рисунок 3.1 – Узагальнена структура програмного додатка розпізнавання текстової інформації зображень

Запропонована структура є включає усі стандартні програмні модуля, які характерні для програмних розробок даного класу. Серед основних модулів слід відмітити такі:

Центральний модуль програмної системи – даний програмний модуль забезпечує інтерфейсний зв'язок між окремими структурними елементами реалізованої програмної системи, до його функцій відносяться функції перевірки цілісності програмного додатку, наявність оновлень чи відключення пошкоджених блоків програми, контроль та переконвертація даних під час передачі інформації між окремими мобілями системи. Вхідними даними даного модуля є вектор параметрів роботи програми. Вихідними – корекційні правки пакетів обміну даних, повідомлення для користувача при необхідності внести тих чи інших змін в роботу програмного додатку в цілому.

Модуль вводу даних – даний модуль має низький пріоритет в системі, оскільки містить функції для отримання вхідних зображень з різних пристроїв, серед яких зовнішні та внутрішні носії даних, фотоапаратура, бази даних цифрових зображень. На даному етапі експлуатація програмної системи не проводиться якась додаткова перевірка чи обробка інформації. Результат роботи даного модуля є пара вхідних зображень у цифровому форматі, що підтримується програмною системою та команда активації для всіх інших програмних модулів системи.

Модуль попередньої обробки – в даному модулі об'єднані функції, що забезпечують підвищення якості аналізу та обробки зображень на більш пізніх етапах аналізу та перетворення зображень. Більшість функцій були взяті з зовнішніх цифрових бібліотек, оскільки під час виконання індивідуального завдання перед виконавцем не ставились завдання розробки та проектування алгоритмів попередньої обробки зображень. Даний системний елемент, зокрема виконує попередню перевірку чи формат та кодування вхідних зображень можуть бути обпрацьовані засобами розробленого програмного засобу, проведені попередній аналіз по визначенню розміру типу кольору, перевірка чи дані зображення не є ідентичними та чи в програмі є попередньо збережена сітка характерних точок для даної пари зображень. Результатом роботи даного модуля є встановлення параметрів початкової роботи програмного додатку.

Модуль налаштування – стандартний структурний елемент будь-якої програмної системи. Функції якого забезпечують можливість встановлення початкових параметрів роботи системи або корегування її роботи під потребу користувача, встановлення роботи самої системи та окремих її алгоритмів. Серед параметрів, які можна встановити слід відмітити: тип та розмір шрифтів інтерфейсу та модулі виводу звітів, параметрів підключення та роботи фото/веб/відеокамер, встановлення параметрів роботи алгоритмів аналізу зображень (розміри ковзного вікна та крок його зміни).

Модуль аналізу зображень – основний функціональний вузол програмної системи. В ньому спроектовані та реалізовані основні функції по розпізнаванні програмних текстів. Він активує функції сегментації, контурного аналізу, скелетизації, цей програмний модуль розроблений забезпечує основний інтерфейс між програмним забезпеченням та вхідним зображенням. В ньому реалізовано доступ до деяких параметрів програми у ручному режимі, щоб надати користувачеві можливість вносити корективи в роботу програм при очевидно негативних результатах. Оскільки, інколи важко програмно підібрати усі параметри роботи програмної системи, тому при анатдо складних випадках вхідного зображення (зашумленість, нестандартність кольорів, тексту та фону тощо) користувач має можливість внести корективи в роботу системи. Такай підхід дозволяє підійти до вирішення данаї задачі є максимально ефективним та дозволяє користувачеві у повній мірі отримати результат в залежності його потреб. Результатом роботи даного модуля є масив описів символів, що були присутні на вхідному зображенні.

Модуль розпізнавання зображення – даний модуль приймає на вхід множину отриманих описів символів, а також масив еталонних моделей. Після чого проводиться послідовне порівняння вхідних символів з елементами масиву еталонів. Після порівняння вхідному символу присвоюється номер класу, відсоток подібності до якого був найбільшим. Після завершення процесу порівняння, алгоритм проводить підрахунок точності класифікації, а саме

визначає який рівень побідності найчастіше зустрічався під час класифікації. Даний рівень приймає за проміжний рівень якості розпізнавання.

Модуль роботи з БД етталонних зображень – допоміжний модуль, що організовує роботу головних функцій та класів по роботі з масивом етталонних моделей. Серед основних функційце: перевірка наявності масиву етталонів, його цілісність та повнота, можливість зміни окремих елементів масиву. При виникненні помилок при роботі з базою даних, виводить повідомлення у вигляді систем повідомлення.

Модуль корекції масиву етталонів – допоміжний елемент програмної розробки, забезпечує можливість вносити корективи в базу даних етталонів. Проводить обчислення параметрів нових моделей та активує етап повторного аналізу та розпізнавання цифрових зображень.

Модуль виводу результатів – призначений для візуального відображення результатів розпізнавання цифрових зображень, а також забезпечує можливість зберігання їх в форматі doc, що є незаперечною перевагою розробки, оскільки саме цей текстовий редактор на сьогоднішній день має найбільшу популярність.

Спроектвана та розроблена ієрархічно-модульна структура програмного додатку аналізу та перетворення зображень у мовній мірі формує цілісну систему та має всі модливості для виконання поставлених завдань по отриманні, попередній обробці, аналізі, перетворенні та виводі результатів обробки вхідних зображень. Окрім того використаний модульний підхід дозволяє реагувати на зміни в роботі програмної системи додаючи наві модулі чи модифікуючи вже присутні, при йьому роботоздатність та цілісність системи не буде пошкоджена.

Перед процесом реалізації розглянутих алгоритмів було проведено моделювання розробленого прогламного додатку за допомогою сучасних програмних пакетів.

На стартовому етапі моделювання проведено аналіз набору можливих дій та доступних функцій і параметрів користувача при роботі з розробленою системою. Приклад діаграми прецедентів наведено на рисунку 3.2.



Рисунок 3.2 – Діаграма прецедентів програмної системи

Як видно з наведеної діаграми користувач програми достатньо великий набір функцій для внесення коректив в роботу самої програмної системи. Серед яких можливість встановлення параметрів роботи програмної системи на різних етапах, а також він повноцінно може впливати на сегментацію та контурний аналіз на попередньому етапі обробки зображення, що в свою чергу підвищує якість результату роботи програмної системи. Заборона доступу користувача до внутрішніх параметрів роботи системи розпізнавання, з однієї сторони підвищує швидкість роботи програми. Окрім того, це дозволяє зменшити кількість хибних спрацювань від втручання в роботу непрофесійних

користувачів або користувачів з низьким рівнем навиків роботи з програмними продуктами даного типу. А сам процес опанування програмою є достатньо короткий та не вимагає від нових користувачів глибоких попередніх знань комп'ютерної техніки. Достатньо високий рівень автоматизації роботи програмного додатку забезпечує можливість використання її для роботи та навчання користувачів без досвіду роботи.

На другому кроці моделювання було проведено аналіз послідовності взаємодії між основними групами процесів та об'єктів, що будуть відбуватись в процесі роботи програми. Результат моделювання наведено на рисунку 3.3.



Рисунок 3.3 – Діаграма послідовності програмної системи

Як видно з наведеної діаграми, процес функціонування програмної реалізації має послідовний характер та виконує усі дії певній чіткій послідовності, що з однієї сторони сповільнює процес опрацювання зображення, проте запропонований алгоритм надає можливість при наявності відповідних апаратних засобів провести процес розпаралелення задачі опрацювання зображення та прискорити процедуру опрацювання зображення. Проте в даній реалізації для використання програмного додатку для його роботи буде достатньо робочої станції з середніми технічними показниками

На етапі моделювання було розглянуто основні фактори. Які можуть впливати на роботу системи, а також послідовність функцій які має виконати користувач для досягнення бажаного результату.

Наступним етапом проектування та розробки програмної системи є етап розробки користувацького інтерфейсу для підвищення зручності користування програмою. Під час проектування інтерфейсу головного вікна розробки враховувались усі можливості та побажання користувачів відносно швидкого доступу до окремих функцій та мінімально можливої кількості пунктів меню. А з іншої сторони вдалось уникнути значного завантаження головного вікна програми. При цьому основний простір на головному вікні було віддано областям для відображення вхідних зображень та регілну для візуалізації результатів роботи програми. Результат проектування графічного інтерфейсу користувача наведено на рисунку 3.4.

При проектуванні зовнішнього інтерфейсу було враховано те, що програмна система повинна працювати з мінімальним втручанням ззовні, через те для користувача доступна мінімальна кількість активних функціональних елементів. Основне функції, що необхідно виконувати в процесі налаштування роботи програми реалізовані у вигляді інтерактивних точок, що можна переміщувати по зображенні в ручному режимі з метою отримати сітку відповідності з максимальною точністю відображення відповідних точок. Мінімізація кількості активних компонентів на екрані дозволила збільшити

корисну площу на головному вікні програми, що в свою чергу позитивно вплинуло на зручність при роботі з програмним додатком.

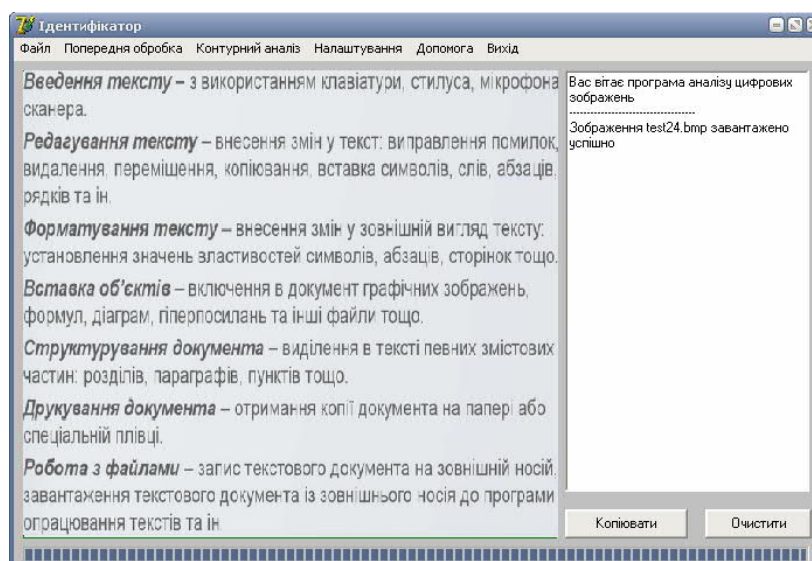


Рисунок 3.4 – Зовнішній вигляд програмної системи

Сам інтерфейс складається з трьох складних елементів:

- головного вікна програмної системи для візуалізації вхідних зображень. Дана область також забезпечує можливість користувача переглядати вхідне цифрове зображення, проводити його завантаження та переглядати зміни під час попередньої обробки зображення ;

- вікно відображення результатів роботи програми (результати візуалізуються тексту, що відображається у правій частині робочого вікна програмної системи. Користувач може зберегти отримані результати у зручний для себе текстовий формат для продовження роботи над розпізнаними даними;

- панель активних перемикачів (набір перемикачів та активаторів функції програмної системи, необхідних для запуску, зупинки або корекції роботи програмної системи).

Даний зовнішній вигляд програмної системи достатньо зручний та швидкий в опануванні, що є безперечною перевагою під час роботи з програмним продуктом.

Для вирішення поставлених задач при аналізі та розпізнаванні цифрових зображень користувачеві необхідно здійснити декілька кроків. Відповідну послідовність кроків наведено на рисунку 3.5:

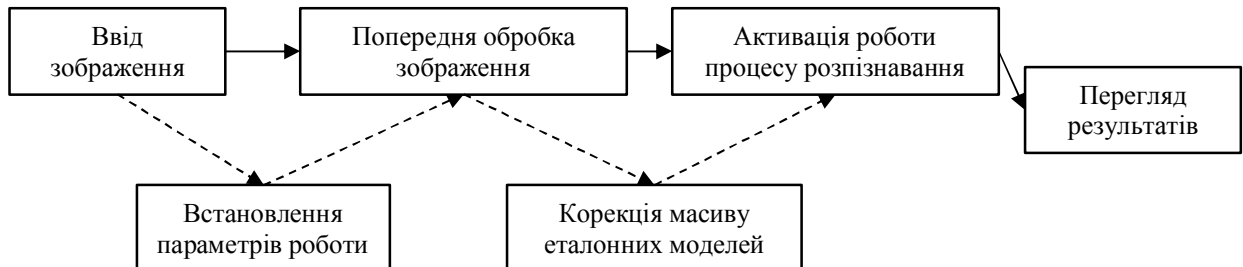


Рисунок 3.5 – Послідовність дій користувача при роботі з програмною системою перетворення елементів зображення

Як показано зі схеми дій користувача, при використанні програмної системи користувач повинен виконати три основні дії, інші можуть не виконуватись, якщо результат роботи програми є задовільним. Серед обов'язкових етапів роботи з програмною системою, що включають старт програмного додатку шляхом завантаження вхідних зображень, перевірки коректності перевірки та встановлення параметрів попередньої обробки та процесу розпізнавання та перегляду отриманих результатів в кінці процесу обробки. Окрім того користувач може виконати ще дві додаткові дії, при необхідності внесення змін при негативному або не достатньо хорошому результаті роботи програми.

3.2 Програмні модулі системи перетворення елементів зображення

Для реалізації програмного додатку було використано технічні можливості програмного середовища Delphi. За допомогою технологій обробки

зображень даної мови програмування можна написати програмне забезпечення для керування яскравістю зображення, контрастом, знаходженням крайової характеристики, надання ефекту тиснення або змінити зображення на вигляд олійного живопису.

Модуль попередньої обробки зображень. Обробка зображення означає виконання над ним певних дій для досягнення певного результату (покращення якості, внесення змін на зображенні, виділення областей, тощо). Виділення однотипних областей є однією з важливіших задач в цифровій обробці зображень. На сьогоднішній день відомо багато алгоритмів сегментації об'єктів, проте більшість з них ґрунтуються на аналіз функції яскравості.

На основі апріорної інформації, що ядро клітини на гістологічних зображеннях завжди має меншу яскравість ніж фон, то для визначення ядер можна використати алгоритм порогової сегментації. Знаходження ядер клітин в автоматичному режимі призводить до збільшення швидкості обробки зображення та зменшення впливу людського фактору на процес обробки та аналізу біомедичних зображень.

Оскільки вхідне зображення знаходиться в базисі RGB, то для проведення сегментації необхідно отримати представлення вхідного зображення на основі яскравості. Дана операція відбувається в процедурі `FontColorSell(integer, integer, integer)` Дане представлення не завжди є візуально інформативним для людського ока, проте дозволяє швидко та якісно працювати з зображеннями з використанням обчислювальної техніки. На сьогоднішній день є багато підходів та базисів ($L^*a^*b^*$, YIQ, HLS, HSB), що дозволяють розглядати зображення як набір яскравостей кожної точки. На основі аналізу результатів проведених експериментів для отримання яскравісного представлення зображення було використано базис HLS (тон, яскравість, насиченість). Перевід з базису RGB в HLS здійснюється за наступними формулами:

$$H = \arcsin\left(\sqrt{\frac{3}{2}}(G - R)/S\right),$$

$$L = (R + G + B)/3,$$

$$S = \sqrt{R^2 + G^2 + B^2 - RG - GB - RB}.$$

Приклад роботи даного алгоритму наведено на рисунку 3.6:

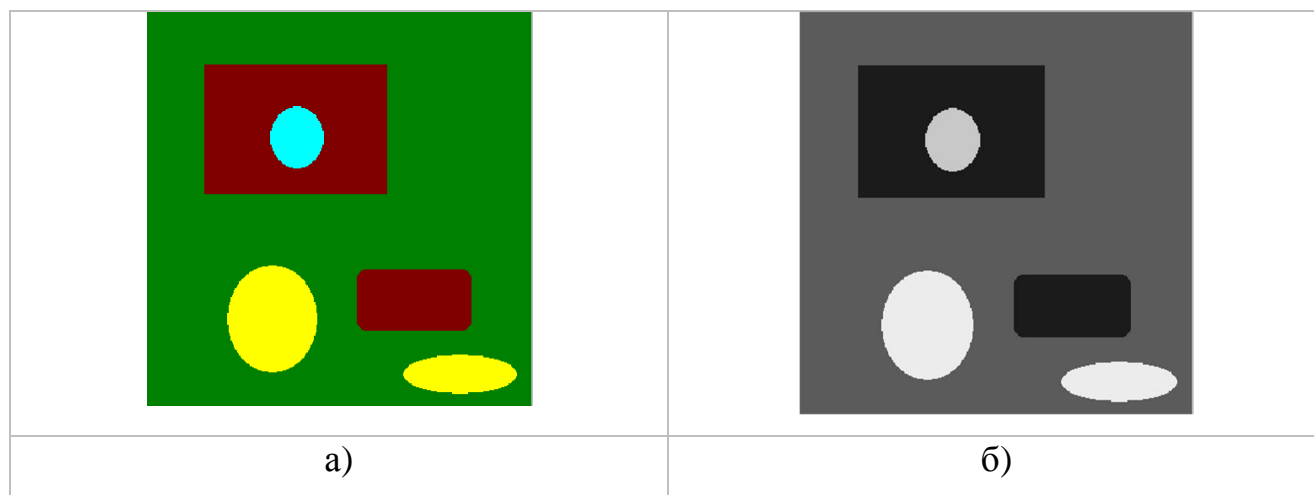


Рисунок 3.6 – Приклад перекодування кольорового (а) зображення монохромне(б)

Корекція яскравості зображення. Візуальний аналіз растрових зображень може ускладнюватись через низьку яскравість окремих ділянок зображення або їх низьку контрастність порівняно з іншими елементами зображення. Методи просторового оброблення зображень задаються таким рівнянням:

$$g(x,y)=T[f(x,y)],$$

де $f(x,y)$ – піксель вхідного зображення,

$g(x,y)$ – піксель вихідного зображення,

T – деякий оператор або перетворення над f , визначене в околі пікселя (x,y) , тобто методи просторового оброблення зображень здійснюють перетворення числових значень пікселів зображення.

У випадку перетворення яскравості зображення всі маніпуляції здійснюються над поточним пікселем зображення, тому окіл пікселя (x,y) має розмір 1×1 , значення $g(x,y)$ залежить лише від значення f , а T називають функцією перетворення яскравості. До функцій перетворення яскравості належить γ -корекція. Для кольорових зображень під терміном яскравість вважається значення кольорової компоненти зображення в даному діапазоні кольору.

Інверсія вхідного зображення. Для отримання інвертованого вхідного зображення необхідно до кожного значення пікселя кольорового каналу (для кольорових зображень) та яскравості (для монохромних) відняти 256 по модулю.

$$f_i(x, y) = |256 - f(x, y)|,$$

де $f(x, y)$ - значення яскравості пікселя до інверсії.

Дана операція дозволяє в деяких випадках покращити візуальне сприйняття вхідного зображення.

Модуль контурного аналізу. Даний модуль є одним з основних в програмній системі ідентифікації об'єктів на зображенні, оскільки від результатів роботи даного модуля в основному залежить достовірність отриманих під час сегментації даних. Тому операції в даному модулі виконуються в декілька етапів.

Бінаризація. На даному етапі вхідне зображення перетворюється з множини точок на масив бінарних значень, де «1» - відповідають об'єктам, «0» - фону.

$$Im \xrightarrow{Binary} Imb .$$

Бінаризацію зображень виконується на основі формули:

$$Im(x, y) = \begin{cases} 0, & Im(x, y) \leq T \\ 1, & Im(x, y) > T \end{cases}$$

де Im – вхідне зображення.

$Im(x, y)$ – точка вхідного зображення з координатами (x, y) .

T – порогове значення алгоритму бінаризації, задається користувачем на основі деякої апріорної інформації.

Серед переваг бінаризованого представлення зображень:

- зменшення розмірів файлів, для збереження зображення;
- можливість використання швидких алгоритмів обробки зображень, наприклад підкреслення контурів на основі деяких масок;
- автоматичне розділення зображення на область фону та множину об'єктів, що знаходяться на зображенні.

Результатом виконання даного етапу є бінаризоване вхідне зображення (рисунок 3.7).

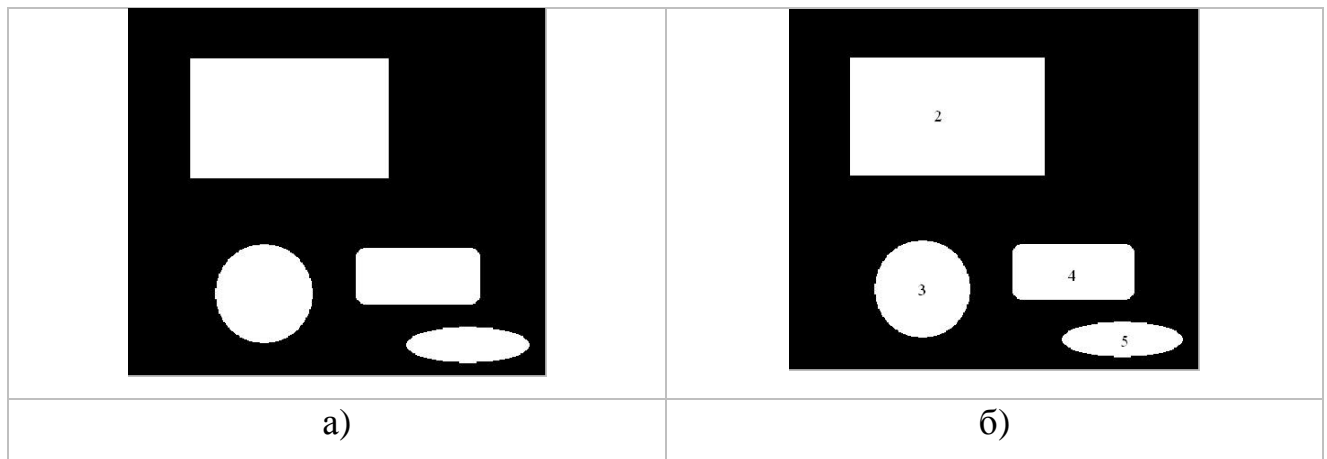


Рисунок 3.7 – Приклад бінаризації вхідного зображення(а), приклад розфарбування (б)

Розфарбування. Проводиться процедура ідентифікації кожної точки вхідного зображення. Ідентифікаційних номер присвоюється відповідно до

номера однорідної області. Нумерація областей проводиться зверху вниз, зліва на право.

$$\text{Im } b \xrightarrow{\text{Paint}} \text{Im } p.$$

Результатом виконання даного етапу є бінаризоване вхідне зображення з пронумерованим однорідними областями, що відповідають об'єктам на зображенні. Як правило, ідентифікатор №1 присвоюється області, що відповідає фону зображення.

Виділення контурів об'єктів. Визначення контуру відбувається на основі „алгоритму проходження контуром з зворотнім ходом”. Результатом роботи алгоритму є масив контурів об'єктів. Контури зберігаються у вигляді двовимірних масивів.

$$\text{Im } p \xrightarrow{\text{contur_tracind}} \tilde{N}_i.$$

Окрім структури контуру важливою є також інформація про форму об'єкта. В таблиці приведено основні типи форм об'єктів та якісний опис їхніх функції доцентрових відстаней. Функція доцентрових відстаней $R(s)$ визначається як відстань від центру об'єкта до кожної точки контуру і обчислюється за формулою:

$$R(s) = \sqrt{(x_c - x_o)^2 + (y_c - y_o)^2}.$$

Серед інших типів форм що виділяються при дослідженні об'єктів на біомедичних зображеннях слід відмітити: квадратну, прямокутну, ромбічну, трикутну, трикутну-рівнобедренну та інші.

Наведений перелік реалізованих функцій демонструє внутрішню будову програмного додатку та механізмів, що відбуваються під час його

функціонування. Як показав процес написання програмного коду, обрано мова програмування та набір бібліотек у повній мірі забезпечив можливість створення програмного додатку для вирішення усіх завдань, що були поставлені при виконанні магістерської роботи.

3.3 Тестування та аналіз реалізованої системи

Для тестування програмної системи розпізнавання текстової інформації на цифрових зображеннях було використано в якості тестової робочої станції персональний комп'ютер з наступними технічними характеристиками:

- Zalman Z1 Black + блок живлення Chieftec APS-550SB;
- HDD WD Caviar 1TB;
- videocard Asus PH-GTX1060-3G;
- МП AMD Ryzen 3 2200G BOX 120;
- ОЗУ 4Gb 2400GHz ;
- motherboard Asus B350M-E 90.

Технічні характеристики монітора, що використовувався для відображення зображень:

- діагональ монітора 19";
- тип матриці - TN;
- роздільна здатність 1920x1080;
- інтерфейси HDMI;
- частота оновлення кадрів монітора 65 Гц;
- відношення сторін 16: 9.

Технічні параметри комп'ютера в повній мірі забезпечують можливість проведення тестування спроектованого та реалізованого програмного додатку

та дозволить отримати тестові вибірки для проведення аналізу коректності роботи програмної системи та порівняння її з програмами аналогами.

Для проведення тестування рохробленої програм було підібрано різні групи вхідних зображень, що значно відрізняються один від одного своїм вмістом та характеристиками:

- чорно-білі зображення без явних спотворень тексту (чорні букви на білому фоні (рисунок 3.7а));
- чорно-білі зображення з спотвореним текстом (букви знаходяться під деяким кутом до осі абсцис(рисунок 3.7б));
- кольорові зображення без явних спотворень тексту (букви і фон різні за кольором, іноді можуть бути достатньо подібними (рисунок 3.7в));
- кольорові зображення з спотвореним текстом (текст на зображенні може бути написаний різними шрифтами та кольорами (рисунок 3.7г)).

<ul style="list-style-type: none"> - зміна розмірів графічних зображень; - переміщення графічних зображень; - обтискання графіки текстом; - обрізання зображень; - обрамлення зображень; - заливання графічного зображення; - обертання фігури та об'єкту WordArt; - перетворення зображення у водний знак; - перетворення об'єкту WordArt; - робота з групами графічних об'єктів. 	
а)	б)
	<p>Набір операцій, який виконується над об'єктами системи опрацювання текстів</p> <p>Введення тексту – з використанням клавіатури, стилуса, мікрофона, сканера.</p> <p>Редагування тексту – внесення змін у текст: виправлення помилок, видалення, переміщення, копіювання, вставка символів, слів, абзаців, рядків та ін.</p> <p>Форматування тексту – внесення змін у зовнішній вигляд тексту: установлення значень властивостей символів, абзаців, сторінок тощо.</p> <p>Вставка об'єктів – включення в документ графічних зображень, формул, діаграм, гіперпосилань та інші файли тощо.</p> <p>Структурування документа – виділення в тексті певних змістових частин: розділів, параграфів, пунктів тощо.</p> <p>Друкування документа – отримання копії документа на папері або спеціальній плівці.</p> <p>Робота з файлами – запис текстового документа на зовнішній носій, завантаження текстового документа із зовнішнього носія до програми опрацювання текстів та ін.</p>
в)	г)

Рисунок 3.8 – Приклад різноманітності вхідних зображень

В результаті аналізу можливих чинників впливу результати проведення тестування також було згруповано за декількома різними критеріями, для того щоб визначити параметри встановлення фотофіксуючої апаратури для максимально коректної роботи розробленої системи.

Для проведення першої групи тестів було відібрано зображення, що мають мінімальні спотворення, кольорова гамма перед відома, тобто параметри для роботи програмної системи можна встановити по замовчуванню та отримати відмінні результати (рисунок 3.9).

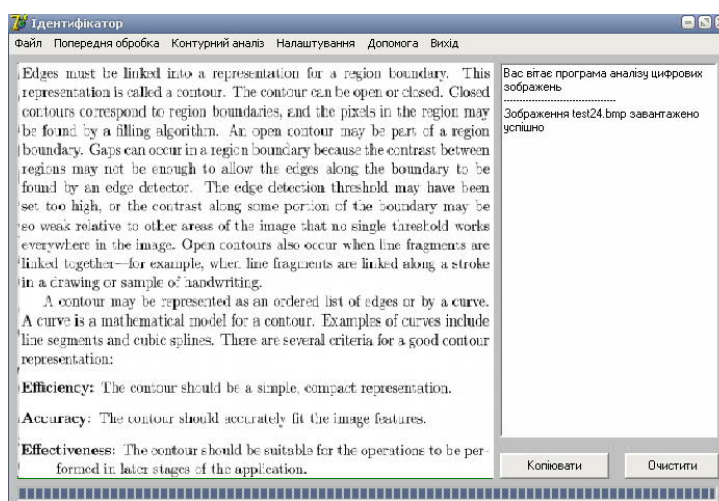


Рисунок 3.9 – Приклад роботи програми

В результаті тестування даної групи було зроблено наступні висновки, програмна система швидко і без додаткової корекції розпізнала запропонований фрагмент тексту, точність розпізнавання більше 99%, похибки розпізнавання були помічені при наявності нестандартного форматування або при розриві слів за допомогою переносу або при присутності специфічних символів, що рідко використовуються при друці.

Під час тестування другої групи зображень, в якості тестової вибірки були відібрані зображення на яких чітко промальовуються афінні спотворення вхідного зображення, нахили під різними кутами, проте параметри кольорової гами залишаються статичними, чорні символи на білому фоні. Результат розпізнавання напряму залежав від того на який саме кут було повернуто текст,

і наскільки дане спотворення впливає на саме зображення. Результати тестування були значно нижчими і в деяких випадках для отримання допустимої якості розпізнавання тексту необхідно було втручання оператора. Проте при невеликих кутах нахилу програма успішно проводить процес розпізнавання.

При інших групах зображення програма успішно справлялась з поставленими завданнями, на основі чого можна зробити висновки, що основним параметром який впливає на якість результату розпізнавання є чіткість відображення символів на екрані, а кольорова гама має набагато менший вплив.

В результаті проведених експериментів та на основі отриманих дани було сформовано усереднені дані, які значно характеризують процес роботи з запропонованим алгоритмом. Під час проведення тестових запитів було отримано підтвердження гіпотези, що саме адаптивний масив еталонних моделей в значній мірі дозволяє підвищити рівень якості процесу розпізнавання зображень в системах аналізу текстових документів на цифрових зображеннях. Результати досліджень наведено в таблиці 3.1.

Таблиця 3.1 – Узагальнена таблиця результатів тестування

Параметри вхідних зображень	Якість розпізнавання	Час роботи, с	Візуальна оцінка розпізнавання
Колір: чорні букви на білому фоні Спотворення: відсутні	дуже висока, від 98%	2-4	Відміння якість, дрібні помилки
Колір: чорні букви на білому фоні Спотворення: текст повернутий на деякий кут відносно осі OX	Залежить від кута	6-15	Якість напряму залежить від кута нахилу тексту
Колір: кольорові букви та фоні Спотворення: відсутні	висока, від 95%	5-8	Відміння якість, присутні помилки при подібності кольору фону та тексту
Колір: кольорові букви та фоні Спотворення: присутні	Залежить від рівня спотворення	20+	Необхідна допомога оператора, присутні нерозпізнані символи

В результаті проведених експериментів було виділено класи зображень для яких розроблений алгоритм показує відмінні результати. Похибки при роботі алгоритму прослідковуються при значних спотвореннях на зображенні, що потребують значної попередньої обробки вхідних зображень, проте задача по розробці та удосконаленні алгоритмів попередньої обробки не була поставлена під час виконання магістерської роботи.

3.4 Аналіз системи захисту доступу до системи

Оскільки програмна система може перетворювати інформацію з паперових носіїв в електронній формат, то доцільно розробити деяку систему обмеження доступу до окремих елементів програмного забезпечення. Система захисту інформації - сукупність спеціальних заходів правового та адміністративного характеру, організаційних заходів, фізичних і технічних засобів захисту, а також спеціального персоналу, призначеного для забезпечення безпеки інформації.

Ідентифікація в інформаційних системах - процедура, в результаті виконання якої для суб'єкта ідентифікації виявляється його ідентифікатор, однозначно ідентифікує цього суб'єкта в інформаційній системі. Для виконання процедури ідентифікації в інформаційній системі суб'єкту попередньо повинен бути призначений відповідний ідентифікатор (тобто проведена реєстрація суб'єкта в інформаційній системі).

Процедура ідентифікації безпосередньо пов'язана з аутентифікацією: суб'єкт проходить процедуру аутентифікації, і якщо аутентифікація успішна, то інформаційна система на основі факторів аутентифікації визначає ідентифікатор суб'єкта. При цьому достовірність ідентифікації повністю визначається рівнем достовірності виконаної процедури аутентифікації. Приклад реалізації процедури ідентифікації наведено на рисунку 3.7.

Логін – алфавітно-цифрова послідовність символів, що однозначно ідентифікує користувача і в той же час разом із паролем використовується операційною системою для встановлення користувачу дозволу до користування програмною розробкою та визначення його прав доступу до ресурсів системи. Логін має бути унікальним в межах даної операційної або програмної системи.

Пароль – це рядок символів, за допомогою якого користувач входить до системи, отримує доступ до файлів, програм та інших ресурсів.

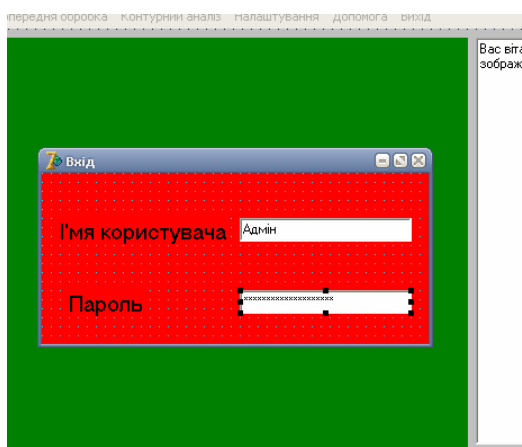


Рисунок 3.7 – Приклад вікна входу в систему

Комбінація логін/пароль є достатнім засобом для забезпечення ідентифікації Користувача в системі. На відміну від логіна пароль є секретом Користувача. Він не може передаватись нікому за будь-яких обставин. Правильно організоване збереження і опрацювання виробничих цифрових інформаційних ресурсів убезпечить від необхідності отримувати доступ до персонального профілю Користувача, який у конкретний момент часу не може виконувати виробничі обов'язки. Захист паролю убезпечить, також, від можливостей виконання несанкціонованих дій від імені дійсного власника логіну.

Створення цифрового облікового запису у поточний час користувачем під час першої активації роботи з програмою у присутності адміністратора. Новим

власникам такого запису надеться комбінація логін/пароль яка є унікальною для системи в цілому. Після отримання паролю користувач повинен дотримуватись правил, що забезпечать не розголошення даної інформації та використовувати тільки у виробничих цілях.

3.5 Висновки до розділу

Розроблено та проведено моделювання програмної системи розпізнавання текстової інформації на основі контурного аналізу, що дозволило програмно реалізувати систему обробки та розпізнавання текстової інформації на зображеннях.

Проведно тестування розробленої програмної системи обробки та розпізнавання зображень, на основі різних тестових груп вхідних зображень, що оцінити точність розпізнавання текстової інформації за допомогою розробленого алгоритму та встановити параметри роботи програмної системи.

ВИСНОВКИ

На основі аналізу сучасних програмних систем обробки цифрових зображень та аналізу алгоритмів розпізнавання тексту на цифрових зображеннях можна розбити наступні висновки:

1. Проведено дослідження та класифікацію методів та алгоритмів контурного аналізу, що дозволило виділити основні напрямки та алгоритми які використовуються опису цифрових зображень.

2. Проаналізовано алгоритми обробки цифрових зображень, що дозволило вигнати групу алгоритмів для отримання компактного опису об'єктів на цифрових зображень, а також їх переваги та недоліки.

3. Проведено аналіз програмних систем розпізнавання текстової інформації на цифрових зображень який дозволив виділити основні структурні модулі, а встановити інтерфейси обміну даними між ними.

4. Проведено аналіз алгоритмів розпізнавання текстових зображень за допомогою контурного аналізу, що дало можливість обрати математичне обґрунтування для проведення розпізнавання текстових цифрових зображень.

5. Розроблено алгоритм розпізнавання текстової інформації на цифрових зображень на основі контурного аналізу та процесу стоншення образів об'єктів, що дозволило спроектувати програмний додаток для обробки цифрових зображень.

6. Розроблено та проведено моделювання програмної системи розпізнавання тексту на цифрових зображеннях на основі стоншення їх контурів, що дозволило програмно реалізувати систему обробки зображень та провести її тестування.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Баженова И.Ю. Язык программирования С++ АО "Диалог-МИФИ", 2017. 366с.
2. Бартлетт Н. Программирование на Delphi Путеводитель. The Coriolis Group, Inc., 1996, Издательство НИПФ "ДиаСофт Лтд.", 2016. 116с.
3. Вебер Дж. Технология С++ в подлиннике. QUE Corporation, 2016, "ВНУ-Санкт-Петербург", 2017. 256с.
4. Волш А. И. Основы программирования на С++ для World Wide Web. IDG Books Worldwide, Inc., 1996, Издательство "Диалектика", 2016. 458с.
5. Марков А. С. «Базы данных. Введение в теорию и методологию. Финансы и статистика». 2016. С.24-35.
6. Абрамов С. А. Задачи по программированию. М.: Наука, 2018. 256с.
7. Березин Б.И., Начальный курс Delphi. М.: ДИАЛОГ-МИФИ, 2016. 331с.
8. Бондарев В.М. Основы программирования. Харьков: Фолио, Ростов н/Д: Феникс, 2017. 446с.
9. Вирт Н. Алгоритмы и структуры данных. М.: Мир, 2019. 345с.
10. Гладков В. П. Задачи по информатике на вступительном экзамене в вуз и их решения: Учебное пособие. Пермь: Перм. техн. ун-т, 2014. 516с.
11. Грогоно П. Программирование на языке Delphi. М.: Мир, 2012. 216с.
12. Дагене В.А. 100 задач по программированию. М.: Просвещение, 2013. 106с.
13. Джамса К. Библиотека программиста Java. Jamsa Press, 2016, ООО "Попурри", 2016. 656с.
14. Марков А. С. «Базы данных. Введение в теорию и методологию. Финансы и статистика». 2016. Р. 24-35.
15. Заварыкин В.М. Основы информатики и вычислительной техники. М.: Просвещение, 2019. 556с.

16. Касаткин В. Н. Информация. Алгоритмы. ЭВМ. М.: Просвещение, 2011. 219с.
17. Кен А. Язык программирования Delphi. Addison-Wesley Longman, U.S.A., 1996, Издательство "Питер-Пресс", 2017. 378с.
18. Керниган Б. Язык программирования Delphi. Пер. с англ. М.: Финансы и статистика, 2012. 391с.
19. Ляхович В.Ф. Руководство к решению задач по основам информатики и вычислительной техники. М.: Высшая школа, 2014. 127с.
20. Мейнджер Дж. Delphi Основы программирования. McGraw-Hill, Inc., 1996, Издательская группа ВНУ, Киев, 2017. 346с.
21. Миков А. И. Информатика. Введение в компьютерные науки. Пермь: Изд-во ПГУ, 2018. 442с.
22. Могилев А. В. Информатика: Учеб. пособие для студ. пед. Вузов. М.: Изд. центр «Академия», 2019. 629с.
23. Нотон П. JAVA: Справ. руководство. М.: БИНОМ: Восточ. Кн. Компания, 2016: Восточ. Кн. Компания. 447с.
24. Нотон П. Полный справочник по Java. McGraw-Hill, 1997, Издательство "Диалектика", 2017. 556с.
25. Ренеган Э. Дж. 1001 адрес WEB для программистов : Новейший путеводитель программиста по ресурсам World Wide Web: Пер.. Минск: Попурри, 2017. 512с. ил.
26. Родли Дж. Создание Java-апплетов. The Coriolis Group, Inc., 1996, Издательство НИПФ "ДиаСофт Лтд.", 2016. 466с.
27. Секреты программирования для Internet на Java. Ventana Press, Ventana Communications Group, U.S.A., 2016, Издательство "Питер Пресс", 2017. 396с.
28. Семакина И. Г. Информатика. Задачник-практикум: В 2 т.. М.: Лаборатория Базовых Знаний, 2019. 476с.
29. Сокольский М.В. Все об Intranet и Internet. М.: Элиот, 2018. 254с.
30. Тассел Д. Стиль, разработка, эффективность, отладка и испытание программ. М.: Мир, 2011. 56с.

31. Тюрин Ю.Н. Анализ данных на компьютере. М.: ИНФРА-М, Финансы и статистика, 2015. 384с.
32. Флэнэген Д. Java in a Nutshell. O'Reilly & Associates, Inc., 1997, Издательская группа BHV, Киев, 2018. 473с.
33. Чен М.С. Программирование на C++:1001 совет:Наиболее полное руководство по Java и Visual J++ :Пер.с англ. Минск:Попурри, 2017. 640с.ил.
34. Эферган М. C++: справочник. QUE Corporation, 2017, Издательство "Питер Ком", 1998. 256с.
35. G. Yang. «Human face detection in a complex background. Pattern Recognition », 27 (1): 2014. P.53-63.
36. Kotropoulos C. «Acoustics, Speech, and Signal Processing», 2017. ICASSP-97, 2017. IEEE International Conference on pp.2537-2540 v. 4
37. Leung TK. «Finding Faces in Cluttered Scenes Using Random Labeled Graph Matching» 2015.p P.83-95.
38. Yow KC. Feature-based human face detection. Image and vision computing 15 (9), 2017. P.713-735.
39. Sinha, P. Perceiving and Recognizing threedimensional forms. PhD thesis, Massachusetts Institute of Technology, 2016. 278p.
40. Lanitis, A «Image Anal. Classifying variable objects using a flexible shape model »Image Processing and its Applications, 2015., P.70-74.
41. Viola P. «Rapid Object Detection using a Boosted Cascade of Simple Features», proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR 2001), 2011., vol. 1, 518p.
42. Jones MJ. «Robust real-time face detection», International Journal of Computer Vision, vol. 57, no. 2, 2014., P.137-154.
43. Buchatskiy AN. «Selection of the Optimal Color Space for Reducing False Positives Rate in the Viola-Jones Method», Актуальні проблеми інфотелекомунікацій в науці та освіті, II Міжнародна науково-технічна та науково-методична конференція. Санкт-Петербург, 2013.

44. Ethan R. ORB: an efficient alternative to SIFT or SURF. Computer Vision (ICCV), IEEE International Conference on. IEEE, 2011. P. 2564–2571.

45. Stefan L. BRISK: Binary Robust Invariant Scalable Keypoints. Computer Vision (ICCV), 2011. P. 2548–2555.

46. Pablo F. Fast Explicit Diffusion for Accelerated Features in Nonlinear Scale Spaces. In British Machine Vision Conference (BMVC), 2013.

47. Martin A. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. Comm. Of the ACM24: 2001. P.381–395,.

48. Патин М.В. Сравнительный анализ методов поиска особых точек и дескрипторов при группировке изображений по схожему содержанию. Молодой ученый. 2016. №11. С. 214-221.

49. Гонсалес Р. Цифровая обработка изображений в среде MATLAB //М.: Техносфера. 2016. 616с.

50. Методичні рекомендації до виконання дипломної роботи з освітньо-кваліфікаційного рівня “Магістр”. Спеціальність „Комп’ютерні системи та мережі”/О.М. Березький, Л.О. Дубчак, Г.М. Мельник /Під ред. О.М. Березького Тернопіль: ТНЕУ, 2018. 41с.